M. Huba, S. Skogestad, M. Fikar, M. Hovd, T. A. Johansen, B. Rohal'-Ilkiv

Editors

Selected Topics on Constrained and Nonlinear Control

Preprints

STU Bratislava – NTNU Trondheim

Compilation: Miroslav Fikar
Cover: Tatiana Hubová
Printed and bounded in Slovakia by Miloslav Roubal ROSA, Dolný Kubín and Tlačiareň Vrábel, Dolný Kubín

ISBN: 978–80–968627–2–6

Preface

This preprints was created within the NIL-I-007-d Project "Enhancing NO-SK Cooperation in Automatic Control" (ECAC) carried out in 2009-2011 by university teams from the Slovak University of Technology in Bratislava and from the Norwegian University of Science and Technology in Trondheim. As it is already given by the project title, its primary aim was enhancing cooperation in academic research in the automatic control area in the partner institutions. This was achieved by supporting broad spectrum of activities ranging from student mobilities at the MSc. and PhD. level, staff mobilities, organization of multilateral workshop and conferences, joint development of teaching materials and publishing scientific publications. With respect to the original project proposal, the period for carrying out the foreseen activities was reasonably shortened and that made management of the all work much more demanding. Despite of this, the project has reached practically all planned outputs – this preprints represents one of them – and we believe that it really contributes to the enhancement of Slovak-Norwegian cooperation and to improvement of the educational framework at both participating universities. Thereby, I would like to thank all colleagues participating in the project activities at both participating universities and especially professor Sigurd Skogestad, coordinator of activities at the NTNU Trondheim, associate professor Katarína Žáková for managing the project activities, and professor Miroslav Fikar for the patient and voluminous work with collecting all contribution and compilation of all main three project publications (textbook, workbook, and workshop preprints).

Bratislava
2.1.2011

Mikuláš Huba
Project coordinator

Contents

Department of Engineering Cybernetics, NTNU Trondheim

Regularized Nonlinear Moving Horizon Observer - Theory and application to automotive friction estimation

Sui Dan and Tor A. Johansen

Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

Abstract: A constrained moving horizon observer is described and analysed for nonlinear discrete-time systems. The algorithm is proved to converge exponentially under a detectability assumption and the data being persistently exciting. However, in many practical estimation problems, such as combined state and parameter estimation, the data may not be persistently exciting. The algorithm therefore has regularization mechanisms to ensure robustness and graceful degradation of performance in time periods when the data are not exciting. This includes the use of a priori estimates in the moving horizon cost function, and the use of thresholded singular value decomposition to avoid ill-conditioned or ill-posed inversion of the associated nonlinear algebraic equations that define the moving horizon state estimate. The latter regularization relies on monitoring of the rank of an estimate of a Hessian-like matrix and conditions for exponential convergence are given. The method is in particular useful with augmented state space models corresponding to mixed state and parameter estimation problems, or dynamics that are not asymptotically stable, as illustrated with simulation examples. The main example considers wheel slip estimation for automotive applications using nonlinearly overparameterized tyre friction models where persistence of excitation does not hold.

1. INTRODUCTION

The state estimation problem of nonlinear discrete-time systems is investigated. A least-squares optimal state estimation problem can be formulated by minimizing a properly weighted least-squares criterion defined on the full data history horizon, subject to the nonlinear model equations, Moraal and Grizzle (1995b), Rao et al. (2003). This is, however, impractical as infinite memory and processing will be needed as the amount of data grows unbounded with time. Alternatively, a well known sub-optimal estimator is given by an Extended Kalman Filter (EKF) which approximates this least-squares problem and defines a finite memory recursive algorithm suited for real-time implementation, where only the last measurement is used to update the state estimate, based on the past history being approximately summarized by estimates of the state and the error covariance matrix, Gelb (2002). Unfortunately, the EKF is based on various stochastic assumptions on noise and disturbances that are rarely met in practice, and in combination with nonlinearities and model uncertainty, this may lead to unacceptable performance of the EKF. A possible better use of the dynamic model and past history when updating the state estimate is made by a Moving Horizon State Estimator (MHE) that makes use of a finite memory moving window of both current and historical measurement data in the least-squares criterion, possibly in addition to a state estimate and covariance matrix estimate to set the initial conditions at the beginning of the data window, see Rao et al. (2003), Moraal and Grizzle (1995b), Alessandri et al. (1999), Alessandri et al. (2008) for different formulation relying on somewhat different assumptions. Such an MHE can also be considered a sub-optimal approximation to an estimator that uses the full history of past data, and some empirical studies, Haseltine and Rawlings (2005) show that the MHE can perform better than the EKF in terms of accuracy and robustness. It should also be mentioned that other variations of the Kalman filter, such as particle filters and the unscented Kalman filter, also show great promise for nonlinear state estimation Rawlings and Bakshi (2006), Kandepu et al. (2008), Bølviken et al. (2001).

A direct approach to the deterministic discrete-time nonlinear MHE problem is to view the problem as one of inverting a sequence of nonlinear algebraic equations defined from the state update and measurement equations, and some moving time horizon. In principle, this approach avoids the use of a covariance matrix estimate, or any other historical information beyond the data window and a priori state estimate, and leads to a conceptually simple problem formulation and tuning parameters. Such discrete-time observers are formulated in the context of numerical nonlinear optimization and analyzed with respect to stability in Moraal and Grizzle (1995b), Alessandri et al. (2008). Some earlier contributions based on similar ideas are given in Glad (1983), Zimmer (1994), while Biyik and Arcak (2006) provides results on how to use a continuous time model in the discrete time design. As pointed out in Grossman (1999), the dead beat type of design philosophy (Moraal and Grizzle (1995b)) does not explicitly take into account robustness to noise, and some modifications are required as proposed in Grossman (1999). It should be mentioned that common to all methods is the use of numerical methods subject to the underlying assumption that local minima and multiple solutions may restrict convergence properties to be only local.

Uniform observability (in some form, see also Raff et al. (2005), Alamir (1999)) is assumed for stability or convergence proofs in the above mentioned references, including the EKF, Reif et al. (1998), Reif and Unbehauen (1999). Uniform observability means that the system and data are such that the problem of inverting the nonlinear algebraic equations is well-posed in the sense of Tikhonov and Arsenin (1977), i.e. that the state estimate solution exists, is unique and depends continuously on the measurement data. In the context of optimization this is commonly referred to as stability of the solution, that can be guaranteed under certain regularity and rank conditions, Fiacco (1983). This robustness is essential in any practical application since otherwise the estimates will be impossible to compute, and will be divergent or highly sensitive to imperfections such as numerical round-off errors, quantization and measurement noise. However, uniform observability is a restrictive assumption that is likely not to hold in certain interesting and important state estimation applications. This is in particular true for combined state and parameter estimation problems where the state space model is augmented with the unknown parameters, Gelb (2002), and convergence of the parameter estimates will depend on the information contents in the data, typically formulated as a condition for persistently exciting (PE) input data appearing in adaptive control and estimation, e.g. Krstic et al. (1995), or boundedness of the EKF covariance matrix estimate, Reif et al. (1998), Reif and Unbehauen (1999). In many practical applications the data will be sufficiently exciting for significant periods of time, but may in some time intervals contain insufficient excitation and information. It should also be noted that with some exceptions (e.g. Panteley et al. (2001), Sedoglavic (2002)), both uniform observability and PE conditions are difficult to verify a priori.

In this paper we consider strongly detectable systems Moraal and Grizzle (1995a), and the objective and novel contribution of the present work is to provide and study an MHE method based on Alessandri et al. (2008) and others with alternative weighting and regularization to achieve satisfactory practical performance also when the condition of uniform observability is violated due to temporarily lack of persistence of excitation, or the system not being observable. The relaxation to detectability was envisioned in Alessandri et al. (2008), although no proofs of the convergence were given. Following the spirit of Moraal and Grizzle (1995a) we introduce practical regularization mechanisms that monitor and estimate the information contents and degree of excitation in the data, and take corresponding action by adaptively weighting the measured data and a priori estimates from the dynamic model. Although the MHE formulation based on Alessandri et al. (2008) does not rely on an explicit uncertainty estimate in terms of a covariance matrix estimate (unlike formulations that may apply an arrival cost estimate, Rao et al. (2003)), the monitoring of persistent excitation in the moving horizon nonlinear observer relies on a related Hessian matrix estimate. This makes the approach similar in spirit to well known modifications of the EKF and Recursive Least Squares estimation methods that rely on monitoring and resetting of the covariance matrix estimate, directional forgetting and using singular value decomposition for numerically robust matrix inversion. Preliminary results are presented in Sui and Johansen (2010b), and further results on pre-filtering using EKF in combination with the MHE strategy and other examples are given in Poloni et al. (2010).

The outline of the paper is as follows: After the introduction, a description of the nonlinear moving horizon estimation problem and the relevant assumptions are given in Section 2, together with an analysis of its convergence under strong observability and with informative data. Section 4 extends the observer to have graceful degradation and practical performacne also for the case when the data are not informative and other assumptions are violated, which is followed by two numerical examples presented in Section 4. Final discussion and conclusions are given in Section 5.

The following notation and nomenclature is used. For a vector $x \in \mathbb{R}^n$, let $||x|| = \sqrt{x^T x}$ denote the Euclidean norm. Recall that the induced matrix norm $||M||$ equals the largest singular value of M. For two vectors $x \in R^n$ and $y \in R^m$ we let $\mathrm{col}(x,y)$ denote the column vector in R^{n+m} where x and y are stacked into a single column. The Moore-Penrose pseudo-inverse Golub and van Loan (1983) of a matrix M is denoted M^+ and we recall that for a matrix M of full rank it is given by $M^+ = (M^T M)^{-1} M^T$ while in general it is defined as $M^+ = VS^+U^T$ where $M = USV^T$ is a singular value decomposition where S is a diagonal matrix with the singular values $\sigma_1, ..., \sigma_n$ on the diagonal, and S^+ is the diagonal matrix $S = \mathrm{diag}(1/\sigma_1, ..., 1/\sigma_r, 0, ..., 0)$ where $r \leq n$ of the singular values are non-zero. The composition of two functions f and g is written $f \circ g(x) = f(g(x))$. Finally, a function $\varphi : R^+ \to R$ is called a K-function if $\varphi(0) = 0$ and it is strictly increasing.

2. NONLINEAR MHE PROBLEM FORMULATION

Consider the following discrete-time nonlinear system:

$$x_{t+1} = f(x_t, u_t) \tag{1a}$$

$$y_t = h(x_t, u_t), \tag{1b}$$

where $x_t \in \mathbb{X} \subseteq R^{n_x}$, $u_t \in \mathbb{U} \subseteq R^{n_u}$ and $y_t \in R^{n_y}$ are respectively the state, input and measurement vectors, and t is the discrete time index. The sets $\mathbb{X}$ and $\mathbb{U}$ are assumed to be convex and compact. The $N+1$ consecutive measurements of outputs and inputs until time t are denoted as $Y_t = \mathrm{col}(y_{t-N}, y_{t-N+1}, \cdots, y_t)$ and $U_t = \mathrm{col}(u_{t-N}, u_{t-N+1}, \cdots, u_t)$. To express Y_t as a function of x_{t-N} and U_t, denote $f^{u_t}(x_t) = f(x_t, u_t)$ and $h^{u_t}(x_t) = h(x_t, u_t)$, and note from (1b) that the following algebraic map can be formulated, Moraal and Grizzle (1995b):

$$\begin{aligned} Y_t &= H(x_{t-N}, U_t) \\ &= H_t(x_{t-N}) \\ &= \begin{bmatrix} h^{u_{t-N}}(x_{t-N}) \\ h^{u_{t-N+1}} \circ f^{u_{t-N}}(x_{t-N}) \\ \vdots \\ h^{u_t} \circ f^{u_{t-1}} \circ \cdots \circ f^{u_{t-N}}(x_{t-N}) \end{bmatrix}. \end{aligned} \tag{2}$$

Definition 1. Moraal and Grizzle (1995b) The system (1) is *N-observable* if there exists a K-function φ such that for all $x_1, x_2 \in \mathbb{X}$ there exists a feasible $U_t \in \mathbb{U}^{N+1}$ such that

$$\varphi(||x_1 - x_2||^2) \leq ||H(x_1, U_t) - H(x_2, U_t)||^2.$$

Definition 2a. The input $U_t \in \mathbb{U}^{N+1}$ is said to be *N-exciting* for the N-observable system (1) at time t if there exists a K-function φ_t that for all $x_1, x_2 \in \mathbb{X}$ satisfies

$$\varphi_t(||x_1 - x_2||^2) \leq ||H(x_1, U_t) - H(x_2, U_t)||^2.$$

From Proposition 2.4.7 in Abraham et al. (1983), we have

$$H(x_1, U_t) - H(x_2, U_t) = \Phi_t(x_1, x_2)(x_1 - x_2), \tag{3}$$

where

$$\Phi_t(x_1, x_2) = \int_0^1 \frac{\partial}{\partial x} H((1-s)x_2 + s x_1, U_t) ds. \tag{4}$$

Like in the linear case, an observability rank condition can be formulated (see also Moraal and Grizzle (1995b), Alessandri et al. (2008), Fiacco (1983) and others for similar results):

Lemma 1. If $\mathbb{X}$ and $\mathbb{U}$ are compact and convex sets, the functions f and h are twice differentiable on $\mathbb{X} \times \mathbb{U}$ and the Jacobian matrix $\frac{\partial H}{\partial x}(x, U_t)$ has full rank (equal to n_x) for all $x \in \mathbb{X}$ and some $U_t \in \mathbb{U}^{N+1}$, then the system is N-observable and the input U_t is N-exciting for the system (1) at time t.

Proof. Due to the observability rank condition being satisfied, $\Phi_t^T(\cdot)\Phi_t(\cdot) > 0$ and the system of nonlinear algebraic equations (3) can be inverted as follows:

$$x_1 - x_2 = \Phi_t^+(x_1, x_2)\big(H(x_1, U_t) - H(x_2, U_t)\big),$$
$$\Rightarrow \frac{1}{\pi_t^2(x_1, x_2)}\|x_1 - x_2\|^2 \leq \|H(x_1, U_t) - H(x_2, U_t)\|^2,$$

where $\pi_t(x_1, x_2) = \|\Phi_t^+(x_1, x_2)\|$. This proves that the conditions in Definitions 1 and 2a hold with $\varphi(s) = s/\overline{p}^2$ where $\overline{p} = \sup_{x_1, x_2 \in \mathbb{X}, U_t \in \mathbb{U}^{N+1}} \|\Phi_t^+(x_1, x_2)\|$ is bounded due to f and h are twice differentiable on the compact set $\mathbb{X} \times \mathbb{U}$. □

Define the N-information vector at time t as

$$I_t = \text{col}(y_{t-N}, \ldots, y_t, u_{t-N}, \ldots, u_t).$$

When a system is not N-observable, it is not possible to reconstruct exactly all the state components from the N-information vector. However, in some cases one may be able to reconstruct exactly at least some components, based on the N-information vector, and the remaining components can be reconstructed asymptotically. This corresponds to the notion of detectability, where we suppose there exists a coordinate transform $\mathbb{T} : \mathbb{X} \to \mathbb{D} \subseteq R^{n_x}$, where $\mathbb{D}$ is the convex hull of $\mathbb{T}(\mathbb{X})$:

$$d = \text{col}(\xi, z) = \mathbb{T}(x) \tag{5}$$

such that the following dynamics are equivalent to (1) for any initial condition in $\mathbb{X}$ and inputs in $\mathbb{U}$,

$$\xi_{t+1} = F_1(\xi_t, z_t, u_t) \tag{6a}$$
$$z_{t+1} = F_2(z_t, u_t) \tag{6b}$$
$$y_t = g(z_t, u_t). \tag{6c}$$

This transform effectively partitions the state x into an observable state z and an unobservable state ξ. The following strong detectability definition is taken from Moraal and Grizzle (1995a):

Definition 3. The system (1) is *strongly N-detectable* if
(1) there exists a coordinate transform $\mathbb{T} : \mathbb{X} \to \mathbb{D}$ that brings the system in the form (6);
(2) the sub-system (6b)-(6c) is N-observable;
(3) the sub-system (6a) has uniformly contractive dynamics, i.e. there exists a constant $L_1 < 1$ such that for all $\text{col}(\xi_1, z) \in \mathbb{D}, \text{col}(\xi_2, z) \in \mathbb{D}$ and $u \in \mathbb{U}$, the function F_1 satisfies

$$||F_1(\xi_1, z, u) - F_1(\xi_2, z, u)||' \leq L_1||\xi_1 - \xi_2||'. \tag{7}$$

with a suitable norm $||\cdot||'$.

It is remarked that since there is considerable freedom in the choice of transform $\mathbb{T}$ and the norm $||\cdot||'$, the contractivity assumption in part 3 of the definition is not very restrictive. For linear systems, it is equivalent to the conventional detectability definition with $||x||' = \sqrt{x^T P x}$ for $P = P^T > 0$.

Definition 2b. The input U_t is said to be *N-exciting* for a strongly N-detectable system (1) at time t if it is N-exciting for the sub-system (6b)-(6c) at time t.

The concept of N-exciting input imposes requirements that may be difficult to assess a priori. In section 3 we will study how N-excitation can be monitored online, and used in modifications to the basic MHE when this requirement is violated because the input data are not N-exciting at all times. If the input U_t is not N-exciting at certain points in time, the state estimation inversion problem (Moraal and Grizzle (1995b)) will be ill-posed (the solution does not exist, is not unique, or does not depend continuously on the data) or ill-conditioned (the unique solution is unaccepably sensitive to perturbations of the data), and particular consideration is required to achieve a robust estimator. Such modifications are generally known as regularization methods, see Tikhonov and Arsenin (1977). A common method, Tikhonov and Arsenin (1977), is to augment the cost function with a penalty on deviation from a priori information and makes the estimated solution degrade gracefully when U_t is not N-exciting. [1] We utilize an explicit regularization approach that weights open loop predictions made with the underlying model (1) similar to Alessandri et al. (2008). This will have similar filtering effect as reducing the feedback gain of a standard nonlinear observer or detuning the gain of an EKF through online tuning of the process noise covariance matrix. Further regularization will be motivated later, and introduced in section 3.

A convergent estimator is pursued by the following constrained, weighted, and regularized least-squares criterion

$$J(\hat{x}_{t-N,t}, \bar{x}_{t-N}, I_t) = \|W_t(Y_t - H_t(\hat{x}_{t-N,t}))\|^2 + \|M_t(\hat{x}_{t-N,t} - \bar{x}_{t-N})\|^2 \tag{8a}$$
$$s.t.\ \hat{x}_{t-N,t} \in \mathbb{X}, \tag{8b}$$

with M_t and W_t being time-varying weight matrices. Let $J_t^o = \min_{\hat{x}_{t-N,t}} J(\hat{x}_{t-N,t}, \bar{x}_{t-N}, I_t)$ subject to (10)-(8b), let $\hat{x}_{t-N,t}^o$ be the associated optimal estimate, and the estimation error is defined as

$$e_{t-N} = x_{t-N} - \hat{x}_{t-N,t}^o. \tag{9}$$

The state estimates on the remainder of the horizon are given by

$$\hat{x}_{i+1,t} = f(\hat{x}_{i,t}, u_i),\ \ i = t-N, \ldots, t-1 \tag{10}$$

It is assumed that an a priori estimator is determined as

$$\bar{x}_{t-N} = f(\hat{x}_{t-N-1,t-1}^o, u_{t-N-1}). \tag{11}$$

This formulation is a slight extension of Alessandri et al. (2008) with some additional flexibility provided by the time-varying weighting matrices W_t and M_t, which will be exploited in section 3. The condition $W_t^T W_t > 0$ may not be sufficient for uniqueness of a solution when the input is not N-exciting. However, the condition $M_t^T M_t > 0$ is generally sufficient to guarantee that the problem has a unique solution $\hat{x}_{t-N,t}^o$. This means that the second term of (8a) can be viewed as a regularization term and the matrix M_t containing regularization parameters.

We remark that the formulation does not account for model error or disturbances (or process noise) since in (8a) the dynamic model is assumed to hold perfectly in the predictions. This can easily be relaxed by introducing additional error variables to be optimized, like in Rao et al. (2003).

We first study the convergence of the MHE for the case when the input is assumed to be N-informative at all time, since this will help us understand and handle cases when this does not hold, too.

[1] Alternative regularization methods exist, and one implicit regularization method is to rely on the regularizing effect of an iterative approach that converges to a solution only asymptotically as $t \to \infty$ (and not converges to a solution at each individual time t), see e.g. Tautenhahn (1994). Hence, a regularizing effect is also achieved with the iterative sub-optimal variants described in Moraal and Grizzle (1995b); Alessandri et al. (2008).

The following are assumed throughout this paper:

(A1) The set $\mathbb{U}$ is compact and convex, and the output sequence $\{y_t\}$ and the input sequence $\{u_t\}$ are bounded.
(A2) For any $\mathrm{col}(\xi_1, z_1) \in \mathbb{T}(\mathbb{X})$ and $\mathrm{col}(\xi_2, z_2) \in \mathbb{T}(\mathbb{X})$, then $\mathrm{col}(\xi_1, z_2) \in \mathbb{T}(\mathbb{X})$.
(A3) The convex and compact set $\mathbb{X}$ is controlled invariant, *i.e.* $f(x_t, u_t) \in \mathbb{X}$ for all $x_t \in \mathbb{X}$ and the control u_t for all $t \geq 0$.
(A4) The initial state $x_0 \in \mathbb{X}$, and $\bar{x}_0 \in \mathbb{X}$.
(A5) The functions f and h are twice differentiable on $\mathbb{X} \times \mathbb{U}$, and the functions F_1, F_2 and g are twice differentiable on $\mathbb{D} \times \mathbb{U}$.
(A6) $\mathbb{T}(x)$ is continuously differentiable and bounded away from singularity for all $x \in \mathbb{X}$ such that $\mathbb{T}^{-1}(x)$ is well defined.
(A7) The system (1) is strongly N-detectable and the input U_t is N-exciting for all time $t \geq 0$.

In the stability analysis we will need to make use of the coordinate transform (5) into observable and unobservable states, although we emphasize that knowledge of this transform is not needed for the implementation of the observer. To express Y_t as a function of z_{t-N} and U_t (note that it does not depend on the unobservable states), the following algebraic mapping can be formulated similar to the mapping H:

$$\begin{aligned} Y_t &= G(z_{t-N}, U_t) \\ &= G_t(z_{t-N}) \\ &= \mathrm{col}(g^{u_{t-N}}(z_{t-N}), \cdots, g^{u_t} \circ F_2^{u_{t-1}} \circ \cdots \circ F_2^{u_{t-N}}(z_{t-N})). \quad (12) \end{aligned}$$

First, we establish lower and upper bounds on the optimal cost function J_t^o:

Lemma 2. Let

$$\begin{aligned} \Phi_t^o(z_{t-N}, \hat{z}_{t-N,t}^o) &= \int_0^1 \frac{\partial}{\partial z} G((1-s)z_{t-N} + s\hat{z}_{t-N,t}^o, U_t)ds, \\ p_{z,t} = p_t(z_{t-N}, \hat{z}_{t-N,t}^o) &= \|(W_t \Phi_t^o(z_{t-N}, \hat{z}_{t-N,t}^o))^+\| \end{aligned}$$

Then for all $x_{t-N}, \hat{x}_{t-N,t}^o \in \mathbb{X}$

$$J_t^o \geq 1/p_{z,t}^2 \|z_{t-N} - \hat{z}_{t-N,t}^o\|^2 \quad (13)$$

Proof. Using the fact that the system (1) can be transformed using (5), there exist $d_{t-N} = \mathbb{T}(x_{t-N})$, $\hat{d}_{t-N,t}^o = \mathbb{T}(\hat{x}_{t-N,t}^o)$ and $\bar{d}_{t-N} = \mathbb{T}(\bar{x}_{t-N})$ such that in the new coordinates, the system is in the form of (6a)-(6c). Note that the least squares term on the right-hand side of expression (8a) in the new coordinations can be rewritten as

$$\|W_t(Y_t - G_t(\hat{z}_{t-N,t}^o))\|^2 = \|W_t(G_t(z_{t-N}) - G_t(\hat{z}_{t-N,t}^o))\|^2.$$

From arguments similar to Lemma 1, it is clear that W_t can be chosen such that $p_{z,t}^2$ is uniformly bounded by any chosen positive number, and

$$\|W_t(Y_t - G(\hat{z}_{t-N,t}^o, U_t))\|^2 \geq 1/p_{z,t}^2 \|z_{t-N} - \hat{z}_{t-N,t}^o\|^2. \quad (14)$$

Taking zero as the lower bound on the second term of (8a) we get (13). □

Lemma 3. Let

$$\begin{aligned} L_2 &= \max_{d \in \mathbb{D}, u \in \mathbb{U}} \|\frac{\partial F_1}{\partial z}(\xi, z, u)\|, \\ L_3 &= \max_{x \in \mathbb{X}, u \in \mathbb{U}} \|\frac{\partial f}{\partial x}(x, u)\|, \\ k_T &= \max_{x \in \mathbb{X}} \|\frac{\partial \mathbb{T}}{\partial x}(x)\|, \\ k_{T^{-1}} &= \max_{x \in \mathbb{X}} \|\frac{\partial \mathbb{T}^{-1}}{\partial x}(x)\|, \\ k_M &= \sup_t \|M_t\| \end{aligned}$$

Then for all $x_{t-N}, \bar{x}_{t-N}, \hat{x}_{t-N,t}^o \in \mathbb{X}$

$$\begin{aligned} J_t^o \leq k_M^2 L_3^2 k_{T^{-1}}^2 (&\|\xi_{t-N-1} - \hat{\xi}_{t-N-1,t-1}^o\|^2 \\ &+ \|z_{t-N-1} - \hat{z}_{t-N-1,t-1}^o\|^2) \end{aligned} \quad (15)$$

Proof. First, we remark that the Lipschitz-like constants are well defined due to (A5) and (A6) and the compactness of $\mathbb{X}$ and $\mathbb{U}$. Since (A3) and (A4) hold, x_{t-N} is a feasible solution of the MHE problem (8). From the optimality of $\hat{x}_{t-N,t}^o$, we have $J_t^o \leq J(x_{t-N}, \bar{x}_{t-N}, I_t)$. It is easy to see that $\|W_t(Y_t - H(x_{t-N}, U_t))\|^2 = \|W_t(Y_t - G(z_{t-N}, U_t))\|^2 = 0$, and

$$\begin{aligned} \|M_t(x_{t-N} - \bar{x}_{t-N})\|^2 &\leq k_M^2 \|x_{t-N} - \bar{x}_{t-N}\|^2 \\ &\leq k_M^2 L_3^2 \|x_{t-N-1} - \hat{x}_{t-N-1,t-1}^o\|^2 \\ &\leq k_M^2 L_3^2 k_{T^{-1}}^2 (\|\xi_{t-N-1} - \hat{\xi}_{t-N-1,t-1}^o\|^2 \\ &\qquad + \|z_{t-N-1} - \hat{z}_{t-N-1,t-1}^o\|^2). \end{aligned}$$

and the result follows. □

Theorem 1. Let

$$\begin{aligned} q_{z,t} &= k_M L_3 k_{T^{-1}} p_{z,t}, \\ \bar{q}_z^2 &= \min\left(\frac{\mu}{2(1+\mu)}, \frac{\gamma(1-L_1^2)}{2(1+\mu)}\right), \\ \mu &= \frac{1-L_1^2}{3L_1 L_2}, \\ \gamma &= \frac{1-L_1}{3L_2^2(1+2L_1^2)}. \end{aligned}$$

By choosing appropriate weight matrices W_t and M_t, then $q_{z,t} \leq \bar{q}_z$ and the observer error dynamics is uniformly exponentially stable for any $x_0, \bar{x}_0 \in \mathbb{X}$.

Proof. From the lower and upper bounds in Lemmas 2 and 3, and the inequality $\sqrt{\|\xi\|^2 + \|z\|^2} \leq \|\xi\| + \|z\|$, we have for all $x_{t-N}, \hat{x}_{t-N,t}^o \in \mathbb{X}$ that

$$\begin{aligned} \|z_{t-N} - \hat{z}_{t-N,t}^o\| &\leq q_{z,t} \|z_{t-N-1} - \hat{z}_{t-N-1,t-1}^o\| \\ &\quad + q_{z,t} \|\xi_{t-N-1} - \hat{\xi}_{t-N-1,t-1}^o\|. \end{aligned} \quad (16)$$

Since $\mathbb{X}$ is positively invariant, then $\bar{x}_{t-N} \in \mathbb{X}$, and $\bar{x}_{t-N}$ is a feasible solution. From (A2), we know $\mathrm{col}(\bar{\xi}_{t-N}, \hat{z}_{t-N,t}^o)$ is also a feasible solution. Considering the cost function of the MHE problem (8), it is clear that $\mathrm{col}(\bar{\xi}_{t-N}, \hat{z}_{t-N,t}^o)$ is also an optimal solution, since the first term does not depend on the unobservable states and the second term is zero, i.e. $\hat{\xi}_{t-N,t}^o = \bar{\xi}_{t-N}$. Then from (A7),

$$\begin{aligned} \|\xi_{t-N} - \hat{\xi}_{t-N,t}^o\| &\leq L_1 \|\xi_{t-N-1} - \hat{\xi}_{t-N-1,t-1}^o\| \\ &\quad + L_2 \|z_{t-N-1} - \hat{z}_{t-N-1,t-1}^o\|. \end{aligned} \quad (17)$$

Let $s_{1,t} = \|z_{t-N} - \hat{z}^o_{t-N,t}\|$, and $s_{2,t} = \|\xi_{t-N} - \hat{\xi}^o_{t-N,t}\|$. Then combining (16) and (17) gives

$$\begin{pmatrix} s_{1,t} \\ s_{2,t} \end{pmatrix} \leq \begin{pmatrix} q_{z,t}, & q_{z,t} \\ L_2, & L_1 \end{pmatrix} \begin{pmatrix} s_{1,t-1} \\ s_{2,t-1} \end{pmatrix}. \tag{18}$$

Since $s_{i,t} \geq 0$, it follows that $s_{i,t} \leq \bar{s}_{i,t}$, $i = 1,2$, where we define the 2nd order linear time-varying system

$$\begin{pmatrix} \bar{s}_{1,t} \\ \bar{s}_{2,t} \end{pmatrix} = \begin{pmatrix} q_{z,t}, & q_{z,t} \\ L_2, & L_1 \end{pmatrix} \begin{pmatrix} \bar{s}_{1,t-1} \\ \bar{s}_{2,t-1} \end{pmatrix}, \begin{pmatrix} \bar{s}_{1,0} \\ \bar{s}_{2,0} \end{pmatrix} = \begin{pmatrix} s_{1,0} \\ s_{2,0} \end{pmatrix}.$$

Consider a Lyapunov function candidate $V(s_1,s_2) = \bar{s}_1^2 + \gamma\bar{s}_2^2$ with $\gamma > 0$. It follows that

$$\begin{aligned} & V(\bar{s}_{1,t},\bar{s}_{2,t}) - V(\bar{s}_{1,t-1},\bar{s}_{2,t-1}) \\ & \quad = -(1 - q_{z,t}^2 - \gamma L_2^2)\bar{s}_{1,t-1}^2 - (\gamma - q_{z,t}^2 - \gamma L_1^2)\bar{s}_{2,t-1}^2 \\ & \quad + 2(q_{z,t}^2 + \gamma L_1 L_2)\bar{s}_{1,t-1}\bar{s}_{2,t-1}. \end{aligned}$$

According to Young's inequality, for any $\mu > 0$,

$$\begin{aligned} & V(\bar{s}_{1,t},\bar{s}_{2,t}) - V(\bar{s}_{1,t-1},\bar{s}_{2,t-1}) \\ & \quad \leq -(1 - q_{z,t}^2 - \gamma L_2^2)\bar{s}_{1,t-1}^2 - (\gamma - q_{z,t}^2 - \gamma L_1^2)\bar{s}_{2,t-1}^2 \\ & \quad + (q_{z,t}^2 + \gamma L_1 L_2)/\mu\bar{s}_{1,t-1}^2 + (q_{z,t}^2 + \gamma L_1 L_2)\mu\bar{s}_{2,t-1}^2 \\ & \quad \leq -\delta_1\bar{s}_{1,t-1}^2 - \delta_2\bar{s}_{2,t-1}^2. \end{aligned}$$

where

$$\delta_1 = 1 - (1 + 1/\mu)q_{z,t}^2 - \gamma\left(L_2^2 + L_1 L_2/\mu\right) \tag{19}$$

$$\delta_2 = \gamma(1 - L_1^2) - (1 + \mu)q_{z,t}^2 - \gamma L_1 L_2 \mu \tag{20}$$

First, choose μ such that $L_1 L_2 \mu = \frac{1}{3}(1 - L_1^2)$. Then the first term of δ_2 dominates its third term by a factor 3, and

$$\mu = \frac{1 - L_1^2}{3L_1 L_2} > 0. \tag{21}$$

Second, choose γ such that $\gamma(L_1 L_2/\mu + L_2^2) = \frac{1}{3}$, which leads to the first term of δ_1 dominating its third term by a factor 3, and

$$\gamma = \frac{1 - L_1^2}{3L_2^2(1 + 2L_1^2)} > 0. \tag{22}$$

Third, since $q_{z,t}$ is chosen such that the first terms of both δ_1 and δ_2 dominate their second terms by a factor 2, respectively, and we have

$$(1 + 1/\mu)q_{z,t}^2 \leq \frac{1}{2} \ \Rightarrow q_{z,t}^2 \leq \frac{\mu}{2(1+\mu)},$$

$$(1+\mu)q_{z,t}^2/\gamma \leq \frac{1}{2}(1 - L_1^2) \ \Rightarrow q_{z,t}^2 \leq \frac{\gamma(1 - L_1^2)}{2(1+\mu)},$$

such that $\delta_1 > 0$ and $\delta_2 > 0$. There always exists a matrix M_t with some sufficiently small k_M and a matrix W_t for some sufficiently small $p_{z,t}$ such that $q_{z,t} \leq \bar{q}_z$ such that $\delta_1 > 0$ and $\delta_2 > 0$, and the 2nd order LTV system is uniformly exponentially stable for the given initial conditions. Since $s_{i,t} \leq \bar{s}_{i,t}$ and (A6) holds, the error dynamics is also uniformly exponentially stable for any $x_0, \bar{x}_0 \in \mathbb{X}$. □

Assumption (A2) is used in the proof to ensure that a feasible solution $\bar{x}_{t-N}$ remains feasible in the transformed coordinates when the observable states are replaced by their optimal values. This assumption is trivially satisfied for any N-observable system. For systems that are not N-observable, but N-detectable, it will still hold trivially in many cases as illustrated in Example 1 later. Like many other assumptions in this paper, such as (A7), it will not be trivial to verify unless $\mathbb{T}$ is known. However, Theorem 1 remains of value in such cases since it provides a qualitative understanding of the method. Hence, the theory provides a guideline, rather than replacement, for practical tuning as illustrated in the examples.

If the data are not N-exciting, the second term of the observer cost function dominates and the observer degenerates to an open loop observer for the state combinations that are not excited, provided M_t has full column rank in the sub-space corresponding to the linear combination of states not being excited. With $M_t^T M_t > 0$ this is trivially satisfied. This may be a satisfactory solution if the system has open loop asymptotically stable dynamics within the region of operation, since the observer may still converge and give accurate estimates. In practise, the accuracy will then depend entirely on the accuracy of the model. If the system is not open loop asymptotically stable, and in particular if there are significant model errors, this approach may not be satisfactory since errors will be allowed to accumulate without the presence of feedback from measurements. This will be the case in a mixed parameter and state estimation problem with the state space $x = \text{col}(\chi, \theta)$ corresponds to the system state χ and the unknown parameters θ and the augmented dynamics

$$\chi_{t+1} = f(\chi_t, \theta_t, u_t) \tag{23}$$

$$\theta_{t+1} = \theta_t \tag{24}$$

$$y_t = h(\chi_t, \theta_t) \tag{25}$$

Regardless of the system dynamics f, the augmented parameter dynamics $\theta_{t+1} = \theta_t$ are not asymptotically stable and estimates may drift off due to integrated errors (see Example 1 later). In the next section we introduce further methods for weighting and regularization that degrade gracefully when data are not N-exciting, which are particularly useful when the system is not asymptotically stable and there are model errors, as in the case with mixed state and parameter estimation.

3. ADAPTIVE WEIGHTING AND REGULARIZATION WITHOUT PERSISTENCE OF EXCITATION

In order to implement excitation-sensitive regularization, it is essential to be able to monitor if the data are N-exciting or not. For N-observable systems, the condition $p_{z,t} \leq \overline{\alpha} \leq \bar{q}_z/(k_M L_3 k_{T^{-1}})$ will depend on the existence of a (not too small) $\varepsilon > 0$ such that

$$\Phi_t^T(x_{t-N}, \hat{x}^o_{t-N,t})\Phi_t(x_{t-N}, \hat{x}^o_{t-N,t}) \geq \varepsilon I > 0 \tag{26}$$

for all $t \geq 0$, where

$$\Phi_t(x_1,x_2) = \int_0^1 \frac{\partial}{\partial x} H((1-s)x_1 + sx_2, U_t)ds \tag{27}$$

This condition comes from the requirement of U_t being N-exciting at all t and is similar to a PE condition. Unfortunately, since $\Phi_t(x_{t-N}, \hat{x}^o_{t-N,t})$ depends on the unknown x_{t-N} we cannot compute $p_t(x_{t-N}, \hat{x}^o_{t-N,t}) = \|(W_t\Phi_t(x_{t-N}, \hat{x}^o_{t-N,t}))^+\|$ exactly at any point in time to monitor if U_t is N-exciting. Instead, we have to rely on some approximation or estimate of $p_t(\cdot)$. If it is assumed that $||e_{t-N}||$ is small, then

$$\Phi_t(x_{t-N}, \hat{x}^o_{t-N,t}) \approx \Phi_t(\hat{x}^o_{t-N,t}, \hat{x}^o_{t-N,t}) = \frac{\partial H}{\partial x}(\hat{x}^o_{t-N,t}, U_t)$$

and we can use $\hat{p}_t(\hat{x}^o_{t-N,t}) = \|(\frac{\partial H^T}{\partial x}(\hat{x}^o_{t-N,t}, U_t)\frac{\partial H}{\partial x}(\hat{x}^o_{t-N,t}, U_t))^+ \frac{\partial H^T}{\partial x}(\hat{x}^o_{t-N,t}, U_t)W_t^+\|$ to approximate $p_t(x_{t-N}, \hat{x}^o_{t-N,t})$. Consider a singular value decomposition (SVD), Golub and van Loan (1983)

$$\frac{\partial H}{\partial x}(\hat{x}^o_{t-N,t}, U_t) = U_t S_t V_t^T. \tag{28}$$

Any singular value (diagonal element of the matrix S_t) that is zero or close to zero indicates that a component is unobservable or the input is not N-exciting. Moreover, the corresponding row of the V_t matrix will indicate which components cannot be estimated. The Jacobian has the structural property that its rank will be no larger than $\dim(z) = n_z \leq n_x$, due to certain components being unobservable. In addition, its rank can be reduced by data being not N-exciting as discussed in Lemma 1. The N-excitation of data may therefore be monitored through the robust computation of the rank of the Jacobian matrix using the SVD, Golub and van Loan (1983).

In general, we do not want to make the estimation of the unobservable or unexcited components depend on the measured data. This is effectively achieved by utilizing the SVD in order to compute a "robust pseudo-inverse" where the inverse of *small* singular values is set to zero rather than let grow unbounded.

We know that convergence depends on W_t being chosen such that $p_{z,t}$ is bounded by a sufficiently small number. To pursue this objective, one may choose W_t such that, whenever possible,

$$\|(W_t\Phi_t(\hat{x}^o_{t-N,t},\hat{x}^o_{t-N,t}))^+\| = \alpha, \tag{29}$$

where $\alpha > 0$ is a sufficiently small scalar. In order to give zero weight on data for components that are either unobservable or unexcited, we modify this ideal design equation into the more practical and realistic design objective

$$\|(W_tU_tS_{t,\delta}V_t^T)^+\| = \begin{cases} \alpha, \text{ if } ||S_t|| \geq \delta \\ 0, \text{ otherwise} \end{cases} \tag{30}$$

where the thresholded pseudo-inverse $S^+_{t,\delta} = \mathrm{diag}(1/\sigma_{t,1},\dots,1/\sigma_{t,\ell},0,\dots,0)$ where $\sigma_1,\dots,\sigma_\ell$ are the singular values larger than some $\delta > 0$ and the zeros correspond to small singular values whose inverse is set to zero, Golub and van Loan (1983). This leads to

$$W_t = (1/\alpha)V_tS^+_{t,\delta}U_t^T \tag{31}$$

satisfying

$$\|(W_t\Phi_t(\hat{x}^o_{t-N,t},\hat{x}^o_{t-N,t}))^+\| \leq \alpha. \tag{32}$$

Here M_t is chosen as

$$M_t = \beta I_{n_x}, \tag{33}$$

where $\beta \geq 0$ is a scalar.The following result shows that the W_t defined in (32) satisfies the conditions of Theorem 1 locally.

Theorem 2. If W_t is chosen according to (31) with δ being sufficiently small and $0 < \alpha < \bar{q}_z/(L_3k_{T-1}k_T\beta)$, then W_t is bounded and the observer error dynamics is locally uniformly exponentially stable.

Proof. Boundedness of W_t follows directly from $\alpha, \delta > 0$. Since δ is sufficiently small and the data are N-exciting, we assume without loss of generality that the matrix $W_t\Phi^o_t(z_{t-N},\hat{z}^o_{t-N,t})$ has full rank. Using similar arguments as Lemma 1, it is easy to show that (14) in the proof of Theorem 1 is still valid.

$$Y - H(\hat{x}^o_{t-N,t},U_t) = \Phi_t(x_{t-N},\hat{x}^o_{t-N,t})(x_{t-N} - \hat{x}^o_{t-N,t}),$$
$$Y - G(\hat{z}^o_{t-N,t},U_t) = \Phi^o_t(z_{t-N},\hat{z}^o_{t-N,t})(z_{t-N} - \hat{z}^o_{t-N,t}).$$

To simplify the notation, let $\Phi_{x,t} = \Phi_t(x_{t-N},\hat{x}^o_{t-N,t})$ and $\Phi_{z,t} = \Phi^o_t(z_{t-N},\hat{z}^o_{t-N,t})$. Since $Y - H(\hat{x}^o_{t-N,t},U_t) = Y - G(\hat{z}^o_{t-N,t},U_t)$,

$$\Phi_{z,t}(z_{t-N} - \hat{z}^o_{t-N,t}) = \Phi_{x,t}(x_{t-N} - \hat{x}^o_{t-N,t}).$$

It is known that

$$d_{t-N} - \hat{d}^o_{t-N,t} = \Gamma_t(x_{t-N} - \hat{x}^o_{t-N,t}),$$

where $\Gamma_t = \int_0^1 \frac{\partial}{\partial x}\mathbb{T}((1-s)x_{t-N} + \hat{s}\hat{x}^o_{t-N,t})ds$. Together with $z = \eta d$, where $\eta = [\mathbf{0}_{n_z\times(n_x-n_z)}, I_{n_z\times n_z}]$, we have

$$\Phi_{z,t}\eta(d_{t-N} - \hat{d}^o_{t-N,t}) = \Phi_{x,t}\Gamma_t^{-1}(d_{t-N} - \hat{d}^o_{t-N,t}).$$

It follows that

$$\begin{aligned} & W_t\Phi_{z,t}\eta = W_t\Phi_{x,t}\Gamma_t^{-1} \\ \Rightarrow & (W_t\Phi_{z,t})^+ = \eta\Gamma_t(W_t\Phi_{x,t})^+ \\ \Rightarrow & ||(W_t\Phi_{z,t})^+|| \leq ||\eta|| \cdot ||\Gamma_t|| \cdot ||V_tS^+_{t,\delta}U_t^TW_t^+||. \end{aligned} \tag{34}$$

It is known that $||\eta|| = 1$ and $||\Gamma_t|| \leq k_T$, then from (32)

$$p_{z,t} = ||(W_t\Phi_{z,t})^+|| \leq k_T||V_tS^+_{t,\delta}U_t^TW_t^+|| \leq k_T\alpha.$$

From Theorem 1, to obtain conditions on the Lyapunov function, $q_{z,t} = k_{T-1}\beta L_3p_{z,t} \leq \bar{q}_z$. Therefore, the convergence condition is implied by $\alpha < \bar{q}_z/(L_3k_{T-1}k_T\beta)$. Note that (34) and the following arguments only holds for $||e_t||$ in a neighborhood of the origin, therefore only local exponential convergence results. □

The tuning parameters with this adaptive choice of W_t and M_t are the non-negative scalars α, δ and β. It is worthwhile to notice that since they are scalars, a successful tuning of the observer will depend on appropriate scaling of the variables and model equations.

When the data are not considered N-exciting at some time instant, then δ should be tuned such that the corresponding singular values of $\Phi_t(\cdot)$ will be less than δ such that W_t defined by (31) will not have full rank. The means that the error in the corresponding state combinations will not be penalized by the first term in the criterion, and due to the second term the estimates will be propagated by the open loop model dynamics. In case of augmented parameter states $\theta_{t+1} = \theta_t$ this means that they are essentially frozen at their value from the previous sample.

Both $\beta \geq 0$ and $\delta \geq 0$ could be considered as regularization parameters that must be chosen carefully in order to tune the practical performance of the observer. In the ideal case with a perfect model, no noise, no disturbances and N-exciting data at all sampling instant one could choose $\delta = \beta = 0$. As a practical tuning guideline we propose to first choose $\beta > 0$ in order to achieve acceptable filtering and performance with typical noise and disturbance levels for typical cases when the data are N-exciting. Second, $\delta > 0$ is chosen in order to achieve acceptable performance also in operating conditions when the data are not N-exciting.

4. EXAMPLES

4.1 *Example 1 - mixed state and parameter estimation*

Consider the following nonlinear system

$$\dot{x}_1 = -2x_1 + x_2 \tag{35a}$$
$$\dot{x}_2 = -x_2 + x_3(u - w) \tag{35b}$$
$$\dot{x}_3 = 0 \tag{35c}$$
$$y = x_2 + v. \tag{35d}$$

One may think of x_3 as a parameter representing an unknown gain on the input, where the third state equation is an augmentation for the purpose of estimating this parameter. It is clear that x_1 is not observable, but corresponds to a stable sub-system. It is also clear that the observability of x_3 will depend on the excitation u, while x_2 is generally observable.

The same observability and detectability properties hold for the discretized system with sampling interval $t_f = 0.1$. When $u = 0$

for all time, the rank of $\frac{\partial H}{\partial x}(\hat{x}^o_{t-N,t}, U_t)$ is 1. When u is white noise, the rank of $\frac{\partial H}{\partial x}(\hat{x}^o_{t-N,t}, U_t)$ is 2 almost always.

In this simulation example we choose $N = 2$ such that the moving window has length $N+1 = 3$. The base case is defined as follows. We use the adaptive weighting law (31) with $\alpha = 1$, $\delta = 0.1$. Measurement noise, with independent uniformly distributed $v \in [-0.5, 0.5]$, is added to the base case. The input is chosen with periods without informative data as follows: During $0 \le t < 30t_f$, $u = 0$. During $30t_f \le t < 60t_f$, u is discrete-time white noise. During $60t_f \le t \le 120t_f$, $u = 0$. In the simulation, true system has an input disturbance with $w = 0.15$, and the model used in the MHE observer has no explicit knowledge of the input disturbance. In the following figures, true states are shown in solid line; estimated states of proposed work are shown in dash-dot line; estimated states using the alternative setting with fixed W_t are shown in dash line. The following initial conditions are used: $x_0 = [4, -7, 2], \bar{x}_0 = [3, -5.9, -1]$.

- Case 1: Default settings are used for the proposed work; for the alternative method, choose $W_t = 4I$ and $\beta = 1$. The simulation result is shown in Figure 1.
- Case 2: Choose $\beta_i = 0, i = 0, 1, 2$ for the proposed work; for the alternative method, choose $W_t = I$ and $\beta = 0$. The simulation result is shown in Figure 2.
- Case 3: The measurements are generated using the true system while the model in the observer differs by adding 10% error on all model parameters. Choose $\alpha = 0.03$, $\beta_i = 0.01$ and $\delta = 0.2$ for the proposed work; for the method of work (Alessandri et al. (2008)), choose $W_t = 4I$ and $\beta = 1$. The simulation result is shown in Figure 3.

The example shows that the adaptive weighting with the thresholded singular value inversion effectively freezes the unexcited parameter estimate and thereby avoids the parameter estimate drift that otherwise may result due to unmatched model error (input disturbance) when there are no excitations. This is due to the estimator degrading to integrated in an open loop fashion the parameter model $\dot{\theta} = 0$ in this case. Additional regularization is achieved by $\beta > 0$ since otherwise the parameter estimation will be mainly dominated by noise, as shown by case 2.

4.2 *Example 2 - Wheel slip and tyre-road friction estimation*

An anti-lock brake system (ABS) controls the slip of each wheel of a vehicle to prevent it from locking such that a high friction is achieved and steerability is maintained during hard braking. ABS brakes are characterized by robust adaptive behavior with respect to highly uncertain tyre characteristics and fast changing road surface properties and they have been commercially available in cars for 30 years (Burckhardt (1993); SAE (1992)).

Since the vehicle forces transferred from the tires to the road determine the vehicle motion, accurate information about road surface properties (dry, wet, snow, ice, etc.) has a significant importance in ABS and other automotive active safety systems. However, such forces and road surface properties are usually difficult to measure, since sensors are too complex and expensive for use in production cars. Therefore, it is necessary to estimate them from the computed or measurable signals such as angular wheel speed and the torque acting on the wheel's axis. In order to take advantage of a friction model to estimate the longitudinal wheel slips and speed for use in a wheel slip control system, one should also estimate one or more parameters of the friction model Johansen et al. (2003); Alvarez et al. (2005).

Consider the longitudinal dynamics corresponding to one wheel and 1/4 of a vehicle mass (quarter-car model). The wheel slip dynamics are important in the control algorithm of an anti-lock brake systems (ABS), Burckhardt (1993); Johansen et al. (2003)

$$\dot{v} = -\frac{1}{m} F_z \mu_1(\lambda), \tag{36a}$$

$$\dot{\lambda} = -\frac{1}{v}\left(\frac{1}{m}(1-\lambda) + \frac{r^2}{J}\right) F_z \mu_1(\lambda) + \frac{1}{v}\frac{r}{J} T_b, \tag{36b}$$

$$y = \frac{v(1-\lambda)}{r}, \tag{36c}$$

where v is the longitudinal speed, $\lambda = (v - \omega r)/v$ is longitudinal tyre slip, T_b is torque acting on wheel axis, $F_z = mg$ is the vertical force, μ_1 is the friction coefficient, and y is the angular wheel speed measurement. This application requires a combined state and parameter estimator since only the angular wheel speed $y = \omega$ is measured, such that both λ and v needs to be estimated togehter with parameters of the tire-road friction model that defines the friction coefficient μ_1. The parameters used in the paper are given in Table 1.

Para.	Description	Value	Unit
m	Quarter vehicle mass.	325	kg
J	Moment of inertia of the wheel.	1	kgm^2
r	Wheel radius.	0.345	m
g	Acceleration of gravity.	9.81	m/s^2
F_z	Vertical force.	3188	kgm^2/s^2
v	Longitudinal speed.		m/s
ω	Angular speed.		rad/s
λ	Longitudinal tyre slip.		m/s
T_b	Torque acting on wheel axis.		Nm
F_x	Friction force between wheel and road.		N

Table 1. Model vaiables. The numeric values are nominal values used in the simulation case study.

In the example the friction coefficient $\mu_1(\lambda)$ is a nonlinear function of the longitudinal slip λ with

$$\mu_1(\lambda) = \theta \sin(C(\arctan B\lambda - E(B\lambda - \arctan(B\lambda)))). \tag{37}$$

and the parameters B, C, E and θ characterize the tire and the road surface. Typical values of parameters B, C, E and θ are given by Matuško et al. (2003)

Dry asphalt: $B = 10.38,\ C = 1.65,\ E = 0.65663,\ \theta = 1$

Snow: $B = 14.395,\ C = 0.9,\ E = -6.439,\ \theta = 0.3$

The longitudinal slip $\lambda = (v - \omega r)/v$ describes the normalized difference between the vehicle's longitudinal velocity v and the speed of the wheel perimeter ωr. The slip value of $\lambda = 0 (v = \omega r)$ characterizes the free motion of the wheel where no friction force F_x is exerted. If the slip attains the value $\lambda = 1$ then the wheel is locked ($\omega = 0$). The typical friction curves $\mu(\lambda)$ are shown in Figure 4. The friction coefficient μ is generally a differentiable function with the property $\mu(0) = 0$ and $\mu(\lambda) > 0$ for $\lambda > 0$. Figure 4 shows how μ increases with slip λ up to some value, where it attains its maximum value. For higher slip values, the friction coefficient will decrease to a minimum value where the wheel is locked and only the sliding friction will act on the wheel. The dependence of friction on the road condition is also exemplified in Figure 4. For wet or icy roads,

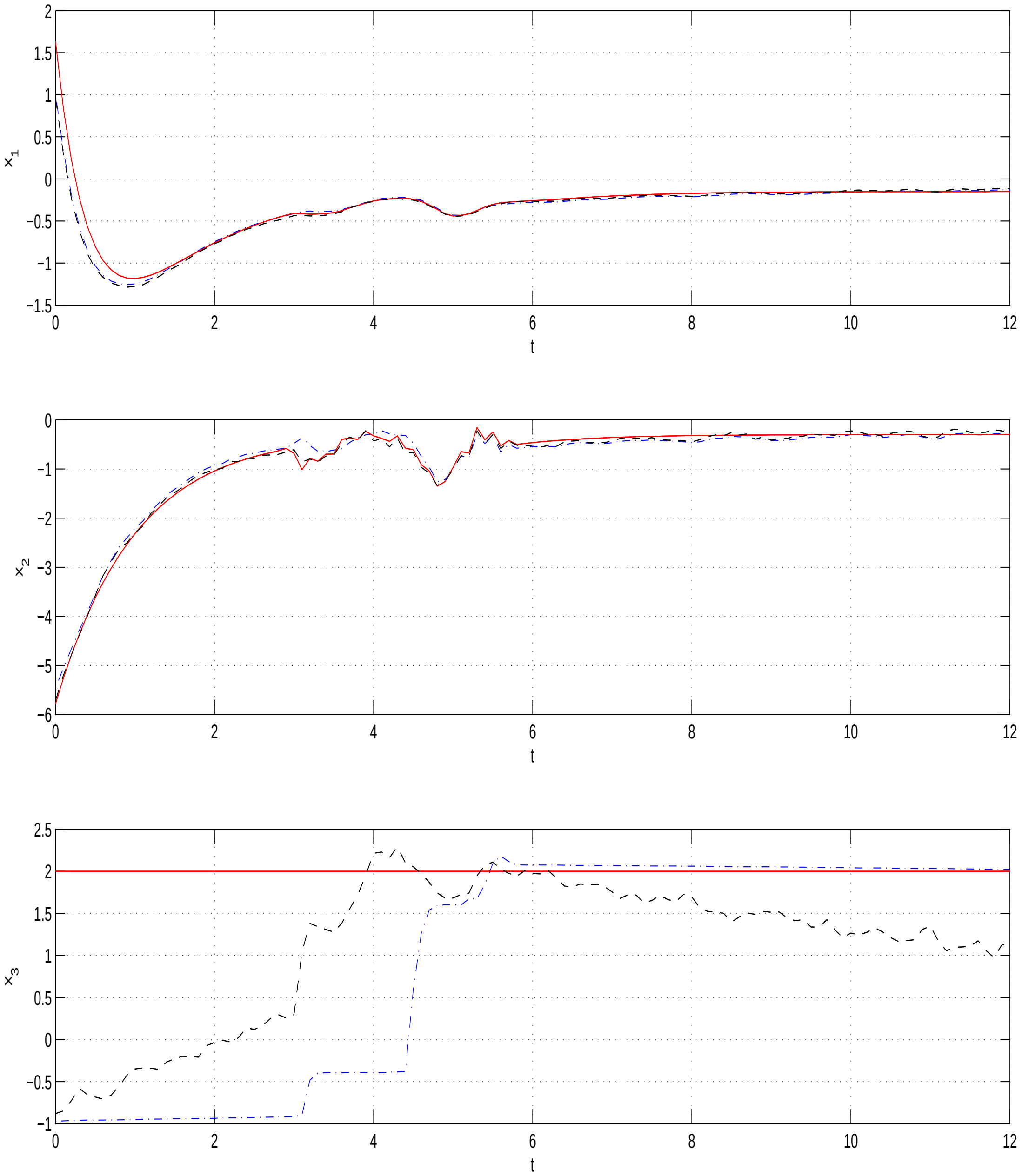


Fig. 1. Simulation results of example 1, case 1.

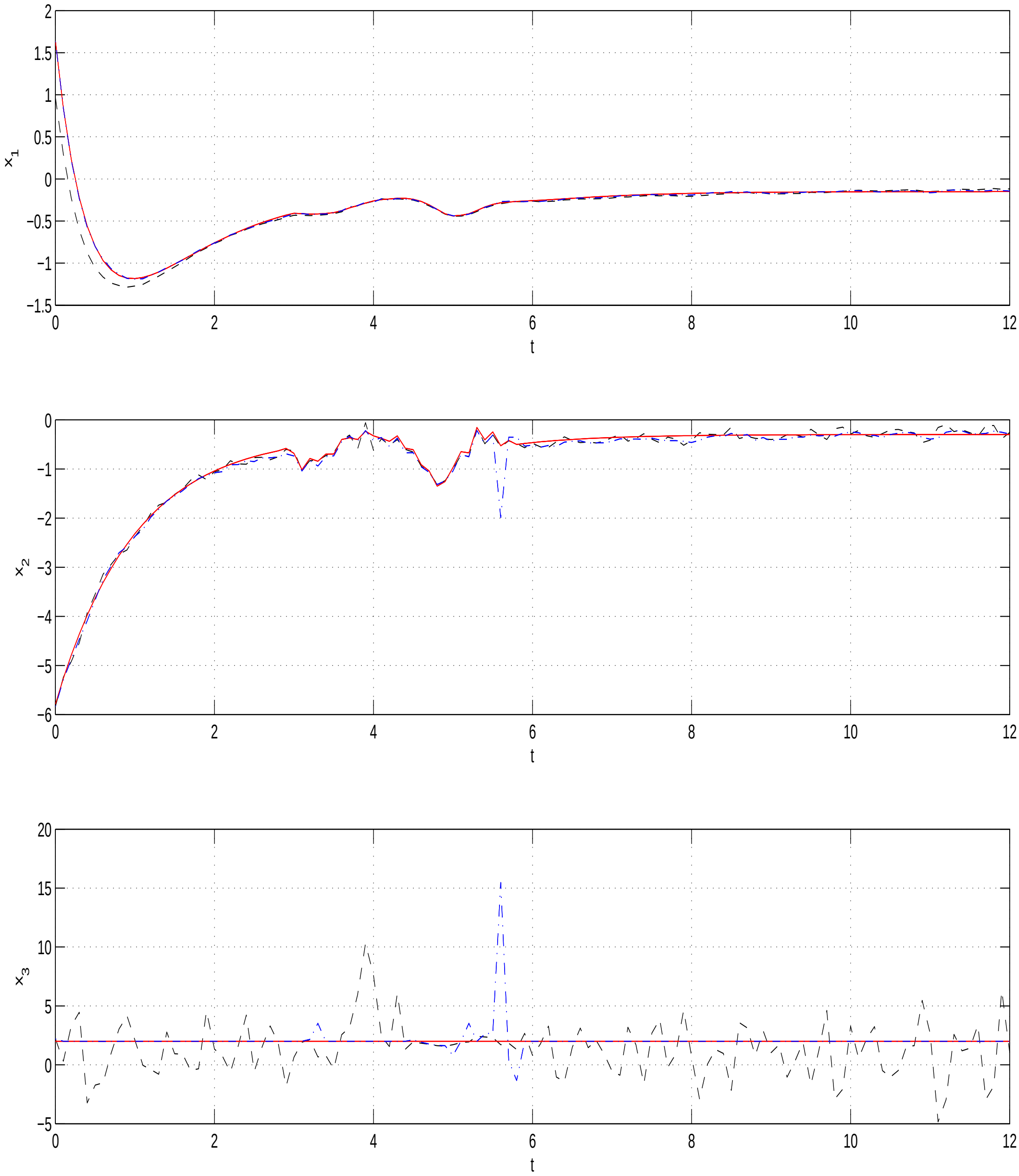


Fig. 2. Simulation results of example 1, case 2.

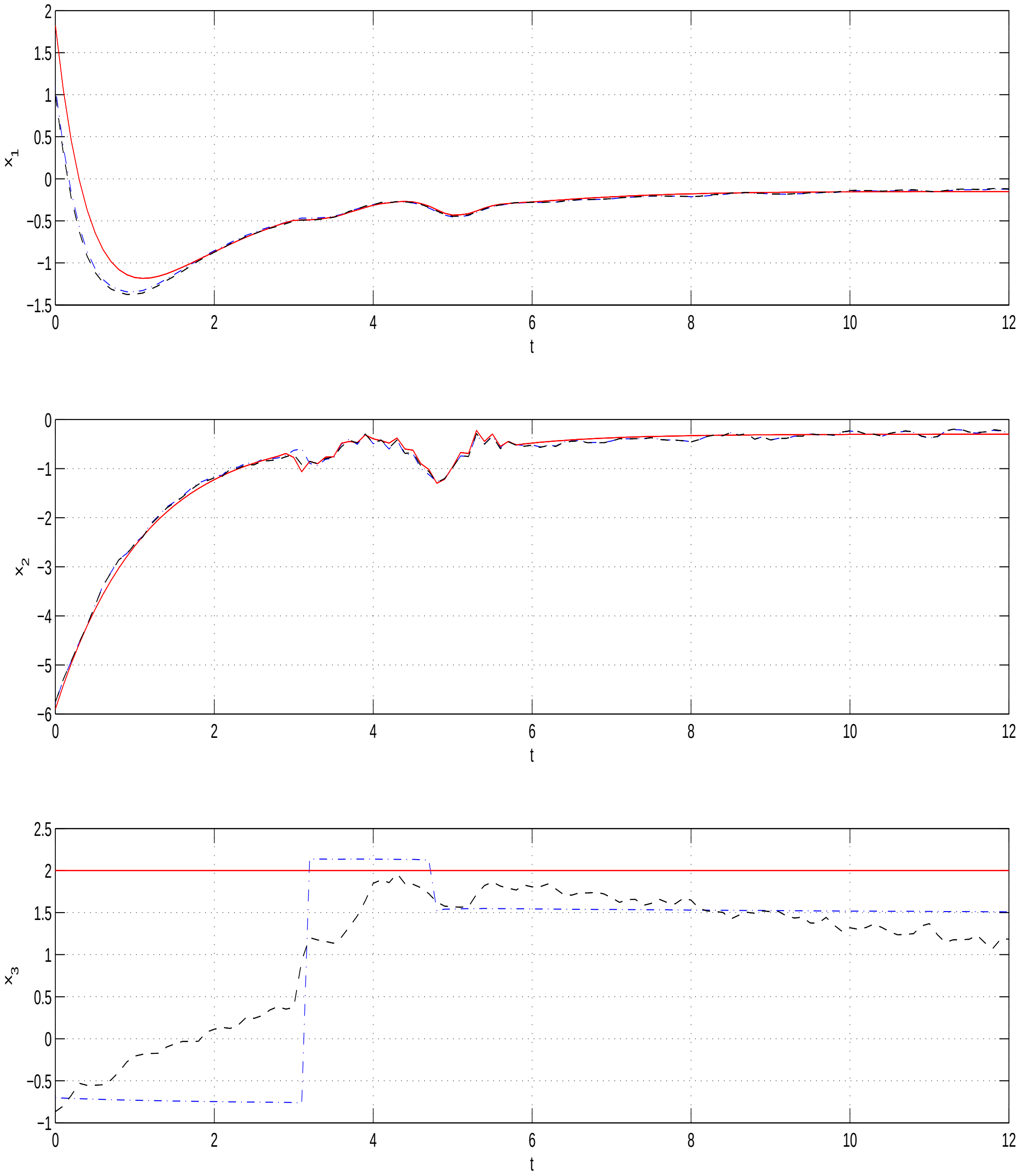


Fig. 3. Simulation results of example 1, case 3.

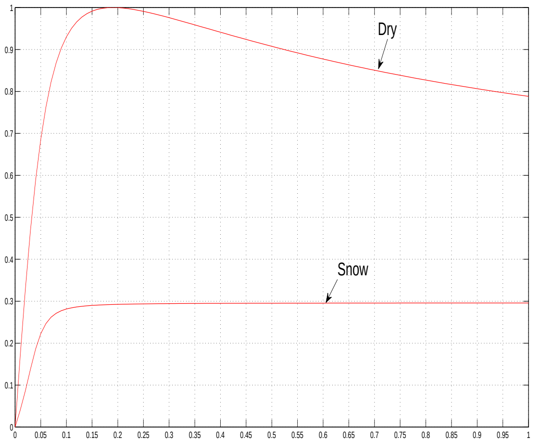


Fig. 4. Typical friction curves $\mu_1(\lambda)$.

the maximum friction is small and the right part of the curve is typically flatter.

We consider the tyre/road maximum friction coefficient θ as an augmented state. We get the augmented wheel slip dynamics

$$\dot{v} = -\frac{1}{m} F_z \mu(\lambda,\theta), \tag{38a}$$

$$\dot{\lambda} = -\frac{1}{v}\left(\frac{1}{m}(1-\lambda)+\frac{r^2}{J}\right) F_z \mu(\lambda,\theta) + \frac{1}{v}\frac{r}{J} T_b, \tag{38b}$$

$$\dot{\theta} = 0. \tag{38c}$$

In general, some reasonable constraints of parameters should be added into the MHE problem. Here the constraints are given as below:

$$v(t) \geq 1.0, \;\; 0 \leq \lambda(t) \leq 1, \;\; 0 \leq \theta(t) \leq 1.$$

and the system is discretized using the standard Euler method. We remark that the lower bound on $v(t)$ is conventional, since the ABS application will handle low speed as an exception where controllability is lost due to the singularity at $v = 0$ Johansen et al. (2003). In the example we choose the initial conditions $v(0) = 20, \lambda(0) = 0.01$, and the true values θ, B, C and E are given according to the different scenarios for dry asphalt and snow. In the simulation, choose the initial a priori estimates $\bar{v}(0) = 19, \bar{\lambda}(0) = 0, \bar{\theta}(0) = 0.6$, $\bar{B}(0) = 12$, $\bar{C}(0) = 1.3$ and $\bar{E}(0) = 0$. The horizon is chosen as $N = 10$. The sampling interval $t_f = 0.01$ s, and Gaussian white noise with variance 0.2 rad/s is applied to the measurements. We choose $\alpha = 0.01$ and $\beta_i = 1$, and W_t is chosen according to (31) with $\delta = 0.1$.

- Case 1, for dry asphalt, the simulation result is shown in Figure 5.
- Case 2, for snow, the simulation result is shown in Figure 6.

We observe that although the estimate of the single parameter θ converges to an accurate estimate, the observer still fails in estimating the velocity and wheel slip in Case 2 due to the inaccurate fixed values of B, C and E used in the model. Hence, there is a potential benefit of estimating also B, C and E.

Next, it is assumed that the parameters B, C, E, θ are unknown. With this parameterization one has to expect that the model will be over-parameterized such that the persistence of excitation condition (and uniform observability) will not hold. This challenging parameterization is chosen in order to illustrate the power of the proposed method, and in particular that the algorithm will accurately detect the excitation level of the data at any time and adapt the weights accordingly when using W_t defined by (31). Therefore, the proposed MHE algorithm is applied to the combined state and parameter estimation problem. Considering the parameters θ, E, C, B as augmented states, the states B, C, E, θ are added,

$$\dot{B} = 0, \; \dot{C} = 0, \; \dot{E} = 0, \; \dot{\theta} = 0. \tag{39}$$

The constraints on the states are given as

$$\begin{aligned} 1 \leq& v(t) \leq 30, \\ 0 \leq& \lambda(t) \leq 1, \\ 0 \leq& \theta(t) \leq 1, \\ 9 \leq& B(t) \leq 15.5, \\ 0 \leq& C(t) \leq 3, \\ -7.5 \leq& E(t) \leq 2. \end{aligned}$$

Here we choose $\alpha = 0.01$ and $\beta = 1$. W_t is chosen with $\delta = 0.8$.

- Case 3, for dry asphalt, the simulation result is shown in Figure 7.
- Case 4, for snow, the simulation result is shown in Figure 8.

In the figures, the true states are shown in solid lines and estimated states are shown in dash lines. It is interesting to observe that the estimation is robust and that the SVD thresholding effectively prevents the estimates of (B, C, E) from drifting and becoming highly incorrect when there is not much excitation or they are poorly observable. There are slightly more excitations in the dry asphalt case (stronger braking and higher wheel slips) and the adaptive weighting makes more attempts to estimate the parameters C and E in this case, compared to the show scenario. The parameter B is in both cases not excited, while good estimates of the most important vairiables λ and θ are achieved in both scenarios. Further results and discussions are given in Sui and Johansen (2010a).

5. DISCUSSION AND CONCLUSIONS

Theoretical and practical properties of a regularized nonlinear moving horizon observer were demonstrated in this paper. Although no convergence problems due to local minima were encountered in the simulation example in this paper, it is important to have in mind that the method will rely on a sufficiently accurate guess of the initial a priori estimate in cases when sub-optimal local minima exist.

The main feature of the proposed method is systematic handling of nonlinear systems that are neither uniformly observable, nor persistently excited, and may not be asymptotically stable. This is a typical situation with mixed parameter and state estimation with an augmented state space model. With the exception of the preliminary results in Sui and Johansen (2010b); Moraal and Grizzle (1995a), this is to the best of the authors knowledge, an important issue not studied in depth in any other nonlinear moving horizon observer. The examples show that the method can be successfully tuned and applied in challenging cases when the uniform observability and persistence of excitation conditions are not fulfilled, even with a highly over-parameterized model, without the need for careful a priori analysis of observability and persistence of excitation conditions. By proper scaling and

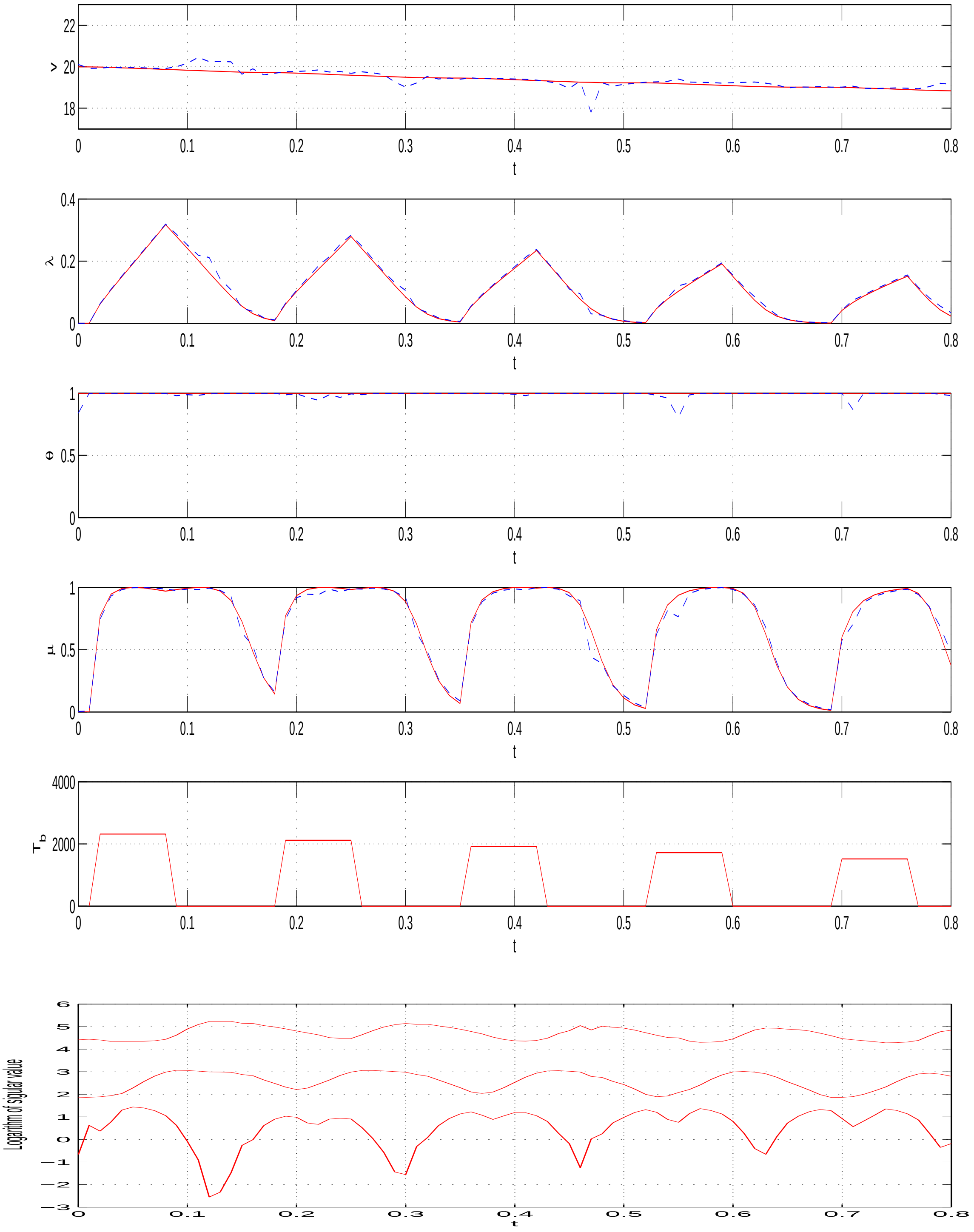


Fig. 5. Example 2: Simulation results of case 1, dry asphalt, one unknown parameter.

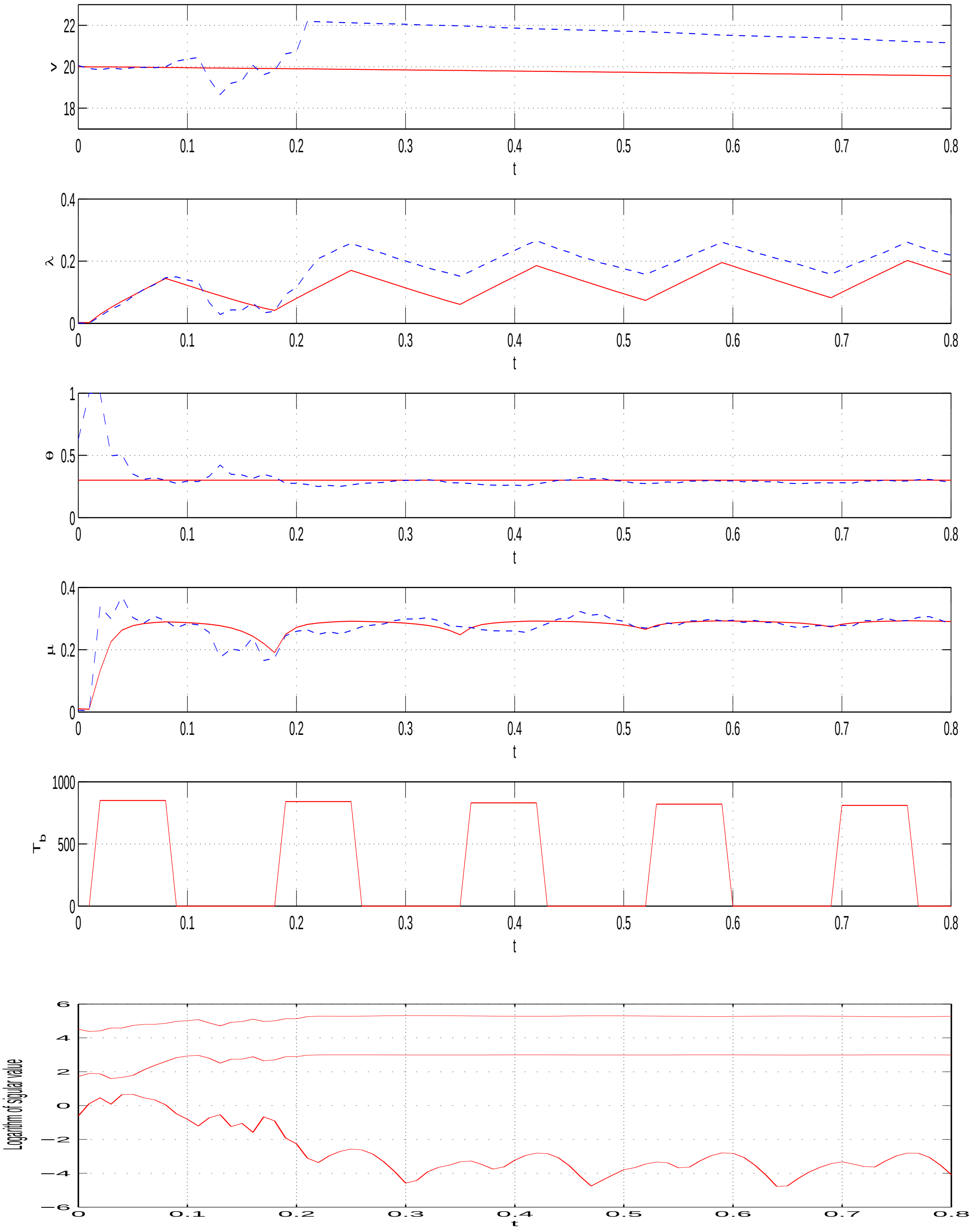


Fig. 6. Example 2: Simulation results of case 2, snow, one unknown parameter.

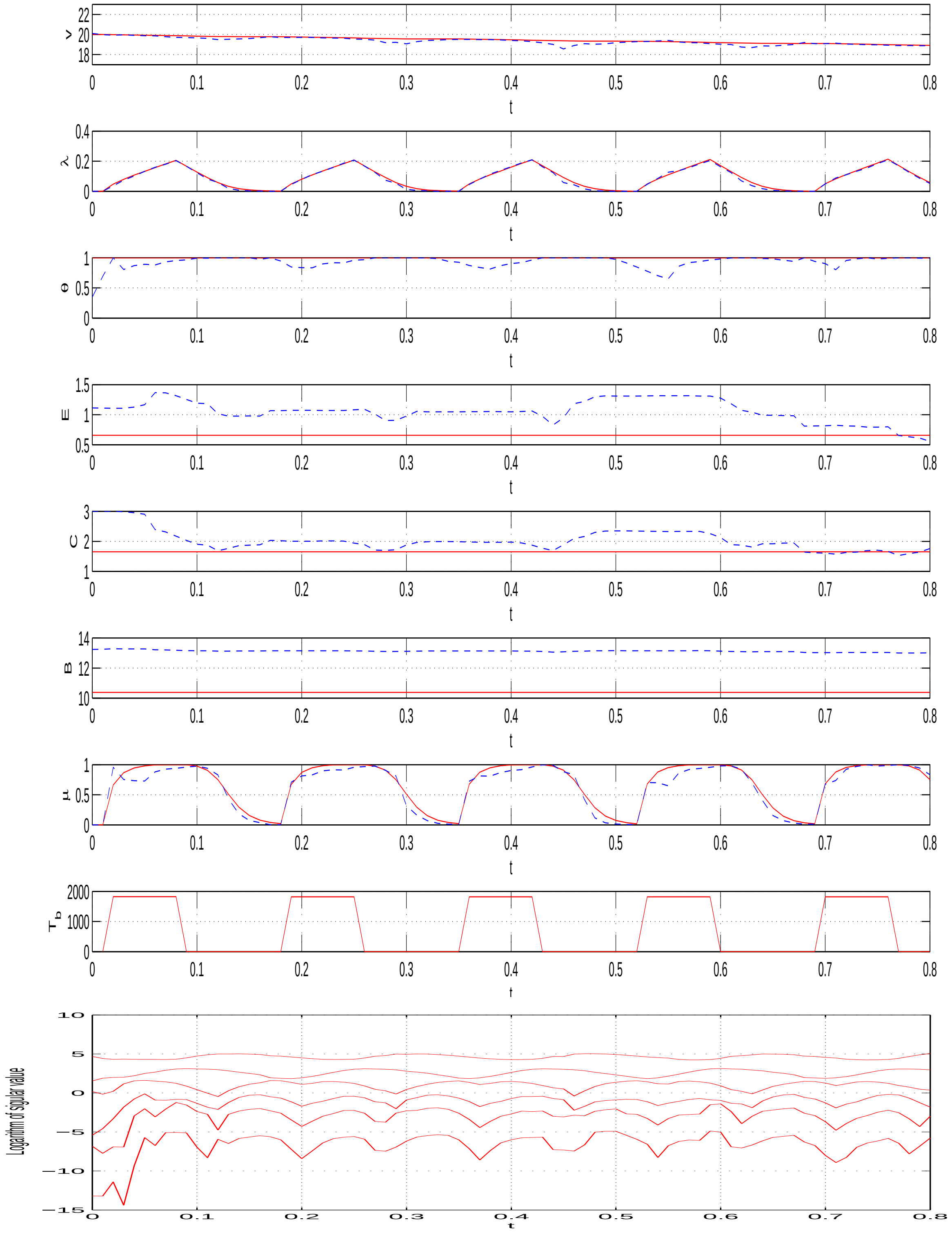


Fig. 7. Example 2: Simulation results of case 3, with dry asphalt road conditions and 4 unknown parameters.

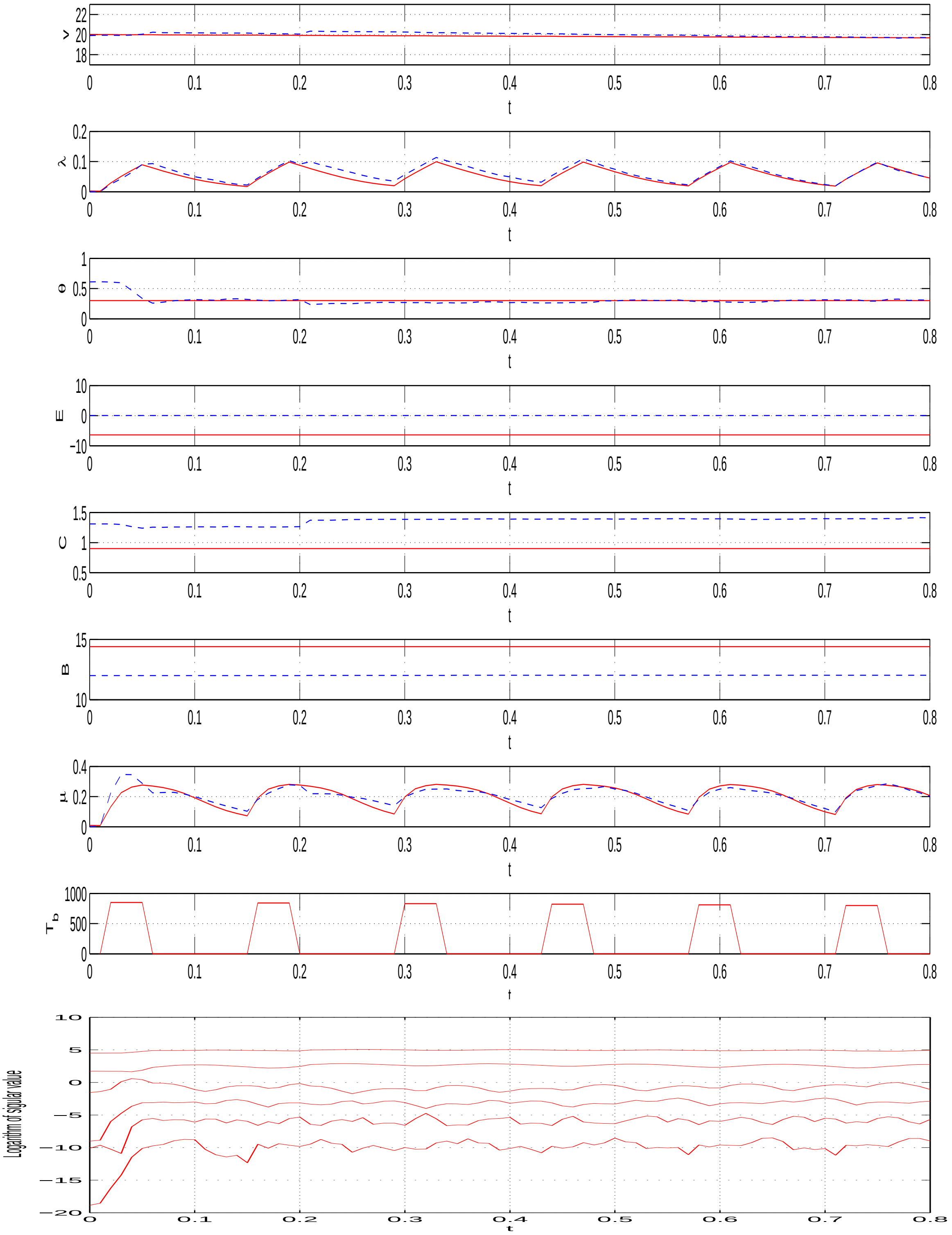


Fig. 8. Example 2: Simulation results of case 4, with snow road conditions and 4 unknown parameters.

tuning, the algorithm can automatically adapt to the level of excitation.

ACKNOWLEDGEMENTS

This research was mainly supported by the Research Council of Norway under the Strategic University Program on Computational Methods in Nonlinear Motion Control.

REFERENCES

Abraham, R., Marsden, J.E., and Ratiu, T. (1983). *Manifolds, Tensor Analysis, and Applications*. Springer-Verlag New York.

Alamir, M. (1999). Optimization based non-linear observers revisited. *Int. J. Control*, 72, 1204–1217.

Alessandri, A., Baglietto, M., and Battistelli, G. (2008). Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44, 1753–1765.

Alessandri, A., Baglietto, M., Parisini, T., and Zoppoli, R. (1999). A neural state estimator with bounded errors for nonlinear systems. *IEEE Transactions on Automatic Control*, 44, 2028 – 2042.

Alvarez, L., Yi, J., Horowitz, R., and Olmos, L. (2005). Dynamic friction model-based tire-road friction estimation and emergency braking control. *Transactions of the ASME*, 127, 22–32.

Biyik, E. and Arcak, M. (2006). A hybrid redesign of Newton observers in the absence of an exact discrete time model. *Systems and Control Letters*, 55, 429–436.

Bølviken, E., Acklam, P.J., Christophersen, N., and Størdal, J.M. (2001). Monte Carlo filters for non-linear state estimation. *Automatica*, 37, 177–183.

Burckhardt, M. (1993). Fahrwerktechnik: Radschlupfregelsysteme. *Vogel Verlag*.

Fiacco, A.V. (1983). *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press.

Gelb, A. (2002). *Applied Optimal Estimation*. MIT Press, 17 edition.

Glad, S.T. (1983). Observability and nonlinear dead beat observers. In *IEEE Conf. Decision and Control, San Antonio*, 800–802.

Golub, G.H. and van Loan, C.F. (1983). *Matrix computations*. Oxford University Press.

Grossman, W.D. (1999). Enhancing noise robustness in discrete-time nonlinear observers. In *Proc. American Control Conference, San Diego*, 3489–3494.

Haseltine, E.L. and Rawlings, J.B. (2005). Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Ind. Eng. Chem. Res*, 44, 2451–2460.

Johansen, T.A., Petersen, I., Kalkkuhl, J., and Ludemann, J. (2003). Gain-scheduled wheel slip control in automotive brake systems. *IEEE Trans. Control Systems Technology*, 11, 799–811.

Kandepu, R., Foss, B., and Imsland, L. (2008). Applying the unscented Kalman filter for nonlinear state estimation. *J. Process Control*, 18, 753–768.

Krstic, M., Kanellakopoulos, I., and Kokotovic, P. (1995). *Nonlinear and Adaptive Control Design*. Wiley and Sons.

Matuško, J., Petrović, I., and Perić, N. (2003). Application of extended kalman filter for road condition estimation. *Automatica*, 44, 59–65.

Moraal, P.E. and Grizzle, J.W. (1995a). Asymptotic observers for detectable and poorly observable systems. In *IEEE Conf. Decision and Control, New Orleans*, 109–114.

Moraal, P.E. and Grizzle, J.W. (1995b). Observer design for nonlinear systems with discrete-time measurement. *IEEE Transactions Automatic Control*, 40, 395–404.

Panteley, E., Loria, A., and Teel, A. (2001). Relaxed persistency of excitation for uniform asymptotic stability. *IEEE Trans. Automatic Control*, 46, 1874–1886.

Poloni, T., Rohal-Ilkiv, B., and Johansen, T.A. (2010). Damped one-mode vibration model state and parameter estimation via pre-filtered moving horizon observer. In *5th IFAC Symposium on Mechatronic Systems, Boston*.

Raff, T., Ebenbauer, C., Findeisen, R., and Allgöwer, F. (2005). Remarks on moving horizon state estimation with guaranteed convergence. In T. Meurer, K. Graichen, and E.D. Gilles (eds.), *Control and Obsever Design for Nonlinear Finite and Infinite Dimensional Systems*, 67–80. Springer-Verlag, Berlin.

Rao, C.V., Rawlings, J.B., and Mayne, D.Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximation. *IEEE Transactions Automatic Control*, 48, 246–258.

Rawlings, J.B. and Bakshi, B.R. (2006). Particle filtering and moving horizon estimation. *Computers and Chemical Engineering*, 30, 1529–1541.

Reif, K., Sonnemann, F., and Unbehauen, R. (1998). EKF-based nonlinear observer with a prescribed degree of stability. *Automatica*, 34, 1119–1123.

Reif, K. and Unbehauen, R. (1999). The extended Kalman filter as an exponential observer for nonlinear systems. *IEEE Transactions on Signal Processing*, 47, 2324–2328.

SAE (1992). Antilock brake review. *Tech. Rep. J2246, Society of Automative Engineerings, Warrendale PA.*

Sedoglavic, A. (2002). A probabilistic algorithm to test local algebraic observability in polynomial time. *J. Symbolic Computation*, 33, 735755.

Sui, D. and Johansen, T.A. (2010a). Moving horizon estimation for tire-road friction during braking. In *IEEE Multiconference of Systems and Control, Yokohama*.

Sui, D. and Johansen, T.A. (2010b). Regularized nonlinear moving horizon observer for detectable systems. In *IFAC NOLCOS, Bologna, Italy*.

Tautenhahn, U. (1994). On the asymptotic regularization of nonlinear ill-posed problems. *Inverse Problems*, 10, 1405–1418.

Tikhonov, A.N. and Arsenin, V.Y. (1977). *Solutions of Ill-posed Problems*. Wiley.

Zimmer, G. (1994). State observation by on-line minimization. *Int. J. Control*, 60, 595–606.

Comments – Remarks

Multi-level Programming for Designing Penalty Functions for MPC Controllers

Morten Hovd [*]

[*] *Engineering Cybernetics Department, Norwegian University of Technology and Science, N-7491 Trondheim, Norway (e-mail:morten.hovd@itk.ntnu.no)*

Abstract: The paper develops procedures for calculating the maximal values of the 1-norm and the infinity-norm of the Lagrangian multipliers for QP problems. These can be used in MPC design to design penalty functions for exact soft constraints, thus ensuring that the constraints will be violated only if there exists no solution where all constraints are satisfied.

1. INTRODUCTION

Model Predictive Control (MPC) has been a remarkable industrial success, with thousands of installations worldwide (Qin and Badgwell (2003)). A distinguishing feature of MPC controllers is the relative ease with which constraints in both states/outputs and inputs are handled. Nevertheless, such constraints may introduce many complexities that an industrial MPC controller need to address. There has been particular focus on the effect of hard output constraints on stability (Zafiriou and Marchal (1991); de Olivieira and Biegler (1994)) as well at the use of soft constraint formulations to ensure a feasible optimization problem, see (Scokaert and Rawlings (1999); Vada (2000); Hovd and Braatz (2001)) and references therein.

A fairly typical MPC formulation may be expressed as

$$\min_{u_0,u_1,\cdots,u_{N-1}} \sum_{k=0}^{N-1} \left(u_k^T R u_k + x_k^T Q x_k\right) \tag{1}$$

$$+x_N^T Q_f x_N$$

$$s.t.\ G_k x_k + H_k u_k \le b_k,\ k \in [0,\dots,N] \tag{2}$$

$$x_{k+1} = Ax_k + Bu_k,\ \ x_0 = given \tag{3}$$

$$Q \succeq 0,\ Q_f \succeq 0\ R \succ 0 \tag{4}$$

It is now fairly well known how ensure that this problem corresponds to a constrained infinite horizon problem, details may be found in e.g. (Rossiter (2003)).

For compactness of notation, we will in the following assume that the future states are eliminated from the MPC constraints, and that the resulting MPC problem is expressed as:

$$\min_u \ \left(0.5u^T Hu + x_0^T Fu\right) \tag{5}$$

$$x_0 = given$$

$$Gu \le W + Ex_0 \tag{6}$$

where $u = \left[u_0^T, u_1^T, \cdots, u_{N-1}^T\right]^T$.

The MPC formulation shown above is a socalled *hard constrained* problem. There may be initial states x_0 for which there exists no input sequence $\{u_k\}$ for which the constraints are fulfilled. In such a situation the optimization solver will find no solution, and what input to apply to the plant will not be defined. This is in general considered unacceptable in industrial practice. Practical MPC implementations therefore include some way of relaxing the constraints to ensure that the optimization problem is always feasible and the input to the plant is always well defined. There are several ways of doing this (Scokaert and Rawlings (1999)), one of the simplest and most common is to use *soft constraints*. When using soft constraints, the MPC formulation includes a variable in the constraint equations which allows relaxing (some of) the constraints, while the optimization cost function includes terms which penalize the constraint violation. Thus, with a soft constraint formulation, (1) is replaced by

$$\min_{u_0,u_1,\cdots,u_{N-1},\epsilon} \sum_{k=0}^{N-1} \left(u_k^T R u_k + x_k^T Q x_k\right) \tag{7}$$

$$+x_N^T Q_f x_N + g(\epsilon)$$

whereas the constraint equations (2) are modified as follows

$$G_k x_k + H_k u_k \le b_k + \epsilon_k,\ k \in [0,\dots,N] \tag{8}$$

$$\epsilon_k \ge 0$$

Remark: Naturally, we will soften constraints only if this is physically meaningful and safe to do so. Input constraints are typically hard constraints given by the physics of the process, and it would then be absurd to soften such constraints. However, many state/output constraints represent operational desirables (product quality specifications, comfort of operators, etc.), and violating such constraints for some period may be acceptable.

The penalty function $g(\epsilon)$ is typically given by

$$g(\epsilon) = c_\epsilon^T \epsilon + \epsilon^T Q_\epsilon \epsilon \tag{9}$$

[*] This paper is based on work submitted to the 2011 IFAC World Congress. The present version is intended for members of the NIL project groups.

A desirable property of the penalty function g is that it should ensure *exact* soft constraints, i.e., that the (hard) constraints will be fulfilled whenever this is possible. Only the linear terms in the penalty function determines whether the soft constraints are exact. The quadratic term should ensures that modified QP problem is a standard QP, but is otherwise held to be of less importance. Typically, the elements of Q_ϵ are therefore small, although one with some more careful choice of Q_ϵ may influence the tradeoff between constraint violations in different variables. This issue will not be pursued any further here. Ensuring that the soft constraints are exact is considered to be of primary importance, and we will therefore focus on the linear term in the penalty function in this paper.

A sufficiently high weight on the linear term in the penalty function will ensure that the soft constraints are exact. However, too high weight is generally not desirable, since that may lead to unnecessarily violent control action should the plant for some reason be outside of the (hard constrained) feasible region.

In the next section, we will briefly state existing criteria for ensuring the soft constraints are exact, and explain why this has generally been considered an intractable problem. In subsequent sections we will detail how to use multi-level programming to ensure that the soft constraints are exact. The resulting optimization problems will be mixed-integer linear programs (MI(L)P's). MILPs are non-convex, but very efficient solvers exist for this class of optimization problems, making it possible to solve problems of non-trivial size.

2. EXACT PENALTY FUNCTIONS IN MPC

Denote the cost function of the optimization problem, as shown in (1), by $f_h(u, x_0)$. The cost function for the soft constrained MPC in (7) is similarly denoted $f_s(u, x_0, \epsilon) = f_h(u, x_0) + g(\epsilon)$. We will here only consider the linear term in $g(\epsilon)$, as it is this term that determines whether the soft constraints are exact. We will assume that this linear term in $g(\epsilon)$ can be expressed in terms of an L_p norm of ϵ. For a vector a, the L_p-norm of a, denoted $\|a\|_p$, is given by

$$\|a\|_p = \left(\sum_i |a_i|^p\right)^{1/p} \tag{10}$$

In MPC, L_1-norm and L_∞-norm penalty functions are frequently used. The L_1-norm is the sum of the absolute values of the vector elements (and the slack variables in the MPC criterion are non-negative), whereas the L_∞-norm is the magnitude of the maximum vector element. These vector norms are therefore easily included in the function $g(\epsilon)$. The L_2-norm is the conventional Euclidian vector length. However, this is not commonly used for (exact) penalty functions, since the linear term in the penalty function then is not a linear function of the vector elements.

The L_1-norm penalty function increases the number of decision variables in the optimization problem by the number of constraints that are relaxed. In contrast, the L_∞-norm penalty function only increases the number of decision variables in the optimization problem by 1 - since the same slack variable can be used for all relaxed constraints. For this reason, L_∞-norm penalty functions are often preferred, although it is shown in (Rao et al. (1998)) that the addition of the L_1-norm optimization variables can be handled at virtually no additional computational cost if problem structure is utilized in the QP solver. On the other hand, the L_∞-norm can result in unexpected behaviour and poor performance if it is used to soften an output constraint for which there is an inverse response. In (Hovd and Braatz (2001)) it was shown how to minimize this problem by using time-dependent weights in the optimization criterion.

For a L_1-norm penalty function the linear term in $g(\epsilon)$ takes the form

$$g_{l,1}(\epsilon) = k\,[\,1\ \cdots\ 1\,]\begin{bmatrix}\epsilon_1\\ \vdots\\ \epsilon_m\end{bmatrix} \tag{11}$$

where k is a scalar and m is the number of constraints that are relaxed. For an L_∞-norm penalty function only a single slack variable is required, and the linear term in $g(\epsilon)$ therefore simplifies to

$$g_{l,\infty}(\epsilon) = k\epsilon \tag{12}$$

In standard optimization textbooks (e.g., Fletcher (1987)) we find conditions for ensuring that the soft constraints are exact. A L_p norm penalty function ensures that the soft constraints are exact, provided that the weight k on the linear term of the penalty function is larger than the maximal value of the dual norm of the Lagrangian multipliers of the corresponding hard-constrained optimization problem. The dual norm of an L_p norm is denoted by an index p_d, such that

$$\frac{1}{p} + \frac{1}{p_d} = 1 \tag{13}$$

Thus, the dual norm of the L_1-norm is the L_∞-norm, and *vice versa*, whereas the L_2-norm is its own dual.

This means that if we use an L_∞-norm penalty function and want to ensure that the soft constraints are exact, we must find the maximal value over the entire feasible region for the L_1-norm of the Lagrangian multipliers of the hard constrained problem. This is a non-convex optimization problem which in general has been considered intractable. In the next section we will briefly introduce multi-level programming, which we will use to reformulate the optimization of the norm of the Lagrangian multipliers into an MI(L)P problem.

3. MULTI-LEVEL PROGRAMMING

Multi-level programming is the generalization of the more common bi-level programming, where the constraints of the main optimization problem involve the solution of another (lower level) optimization problem.

$$\begin{aligned} \min_{y} \ & V_U(y,z) && (14)\\ \text{subjectto } & G_{UI}(y,z) \leq 0 \\ & G_{UE}(y,z) = 0 \\ & z = \arg\min_{z} V_L(y,z) \\ & \text{subjectto } G_{LI}(y,z) \leq 0 \\ & \qquad\qquad G_{LE}(y,z) = 0 \end{aligned}$$

Bi-level programming has been addressed since the 1970's, and the survey (Colson et al. (2005)) lists several contributions in the control area going back to the early 1980's, but due to the inherent difficulty of these problem formulations, they have been used rather sporadically since. However, with increasing availability of computing power, interest in these problems is returning (e.g., Kookos and Perkins (2003), Hovd and Kookos (2005), Jones and Morari (2009), Manum et al. (2009)).

3.1 Replacing lower-level problem with KKT conditions

In this paper, the lower-level optimization problem considered will be an MPC problem. These problems can be assumed to be convex and regular, admitting a unique optimal solution for everywhere in the feasible region for the problem. The lower-level optimization problem can therefore be replaced by its Karush-Kuhn-Tucker conditions (KKT), resulting in

$$\begin{aligned} \min_{y,z,\lambda,\nu} \ & V_U(y,z) && (15)\\ \text{subjectto } & G_{UI}(y,z) \leq 0 && (16)\\ & G_{UE}(y,z) = 0 && (17)\\ & \lambda \geq 0 && (18)\\ & G_{LI}(y,z) \leq 0 && (19)\\ & G_{LE}(y,z) = 0 && (20)\\ & \lambda \times G_L(y,z) = 0 && (21)\\ & \nabla_z \mathcal{L}(y,z,\lambda,\nu) = 0 && (22) \end{aligned}$$

where the × symbol indicate that element k of the vector λ of Lagrangian multipliers multiply constraint equation k in the original lower-level constraints. $\mathcal{L}(y,z,\lambda) = V_L(y,z) + \lambda^T G_{LI}(y,z) + \nu^T G_{LE}(y,z)$ is the Lagrangian function of the lower-level problem. Notice that there are no nonnegativity constraints for the Lagrangian multipliers ν for the equality constraints.

3.2 Reformulating KKT conditions using binary variables

We apply the technique proposed by (Fortuny-Amat and McCarl (1981)) to reformulate the non-convex complementarity conditions 21 using binary variables s:

$$\begin{aligned} \min_{y,z,\lambda,s,\nu} \ & V_U(y,z) && (23)\\ \text{subjectto } & G_{UI}(y,z) \leq 0 && (24)\\ & G_{UE}(y,z) = 0 && (25)\\ & \lambda \geq 0 && (26)\\ & \lambda \leq Ms && (27)\\ & G_{LI}(y,z) \leq 0 && (28)\\ & G_{LE}(y,z) = 0 && (29)\\ & G_{LI}(y,z) \geq -M(1-s) && (30)\\ & \nabla_z \mathcal{L}(y,z,\lambda,\nu) = 0 && (31)\\ & s \in \{0,1\} && (32) \end{aligned}$$

where M is some sufficiently large scalar. The following section will detail how this solution approach can be used to maximize the norm of the Lagrangian multipliers, which is needed in MPC for the design of exact soft constraints.

4. MAXIMIZING THE NORM OF THE LAGRANGIAN MULTIPLIER VECTOR

We will here apply the techniques of the preceding section to the hard-constrained MPC problem in (1) - (3), to find the norm of the Lagrangian multiplier vector. First, the problem of maximizing the L_1-norm, required for making soft constraints exact for a penalty function using the L_∞ norm, is addressed. Thereafter, maximization of the L_∞-norm is addressed.

4.1 Maximizing the L_1-norm

Maximizing the L_1-norm of the Lagrangian multiplier vector for (5) - (6) can be done by solving

$$\begin{aligned} \max_{x_0} \ & \underline{1}^T \lambda && (33)\\ \text{subject to} & \\ & u = \arg\min_{u} \left(0.5u^T H u + x_0^T F u\right) && (34)\\ \text{subject to } & Gu \leq W + Ex_0 && (35) \end{aligned}$$

where $\underline{1}$ denotes a column vector of ones. In a slight contrast to ordinary bilevel programming, we see that the upper-level criterion here does not become well defined until the lower level optimization problem is replaced by its KKT conditions - only then do the Lagrangian multipliers appear explicitly in the problem.

$$\begin{aligned} \max_{x_0} \ & \underline{1}^T \lambda && (36)\\ \text{subject to} & \\ & \lambda \geq 0 && (37)\\ & \lambda \leq Ms && (38)\\ & Gu - W - Ex_0 \leq 0 && (39)\\ & Gu - W - Ex_0 \geq -M(1-s) && (40)\\ & Hu + F^T x_0 + G^T \lambda = 0 && (41)\\ & s \in \{0,1\} && (42) \end{aligned}$$

Constraints (39) are the constraints of the original MPC problem. The presence of these constraints mean that we do not have to calculate the feasible region explicitly. This is a major advantage, since the projection operation involved in calculating the feasible region can be computationally very demanding for large systems. However, although the KKT conditions for the MPC problem (the lower-level problem) uniquely determine the optimal u, they do not uniquely determine the Lagrangian multipliers λ. The direct inclusion of λ as free variables in the maximization will therefore result in unnecessarily large λ's - bounded only by M in (38). We are instead after *small* λ's that fulfill the KKT conditions for the MPC problem. To this end, we insert an additional minimization in the formulation above:

$$\max_{x_0,u,s} \ \underline{1}^T\lambda \tag{43}$$
subject to
$$Gu - W - Ex_0 \geq -M(1-s) \tag{44}$$
$$Gu - W - Ex_0 \leq 0 \tag{45}$$
$$\min_{\lambda} \ 0.5\lambda^T\lambda \tag{46}$$
subject to
$$\lambda \geq 0 \tag{47}$$
$$\lambda \leq Ms \tag{48}$$
$$Hu + F^Tx_0 + G^T\lambda = 0 \tag{49}$$
$$s \in \{0,1\} \tag{50}$$

Proceeding as before with replacing the lower-level optimization problem with its KKT condition, and expressing the complementarity conditions as binary variables, we arrive at

$$\max_{x_0,\lambda,u,s,\delta,\mu,\nu} \ \underline{1}^T\lambda \tag{51}$$
subject to
$$Gu - W - Ex_0 \geq -M(1-s) \tag{52}$$
$$Gu - W - Ex_0 \leq 0 \tag{53}$$
$$\lambda \leq Ms \tag{54}$$
$$\lambda \geq 0 \tag{55}$$
$$Hu + F^Tx_0 + G^T\lambda = 0 \tag{56}$$
$$\lambda + [\,-I \ \ I\,]\,\delta + G\mu = 0 \tag{57}$$
$$\delta \geq 0 \tag{58}$$
$$\delta \leq M\nu \tag{59}$$
$$\begin{bmatrix} -\lambda \\ \lambda - Ms \end{bmatrix} \geq -M(1-\nu) \tag{60}$$
$$s \in \{0,1\} \tag{61}$$
$$\nu \in \{0,1\} \tag{62}$$

Comparing (54), (55) and (60), we observe that for these constraints to be consistent we need

$$\nu = \begin{bmatrix} 1-s \\ 1-s \end{bmatrix} \tag{63}$$

Thus, the final formulation becomes

$$\max_{x_0,\lambda,u,s,\delta,\mu} \ \underline{1}^T\lambda \tag{64}$$
subject to
$$Gu - W - Ex_0 \leq 0 \tag{65}$$
$$Gu - W - Ex_0 \geq -M(1-s) \tag{66}$$
$$\lambda \geq 0 \tag{67}$$
$$\lambda \leq Ms \tag{68}$$
$$Hu + F^Tx_0 + G^T\lambda = 0 \tag{69}$$
$$\lambda + [\,-I \ \ I\,]\,\delta + G\mu = 0 \tag{70}$$
$$\delta \geq 0 \tag{71}$$
$$\delta \leq M\begin{bmatrix} 1-s \\ 1-s \end{bmatrix} \tag{72}$$
$$s \in \{0,1\} \tag{73}$$

We observe that we have retained the complementarity conditions for the MPC problem. The final formulation therefore retains the optimal solution to the MPC problem, with the additional constraint that the Lagrangian multipliers found minimize the 2-norm among the Lagrangian multipliers that satisfy the KKT conditions for the MPC problem. The overall optimization formulation maximizes the corresponding 1-norm over the feasible region.

4.2 Maximizing the L_∞-norm

Finding the maximum of the L_∞-norm of the Lagrangian multipliers requires solving

$$\max_{x_0} \ \gamma \tag{74}$$
subject to
$$\lambda \leq \gamma \tag{75}$$
$$\min_{u} \ 0.5u^THu + x_0^TFu \tag{76}$$
subject to
$$Gu \leq W + Ex_0 \tag{77}$$

Replacing the lower optimization problem with its KKT conditions, we obtain

$$\max_{x_0,\gamma,\lambda,u,s} \ \gamma \tag{78}$$
subject to
$$\lambda \leq \gamma \tag{79}$$
$$\lambda \geq 0 \tag{80}$$
$$\lambda \leq Ms \tag{81}$$
$$Gu - W - Ex_0 \leq 0 \tag{82}$$
$$Gu - W - Ex_0 \geq -M(1-s) \tag{83}$$
$$Hu + F^Tx_0 + G^T\lambda = 0 \tag{84}$$
$$s \in \{0,1\} \tag{85}$$

However, as above we note that the Lagrangian multipliers are not uniquely determined by the KKT conditions for the lower optimization problem, and the γ we are after is the *smallest* γ for which there exists Lagrangian multipliers λ fulfilling the KKT conditions. Therefore, we again insert a lower-level optimization:

$$\max_{x_0,u,s} \quad \gamma \tag{86}$$

$$\text{subject to}$$

$$Gu - W - Ex_0 \leq 0 \tag{87}$$

$$Gu - W - Ex_0 \geq -M(1-s) \tag{88}$$

$$\min_{\gamma,\lambda} \quad 0.5\gamma^2 \tag{89}$$

$$\text{subject to}$$

$$\lambda \leq \gamma \tag{90}$$

$$\lambda \geq 0 \tag{91}$$

$$\lambda \leq Ms \tag{92}$$

$$Hu + F^T x_0 + G^T \lambda = 0 \tag{93}$$

$$s \in \{0,1\} \tag{94}$$

Note that although the optimum of the inserted lower-level optimization problem does not necessarily uniquely define the optimal value of the λ's, it does uniquely define the optimal value of γ, which we are after. Replacing the lower-level optimization problem with the corresponding KKT-conditions, and expressing the the complementarity conditions with binary variables, we obtain

$$\max_{x_0,\gamma,\lambda,u,s,\delta,\mu,\nu} \gamma \tag{95}$$

$$\text{subject to}$$

$$Gu - W - Ex_0 \leq 0 \tag{96}$$

$$Gu - W - Ex_0 \geq -M(1-s) \tag{97}$$

$$\lambda \leq \gamma \tag{98}$$

$$\lambda \geq 0 \tag{99}$$

$$\lambda \leq Ms \tag{100}$$

$$Hu + F^T x_0 + G^T \lambda = 0 \tag{101}$$

$$\begin{bmatrix} 0 \\ \gamma \end{bmatrix} + \begin{bmatrix} I & -I & I \\ -\underline{1}^T & 0 & 0 \end{bmatrix} \delta \tag{102}$$

$$+ \begin{bmatrix} G \\ 0 \end{bmatrix} \mu = 0$$

$$\delta \geq 0 \tag{103}$$

$$\delta \leq M\nu \tag{104}$$

$$\begin{bmatrix} \lambda - \gamma \\ -\lambda \\ \lambda - Ms \end{bmatrix} \geq -M(1-\nu) \tag{105}$$

$$s \in \{0,1\} \tag{106}$$

$$\nu \in \{0,1\} \tag{107}$$

Comparing (99), (100) and (105), we conclude that the binary variable ν must be parameterized as

$$\nu = \begin{bmatrix} \nu_1 \\ 1-s \\ 1-s \end{bmatrix} \tag{108}$$

We thus arrive at the final formulation

$$\max_{x_0,\gamma,\lambda,u,s,\delta,\mu,\nu_1} \gamma \tag{109}$$

$$\text{subject to}$$

$$Gu - W - Ex_0 \leq 0 \tag{110}$$

$$Gu - W - Ex_0 \geq -M(1-s) \tag{111}$$

$$\lambda \leq \gamma \tag{112}$$

$$\lambda \geq 0 \tag{113}$$

$$\lambda \leq Ms \tag{114}$$

$$\lambda - \gamma \geq -M(1-\nu_1) \tag{115}$$

$$Hu + F^T x_0 + G^T \lambda = 0 \tag{116}$$

$$\begin{bmatrix} 0 \\ \gamma \end{bmatrix} + \begin{bmatrix} I & -I & I \\ -\underline{1}^T & 0 & 0 \end{bmatrix} \delta \tag{117}$$

$$+ \begin{bmatrix} G \\ 0 \end{bmatrix} \mu = 0$$

$$\delta \geq 0 \tag{118}$$

$$\delta \leq M \begin{bmatrix} \nu_1 \\ 1-s \\ 1-s \end{bmatrix} \tag{119}$$

$$s \in \{0,1\} \tag{120}$$

$$\nu_1 \in \{0,1\} \tag{121}$$

5. EXAMPLES

The proposed procedure will next be illustrated on two examples.

5.1 *Example 1*

This example taken from Hovd et al. (2009). The system is a double integrator, described by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}$$

with constraints

$$-1 \leq u_k \leq 1$$

$$\begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

The weight matrices used are $Q = I$ and $R = 1$, whereas the prediction horizon $N = 15$ is used, resulting in 58 constraints in the MPC formulation. Maximizing the 1-norm of the Lagrangian multipliers, we find that the maximum is achieved at $x = [-9 \quad 3]^T$. The feasible region and the point where the maximum is obtained are shown in Fig. 1. The corresponding value of $\|\lambda\|_1 = 950$. This value, and the location of the maximum is verified by solving the MPC problem at all vertices of the feasible region.

5.2 *Example 2*

This example is taken from Hovd and Braatz (2001). The discrete-time model is given by

$$A = \begin{bmatrix} 0.9280 & 0.0024 & -0.0031 & -0.0040 \\ 0.0415 & 0.9538 & 0.0119 & 0.0065 \\ -0.0521 & -0.0464 & 0.8957 & -0.0035 \\ -0.0686 & 0.0508 & 0.0318 & 0.9346 \end{bmatrix} \tag{122}$$

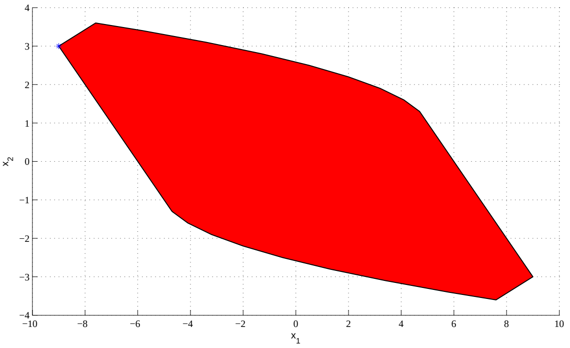


Fig. 1. Feasible region and point where the 1-norm of the Lagrangian multipliers is maximized.

$$B = \begin{bmatrix} 0.0000 & 0.3355 \\ 0.1826 & 0.0074 \\ 0.0902 & -0.0093 \\ 0.0418 & -0.0123 \end{bmatrix} \tag{123}$$

$$C = \begin{bmatrix} 0.0000 & 0.0000 & -0.0981 & 0.2687 \\ 0.0000 & 0.0000 & 0.0805 & 0.3271 \end{bmatrix} \tag{124}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{125}$$

and the constraints are given by

$$\begin{bmatrix} -1 \\ -1 \end{bmatrix} \leq Cx \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \qquad \begin{bmatrix} -1 \\ -1 \end{bmatrix} \leq u \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{126}$$

The state weight is given by $Q = C^T C$, the input weight is $R = I$, and a prediction horizon $N = 10$ is used. This problem has 120 constraints in the MPC formulation, and hence requires 120 binary variables in the MILP formulation for calculating $\|\lambda\|_1$. We find that the maximum value of the norm is achieved at $x = [\,25.5724\ 25.3546\ 9.7892\ -0.2448\,]^T$, and has the value $\|\lambda\|_1 = 38907$. For this example, calculating the feasible region is very computationally demanding, and the result has therefore not been verified by checking the vertices of the feasible region.

6. NUMERICAL ISSUES

Many MPC problems are symmetric in the constraints. In such cases, the Lagrangian multipliers at $x = z$ will be the same as the multipliers at $x = -z$. The problem will thus have (at least) two optima. Any global optimizer will try to discriminate between these optima, potentially resulting in substantial computational effort for no gain. This type of symmetry in the problem may be avoided by adding an additional constraint to the problem - in this work the constraint $u_0 > 0$ has been used for this purpose.

The parameter M should in theory be of little importance, it is just required to be sufficiently large, and any variable equal to M indicates that the value used is too small. However, numerical inaccuracies may be introduced by making M very large. In our (somewhat limited) experience, this inaccuracy is more likely to affect the value of the objective function rather than the location x in state space where the maximum is achieved. It is thus simple to check the value of the objective function by solving the MPC problem at x. Alternatively, one may use different values for M in different constraints, only increasing M for variables whose values are constrained by a too low value of M, and retaining a modest value for M for the remaining variables. This approach proved effective for Example 2 above.

7. CONCLUSIONS

In this paper, procedures for calculating the maximum values of the 1-norm and the infinity-norm of the Lagrangian multipliers of standard QP problems have been developed. The procedures are intended for designing penalty functions for soft constraints in MPC, to find the required weights for making the constraints *exact*. The calculation procedures are formulated as MILP problems, which are known in general to be NP-hard and thus very computationally demanding to solve. However, highly efficient solvers for MILP problems are available, and the number of constraints (and thus the number of integer variables in the MILP formulation) in Example 2 illustrates that the procedures can be applied to some problems of industrial relevance. In this work, the MILP solver in CPLEX is used.

REFERENCES

B. Colson, P. Marcotte, and G. Savard. Bilevel programming: A survey. *4OR, A Quarterly Journal of Operations Research*, pages 87–107, 2005.

N. M. C. de Olivieira and L. T. Biegler. Constraint handling and stability properties of model-predictive control. *AIChE Journal*, 40(7):1138–1155, July 1994.

R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, 2nd edition, 1987.

J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of Operational Research Society*, 32: 783–792, 1981.

M. Hovd and R. D. Braatz. Handling state and output constraints in MPC using time-dependent weights. In *Proceedings of the American Control Conference*, 2001.

M. Hovd and I. Kookos. Calculating dynamic disturbance rejection measures. In *IFAC World Congress*, Prague, Czech Republic, July 2005.

M. Hovd, F. Scibilia, J. M. Maciejowski, and S. Olaru. Verifying stability of approximate explicit mpc. In *Conference on Decision and Control*, Shanghai, China, 2009.

C. N. Jones and M. Morari. Approximate explicit mpc using bilevel optimization. In *European Control Conference*, Budapest, Hungary, 2009.

I. K.. Kookos and J. D. Perkins. On the efficient computation of disturbance rejection measures. *Computers and Chemical Engineering*, pages 95–99, 2003.

H. Manum, C. N. Jones, J. Löfberg, M. Morari, and S. Skogestad. Bilevel programming for analysis of low-complexity control of linear systems with constraints. In *IEEE Conference on Decision and Control*, pages 946–951, Shanghai, P. R. China, 2009.

S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, pages 733–764, 2003.

C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *J. of Optimization Theory and Applications*, 99(3):723–757, December 1998.

J. A. Rossiter. *Model-Based Predictive Control*. CRC Press, London, 2003.

P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE Journal*, 45(8): 1649–1659, August 1999.

J. Vada. *Prioritized Infeasibility Handling in Linear Model Predictive Control: Optimality and Efficiency.* PhD thesis, Engineering Cybernetics Department, NTNU, 2000.

E. Zafiriou and A. L. Marchal. Stability of SISO quadratic dynamic matrix control with hard output constraints. *AIChE Journal*, 37(10):1550–1560, October 1991.

Comments – Remarks

Relaxing PWQ Lyapunov function stability criteria for piecewise linear and piecewise affine systems

Morten Hovd [*] **Sorin Olaru** [**]

[*] *Engineering Cybernetics Department, Norwegian University of Science and Technology, N-7491 Trondheim, Norway*
[**] *Automatic Control Department, Supelec, 3 rue Joliot Curie, 91192 France*

Abstract: The calculation of piecewise quadratic (PWQ) Lyapunov functions is addressed, using the linear matrix inequality (LMI) approach proposed in Johansson and Rantzer (1998) for the stability analysis of PWL and PWA dynamics. Alternative LMI relaxations are proposed. These relaxations are shown to be effective compared to existing relaxations. *Copyright© 2011 Authors*

1. INTRODUCTION

Piecewise linear (PWL) and piecewise affine (PWA) systems appear often in practical control systems whenever piecewise linear components are encountered. Such components include dead-zone, saturation, relays and hysteresis Feng (2002). Model predictive control with constraints is also known to result in piecewise affine closed-loop dynamics. In this work, we address stability verification for PWL and PWA systems, using PWQ Lyapunov functions calculated using the LMI approach introduced by Johansson and Rantzer Johansson and Rantzer (1998); Rantzer and Johansson (2000). This approach has since been extended to discrete-time systems Feng (2002); Ferrari-Trecate et al. (2002).

For systems described by PWL (or PWA) models, the system description can be partitioned into different operating regions, with a given linear (or affine) system model for each region. Clearly, it makes no sense to impose the conventional stability criteria on Lyapunov function for a given system dynamic *outside* the region where that same system dynamic is valid. Still, the LMI formulations of the stability criteria must hold globally. To ameliorate this problem, Johansson and Rantzer Johansson and Rantzer (1998) introduce relaxations to the LMI formulation. These relaxations are such that the LMI formulation still ensures that the Lyapunov function stability criteria hold within the region of validity for each system dynamic, while relaxing the stability criteria outside that region.

The LMI relaxations introduced in Johansson and Rantzer (1998) have also been adopted by other authors Feng (2002); Ferrari-Trecate et al. (2002). In this work, alternative relaxations are proposed, and are shown to be effective. In addition, we illustrate how additional degrees of freedom may be introduced in the LMI formulation by sub-dividing the operating regions for the system dynamics, which in some cases enables the calculation of PWQ Lyapunov functions without requiring LMI relaxations. We can therefore conclude that a polyhedral pre-treatment of the regions of the original partition can decrease the conservativeness of the stability analysis based on semi-definite programming techniques.

2. SYSTEM MODELLING

Let $\{\mathcal{X}_i\}_{i\in I} \subseteq \mathbf{R}^n$ be a partitioning of the state space into non-overlapping (possibly unbounded) polyhedral cells, with I being the index set of the cells. The system dynamics are given by

$$x_{k+1} = A_i x_k + a_i \text{ for} x_k \in \mathcal{X}_i \tag{1}$$

The index set I is partitioned into two subsets, I_0 representing the polyhedral cells containing the origin, and I_1 representing all other cells. It is assumed that $a_i = 0\ \forall i \in I_0$, which clearly is a prerequisite for the origin to be an equilibrium point of the system. Further, when considering PWL systems, it is obviously assumed that $a_i = 0\ \forall i \in I$. Each region $\mathcal{X}_i$ is defined by the linear inequalities

$$E_i x \geq e_i \tag{2}$$

For PWL systems, this essentially completes the description of the system dynamics. However, for PWA systems we will find it convenient to introduce some extra notation to simplify the system description. This is done in the next subsection.

2.1 Simplifying the system description for PWA systems

Following Johansson and Rantzer (1998), we simplify notation for the description of PWA systems by introducing an auxiliary state, such that

$$\bar{x} = \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{3}$$

The definition of the polyhedral cell may then be expressed as

$$\bar{E}_i \bar{x} \geq 0 \text{ with } \bar{E}_i = [\, E_i \ -e_i \,]\,.$$

One may similarly express the system dynamics as linear in this enlarged (or 'lifted') state space. However, to be able to use standard Lyapunov function stability criteria on the enlarged state space, we must clearly be able to set the auxiliary state to zero near the origin. This implies that what linear dynamics to use will depend not only on which region the state is in at present, but also what region the state transits to at the next timestep. For the origin to be asymptotically stable, there must clearly be a positively invariant region around the origin. The assumption that all $\mathcal{X}_{i,i\in I_0}$ have linear dynamics will then also imply that there is a positively invariant region near the origin consisting entirely of cells with linear dynamics. A natural preliminary step in the stability analysis will therefore be to analyze stability for the PWL system described by the dynamics in $\bigcup_{i\in I_0}\mathcal{X}_i$. Assuming that this is proven stable (possibly using the techniques described later in this paper), a positively invariant subset of $\bigcup_{i\in I_0}\mathcal{X}_i$ may be extracted, using e.g. the techniques described in (ref. Hichem???). We will assume that this preliminary analysis has been done, and that the description of the polyhedral cells and the index set I_0 have been modified such that $\bigcup_{i\in I_0}\mathcal{X}_i$ is positively invariant.

Remark. The original motivation for looking into the stability of PWA systems came from stability analysis for MPC controllers, where the MPC design will result directly in I_0 consisting of a single, positively invariant cell Hovd and Olaru (2010).

Thus, (1) can be expressed as

$$x_{k+1} = \bar{A}_{i1}\bar{x} \tag{4}$$

$$\bar{A}_{i1} = \begin{bmatrix} A_i & a_i \\ 0 & 1 \end{bmatrix};\ x_{k+1} \in \mathrm{X}_j, j \in I_1 \tag{5}$$

$$x_{k+1} = \bar{A}_{i0}\bar{x} \tag{6}$$

$$\bar{A}_{i0} = \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix};\ x_{k+1} \in \mathrm{X}_j, j \in I_0 \tag{7}$$

3. LMI FORMULATION OF THE LYAPUNOV STABILITY CRITERIA

Following the development in Feng (2002), we consider PWQ Lyapunov functions. Thus, for each region of the state space we have

$$V_i(x) = \bar{x}^T\bar{P}_i\bar{x};\quad x \in \mathcal{X}_i \tag{8}$$

The closed loop system is then stable provided

$$\bar{P}_i = \bar{P}_i^T \tag{9}$$

$$V_i(x) > 0 \quad \forall x \in \mathcal{X}_i \tag{10}$$

$$V(x_k) > V(x_{k+1}) \tag{11}$$

where (11) should hold for all x_k and x_{k+1} that may occur according to the dynamics in (4 - 7). Let the index i identify the region in which the state is at time k, and the index j identify the region in which the state is at time $k+1$. All possible transitions between regions may then be identified with a set $\mathcal{S}$, where each element of $\mathcal{S}$ consists of a pair (i,j) that is consistent with the partition of the state space and the dynamics (4 - 7).

3.1 Expressing the stability conditions as an LMI

Equation (9) is fulfilled simply by defining $\bar{P}_i$ to be a symmetric (matrix valued) variable. The conditions (10) and (11) will be fulfilled if the following LMIs are fulfilled:

$$\bar{P}_i > 0;\quad \forall i \tag{12}$$

$$\bar{A}_i^T\bar{P}_j\bar{A}_i - \bar{P}_i < 0;\quad \forall (i,j) \in \mathcal{S} \tag{13}$$

where $\bar{A}_i$ refers to either $\bar{A}_{i1}$ or $\bar{A}_{i0}$, as appropriate. However, the condition (10) only has to hold for $x \in X_i$, whereas fulfilling (12) means that it is fulfilled for the entire extended state space (i.e., for all $\bar{x}$). Similarly, condition (11) only has to hold for $x_k \in X_{ij}$, where $X_{ij} \in X_i$ is the subregion of region X_i for which the state moves to X_j in the next timestep. Clearly, X_{ij} is a polyhedron, since both X_i and X_k are polyhedra and the closed loop dynamics is piecewise affine. In contrast, the fulfillment of (13) implies that the condition has to hold for the entire extended state space.

3.2 Relaxing the LMI conditions

Clearly, some way of relaxing the LMI conditions is desirable, to reduce the conservatism resulting from taking conditions on the Lyapunov function that must be fulfilled only in specific regions of the state space, and converting these conditions into LMIs that by default imply that the conditions are fulfilled for the entire state space. To this end, let us introduce the quadratic functions

$$f_i(x) = \bar{x}^T F_i\bar{x};\quad f_i(x) > 0, \forall x \in X_i \tag{14}$$

Note that $f_i(x) < 0$ is allowed for $x \notin X_i$. It is then easy to see that (10) is fulfilled, provided

$$\bar{P}_i - F_i > 0 \tag{15}$$

Similarly, we introduce functions

$$g_{ij}(x) = \bar{x}^T G_{ij}\bar{x};\quad g_{ij}(x) > 0, \forall x \in X_{ij} \tag{16}$$

Then, (11) is fulfilled provided

$$\bar{A}_i^T\bar{P}_j\bar{A}_i - \bar{P}_i + G_{ij} < 0 \tag{17}$$

Let the region X_i be defined by

$$E_i x \geq e_i \Leftrightarrow \bar{E}_i\bar{x} \geq 0 \tag{18}$$

where $\bar{E}_i = [\, E_i\ -e_i \,]$. Similarly, the region X_{ij} is defined by $\bar{E}_{ij}\bar{x} \geq 0$. The relaxations proposed in Rantzer and Johansson (2000) for continuous-time dynamics are then given by

$$F_i = \bar{E}_i^T U_i \bar{E}_i \tag{19}$$

$$G_{ij} = \bar{E}_{ij}^T W_{ij}\bar{E}_{ij} \tag{20}$$

where U_i and W_{ij} are symmetric, non-negative matrices. It appears that the same type of relaxations have been used for discrete-time dynamics by other authors, e.g. Feng (2002); Ferrari-Trecate et al. (2002). It should be clear that the power of the LMI-based technique for finding PWQ Lyapunov functions is strongly dependent on effective relaxations. Motivated by failure in finding PWQ Lyapunov functions, alternative relaxations have been sought. These are presented in the following sections.

4. A NOVEL LMI RELAXATION FOR PWA SYSTEMS

From (15) and (17) it is clear that we only need the relaxations ($f_i(x)$ and $g_{ij}(x)$) to be positive within specific polytopes. Outside those polytopes the functions may be negative, and may thereby make it easier to find a valid solution to the LMIs. The relaxations (19) and (20) do fulfill these requirements, and have proven effective for problems of modest size. However, the resulting relaxation functions are somewhat arbitrary, and there is a possibility that more careful specification of the functional form of the relaxations can be beneficial.

A reasonable choice for a relaxation would seam to be a concave quadratic function centered in a point contained in the interior of the polytope considered. Clearly, the quadratic function should be positive over the polytope. Thus, a reasonable relaxation function would appear to be

$$h(x) = (x - x_0)^T H(x - x_0) + c \tag{21}$$

where H is a symmetric negative definite matrix and c is a scalar that is sufficiently large to make h positive for the entire polytope. Clearly, the relaxation function reaches its maximum at (is 'centered on') $x = x_0$. The function $h(x)$ in (21) may equivalently be expressed as

$$h(x) = \bar{x}^T \begin{bmatrix} H & -Hx_0 \\ -x_0^T H & x_0^T H x_0 + c \end{bmatrix} \bar{x} = \bar{x}^T \left[\bar{H}_i + \bar{C}\right] \bar{x} \tag{22}$$

with

$$\bar{H} = \begin{bmatrix} H & -Hx_0 \\ -x_0^T H & x_0^T H x_0 \end{bmatrix}; \; \bar{C} = \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix} \tag{23}$$

Thus, $\bar{H}$ can be chosen as any symmetric negative definite matrix, provided we add the additional constraints that $h(x) \geq 0$ at all the vertices of the polytope in question.

Remark: From the explanation above, it follows that it is actually only H that should be negative definite. The Schur complement of $\bar{H}$ is

$$x_0^T H x_0 - x_0^T H (H)^{-1} H x_0 = 0$$

and thus $\bar{H}$ should be negative *semi*-definite. However, a simple reformulation of (22) gives

$$h(x) = \bar{x}^T \left[\bar{H} + \bar{C}\right] \bar{x} = \bar{x}^T \left[\left(\bar{H} + \bar{C}_a\right) + \bar{C}_b\right] \bar{x} \tag{24}$$

Thus, we can 'move part of the constant c into $\bar{H}$', to make $\bar{H} + \bar{C}_{i,a}$ semi-definite. The non-negativity at the vertices and the concavity of the relaxation function then ensures that the relaxation is of the correct form.

5. RELAXING PWL SYSTEM STABILITY CRITERIA

For piecewise linear systems defined on polytopic conical regions (with the vertex at the origin), we may relax the LMI-type stability criteria without introducing the auxiliary state. This follows since for such conical regions, $e_i = 0$ in 18, and the region is defined by $E_i x \geq 0$. The same LMI formulation as in (15) and (17) may therefore be used, without introducing the auxiliary state. This has been utilized previously in (ref. Lazar thesis?).

Remark: Actually, more general polyhedral regions may be analyzed in the same way, provided each region can be embedded inside (covered by) a polytopic cone with its vertex at the origin.

5.1 A novel relaxation for PWL systems defined on polytopic cones

In this subsection, a novel relaxation will be proposed, for a *pointed* polytopic conical region with its vertex at the origin. Such regions cannot cover an entire halfspace. Thus, there must exist a ray r_m originating at the origin, that is *inside* the polytopic cone, such that any ray originating at the origin which is orthogonal to r_m is fully *outside* the polytopic cone. Let $\mathcal{R}_p = \{r_{pj}\}, 1 \leq j \leq n_x + 1$ denote a set of mutually orthogonal rays that are also orthogonal to r_m.

From the halfplane description of the polytopic cone, $E_i x \geq 0$, the extreme rays r_{ej} can be identified. These are the rays where $n_x - 1$ of the inequalities $E_i x \geq 0$ are fulfilled with equality. Let $\mathcal{R}_e = \{r_{ej}\}$ be the set of extreme rays defining the polytopic conical region. Let v_m be a point on r_m, v_{pj} be a point on r_{pj}, and v_{ej} be a point on r_{ej}. We may then define a relaxation function for the polytopic cone as follows:

$$f(x) = x^T F x \tag{25}$$

$$v_m^T F v_m \geq 0 \tag{26}$$

$$v_{ej}^T F v_{ej} \geq 0 \forall r_{ej} \in \mathcal{R}_e \tag{27}$$

$$v_{pj}^T F v_{pj} \leq 0 \forall r_{pj} \in \mathcal{R}_p \tag{28}$$

Any F which fulfils the above constraints define a valid relaxation function. The constraints require F to have one positive eigenvalue, and $n_x - 1$ negative eigenvalues, and ensure that $f(x)$ is positive for any x inside the polytopic cone.

However, no quadratic relaxation can help for a region that covers an entire halfspace. Even if the origin is not inside the region, the relaxation function has to be positive along any ray starting at the origin and entering the halfspace region considered. Noting that any quadratic relaxation has to be symmetric about the origin, we find that the requirement that the relaxation function is positive within the polytopic cone then means that the relaxation function is positive everywhere. For a region that covers an entire halfspace, the relaxations therefore make the stability criteria *harder* to fulfill. For regions that cover an entire halfspace (or a large part thereof), it would therefore be desirable to be able to introduce additional degrees of freedom in the Lyapunov function optimization. This is addressed in the next subsection.

5.2 Sub-partitioning regions to introduce additional degrees of freedom in the Lyapunov function design

There is no fundamental reason why the partition of the state space used for the definition of the Lyapunov function should be identical to the partition resulting from the regions of validity of the different system dynamics. While it would seem reasonable to change the Lyapunov function where the system dynamics changes, there is no fundamental reason not to use a finer partitioning of the state space for Lyapunov function design than the partition resulting from the regions for the system dynamics. Obviously, the essence is that the Lyapunov function is positive (except at the origin), and has to decrease along the system trajectory in order to guarantee stability. There are (at least) two approaches to such finer partitioning of the state space:

(1) Partitioning based on the system dynamics, such that all states within a new sub-partition use the same number of timesteps to leave the original partition and enter the same of the other regions. To illustrate,

we may denote by P_{S143} the new sub-partitioned region which is within the original partition 1, and stays within original partition 1 until the states after 4 timesteps enter original partition 3.

(2) A more arbitrary approach, where the sub-partitioning of the original partitions have no clear connection to system dynamics.

Both approaches will be illustrated in the examples in the next section.

6. EXAMPLES OF PWQ LYAPUNOV FUNCTIONS USING RELAXED LMI CONDITIONS

The approach described above will next be illustrated on a few examples. In all examples, the calculations have been performed using Matlab and YALMIP Löfberg (2004), with the SeDuMi optimization solver.

6.1 Example 1

Consider first the following simple one-state example, which is included primarily to illustrate how the LMI relaxations work. The system is open loop unstable

$$x_{k+1} = 1.1x_k + u_k \tag{29}$$

and the input is constrained $-2 \le u_k \le 2$. In Hovd et al. (2009) this example was used to design an MPC controller, resulting in closed loop dynamics which can be described using three regions in the state space. The local closed loop dynamics for each of these regions are:

$$\bar{A}_0 = \begin{bmatrix} 0.011 & 0 \\ 0 & 0 \end{bmatrix} \text{ for } R_0 = \{-1.836 \le x_k \le 1.836\}$$

$$\bar{A}_1 = \begin{bmatrix} 1.1 & -2 \\ 0 & 1 \end{bmatrix} \text{ for } R_1 = \{1.836 \le x_k \le 19.75\}$$

$$\bar{A}_2 = \begin{bmatrix} 1.1 & 2 \\ 0 & 1 \end{bmatrix} \text{ for } R_2 = \{-19.75 \le x_k \le -1.836\}$$

Using LMI approach to find a PWQ Lyapunov function, we find that the system is stable in closed loop, although both region 1 and region 2 have unstable local dynamics. Simple inspection will show that the closed loop is stable for $-20 < x < 20$. The LMI approach can prove stability for nearly the same region of the state space - numerical problems occur when including states very close to ± 20.

The importance of relaxing the LMI stability criteria can be illustrated by Fig. 1. The figure shows the relaxed and unrelaxed solutions to the Lyapunov function stability criterion for states originating in region 1 and staying in region 1 at the next timestep. This corresponds to states $3.487 < x < 19.75$. It can be observed that although the unrelaxed solution fulfills the criterion (11) for the relevant part of the state space, the LMI condition (13) fails to hold globally. In contrast, the relaxed solution fulfills (17) over the entire state space

6.2 Example 2

This example is a slight modification of an example in Hovd and Braatz (2001). The system is described by

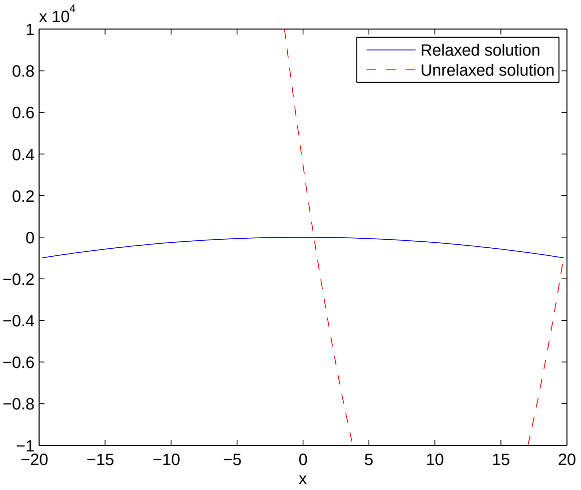


Fig. 1. Relaxed and unrelaxed Lyapunov function stability criteria for Example 1 (the LHS of (17) and (13), respectively), for states originating in region 1 and staying in region 1. The criteria are here evaluated for $\bar{x} = [x\ 1]^T$. The unrelaxed solution holds for the required region $3.487 < x < 19.75$, while the relaxed solution holds globally.

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

with

$$A = \begin{bmatrix} 2 & -1.45 & 0.35 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix};\ B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};\ C = [\,-1\ \ 0\ \ 2\,]$$

In Hovd and Olaru (2010) this example is used for the design of an approximate explicit MPC controller. The input constraints are given by $-2 \le u_k \le 2$, whereas the output constraints are $-1 \le y_k \le 1$. Further information about the MPC formulation can be found in Hovd and Olaru (2010).

The initial MPC design results in 199 regions. Through merging regions with identical affine dynamics, the number of regions can be reduced to 147. After this merging, there are 1478 transitions between regions (including 'transitions' where $i = j$). Using the LMI approach with the traditional LMI relaxations fails for this case. The LMI constraint (15) fails for 86 regions and the constraint (17) fails for 1353 transitions between regions. It would seem that the LMI analysis is of no help for this example, both for proving stability and for identifying regions of the state space where the control needs to be improved, given that the LMI conditions fail for such a high number of regions and transitions between regions. This result shows the limitations of the stability analysis applied. We therefore attempt to find a PWQ Lyapunov function using the alternative relaxation formulation. Of the 147 regions and 1478 transitions between regions, the relaxed solution now fails to fulfill the criteria only for 4 transitions between regions. Closer inspection shows that these 'transitions' actually represents the state staying in the same region, i.e., 'transitions' where $i = j$. Inspecting the four regions in question, it turns out that each of them has a fixed point inside the region, and thus the system is indeed not asymptotically stable. This happens even though the input in within each region corresponds to linear interpolation between the inputs at the vertices of the (simplicial) region, and the input at each vertex being the optimal input for an MPC formulation with guaranteed closed loop stability. With the new relaxations attention is thus effectively focused on the regions where the control was inadequate.

Having identified four regions where the dynamics do not fulfill the stability criteria, the MPC design is refined in these regions. This results in a total of 155 regions and 1768 transitions between regions. With this refined MPC design, the closed loop system is found to be stable.

6.3 Example 3.

A PWL system is defined by

$$A_1 = \begin{bmatrix} 2-k_1 & -(1+k_2) \\ 1 & 0 \end{bmatrix};\ x_1 \geq 0$$

$$A_2 = \begin{bmatrix} 2 & -(1+k_1+k_2) \\ 1 & 0 \end{bmatrix};\ x_1 < 0$$

This system may be stable or unstable depending on the values of the parameters k_1 and k_2. Here, the parameter values in Table 1 will be considered.

Table 1. Parameter sets for example 3.

Parameter set	k_1	k_2
1	1.2	-0.8
2	0.75	-0.4
3	1.0	-0.7
4	1.5	-0.98
5	0.5	-1.0
6	1.2	-1.2
7	1.5	-0.82
8	2.5	-0.8

It is easily verified by simulation that set 1-4 result in stable dynamics, whereas sets 5-8 result in unstable dynamics. However, we are seeking a Lyapunov function based proof of stability. Applying the LMI techniques for this purpose is unsuccessful, and the relaxations do not help - as should be expected from the discussion above - since each of the regions are half-spaces.

We therefore attempt sub-partitioning the original regions, as proposed in Section 5.2. Both approaches work well, proving stability for the stable cases, and failing to do so for the unstable cases. Depending on system dynamics, the first approach can result in a high number of sub-partitioned regions, unless one only defines new sub-partitions for parts of the original partition where the state leaves the original partition after a fixed maximum number of timesteps. For the second approach, the regions were sub-divided such that each new region had the same angle at its vertex at the origin. For this case some 'trial-and-error' was necessary in order to find how many new sub-partitions were required to be able to prove stability. For the stable cases, stability was proven when sub-partitioning the original regions into 2-8 new regions, whereas no stabilizing PWQ Lyapunov function could be found (as expected) for any of the unstable cases, even after subdividing each original region into 64 subregions.

Remark. Whereas the LMI-based approach is effective in finding PWQ Lyapunov functions (when they exist), it is not suitable for proving *instability.* There will always be the possibility that a Lyapunov function could be found with finer sub-partitioning of the regions, or by considering a different class of Lyapunov functions, like e.g., the polytopic Lyapunov functions studied in Blanchini and Miani (2008). Shortcut methods that can prove instability can therefore be useful. The most basic such test, would be to check that the state cannot 'blow up' while staying within a single region of the state space. For PWL systems defined on conical regions, such as in this example, a real-valued eigenvalue $\lambda_i > 1$ combined with an eigenvector that is inside the same conical region is such a simple proof of instability. Thus, we can easily conclude that that Case 5 is unstable for this example.

6.4 Example 4.

We here consider Example 3.2 in Ahmadi and Parrilo (2008). The system is piecewise linear and given by

$$x_{k+1} = \begin{cases} A_1 x_k & \mathit{for}\ x_k^T H x_k > 0 \\ A_2 x_k & \mathit{for}\ x_k^T H x_k \leq 0 \end{cases} \tag{30}$$

where $A_1 = \lambda e^{2A_1^C}$, $A_2 = \frac{1}{\lambda} e^{2A_2^C}$, $\lambda \geq 1$ is a scaling factor, and

$$A_1^C = \begin{bmatrix} -0.1 & 5 \\ -1 & -0.1 \end{bmatrix} \qquad A_2^C = \begin{bmatrix} -0.1 & 1 \\ -5 & -0.1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

We see that the state space is divided into four conical regions (which pairwise share the same dynamics), along the lines $x_1 - x_2 = 0$ and $x_1 + x_2 = 0$. In Ahmadi and Parrilo (2008), this example is used to illustrate the use of non-monotone Lyapunov functions, and it was found that the system could be proven stable (using both non-monotone Lyapunov functions with a 'horizon' of 2, and using unrelaxed PWQ Lyapunov functions) for $\lambda \in [1, 1.221)$. We sub-partition the original regions into 3 sub-regions, corresponding to parts of the original regions for which the state leaves the region after 1, 2, or more than 2 timesteps. The relaxations for PWL systems are then used for the sub-partitioned system (both the traditional and the new relaxations give the same result), and it is found that the system can be proven stable for $\lambda \in [1, 1.587)$. Although this quite nicely illustrates that sub-dividing regions and relaxing the LMI criteria are powerful tools, we do not claim that this constitutes a fair comparison between the methods studied here and the non-monotone Lyapunov function approach of Ahmadi and Parrilo (2008). It is quite conceivable that the proven region of stability in Ahmadi and Parrilo (2008) could be increased by allowing a longer 'horizon' over which decrease of the Lyapunov function is imposed.

6.5 Example 5.

A 3-state PWL system is addressed, in which the state space is partitioned into 8 polytopic conical regions along the coordinate system axes of the state space. The model is given in the Appendix. With the new relaxations the system is proven stable, whereas no Lyapunov function is found using the traditional relaxations (with the same partitioning of the state space). Testing a number of examples, we find several 3-state examples for which the new relaxations prove stability whereas the traditional relaxations fail to do so. In no case have we found the opposite to be true.

7. CONCLUSIONS

Novel relaxations for LMI-based stability verification of piecewise affine and piecewise linear systems have been proposed. A simple way of sub-dividing the original state space partitions to obtain additional degrees of freedom in the optimization formulation is also proposed. The idea of further subdividing the partitions has previously been proposed in Ohta and Yokohama (2010), but the procedure for chosing where to position the additional hyperplanes is much simpler here.

The usefulness of the proposed tools have been demonstrated by simulations.

REFERENCES

A. A. Ahmadi and P. A. Parrilo. Non-monotonic lyapunov functions for stability of discrete time nonlinear and switched systems. In *Conference on Decision and Control*, Cancun, Mexico, 2008.

F. Blanchini and S. Miani. *Set Theoretic Methods in Control*. Birkhauser, 2008.

G. Feng. Stability analysis of piecewise discrete-time linear systems. *IEEE Trans. Autom. Contr.*, 47:1108–1112, 2002.

G. Ferrari-Trecate, F. A. Cuzzola, D. Mignone, and M. Morari. Analysis of discrete-time piecewise affine and hybrid systems. *Automatica*, 38:2139–2146, 2002.

M. Hovd and R. D. Braatz. On the use of soft constraints in mpc controllers for plants with inverse response. In *Preprints Dycops*, pages 295–300, Jejudo Island, Korea, June 2001.

M. Hovd and S. Olaru. Piecewise quadratic lyapunov functions for stability verification of approximate explicit mpc. In *Submitted to the IEEE Multi-Conference on Systems and Control*, Yokohama, Japan, 2010.

M. Hovd, F. Scibilia, J. M. Maciejowski, and S. Olaru. Verifying stability of approximate explicit mpc. In *Conference on Decision and Control*, Shanghai, China, 2009.

M. Johansson and A. Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43:555–559, 1998.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

Y. Ohta and H. Yokohama. Stability analysis of uncertain piecewise linear systems using piecewise quadratic lyapunov functions. In *IEEE Multi-Conference on Systems and Control*, pages 2112–2117, 2010.

A. Rantzer and M. Johansson. Piecewise linear quadratic optimal control. *IEEE Transactions on Automatic Control*, 45:629–637, 2000.

APPENDIX. THE MODEL IN EXAMPLE 5.

Table A1. The model used in Example 5.

Region	Constraints	A-matrix
1	$x_3 < 0$ $x_2 < 0$ $x_1 < 0$	$A_1 = \begin{bmatrix} -0.94 & 0.22 & 0.36 \\ 0.48 & 0.12 & -0.18 \\ 0.40 & 0.04 & 0.30 \end{bmatrix}$
2	$x_3 > 0$ $x_2 < 0$ $x_1 < 0$	$A_2 = \begin{bmatrix} 0.62 & -0.36 & 0.22 \\ -0.24 & -0.24 & 0.70 \\ -0.46 & -0.58 & -0.90 \end{bmatrix}$
3	$x_3 < 0$ $x_2 > 0$ $x_1 < 0$	$A_3 = \begin{bmatrix} -0.82 & 0.66 & -0.22 \\ -0.50 & 0.68 & -0.94 \\ -0.92 & 0.94 & -0.74 \end{bmatrix}$
4	$x_3 > 0$ $x_2 > 0$ $x_1 < 0$	$A_4 = \begin{bmatrix} 0.52 & -0.08 & 0.56 \\ -0.52 & 0.78 & -0.32 \\ -0.92 & 0.04 & -0.46 \end{bmatrix}$
5	$x_3 < 0$ $x_2 < 0$ $x_1 > 0$	$A_5 = \begin{bmatrix} 0.82 & -0.28 & 0.42 \\ 0.50 & 0.64 & 0.30 \\ 0.16 & 0.64 & -0.14 \end{bmatrix}$
6	$x_3 > 0$ $x_2 < 0$ $x_1 > 0$	$A_6 = \begin{bmatrix} 0.22 & -0.80 & -1.00 \\ -0.86 & 0.74 & 0.80 \\ -0.48 & 0.78 & 0.74 \end{bmatrix}$
7	$x_3 < 0$ $x_2 > 0$ $x_1 > 0$	$A_7 = \begin{bmatrix} 0.94 & -0.64 & -0.44 \\ 0.98 & -0.72 & -0.36 \\ -0.36 & 0.34 & 0.02 \end{bmatrix}$
8	$x_3 > 0$ $x_2 > 0$ $x_1 > 0$	$A_8 = \begin{bmatrix} -0.32 & 0.68 & 0.58 \\ 0.54 & -0.90 & -0.36 \\ 0.72 & 0.92 & -0.22 \end{bmatrix}$

Comments – Remarks

Patchy approximate explicit model predictive control

Hoai Nam Nguyen [*] **Sorin Olaru** [**] **Morten Hovd** [***]

[*] *Automatic Control Department, Supelec, 3 rue Joliot Curie, 91192 France*
[**] *Automatic Control Department, Supelec, 3 rue Joliot Curie, 91192 France*
[***] *Engineering Cybernetics Department, Norwegian University of Science and Technology, N-7491 Trondheim, Norway*

Abstract: Multiparametric quadratic programming (MPQP) can be used to construct an off-line solution to constrained linear model predictive control. The result is a piecewise linear state feedback defined over polyhedral cells of the state space. However, with high dimensional problems, coding and implementation of this solution may be very burdensome for the available hardware, due to the high number of polyhedral cells in the state space partition. In this paper we provide an algorithm to find an approximate solution to MPQP, which is obtained by linear interpolation of the exact solution at the vertices of a feasible set and the solution of linear quadratic(LQ) problem. Based on a patchy control technique, we assure robust closed loop stability in the presence of additive measurement noise despite the presence of discontinuities at the switch between the regions in the state space partition.

Keywords: Model predictive control, Multiparametric programming, Smooth patchy Lyapunov function, Feasible set, Piecewise-linear Lyapunov function.

1. INTRODUCTION

Constrained linear model predictive control(MPC) is by now a well-known technique Mayne et al. (2000). This is an optimal control approach, which uses the model of the plant to predict the future evolution over a finite horizon. At the time instant t, using this prediction and the current state, an open loop optimal control problem (typically based on quadratic programming(QP)) is solved. Then, only the first element of the optimal control sequence is applied to the plant. At the time instant $t + 1$ the whole procedure is repeated with a new state vector supposed to be available from measurements or estimation. The implementation of MPC technique requires powerful on-line quadratic programming solvers, which may be very burdensome and represented an obstacle to wider application of MPC.

In Bemporad et al. (2002) it was shown that the constrained linear MPC is equivalent to a multiparametric quadratic program (MPQP), when the state plays the role of a vector of parameters for the optimization problem. The solution is a piecewise affine function of the state over a polyhedral partition of the state space and the MPC computation effort is moved off-line Olaru and Dumur (2005). However for high dimensional problems the explicit solution may be very complex due to the high number of polyhedral cells.

Several solutions have been proposed in the literature for the complexity reduction of explicit formulations leading to simpler polyhedral partition of the state space. In Johansen and Grancharova (2003) the state space partition in orthogonal hypercubes has been suggested, the suboptimal solution being computed as piecewise affine control law which minimizes the loss in the cost function over the hypercube cell. In the interior of each hypercube an approximate solution is obtained based on these data. If necessary, the hypercubes may be further partitioned into smaller hypercubes for achieving the desired accuracy.

A different approach is described in Bemporad and Filippi (2003) where an approximate solution to MPQP is found by relaxing the first order Karush-Kuhn-Tucker optimality conditions by some parameter ϵ. This represents a tuning parameter for the complexity of the controller. In Bemporad and Filippi (2006) the authors propose a method, which splits the state space into simplices, the optimal solution being computed only at the vertices of each simplex and a linear interpolation used inside each simplicial cell.

A variable structure linear state feedback controller, given in terms of the controls at the vertices of the polyhedral state constraint set, was presented in Gutman and Cwikel (1986). In Scibilia et al. (2009), the authors decompose a feasible set into two regions: the region where the linear quadratic regulator is feasible and the rest of the feasible set. The latter region is partitioned in simplices. Inside each simplex, the approximate explicit solution is obtained by linear interpolation of the exact solution at the vertices.

[⋆] This paper is based on work presented at the 2010 International Conference on Control, Automation and Systems 2010, in KINTEX, Gyeonggi-do, Korea. Original paper Copyright ©IEEE. This version is intended for members of the NIL project groups.

In Rossiter and Grieder (2005) an interpolation based control scheme was introduced, the main idea being the use two interpolations, the first one aims at diminishing the loss of optimality while the second one assures closed loop stability.

In this paper, we propose an alternative approach to compute the approximate explicit solution. The main idea is to unite two types of control laws. The first one is defined over a partition of the feasible set into simplicial cones and obtained by interpolating the exact solution at the vertices. The second one is a linear quadratic regulator, defined over a maximal invariant region. The overall result is a patchy control.

This paper is organized as follows. Section 2 introduces explicit MPC concepts for discrete-time linear time-invariant systems while in Section 3 an approximate solution is introduced. A simplicial controller is given Section 3.1 with an associated smooth patchy Lyapunov function described in Section 3.2. Section 3.3 is dedicated to the problem of a constructive patchy partition for robustness. The simulation results are evaluated in Section 4 before drawing the conclusions.

2. EXPLICIT MODEL PREDICTIVE CONTROL

2.1 Model predictive control

Consider the problem of regulating to the origin of discrete-time linear time-invariant system

$$x(t+1) = Ax(t) + Bu(t), \tag{1}$$

where $t \geq 0$ denote the current time, $x(t) \in R^n$ is the state, $u(t) \in R^m$ is the input, $A \in R^{n\times n}$ and $B \in R^{n\times m}$.

Both the control $u(t)$ and the state $x(t)$ are subject to polytopic constraints:

$$\begin{cases} u(t) \in U : U = \{u|H_u u \leq K_u\} \\ x(t) \in X : X = \{x|H_x x \leq K_x\} \end{cases} \forall t \geq 0 \tag{2}$$

where the matrices H_u, H_x and the vectors K_u, K_x are assumed to be constant with $K_u > 0$, $K_x > 0$ such that the origin is contained in the interior of U and X.

Assuming that $x(t)$ is available for measurement, a typical MPC algorithm solves the optimization problem

$$V_{\mathbf{u}}(x) = \min_{\mathbf{u}=(u_t,\ldots,u_{t+N-1})} J(\mathbf{u}, x(t)) \tag{3}$$

subject to $x_{t|t} = x(t)$ and

$$\begin{aligned} &u_{t+k} \in U,\ k = 0,\ldots,N-1 \\ &x_{t+k|t} \in X,\ k = 1,\ldots,N \\ &x_{t+N|t} \in \Omega \\ &x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k},\ k = 0,\ldots,N-1 \end{aligned}$$

where the objective function is given generally in th form of a finite horizon quadratic cost function:

$$J(\mathbf{u}, x(t)) = x_{t+N|t}^T P x_{t+N|t} + \\ + \sum_{k=0}^{N-1} (x_{t+k|t}^T Q x_{t+k|t} + u_{t+k}^T R u_{t+k})$$

In this formulation, N denote the prediction horizon, Q and R are weighting matrices, $Q \geq 0$, $R > 0$. It is assumed that the pair (A, B) is controllable and the pair $(\sqrt{Q}, A)$ is observable. The terminal cost P and terminal set Ω are classical ingredients for stability reinforcement Mayne et al. (2000). Practically P is obtained as a solution of Riccati equation:

$$A^T PA - P - A^T PB(B^T PB + R)^{-1}B^T PA + Q = 0 \tag{4}$$

At each time instant t the optimal control sequence $\mathbf{u}$ is computed, and only the first element of this sequence $u(t) = u_t$ is applied as control action to the plant. At the next time instant, the whole procedure is repeated with new state measurement.

2.2 Terminal set and feasible set

This section provides effective construction procedures for the terminal set Ω and the feasible set X_f depending on the prediction horizon N.

It is well-known that, Ω has to be an invariant region with respect to the system dynamics in closed loop with the linear quadratic regulator. This property assures the feasibility at all time Mayne et al. (2000). The feedback gain associated with this region is defined by:

$$K = -(B^T PB + R)^{-1}B^T PA \tag{5}$$

Denoting $A_c = A + BK$, the terminal set is define as

$$\Omega = \{x \in X : A_c^t x \in X, KA_c^t x \in U, t = 0,\ldots,\infty\} \tag{6}$$

The following theorem Gilbert and Tan (1991) gives simple conditions for finite determination of Ω.

Theorem 1. If the following assumptions hold: i) A_c is asymptotic stable, ii) X is bounded, iii) X has the origin as an interior point, then Ω is finitely determined.

Under this assumption the constructive procedure is used to compute the terminal set, as follows.

Procedure 1: Terminal set computation.

(1) Set $t = 0$, $H_t = H_x$, $K_t = K_x$ and $X_t = X$
(2) Set $X_t^1 = X_t$
(3) Compute a polytope

$$X_t^2 = \{x|H_t A_c x \leq K_t\} \cap \{x|H_u KA_c x \leq K_u\}$$

(4) Set X_t as an intersection:

$$X_t = X_t^1 \cap X_t^2$$

(5) If $X_t = X_t^1$ then stop and set $\Omega = X_t$. Else continue
(6) Set t = t+1, go to step 2

Finite determination properties Gilbert and Tan (1991) assure that the above procedure terminates in finite time and leads to the terminal set in form of a polytope:

$$\Omega = \{x : H_\omega x \leq K_\omega\}$$

Depending on the length of the prediction horizon N, the feasible set is a set of states, which can be steered to the terminal set Ω in N steps. It is apparent that, the number of constraints in the standard form of the optimization problem (3) will increase linearly with N and the complexity of the feasible set does not have an analytic dependence on N, thus placing a practical limitation on the choice of N. The simplest way to determine the feasible set is to reformulate MPC constraints (3) in terms of the input sequence $\mathbf{u}$ and current state $x(t)$:

$$G\mathbf{u} + Ex \leq W \tag{7}$$

where G and E are matrices and W is a vector of suitable dimensions. The linear inequalities (7) describe a polytope in the space R^{n+mN}. Then the feasible set X_f is obtained as the orthogonal projection of this polytope onto the state space.

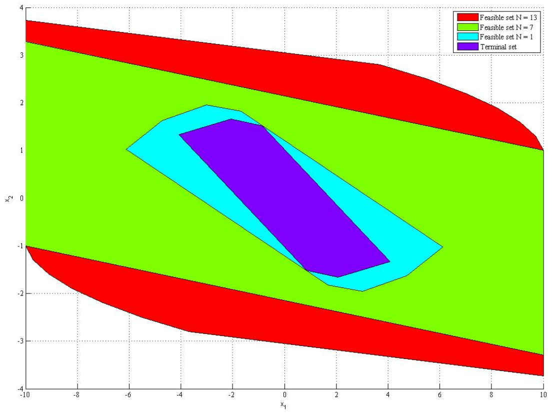


Fig. 1. Terminal set and feasible set

2.3 Explicit solution

With a simple change of coordinates $\mathbf{z} = \mathbf{u} + H^{-1}F^Tx$ quadratic problem (3) can by reformulated as

$$V_{\mathbf{z}}(x) = \min_{\mathbf{z}} \frac{1}{2}\mathbf{z}^T H\mathbf{z} \tag{8}$$

subject to $G\mathbf{z} \leq W + Sx$
where the matrices G and S can be found after simple matrix manipulation (see Bemporad et al. (2002) for details).

The current state vector x can be viewed as a vector of parameters, the reformulated problem can be considered as a multiparametric quadratic problem, since it is a quadratic problem in $\mathbf{z}$ parameterized by x. In parametric programming, the objective is to define the optimal solution $\mathbf{z}$ as an explicit function of the vector of parameters x. For the problem (8), the solution $\mathbf{z}$ has the following properties Bemporad et al. (2002).

Theorem 2. Consider the problem (8). The optimal solution $\mathbf{z}$ (and $\mathbf{u} = \mathbf{z} - H^{-1}F^Tx$) is a continuous piecewise affine function of state x and $V_{\mathbf{z}}(x)$ is a convex and continuous piecewise quadratic function. □

In the sequel any inequality constraint is said to be active for some x if it holds with equality at the optimum.

The following theorem gives an explicit representation of the optimal piecewise affine function of state.

Theorem 3. Consider the problem (8) and arbitrary fixed set of active constraints. Denote $\widetilde{G}$, $\widetilde{W}$ ans $\widetilde{S}$ the submatrices containing the corresponding rows of G, S and W. If the rows of $\widetilde{G}$ are linearly independent, the optimal solution and associated Lagrange multipliers are given by the affine functions:

$$\mathbf{z} = Lx + g \tag{9}$$

$$\lambda = L_1 x + g_1 \tag{10}$$

where

$$\begin{aligned} L &= H^{-1}\widetilde{G}^T(\widetilde{G}H^{-1}\widetilde{G}^T)\widetilde{S}, \\ L_1 &= -(\widetilde{G}H^{-1}\widetilde{G}^T)\widetilde{S}, \\ g &= H^{-1}\widetilde{G}^T(\widetilde{G}H^{-1}\widetilde{G}^T)\widetilde{W}, \\ g_1 &= -(\widetilde{G}H^{-1}\widetilde{G}^T)\widetilde{W}. \end{aligned} \tag{11}$$

In addition, the critical region $CR \in X_f$, where this solution is optimal is described by a polyhedron set

$$\begin{aligned} CR = \{x \in X_f |\ & G(Lx + g) \leq W + Sx, \\ & L_1 x + g_1 \geq 0\} \end{aligned} \tag{12}$$

One can find in the literature Bemporad et al. (2002) effective procedures for the partition of the feasible set X_f in critical regions as in (12). These regions have zero measure intersection and the union covers the feasible region X_f. The same reference Bemporad et al. (2002) provides solutions for degenerate cases when the linear independence condition is violated.

Thus, multiparametric quadratic programming algorithm gives the solution to MPC synthesis in the form of a piecewise affine function of state over polyhedral cells. The necessary on-line effort is reduced to identifying the region containing the current state and evaluate the associated affine feedback law.

3. APPROXIMATE EXPLICIT SOLUTION

The problem of reducing on-line computation, although addressed by MPQP, is not yet solved. In fact, for a high dimensional problem, the number of polyhedral cells may increase exponentially. Coding and implementation of this solution may be prohibitive for the available hardware.

Two types of local control laws will be used in this paper in order to approximate the exact explicit solution for (3). The first one is the LQ controller defined over the terminal set. The second one is a simplicial controller defined over the rest of the feasible set. Simplicial controller is used to steer the state to a point where the LQ controller is applicable. The control strategy results in a hybrid closed loop system.

3.1 Simplicial controller

Given a polytope $X_f \in R^n$, this polytope can be decomposed in a sequence of simplices X_f^k each formed by n vertices $x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}$ and the origin. These simplices have following properties:

- X_f^k has nonempty interior,
- $Int(X_f^k \cap X_f^l) = \emptyset$ if $k \neq l$,
- $\bigcup_k X_f^k = X_f$,

Denote by $X^{(k)} = (x_1^{(k)}\ x_2^{(k)}\ \ldots\ x_n^{(k)})$ the square matrix defined by the vertices generating X_f^k. Since X_f^k has nonempty interior, $X^{(k)}$ is invertible. Let $U^{(k)} = (u_1^{(k)}\ u_2^{(k)}\ \ldots\ u_n^{(k)})$ be the matrix defined by the optimal

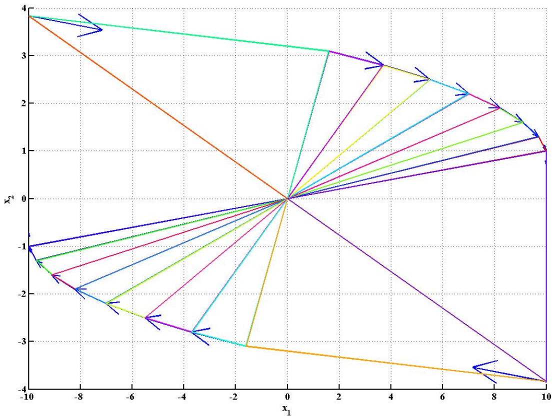


Fig. 2. Simplicial controller and vector field

control values at these vertices. For $x \in X_f^k$ consider the following linear gain K^k:

$$K^k = U^{(k)}(X^{(k)})^{-1} \tag{13}$$

Theorem 4. The piecewise linear control $u = K^k x$ is feasible for all $x \in X_f$.

Proof

For all $x \in X_f$ there exists an index k such that $x \in X_f^k$ and x can be expressed by convex combination of vertices of X_f^k: $x = \sum_{i=1}^n \alpha_i x_i^k$, which is equivalent with

$$x = X^{(k)}\alpha$$

and by consequence $\alpha = (X^{(k)})^{-1}x$, $\alpha \geq 0$ and $\sum_{i=1}^n \alpha_i \leq 1$.

For feasibility one has to ensure $\forall x \in X_f : H_u u \leq K_u$ and $x^+ = Ax + Bu \in X_f$.

With simple manipulations

$$\begin{aligned} H_u u &= H_u U^{(k)}(X^{(k)})^{-1}x = H_u U^{(k)}\alpha \\ &= \sum_{i=1}^n \alpha_i H_u u_i^k \leq \sum_{i=1}^n \alpha_i K_u \leq K_u \end{aligned}$$

and

$$\begin{aligned} x^+ &= Ax + Bu \\ &= A(X^{(k)})\alpha + BU^{(k)}\alpha = \sum_{i=1}^n \alpha(Ax_i^k + Bu_i^k) \end{aligned}$$

$\forall i = \overline{1,n}$ we have $Ax_i^k + Bu_i^k \in X_f$, it follows that $x^+ \in X_f$ □

3.2 Smooth patchy Lyapunov function

In the sequel we call C-set a compact set, containing the origin as an interior point.

The asymptotic stability of MPC guarantees that all solutions starting in X_f with the simplicial controller will reach the terminal set Ω in finite time. Inside the set Ω the LQ controller can b used to stabilize the system (1). That means the resulting switch-controller makes the system globally asymptotically stable in X_f. Indeed the origin is not locally attractive for Fillipov solutions. The reason of this is that there exist an admissible measurement noise, which makes every point on the boundary of Ω an equilibrium point of (1). So the system (1) with above discontinuous controller is very sensitive to measurement noise.

Patchy control Lyapunov functions (PCLFs) are, roughly speaking, objects consisting of several local control Lyapunov function (CLFs) the domain of which cover R^n and have certain weak invariance properties Bressan and Piccoli (2007), Goebel et al. (2009).

Definition: A smooth patchy Lyapunov function for the system (1) and the feasible set X_f consists of a finite countable set $Q \in N$ and a collection of functions V_q and sets Ω_q, $\Omega_q^{'}$, such that:

- Ω_q and $\Omega_q^{'}$ are families of nonempty open subsets of X_f such that $\Omega = \bigcup_{q \in Q} \Omega_q = \bigcup_{q \in Q} \Omega_q^{'}$ and $\overline{\Omega_q^{'}} \subset \Omega_q$, ($\overline{\Omega_q^{'}}$ denotes closure of $\Omega_q^{'}$),
- For each q, V_q is a smooth function defined on a neighborhood of $\Omega_q \setminus \cup_{r>q}\Omega_r^{'}$, such that for all $x \in \Omega_q \setminus \cup_{r>q}\Omega_r$ there exists $u_{q,x}$ such that $V_q(x)$ is a Lyapunov function.

We now return to the system (1) with the feasible set X_f and display a smooth patchy control Lyapunov function for it. For this purpose we need the following:

Definition: Given a C-set F, the Minkowski functional $\Psi_F(x)$ of F is defined as:

$$\Psi_F(x) = \inf\{\lambda \geq 0 | x \in \lambda F\} \tag{14}$$

The function $\Psi_F(x)$ is convex, positively homogeneous of order one. Furthermore it represents a norm for the C-set if and only if F is 0-symmetric Blanchini (1999).

The Minkowski functional Ψ_{X_f} defined over X_f for system (1) with the simplicial controller is positive definite and the property $\Psi_{X_f}(x^+) \leq \Psi_{X_f}(x)$ is guaranteed by the asymptotic stability of MPC for any $x \in X_f$. Furthermore, the asymptotic stability of MPC also guarantees that, there is no state-trajectory such that $\Psi_{X_f}(x)$ is constant on this trajectory.

Let $Q = \{1, 2\}$, $\Omega_2 = \Omega$, $\Omega_2^{'} = \gamma\Omega$, $\Omega_1 = X_f \setminus \Omega_2^{'}$ and $\Omega_1^{'} = X_f \setminus \Omega_2$, where γ is any positive number, $\gamma < 1$. For the simplicial controller one has Ψ_{X_f} as a Lyapunov function. In the same time, by the fact that the matrix $A + BK$ is stable, the associated dynamic is stable. In conclusion, there exist Lyapunov functions for each region Ω_q, $q = 1, 2$ taken independently.

With these elements, we construct a hybrid patchy controller for the system (1) as follows:

- If $x(t) \in \Omega_1^{'} \cap X_f^k$, then $u(t) = K^k x(t)$,
- If $x(t) \in \{\Omega_1 \cap \Omega_2 \cap X_f^k$, and $u(t-1) \neq Kx(t-1)$ then $u(t) = K^k x(t)$ else $u(t) = Kx(t)$,
- If $x(t) \in \Omega_2^{'}$, then $u(t) = Kx(t)$

It is clear that a switching from the simplicial controller to the LQ controller can occur when $x \in \Omega_2^{'}$, while a switch from the LQ controller to the simplicial controller can occur when $x \in \Omega_1^{'}$. This hysteresis-type controller prevents chattering between two controllers and lead the origin of the closed-loop system a global asymptotically stable, robust to measurement noise.

3.3 Constructive patchy partition for robustness

This section addresses the problem of finding a suitable γ for guaranteeing robustness in the presence of measurement noise .

Consider the following discrete-time linear time-invariant system

$$x(t+1) = Ax(t) + Bu(t) + w(t) \tag{15}$$

where w is a disturbance. Assume that the disturbance is persistent, additive and belong to a bounded set W. Assume also that the set W is a C-set. It is well known that if K is a linear matrix gain such that matrix $A_c = A+BK$ is strictly stable, then the trajectory of the system

$$x(t+1) = A_c x(t) + w(t) \tag{16}$$

will converge to a minimal robustly positively invariant(mRPI) set F_∞(see Bertsekas and Rhodes (1971) for details).

If the mRPI set F_∞ is contained in Ω, then there exists an unique, non-empty maximal robustly positively invariant(MRPI) set O_∞. This set O_∞ is a subset of the terminal set Ω as a consequence of the fact that Ω is positively invariant for the disturbance free dynamic.

The robust invariance property of the set O_∞ assures that, once the trajectory of the system enters O_∞, it will remain inside in this set and converge to the mRPI F_∞. Hence for the hybrid patchy controller one can choose the scaling factor $\gamma < 1$ such that $\Omega_2^{'} = \gamma\Omega \subset O_\infty$ or directly $\Omega_2^{'} = O_\infty$.

Below, we present a constructive procedure for computing the MRPI set O_∞.

Procedure 2: MRPI set computation.

(1) Set $t = 0$, $H_t = H_x$, $K_t = K_x$ and $X_t = X$,
(2) Set $X_t^1 = X_t$,
(3) Set $D_x = X_t \ominus W$ and $D_u = U \ominus KW$, where $\ominus$ denotes the Pontryagin difference,
(4) Compute a polytope
$X_t^2 = \{x : |H_{dx} A_c x \leq K_{dx}\} \cap \{x : |H_{du} K A_c x \leq K_{du}\}$
where H_{dx}, K_{dx} correspond to the H-representation of D_x and H_{du}, K_{du} give the H-representation of D_u
(5) Set X_t as an intersection:
$$X_t = X_t^1 \cap X_t^2$$
(6) If $X_t = X_t^1$ then stop and set $O_\infty = X_t$ else continue
(7) Set $t = t+1$ and go to step 2.

4. EXAMPLE

Consider the following discrete-time linear time-invariant system:

$$x(t+1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0.3 \end{pmatrix} u(t) \tag{17}$$

and the MPC problem with weighting matrices $Q = I$ and $R = 1$. The constraints are $-10 \leq x_1(t) \leq 10$, $-5 \leq x_2(t) \leq 5$ and $-1 \leq u(t) \leq 1$. The prediction horizon is $N = 13$.

Figure 3 shows the state space partition and approximate state trajectory of the system considered in the example, using the patchy approximate explicit model predictive control method. The number of regions is $n_p = 19$. Note that switch occurs when $x(t)$ is strictly inside Ω_2.

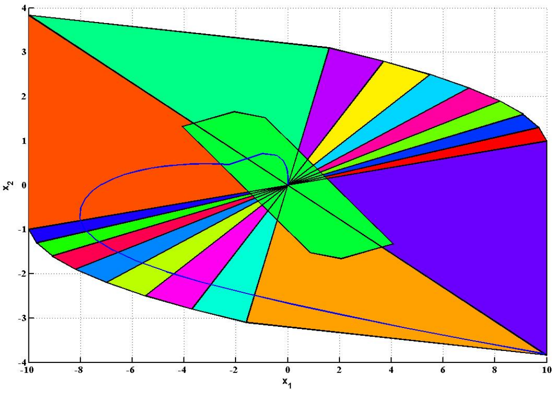


Fig. 3. State space partition and patchy approximate state trajectory. Number of regions $n_p = 19$. Switch occurs when $x(t)$ is strictly inside Ω_2

.

In comparison with this low complexity solution the Figure 4 shows the state space partition and approximate state trajectory of the system, using the method in Scibilia et al. (2009) thus showing the effectiveness of the complexity reduction (the number of regions is $n_p = 25$, and was shown to be one of the best solution available in the literature). The price to be paid by the gain in complexity can be found in the performance deterioration. Indeed, in order to perform a complete comparison of complexity vs. closed loop performance we present in Figure 5 the state space partition and state trajectory of the system, using the explicit model predictive control method. The number of regions in this case is $n_p = 129$ and is the exact solution

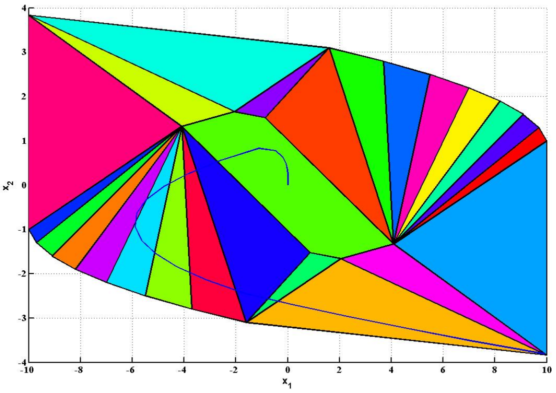


Fig. 4. State space partition and approximate state trajectory via Delaunay tessellation. Number of regions $n_p = 25$

corresponding to the optimal closed loop MPC trajectory.

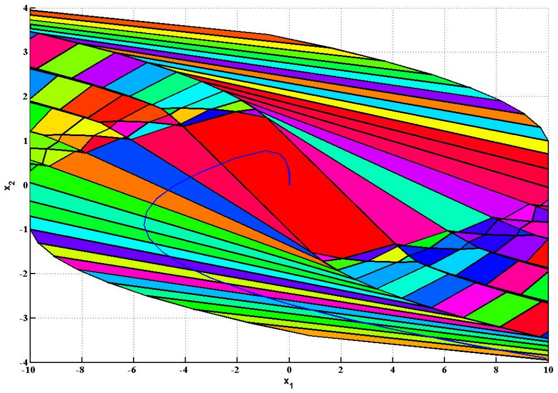


Fig. 5. State space partition and state trajectory via explicit model predictive control. Number of regions $n_p = 129$

With these three control laws we present in Figure 6 the results of the time-domain simulation. The three curves correspond to the explicit MPC method, the patchy approximate explicit MPC method and the method based on Delaunay tesselation Scibilia et al. (2009) respectively. Note that in the case of the patchy approximate explicit MPC, the control law is discontinuous but we do detain a proof of closed loop stability. In the case of the approximate explicit MPC via Delaunay tessellation, the control law is continuous but there is no a priori guarantee of stability, this being achieved by adding more vertices and subsequently increasing the complexity of the state space partition, or, alternatively, by a posteriori analysis of the resulting closed loop piecewise affine dynamic Hovd and Olaru (2010).

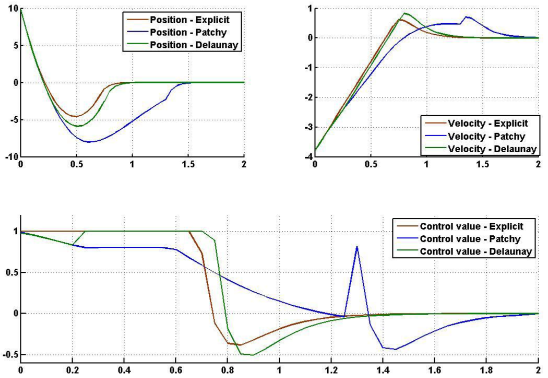


Fig. 6. State and control trajectories of the system considered in the example. The brown one is obtained by using the explicit MPC, the blue one is obtained by using the patchy approximate explicit MPC and the green one is obtained by using the approximate explicit MPC with Delaunay tessellation

5. CONCLUSION

This paper presented an alternative approach to compute an approximate solution for MPC. Based on the patchy technique, this approach united two types of controller: the simplicial and the LQ.

We point out the trade of between the complexity of the state space partition corresponding to the piecewise affine control laws and the closed loop performances. The explicit MPC controller is the reference from the performance point of view but very often turns to be impossibly complex for an effective on-line implementation. The approximate solution based on the interpolation between the feasible frontier and the frontier of the unconstrained LQ region provides a good compromise between the complexity and performance deterioration, these being related principally to the global continuity of the control law. By pushing the simplification to the ultimate bounds, we proposed an approximate solution with virtually simplest piecewise affine structure. It is worth noticing that the stability guarantee associated with this control law despite the presence of discontinuity, this same discontinuity being at the origin of the possible loss of performances in closed loop.

The simulation results show the effectiveness of the proposed methods.

REFERENCES

A. Bemporad and C. Filippi. Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *Journal of optimization theory and applications*, 117(1):9–38, 2003.

A. Bemporad and C. Filippi. An algorithm for approximate multiparametric convex programming. *Computational Optimization and Applications*, 35(1):87–108, 2006.

A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

D.P. Bertsekas and I.B. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7(2): 233–247, 1971.

F. Blanchini. Set invariance in control* 1. *Automatica*, 35 (11):1747–1767, 1999.

A. Bressan and B. Piccoli. *Introduction to the mathematical theory of control.* American Institute of Mathematical Sciences, 2007.

EG Gilbert and KT Tan. Linear systems with state and control constraints: the theory andapplication of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

R. Goebel, C. Prieur, and A.R. Teel. Smooth patchy control Lyapunov functions. *Automatica*, 45(3):675–683, 2009.

P.O. Gutman and M. Cwikel. Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states. *IEEE transactions on Automatic Control*, 31(4):373–376, 1986.

Morten Hovd and Sorin Olaru. Piecewise quadratic Lyapunov functions for stability verification of approximate explicit MPC. *Modeling, Identification and Control*, 31 (2):45–53, 2010. doi: 10.4173/mic.2010.2.1.

TA Johansen and A. Grancharova. Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48(5):810–815, 2003.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and PO Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

S. Olaru and D. Dumur. Avoiding Constraints Redundancy in Predictive Control Optimization Routines. *IEEE Transactions on Automatic Control*, 50(9):1459, 2005.

JA Rossiter and P. Grieder. Using interpolation to improve efficiency of multiparametric predictive control. *Automatica*, 41(4):637–643, 2005.

F. Scibilia, S. Olaru, and M. Hovd. Approximate explicit linear MPC via delaunay tessellation. *The European Control Conference 2009*, 2009.

Comments – Remarks

Robust Feasibility for Constrained Linear Systems with PWA Controllers [⋆]

F. Scibilia [*] **R. R. Bitmead** [**] **S. Olaru** [***] **M. Hovd** [*]

[*] *Department of Engineering Cybernetics, NTNU, O.S. Bragstads plass 2D, 7491 Trondheim, Norway (e-mail: francesco.scibilia@itk.ntnu.no, morten.hovd@itk.ntnu.no).*
[**] *Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0411, USA (e-mail: rbitmead@ucsd.edu)*
[***] *SUPELEC Systems Sciences (E3S) - Automatic Control Department, Gif-sur-Yvette, 91192, France, (e-mail: sorin.olaru@ieee.org)*

Abstract: Piecewise affine (PWA) feedback control laws represent an important class of controller for linear systems subject to linear constraints, with explicit Model Predictive Control approaches being probably the most popular techniques to obtain such control laws. These controllers are usually defined within a polyhedral set of initial states called the feasible set. In the presence of model mismatch, when the controller designed using the nominal model is applied to the real plant, the feasible set may lose its invariance property, resulting violation of constraints. Since the controller is only designed over the feasible set, there is also the technical problem that the control action is undefined if the state moves outside of the feasible set. This work proposes a tool to analyze how uncertainty in the model affects the piecewise affine control law computed using a nominal model. Given the linear system describing the plant and the piecewise affine control law, the algorithm that is presented considers a polytopic model uncertainty defined by the user and constructs the maximal robust feasible set, i.e. the largest subset of the feasible set which is guaranteed to be feasible for any model in the family of models described by the polytopic uncertainty.

Keywords: Polytopic model uncertainty, LTI systems, constraints, piecewise affine controllers.

1. INTRODUCTION

The concept of invariant sets has been shown to play an important role in the control and analysis of constrained systems (Blanchini (1999), Kerrigan and Maciejowski (2000), Gilbert and Tan (1991)). Given an autonomous dynamic system, a subset of the state space is said to be positively invariant if it has the property that, if it contains the system state at some time, then it will also contain it at all future times. The presence of constraints on the state variables defines an admissible set in the state space, i.e., the set of states that satisfies the constraints at the present time. Due to the system dynamics, in general, not all the trajectories originating from admissible initial states will remain in such a set. Conversely, for any initial condition which belongs to a positively invariant subset of the admissible domain, constraint violations are avoided at future times.

Such characterizations have relevant control applications. Consider the discrete-time linear time-invariant system

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$

$$y(t) = Cx(t), \tag{2}$$

and a linear state feedback control law that regulates the system to the origin

$$u(t) = Kx(t), \tag{3}$$

where $x \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}^m$ is the output vector and $u \in \mathbb{R}^r$ is the input vector, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{m\times n}$, K is a constant matrix gain. Suppose that it is required that the closed-loop system satisfies the output and input constraints

$$u_{min} \le u(t) \le u_{max}, \tag{4}$$

$$y_{min} \le y(t) \le y_{max}, \tag{5}$$

for all time instants $t \ge 0$, where y_{min}, y_{max} and u_{min}, u_{max} are constant vectors of suitable dimension. The closed-loop system represents an autonomous system, and the constraints can be easily rewritten as constraints on the state variables, giving the admissible domain in the state space. Then, starting from any initial condition inside a positively invariant subset of the admissible domain will guarantee convergence to the origin without violation of the constraints.

Among the families of positively invariant sets, the polyhedral sets are of particular importance because of their

[⋆] This work is a revised version of Scibilia et al. (2009a) (Copyright © 2009 IEEE) and is, to a large extent, extracted from Scibilia (2010). This version is intended for members of the NIL project groups.

flexibility and the fact that they are often natural expressions of physical constraints. The analysis of feasible positively invariant sets for linear autonomous systems was considered in Gilbert and Tan (1991), where the authors provide a systematic way to construct polyhedral invariant sets.

The concept of invariant sets extends naturally when a control input is present: a set is said control invariant if, for any initial state in the set, it is possible to keep the trajectory inside the set by means of an admissible feedback control law. Invariant sets are central in Model Predictive Control (MPC), the predominant control approach for systems subject to constraints. When linear models and linear constraints are considered, the stability of the nominal closed-loop system can be guaranteed by imposing positively invariant terminal set constraints (Mayne et al. (2000)). The constrained optimization in the MPC problem also characterizes the relative maximal feasible control invariant set (*feasible set* for short), i.e. the largest set of initial conditions such that the objective of the control is obtained without violating the constraints.

Posing the MPC problem as a (multi-)parametric optimization problem, the controller can be given as an explicitly defined continuous PWA function of the state over the feasible set (Bemporad et al. (2002), Tøndel et al. (2003)). Many solutions have also been proposed to obtain PWA controllers as approximations of the optimal explicit MPC controller when this is impractical (Scibilia et al. (2009b), Bemporad and Filippi (2003), Rossiter and Grieder (2005), Johansen and Grancharova (2003)). This explains the importance of PWA feedback state laws in the control of constrained linear systems. Indeed, in the following we will assume that the PWA controller considered is the result of some explicit MPC approach, since this is probably the most common way to obtain such controllers.

Linear models always involve approximations since all real plants are, to some extent, nonlinear, time-varying and distributed (Ikonen and Najim (2002), van den Boom and Haverkamp (2000)). Thus, any controller obtained by model-based design has to deal with the inherent model uncertainty. Model errors can also be introduced when the available model is of prohibitive order for real-time control and model reduction techniques are adopted to obtain a suitable low order model (Hovland et al. (2008), Johansen (2003)). Naturally, the ultimate goal of the control is to meet the performance requirements when implemented in the real plant. In order to meet such a goal, the control law should guarantee acceptable performance not only for the nominal plant model but also for a family of models which includes, by assumption, the real plant.

A popular paradigm used to cope with model uncertainty is polytopic model uncertainty. Polytopic model uncertainty constitutes a flexible and powerful tool to describe families of models and therefore also model uncertainties, and has been studied for many years (Boyd et al. (1994), van den Boom and Haverkamp (2000)). Robustness to model uncertainties in the MPC context has attracted great attention in the literature (Mayne et al. (2000)). An exhaustive review is out of the scope of this paper, it is instead interesting to focus on some relevant previous work. Polytopic uncertainties have been taken explicitly into consideration in the control design, resulting in robust MPC formulations where the constrained optimization problem is modified to a min-max problem which minimizes the worst-case value of the cost function, where the worst-case is taken over the set of uncertain models (Kothare et al. (1996), Kouvaritakis et al. (2000), Cuzzola et al. (2002), Mayne et al. (2000)). The same min-max approach has also been considered in explicit MPC (de la Peña et al. (2004), Grieder et al. (2003), Cychowski et al. (2005)). However, the solutions obtained are in general rather complex and conservative. In Pluymers et al. (2005a) a simpler and less conservative approach was proposed. The nominal MPC formulation is used, and robustness is defined in terms of satisfaction of input and output constraints for all possible uncertainty realization. An explicit implementation based on this approach was proposed in Rossiter et al. (2005).

Polytopic uncertainties are also useful in performance analysis of nominal controller with respect to possible model uncertainties. The work in Pluymers et al. (2005b) considers linear systems controlled by linear feedback controllers and subject to linear state and input constraints, and proposes an algorithm for constructing the largest set of initial condition which is guaranteed to be positively invariant for all possible models in a given polytopic uncertainty set.

This work proposes a tool to analyze how uncertainty on the model affects explicit MPC solutions computed using nominal models. In fact, it has been shown that MPC approaches possess a remarkable level of inherent robustness, and stability and good performance are maintained for sufficiently small uncertainties (Nicolao et al. (1996), Mayne et al. (2000)). However, when constrains are present, it is also necessary to ensure that the uncertainty does not cause any violation of constraints. Given a nominal linear system describing the plant and a PWA feedback control law designed accordingly, the algorithm that is presented considers a polytopic model uncertainty defined by the user and constructs the *maximal robust feasible set* [1] . This is the largest subset of the nominal feasible set which is guaranteed to generate feasible state trajectories for any model in the family of models described by the polytopic uncertainty. Therefore, for any initial condition within the maximal robust feasible set, the closed-loop system is guaranteed to be feasibly stable.

This can be useful, for example, in the case of control systems for plants which are time-varying due to wear, and subject to state and input constraints. In this case, designing a controller which accounts explicitly for the model mismatch may be unnecessarily conservative, decreasing the performance. Instead, a control design based on the nominal model may represent a better choice, resorting to the intrinsic robustness of the nominal controller to deal with the slowly progressive plant variation. Then, the results here presented can be used to investigate whether the constraints may be violated over time.

2. BASIC NOTIONS

2.1 Polytopic Uncertainty

Consider a linear system of the form (1). Model uncertainty can be expressed by saying that

[1] This notation may be in contrast with some work in the literature where a robust feasible set results from using a robust MPC design.

$$[A|B] \in \mathcal{M}, \tag{6}$$

where $\mathcal{M}$ is a polytope in the parameter space defined by its vertices

$$\left\{ \left[A^{(1)} | B^{(1)} \right], ..., \left[A^{(L)} | B^{(L)} \right] \right\}, \tag{7}$$

L is the number of vertices, as

$$\mathcal{M} \triangleq \mathtt{conv} \left(\left\{ \left[A^{(1)} | B^{(1)} \right], ..., \left[A^{(L)} | B^{(L)} \right] \right\} \right). \tag{8}$$

The function $\mathtt{conv}\,()$ refers to the convex hull.
This is equivalent to say that there exist L non-negative coefficients λ_l, $l = 1, ..., L$, $\sum_{l=1}^{L} \lambda_l = 1$, such that

$$[A|B] = \sum_{l=1}^{L} \lambda_l \left[A^{(l)} | B^{(l)} \right]. \tag{9}$$

The case $L = 1$ corresponds to the case of no model uncertainty.
Polytopic model uncertainty is a flexible tool to describe uncertainties. Consider for example $A \in \mathbb{R}^{1\times1}$ and $B = 0$. If the nominal value $A = a_n$ is known to describe the real value, a_r, with an accuracy ε: $a_n - \varepsilon \le a_r \le a_n + \varepsilon$, then $\mathcal{M} = \mathtt{conv}\left(\{a_n - \varepsilon, a_n + \varepsilon\}\right)$.

2.2 Definitions

Consider the polyhedral convex sets $\mathcal{U} \subset \mathbb{R}^r$ and $\mathcal{Y} \subset \mathbb{R}^m$ given as

$$\mathcal{U} = \{u \in \mathbb{R}^r | D_{\mathcal{U}} u \le d_{\mathcal{U}}\} \tag{10}$$

$$\mathcal{Y} = \{y \in \mathbb{R}^m | D_{\mathcal{Y}} y \le d_{\mathcal{Y}}\}. \tag{11}$$

The input and output constraints are of the form

$$u(t) \in \mathcal{U} \tag{12}$$

$$y(t) \in \mathcal{Y}, \tag{13}$$

for all $t > 0$. We will assume also that the origin is an interior point of the sets $\mathcal{U}$ and $\mathcal{Y}$.
Note that the constraints (4, 5) are special cases of (12-13).
Definition 1. (Feasible positive invariance) A positively invariant set $\mathbb{S}$ for a system of the form (1-2) in closed-loop with a particular feedback control law $u(t) = \Phi(x(t))$ is termed *feasible* with respect to constraints (12-13) if

$$\forall x(0) \in \mathbb{S} : u(t) \in \mathcal{U},\ y(t) \in \mathcal{Y} \text{ for } t \ge 0 \tag{14}$$

Definition 2. (Robustly feasible positive invariance) Given a positively invariant set $\mathbb{S}$ for a system of the form (1-2) in closed-loop with a particular feedback control law, $u(t) = \Phi(x(t))$, feasible with respect to constraints (12-13), a subset $\mathbb{S}_R \subseteq \mathbb{S}$ is said to be *robustly feasible* for the family of dynamics in an uncertainty set of form (8) if

$$\forall x(0) \in \mathbb{S}_R : u(t) \in \mathcal{U},\ y(t) \in \mathcal{Y} \text{ for } t \ge 0,\ \forall\, [A|B] \in \mathcal{M} \tag{15}$$

The set $\mathbb{S}_R$ is *maximal* if it also contains all the other robustly feasible sets.

Note that Definition 2 implies that for all $x(0) \in \mathbb{S}_R \subseteq \mathbb{S}$, the state evolution $x(t)$, for all $t > 0$, is contained within $\mathbb{S}$ for any time invariant $[A|B] \in \mathcal{M}$.

3. PROBLEM FORMULATION

Consider the problem of regulating to the origin a plant with a given nominal model of the form (1-2), such that constraints like (12-13) are satisfied. Assuming that the state is available for measurement, the regulation problem is solved by the finite horizon MPC

$$\min_{\mathbf{u}} \left\{ J\left(\mathbf{u}, x\left(t\right)\right) = \|x_N\|_P^2 + \sum_{k=0}^{N-1} \|x_k\|_Q^2 + \|u_k\|_R^2 \right\} \tag{16}$$

$$\begin{aligned} \text{s.t. } & x_0 = x\left(t\right), \\ & x_{k+1} = Ax_k + Bu_k \\ & y_k = Cx_k && k = 0, 1, ..., N \\ & y_k \in \mathcal{Y}, && k = 1, 2, ..., N \\ & u_k \in \mathcal{U}, && k = 0, 1, ..., N-1 \\ & x_N \in \Omega \end{aligned} \tag{17}$$

where: $\|x\|_Q^2 = x^T Q x$; x_k denotes the predicted state vector at time $t + k$ obtained by applying the k first elements of the input sequence $\mathbf{u} \triangleq [u_0, ..., u_{N-1}]$; N is the prediction horizon; $Q \succeq 0$ (positive semidefinite) and $R \succ 0$ (positive definite) are symmetric matrices corresponding to weights on state and input; P is the terminal cost matrix and $x_N \in \Omega$ the terminal constraint, which are defined to guarantee stability (Mayne et al. (2000)). The matrix $P \succ 0$ is the solution of the algebraic Riccati equation resulting from the corresponding unconstrained LQR problem. The terminal set Ω is chosen to be feasible and positively invariant for the closed-loop system with this LQR.
The MPC will regulate the system to the origin for all the initial conditions contained in the feasible set

$$\mathcal{X}_F = \{x \in \mathbb{R}^n | \exists\ \mathbf{u} \text{ satisfying (17)}\}. \tag{18}$$

Note that $\mathcal{X}_F$ is a convex polyhedron due to the nature of the constraints. The feasible set is positively invariant with respect to the closed-loop system, i.e. for any initial state contained in the feasible set, the state evolution of the closed-loop system is also contained in the feasible set for all future times.
There are two ways to implement the constrained optimization problem (16)-(17). The first is to formulate the MPC as a quadratic program (QP) and solve it online at each sampling time. Only the first element of the optimal control sequence is applied to the system, and at the next time step, the computation is repeated starting from the new state and over the shifted horizon. The second way is to formulate the MPC as a multi-parametric QP (mp-QP) which can be solved offline. In this case, the optimal control is given as an explicitly defined continuous piecewise affine (PWA) function depending on the current state, and defined over $\mathcal{X}_F$. The online computation reduces to the simple evaluation of the PWA function. For the cases where the explicit optimal PWA controller is so complex as to be impractical, several approaches have been proposed to obtain approximate continuous PWA controllers. One of such approaches is presented in Scibilia et al. (2009b), where also details about how to obtain the QP and the mp-QP formulations can be found.

The following considers a continuous PWA feedback control law

$$u\left(x\right) = L_j x + g_j,\ \forall\ x \in CR_j, \tag{19}$$

defined over the partition of the feasible set

$$\mathcal{X}_F = \bigcup_{j=1\ldots n_r} CR_j, \tag{20}$$

where $\mathcal{P}_N = \{CR_1, ..., CR_{n_r}\}$ is the collection of polytopic disjoint regions into which $\mathcal{X}_F$ is partitioned.
The controller (19) can represent either the optimal MPC solution or any suitable approximation thereof.

Apart from the natural assumption that the uncertainty set $\mathcal{M}$ contains both the nominal model and the real model, we will make the following assumptions.

A1 Assume that there exists a subset $\mathcal{X}_{in} \subset \mathcal{X}_F$ containing the origin, in which the controller (19) asymptotically stabilizes the system (1) for any time-invariant $[A|B] \in \mathcal{M}$.
A2 Assume that for all initial states $x_0 \in \mathcal{X}_F$, for any $[A|B] \in \mathcal{M}$, if the closed-loop trajectory remains inside $\mathcal{X}_F$ for all future time, then it converges to the origin asymptotically.

Then, the problem tackled is as follows:

> Given a controller of the form (19) computed for a nominal system of the form (1-2). Given an uncertainty set $\mathcal{M}$ of the form (6). Find the *maximal robust feasible set*, i.e. the set of initial conditions $\mathcal{X}_{FR} \subseteq \mathcal{X}_F$ such that for any possible time-invariant $[A|B] \in \mathcal{M}$, the closed-loop system remains feasible at all times.

Assumption A1 guarantees that $\mathcal{X}_{FR}$ will not be an empty set. This assumption is easy to confirm, e.g. the results in Pluymers et al. (2005b) can be used to find a robust positively invariant polyhedral set for the closed-loop system with the LQR.

Assumption A2 is needed to exclude that some system dynamics in the uncertainty set can lead to limit cycles or chaotic behavior in the feasible set. This assumption can be checked by means of a radially unbounded Lyapunov function (possibly dependent on the dynamics). Finding such a function may be difficult. A more immediate, but also conservative, approach is to check that in each region $CR_j \in \mathcal{P}_N$ the following inequality holds for all the vertices $\left[A^{(i)}|B^{(i)}\right]$ of $\mathcal{M}$

$$\|x\| > \frac{\|B^{(i)}g_j\|}{1 - \|A^{(i)} + B^{(i)}L_j\|} \quad \forall x \in CR_j. \tag{21}$$

$\|\cdot\|$ is a norm or the corresponding induced (matrix) norm, depending on the argument. Due to convexity, (21) needs to be checked only on the vertices of each CR_j.

Proposition 3. If condition (21) is satisfied for all regions $CR_j \in \mathcal{P}_N$, then the closed-loop evolutions satisfy

$$\|x(t)\| > \|x(t+1)\| \; \forall x(t) \in \mathcal{X}_F \setminus \{0\} \tag{22}$$

for any system dynamics in the uncertainty set $\mathcal{M}$.

Proof. Inside each region $CR_j \in \mathcal{P}_N$, the closed-loop system is an affine system of the form

$$x(t+1) = \Phi_j x(t) + \varphi_j \tag{23}$$

where $\Phi_j = A + BL_j$ and $\varphi_j = Bg_j$. Then, using (23) in (22) the inequality can be written as

$$\|x(t)\| > \|\Phi_j x(t) + \varphi_j\| \tag{24}$$

The triangle inequality implies that

$$\|\Phi_j x(t) + \varphi_j\| \leq \|\Phi_j x(t)\| + \|\varphi_j\| \tag{25}$$

Thus, requiring that

$$\|x(t)\| > \|\Phi_j x(t)\| + \|\varphi_j\| \tag{26}$$

implies that also (24) is satisfied.
By a property of the induced norm it is

$$\|\Phi_j x(t)\| \leq \|\Phi_j\| \|x(t)\| \; \forall \; x(t) \tag{27}$$

Thus, if

$$\|x(t)\| > \|\Phi_j\| \|x(t)\| + \|\varphi_j\| \tag{28}$$

is satisfied then also (24) is satisfied. So it can be finally seen that if the inequality

$$\|x(t)\| > \frac{\|\varphi_j\|}{1 - \|\Phi_j\|} \tag{29}$$

holds, then $\|x(t)\| > \|x(t+1)\|$.

4. ALGORITHM

This approach follows a simple idea: remove from the feasible set all the initial states which, for any of the uncertain dynamics, lead to an infeasible closed-loop trajectory. Uncertain dynamics here means that the system is described by some time-invariant dynamics contained in the uncertainty set (8). Due to linearity, we need to consider only the vertices of the uncertainty set, which corresponds to considering the worst case dynamics. If the feasible set is robust for the worst case system dynamics, then it will be robust for all the system dynamics in the uncertainty set.
The complication with control laws of the form (19) is that the evolution of the closed-loop system changes depending on where the current state is in the feasible set.
We can now consider how the algorithm explores the feasible set by searching and removing all the initial states that may lead to infeasibility. For each vertex $\left[A^{(i)}|B^{(i)}\right]$ in the uncertainty set $\mathcal{M}$, the algorithm works in two phases.

In the first phase, each region $CR_j \in \mathcal{P}_N$ forming the partition (20) is moved one time step forward, according to the control law associated with the region, to compute the next time-step region. The next time-step region is defined as follows.

Definition 4. (Successor set) The *successor set* of all states which can be reached in one time step from R_j, given system dynamics $[A|B]$, is defined as

$$\begin{array}{l} \mathtt{succ}\,(R_j, [A|B]) = \\ \left\{x^+ \in \mathbb{R}^n | x^+ = (A + BL_j)\, x + Bg_j,\; x \in CR_j\right\}, \end{array} \tag{30}$$

Remark 5. The successor region can be computed simply by applying the control at the vertices of the region and taking the convex hull of the next time-step vertices.

This phase allows the identification of all the (sub)regions in $\mathcal{X}_F$ that in one time step would lead to infeasibility (Fig. 1). It allows also the formation of a *map of reachability*, i.e. for each region $CR_j \in \mathcal{P}_N$ to identify all the regions in $\mathcal{P}_N$ containing states from which it is possible to reach the current region in one time step.

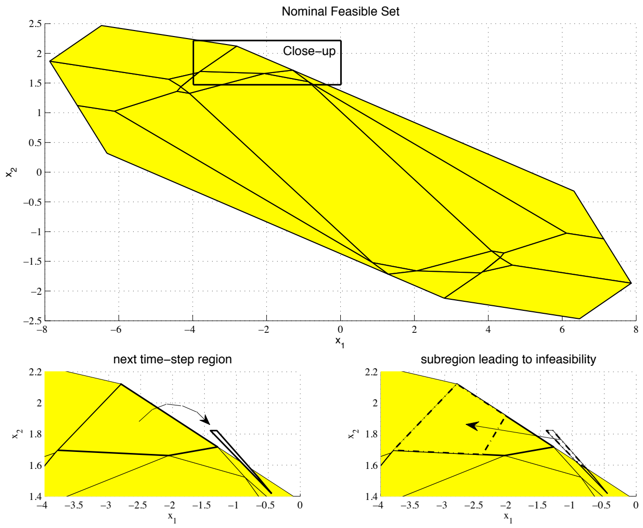


Fig. 1. An example of nominal feasible set partitioned in regions CR_j. In the close-up on the left, the next time-step region is computed. In the close-up on the right the subregion of the feasible set leading to infeasibility is identified.

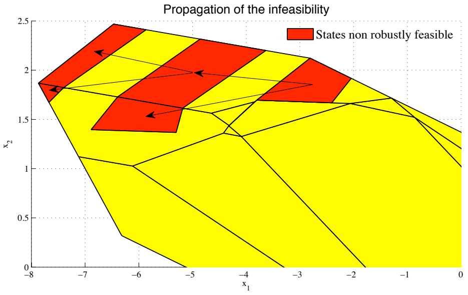


Fig. 2. The subregion not robustly feasible identified in Fig. 1 is propagated backwards in the feasible set, finding all the initial states which would lead to infeasibility.

Certainly, all the states in the (sub)regions that in one time step would lead to infeasibility have to be removed from the feasible set. However, this is clearly not enough, also all the initial states whose closed-loop trajectory moves through these (sub)regions need to be removed. This is done in the second phase of the algorithm, with a mechanism of propagation based on the following definition.

Definition 6. (Predecessor set) Given a region $S \subseteq CR_j \in \mathcal{P}_N$, a region $CR_k \in \mathcal{P}_N$ (CR_j and CR_k may coincide) and system dynamics $[A|B]$, all the states in CR_k for which the next time-step state is in S define the set (the "predecessor" states)

$$\begin{aligned} &\texttt{pred}\left(S, CR_k, [A|B]\right) = \\ &\left\{x \in CR_k | \left(A + BL_k\right) x + Bg_k \in S\right\}. \end{aligned} \tag{31}$$

Remark 7. S and CR_k can be represented as

$$\begin{aligned} S &= \left\{x \in \mathbb{R}^n | D_S x \leq d_S\right\}, \\ CR_k &= \left\{x \in \mathbb{R}^n | D_{CR_k} x \leq d_{CR_k}\right\}. \end{aligned} \tag{32}$$

Then, we can compute the predecessor set as the intersection of a finite number of half-spaces

$$\texttt{pred}\left(S, CR_k, [A|B]\right) = \left\{x \in \mathbb{R}^n | D_{\texttt{pred}} x \leq d_{\texttt{pred}}\right\},$$

where

$$D_{\texttt{pred}} = \begin{bmatrix} D_{CR_k} \\ D_S\left(A + BL_k\right) \end{bmatrix}, \; d_{\texttt{pred}} = \begin{bmatrix} d_{CR_k} \\ d_S - D_S B g_k \end{bmatrix},$$

Using definition 6, all the infeasible (sub)regions in $\mathcal{X}_F$ identified during the first phase of the algorithm are propagated backwards in the feasible set, according to the map of reachability (Fig. 2).

The procedure based on these two phases can be formalized as in the following algorithm. Initially the maximal robust feasible set is initialized as the nominal feasible set. Then, for each vertex of the polytopic uncertainty set the two phases are iterated in sequence.

Algorithm:
Input: the nominal feasible set $\mathbb{X}_F$; the nominal PWA controller and the corresponding feasible set partition $\mathcal{P}_N$; the uncertainty set $\mathcal{M}$.
Output: The maximal robust feasible set $\mathcal{X}_{FR} \subseteq \mathcal{X}_F$.

```
1. Initialize the robust feasible set as $\mathcal{X}_{FR} = \mathcal{X}_F$;
2. For each $\left[A^{(i)}, B^{(i)}\right]$ of $\mathcal{M}$ do
   A. For each $CR_j \in \mathcal{P}_N$ do
      compute $S_{i,j} = \texttt{succ}\left(R_j \cap \mathcal{X}_{FR}, \left[A^{(i)}|B^{(i)}\right]\right)$, the successor region for the remaining points in each region $CR_j$ of the $n_r$ such regions comprising $\mathcal{X}_F$;
      define $Z_{i,j} = S_{i,j} - \mathcal{X}_{FR} \cap S_{i,j}$ and the union of all these sets $\mathcal{Z}_i = \bigcup_j Z_{i,j}$, which represents the set of infeasible states reachable in one step from any point in $\mathcal{X}_{FR}$ for this $[A^{(i)}|B^{(i)}]$;
      build the function $\texttt{rch}_i\left(CR_j\right)$, that gives all the regions containing states from which it is possible to reach $CR_j$ in one time step;
   B. For each $CR_r \in \mathcal{P}_N$ do
      compute $P_{i,r} = \texttt{pred}\left(\mathcal{Z}_i, CR_r, \left[A^{(i)}|B^{(i)}\right]\right) \cap \mathcal{X}_{FR}$;
      define $\mathcal{P}_i = \bigcup_r P_{i,r}$, the admissible predecessor set of $\mathcal{Z}_i$;
      define $\mathcal{P}_N^{rch} = \texttt{rch}_i\left(\mathcal{P}_i\right)$ the set of all the regions containing states which in one time step can reach $\mathcal{P}_i$;
      replace $\mathcal{X}_{FR} = \mathcal{X}_{FR} - \mathcal{P}_i$;
      repeat
         For each $CR_k \in \mathcal{P}_N^{rch}$ do
         compute $P_{i,k} = \texttt{pred}\left(\mathcal{P}_i, R_k, \left[A^{(i)}|B^{(i)}\right]\right) \cap \mathcal{X}_{FR}$;
         replace $\mathcal{P}_i = \bigcup_k P_{i,k}$ and $P_n^{rch} = \texttt{rch}_i\left(\mathcal{P}_i\right)$;
         replace $\mathcal{X}_{FR} = \mathcal{X}_{FR} - \mathcal{P}_i$
      until $\mathcal{P}_i = \emptyset$
```

In general, $\mathcal{X}_{FR}$ is not robustly positively invariant. The set $\mathcal{X}_{FR}$ has the property of containing all and only the states in $\mathcal{X}_F$ which, when used as initial conditions, are guaranteed to have feasible closed-loop trajectories for any possible time-invariant dynamics in the uncertainty set. An initial state x which is not in $\mathcal{X}_{FR}$ does not possess a feasible closed-loop trajectory for all the possible system dynamics. However, this does not mean that x cannot be part of the feasible closed-loop trajectory starting from some state in $\mathcal{X}_{FR}$. (This is discussed further in Section 5).

On the other hand, requiring the positive invariance property of $\mathcal{X}_{FR}$ would have been too conservative and un-

necessary. In fact, it follows that any set with guaranteed positive invariance despite model uncertainty is a subset of $\mathcal{X}_{FR}$. This would unnecessarily limit the possible initial conditions, since we are only interested in guaranteeing that any closed-loop trajectory stays in the feasible set $\mathcal{X}_F$ despite model uncertainty. Moreover, the algorithm for constructing such a positively invariant set would be much more computationally complex than that presented here.
As can also be seen from the numerical examples in Section 5, in general $\mathcal{X}_{FR}$ is not a convex set, though it can be expressed as a finite union of polytopes. This is expected since the piecewise affine control law is a nonlinear controller.

Correctness and convergence of the algorithm are proven by the following theorems.

Theorem 8. The robust feasible set $\mathcal{X}_{FR} \subseteq \mathcal{X}_F$ contains all and only the initial states such that, for any $[A|B] \in \mathcal{M}$, the closed-loop trajectory is feasible.

Proof. To prove the theorem, we show first that if a state $x \in \mathcal{X}_F$ has a closed-loop trajectory that moves outside the feasible set for some $[A|B] \in \mathcal{M}$, then $x \notin \mathcal{X}_{FR}$. Since $[A|B]$ are inside the polytopic uncertainty set, they can be expressed as a convex linear combination of the vertices of $\mathcal{M}$ as in (9). Thus, there is at least a vertex $\left[A^{(i)}|B^{(i)}\right]$, $i \in \{1, ..., L\}$, such that, when used as system dynamics, causes the trajectory starting from x to exit the feasible set, which means that x cannot be in $\mathcal{X}_{FR}$ because it is removed by the algorithm during iteration i at step 2.
It remains to prove that if a state $x \in \mathcal{X}_F$ has feasible closed-loop trajectories for all $[A|B] \in \mathcal{M}$, then $x \in \mathcal{X}_{FR}$. Suppose by contradiction that $x \notin \mathcal{X}_{FR}$. Then there exist some vertex of $\mathcal{M}$ such that the closed-loop trajectory exits the feasible set, which contradicts the assumption that the closed-loop trajectory is feasible for all the dynamics in $\mathcal{M}$. Thus $x \in \mathcal{X}_{RF}$.

Theorem 9. Given the assumptions in Section 3 hold, the algorithm will terminate in a finite number of iterations providing a non-empty robust feasible set $\mathcal{X}_{FR} \subseteq \mathcal{X}_F$.

Proof. The algorithm iterates the two phases A and B for L times, where L is a finite number. Thus, we have to prove that phases A and B execute in finite time. Since $\mathcal{X}_F$ is assumed partitioned into a finite number of polytopes, it is immediate to see that phase A is executed in finite time, and that at each iteration the set of infeasible states $\mathcal{Z}_i$ is described as the union of a finite number of polytopic regions. During phase B, $\mathcal{Z}_i$ is propagated backwards in $\mathcal{X}_F$ according to definition 6. $\mathcal{P}_i$ is initialized as the admissible predecessor set of $\mathcal{Z}_i$, and then iteratively updated in the repeat-until loop within the phase B. Since $\mathcal{Z}_i$ is the union of a finite number of polytopes, $\mathcal{P}_i$ will also have this property for all the iterations. At each iteration, the states comprising $\mathcal{P}_i$ are removed from the current $\mathcal{X}_{FR}$, and once removed they are not considered again in the future iterations. Thus, since $\mathcal{X}_{FR}$ is bounded, eventually $\mathcal{P}_i$ will be an empty set comporting the termination of phase B.
Assumption A1 guarantees that there exist a non-empty region, containing the origin, that will never be in $\mathcal{Z}_i$, thus $\mathcal{X}_{FR}$ will not be empty, and since at all iterations $\mathcal{P}_i$ is the union of a finite number of polytopes, $\mathcal{X}_{FR}$ will be represented as union of polytopic regions.
Assumption A2 guarantees that for any initial state $x \in \mathcal{X}_{FR}$, for any time-invariant $[A|B] \in \mathcal{M}$, the closed-loop system is (feasibly) asymptotically stable.

5. NUMERICAL ILLUSTRATIONS

This section provides examples in order to illustrate the results presented in the previous sections. Here an example is also used to discuss how the presented analysis approach can be related to existing robust control design approaches.

5.1 Robust Feasibility for Optimal Explicit MPC

Consider the double integrator system with input and state constraints. The model of the double integrator is one of the most important in control applications, representing single-degree-of-freedom translational or rotational motion. Thus it can be used to model for instance low-friction, free rigid-body motion, such as single-axis spacecraft rotation and rotary crane motion (Rao and Bernstein (2001)).

The double integrator is given by the continuous-time system

$$\dot{x} = Ax + Bu \tag{33}$$

where $x \in \mathbb{R}^2$, $y \equiv x$, $u \in \mathbb{R}$,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}. \tag{34}$$

The state components x_1 and x_2 can represent for instance the position and velocity, respectively, of a body having mass m. Considering a mass $m = 1$, and discretizing with sampling time 0.3 we obtain the following discrete-time double integrator system matrices

$$A = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0.04 \\ 0.3 \end{bmatrix}. \tag{35}$$

The system is subject to the input constraints $-1 \leq u \leq 1$, and to the velocity constraints $-3 \leq x_2 \leq 3$.
We consider the uncertainty set $\mathcal{M}$ defined by the following vertices

$$A^{(1)} = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix}, \; B^{(1)} = \begin{bmatrix} 0.06 \\ 0.37 \end{bmatrix}, \tag{36}$$

$$A^{(2)} = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix}, \; B^{(2)} = \begin{bmatrix} 0.04 \\ 0.25 \end{bmatrix}. \tag{37}$$

which correspond to the mass being known with an uncertainty of $\varepsilon = 0.2$, i.e. the real mass value is $m = 1 \pm \varepsilon$.

Consider a PWA state feedback controller which represents the optimal solution of the MPC problem (16-17). The weight matrices are chosen as $Q = I$, $R = 1$ and the horizon is $N = 5$.
Fig. 3 shows the nominal feasible set, partitioned into 161 regions, and the portion of the nominal feasible set which is robustly feasible for the uncertainty considered. An initial state in the maximal robust feasible set is shown to generate feasible trajectories for different system dynamics within the uncertainty set. Contrarily, an initial state not

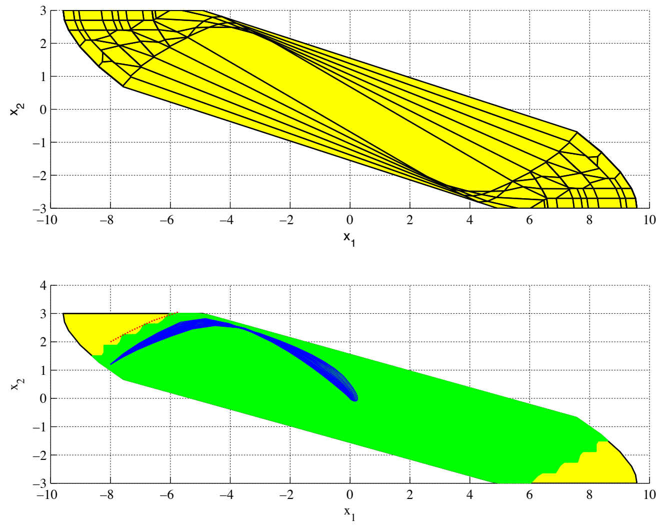


Fig. 3. The upper graph shows the nominal feasible set and its partition for the optimal explicit MPC. The graph below shows the maximal robust feasible set within the feasible set. The feasible state trajectories for different system dynamics in the uncertainty set (blue solid lines) all start from initial position/speed $x_0 = [-8\ 1.2]^T$. The infeasible state trajectory (red dot line) starts from initial position/speed $x_0 = [-8\ 2]^T$ and is given by the system dynamics $\left[A^{(1)}|B^{(1)}\right]$.

in the maximal robust feasible set is shown to originate an infeasible trajectory: when the trajectory exits the feasible set, the control input is undefined.

Remark 10. Some of the feasible trajectories originating inside $\mathcal{X}_{FR}$ may contain states which are not in the set $\mathcal{X}_{FR}$ (but still in $\mathcal{X}_F$). This at first may seem nonsense, but it is perfectly reasonable if one considers that the real system is assumed uncertain but still time invariant: a state $\tilde{x} \notin \mathcal{X}_{FR}$ belonging to the closed-loop trajectory starting from $x \in \mathcal{X}_{FR}$ for certain system dynamics $\left[\tilde{A}|\tilde{B}\right]$ means that $\tilde{x}$ is a robustly feasible initial condition for a part of the uncertainty set including $\left[\tilde{A}|\tilde{B}\right]$, but this is not true for all the possible system dynamics and thus $\tilde{x}$ cannot be included in the set of allowed initial condition $\mathcal{X}_{FR}$.

5.2 Robust Feasibility for Approximate Explicit MPC

Consider the same regulation problem of the previous section, and a PWA state feedback controller which represents the approximate MPC solution computed according to the results in Scibilia et al. (2009b). This PWA controller is optimal for the portion of the feasible set where constraints are not active, on the remaining part of the feasible set the optimal explicit MPC solution is replaced by an approximation based on Delaunay tessellations and computed from a finite number of samples of the exact solution. Finer tessellations can be obtained so as to achieve desired tolerance with the cost function approximation error. Note that for this simple example no extra samples have been introduced since both stability and good performance can be easily proven by post processing the simplest PWA solution.

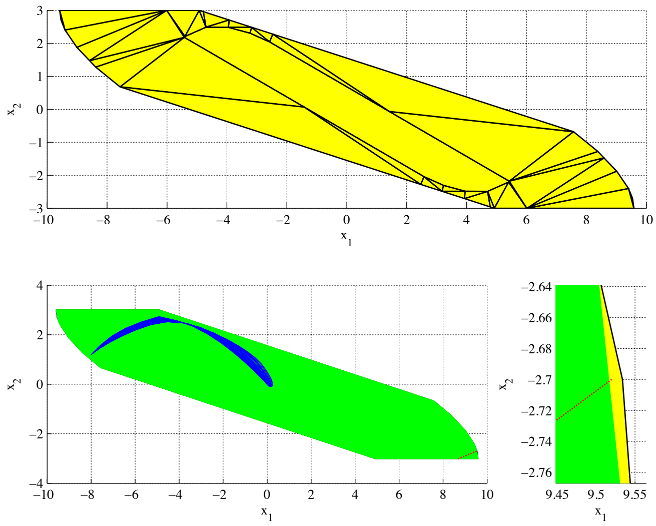


Fig. 4. The upper graph shows the nominal feasible set and its partition for the approximate explicit MPC. The graph below shows the maximal robust feasible set within the nominal feasible set (emphasized in the close-up) for a mass uncertainty $\varepsilon = 0.2$. The feasible state trajectories for different system dynamics in the uncertainty set (blue solid lines) all start from initial position/speed $x_0 = [-8\ 1.2]^T$. The infeasible state trajectory (red dot line) starts from initial position/speed $x_0 = [9.52\ \ -2.7]^T$ and is given by the system dynamics $\left[A^{(1)}|B^{(1)}\right]$.

Fig. 4 presents the feasible set with its partition into 41 regions and the maximal robust feasible set. As can be noted from the close-up, only a minimal part of the nominal feasible set is removed, almost the entire feasible set remains feasible under the uncertainty considered.

It is interesting to note from the simulations that for the case of the double integrator, the closed-loop system with the approximate explicit MPC is characterized by more robust feasibility to model uncertainty than the closed-loop system with the optimal MPC. This can also be seen from Fig. 5, where the mass uncertainty $\varepsilon = 0.5$ is considered.

5.3 Relation to Existing Robust MPC Approaches

The approach proposed in this work represents a tool to analyze the feasibility robustness of nominal explicit MPC approaches (or in general, PWA feedback control laws) with respect to model uncertainty. This section discusses how this relates to a robust MPC design instead, illustrating it by a simple example.

The robust MPC design considered is the one presented in Pluymers et al. (2005a) (Rossiter et al. (2005)), which is based on a nominal MPC formulation where robustness is defined in terms of satisfaction of input and output constraints for all possible uncertainty realization. Given the connection with the nominal MPC design, it is reasonable to believe that this approach represents a better comparison than other robust MPC approaches based on min-max optimization problems.

The robust MPC can be summarized as follows. At each time step, the algorithm minimizes a cost function like

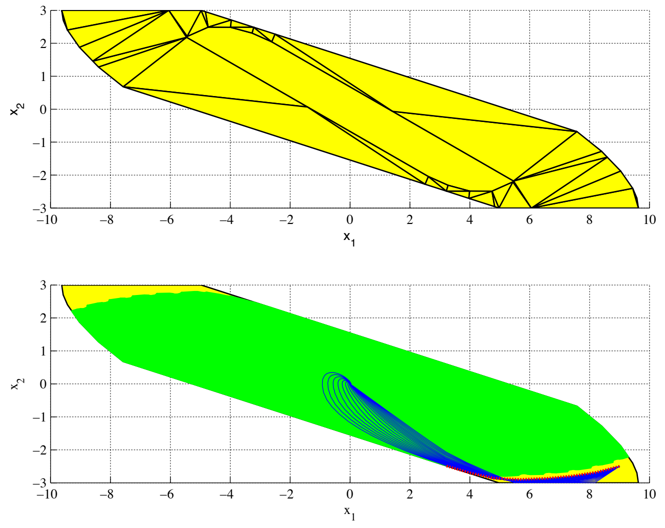


Fig. 5. The upper graph shows the nominal feasible set and its partition for the approximate explicit MPC. The graph below shows the maximal robust feasible set within the nominal feasible set for a mass uncertainty $\varepsilon = 0.5$. The state trajectories for different system dynamics in the uncertainty set all start from initial position/speed $x_0 = [9 \ -2.5]^T$. Since x_0 is outside the robust feasible set, only for some of the system dynamics the trajectories are feasible (blue solid lines). There are system dynamics in the uncertainty set which lead to infeasible trajectories (red dot line).

(16), where the nominal model is used for the future predictions along the horizon. The minimization is subject to constraints like (12-13) which, for robust constraints handling, are applied to all possible predictions according to the following k-step ahead prediction

$$x_k = \prod_{i=0}^{k-1} A_i x_0 + \sum_{j=0}^{k-1} \prod_{l=j+1}^{k-1} A_l B_j u_j \tag{38}$$

where $[A_i, B_i] \in \mathcal{M}$. A terminal constraint is imposed, where the (robust) terminal set is chosen as the largest set of initial condition which is guaranteed to be positively invariant for all possible models in $\mathcal{M}$, assuming the nominal LQR as controller (Pluymers et al. (2005b)). The resulting optimization problem remarkably remains a QP, even if, with respect to the QP resulting from the nominal MPC, more complexity in terms of number of constraints is needed in order to achieve robustness. A multi-parametric QP solution to this robust MPC is proposed in Rossiter et al. (2005). For more details the reader is referred to Pluymers et al. (2005a) and Rossiter et al. (2005).

Note that this robust MPC design is able to deal with linear parameter varying (LPV) systems, while the approach presented here considers uncertain linear parameter invariant systems.

Consider the simple example used in Pluymers et al. (2005a) which has polytopic uncertainty set defined by

$$A^{(1)} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \ B^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{39}$$

$$A^{(2)} = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \ B^{(2)} = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}. \tag{40}$$

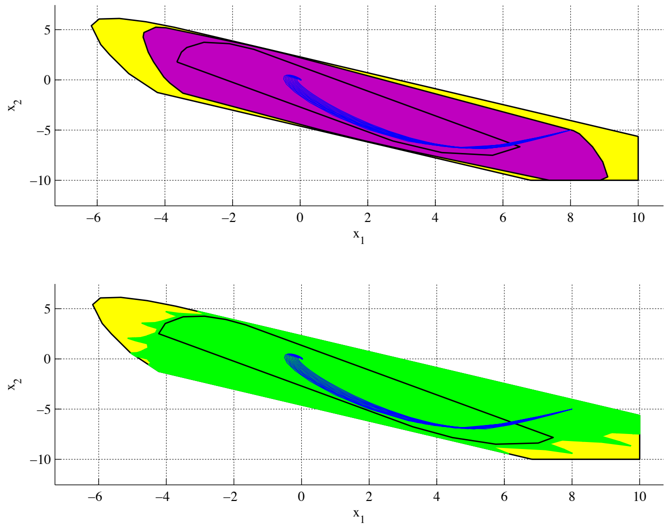


Fig. 6. The upper graph shows the feasible set for the nominal MPC (yellow) and the feasible set for the robust MPC (magenta) with depicted in its interior the robust terminal set. The graph below shows the feasible set for the nominal MPC (yellow), the maximal robust feasible set (green) and the nominal terminal set. In both graphs, several state trajectories are plotted starting all from the initial state $x_0 = [8 \ -5]^T$, for the same different system dynamics.

and the nominal model defined as

$$A = \frac{1}{2}(A^{(1)} + A^{(2)}), \quad B = \frac{1}{2}(B^{(1)} + B^{(2)}). \tag{41}$$

The system is subject to the input constraint $-1 \le u \le 1$, and to the state constraints $[-10 \ -10]^T \le x \le [10 \ 10]^T$. For this system, the robust MPC and the nominal MPC are formulated both with weight matrices chosen as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad R = 3. \tag{42}$$

and horizon $N = 3$.

Fig. 6 illustrates the feasible set resulting from the robust MPC. The same figure also shows the portion of the nominal feasible set which is robustly feasible with the nominal MPC. Both robust and nominal MPC give the same performance, as it can be qualitatively seen from the closed-loop trajectories obtained for the same set of different time-invarying dynamics.

It is not hard to identify regions of initial states for which the nominal MPC would not be sufficient, while instead the robust MPC would be. However, it is also immediate to identify considerably larger regions of initial states which would be satisfactorily controlled by the nominal MPC and which are instead excluded by the feasible set with the robust MPC. Then, assuming that the set of initial conditions of interest is within the maximal robust feasible set from the nominal MPC, the analysis method presented in this paper can be used to decide that the nominal controller is enough and therefore there is no need for the supplementary complexity associated with the robust control design. Of course, this does not exclude a number of cases where the robust design is instead necessary.

The analysis tool presented in this paper may be useful, for example, in the practical case of a crane which has to move objects whose weight may be within a given range, satisfying constraints on position and speed. Reasonably, the parameters of the crane model can be expected to change for each possible weight (cf. Section 5.1). However, once the object has been fixed, from the point of view of the controller the model remains time invariant for the whole operation (until a new object is considered). In this case, a controller design based on a nominal model (for example one which considers the average weight) may be considered satisfactory, after the associated maximal robust feasible set has guaranteed that constraints will not be violated for any possible weight.

6. CONCLUSIONS

This work has proposed a tool for analyzing how uncertainty in the real plant affects the nominal PWA feedback law, thereby providing the maximal subset of the state space which contains safe initial conditions under the model uncertainty considered. The maximal robust feasible set thus obtained is, in general, non-convex. It is not required to be robustly positive invariant, and is computed in finite time. Moreover, any subset, and thus any convex subset, still preserves the property of being robustly feasible.

This result may be used to decide whether or not a nominal design can be used without resorting to a more complex robust design. On the other hand, it can also be seen as an enabling technology for several future approaches to the problem of enhancing the robustness of (approximate) explicit MPC solutions towards model uncertainty. If the maximal robust feasible set does not cover the portion of state space of interest, the next step could be to consider just the regions that do not satisfy the robust feasibility condition, and search for suitable controllers for those regions. One approach could be to define and solve a new explicit MPC problem for each infeasible region, with proper constraints ensuring robust feasibility, and the maximal robust feasible set as the new terminal set.

Assumption A2 is needed to exclude the possibility of limit cycles or chaotic behavior of the uncertain system in closed-loop with the controller, originally designed for the nominal system. The assumption is easy to check, but rather conservative. Future research can be directed to reduce this conservativeness.

An interesting future work would be the extension to LPV systems. This could be achieved propagating back the infeasible regions from the phase A of the algorithm for all the possible uncertain realizations in the polytopic uncertainty set. This however would reasonably result in heavier computational loads.

The inclusion of robustness with respect to disturbances represents another future work of interest.

REFERENCES

Bemporad, A. and Filippi, C. (2003). Suboptimal explicit receding horizon control via approximate multiparametric quadratic progamming. *Journal of Optimization Theory and Applications*, 117(1), 9–38.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Blanchini, F. (1999). Set invariance in control. *Automatica*, 35, 1747–1767.

Boyd, S., Ghaoui, L.E., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*. SIAM.

Cuzzola, F.A., Geromel, J.C., and Morari, M. (2002). An improved approach for constrained robust model predictive control. *Automatica*, 38, 1183–1189.

Cychowski, M.T., Ding, B., and O'Mahony, T. (2005). An orthogonal partitioning approach to simplify robust model predictive control. In *Proceedings of the 13th Mediterranean Conference on Control and Automation*, 877–882.

de la Peña, D.M., Bemporad, A., and Filippi, C. (2004). Robust explicit MPC based on approximate multiparametric convex programming. In *Proceedings of the 43rd IEEE Conference on Decision and Control*.

Gilbert, E.G. and Tan, K.T. (1991). Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9).

Grieder, P., Parrillo, P.A., and Morari, M. (2003). Robust receding horizon control - analysis and synthesis. In *Proceedings of the 42nd IEEE Conference on Decision and Control*.

Hovland, S., Willcox, K.E., and Gravdhal, J. (2008). Explicit MPC for large-scale systems via model reduction. *AIAA J. Guidance, Control and Dynamics*, 31(4).

Ikonen, E. and Najim, K. (2002). *Advanced process identification and control*. Control Engineering Series. Marcel Dekker, Inc.

Johansen, T.A. (2003). Reduced explicit constrained linear quadratic regulators. *IEEE Transactions on Automatic Control*, 48(5), 823–828.

Johansen, T.A. and Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48, 810–815.

Kerrigan, E.C. and Maciejowski, J.M. (2000). Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proceedings of the 39th IEEE Conference on Decision and Control*.

Kothare, M.V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10), 1361–1379.

Kouvaritakis, B., Rossiter, J.A., and Schuurmans, J. (2000). Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8), 1545–1549.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789–814.

Nicolao, G.D., Magni, L., and Scattolini, R. (1996). On the robustness of receding-horizon control with terminal constraints. *IEEE Transactions on Automatic Control*, 41(3), 451–453.

Pluymers, B., Rossiter, J.A., Suykens, J., and Moor, B.D. (2005a). A simple algorithm for robust MPC. In *Proceedings of the 16th IFAC World Congress*.

Pluymers, B., Rossiter, J.A., Suykens, J.A.K., and Moor, B.D. (2005b). The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. *Proceedings of the American Control Conference*.

Rao, V.G. and Bernstein, D.S. (2001). Naive control of the double integrator. *IEEE Control Systems Magazine*, 21, 86–97.

Rossiter, J.A. and Grieder, P. (2005). Using interpolation to improve efficiency of multiparametric predictive control. *Automatica*, 41(4), 637–643.

Rossiter, J.A., Pluymers, B., Suykens, J., and Moor, B.D. (2005). A multi parametric quadratic programming solution to robust predictive control. In *Proceedings of the 16th IFAC World Congress*.

Scibilia, F. (2010). *Explicit Model Predictive Control: Solutions via Computational Geometry*. Ph.D. thesis, Norwegian University of Science and Technology.

Scibilia, F., Bitmead, R.R., Olaru, S., and Hovd, M. (2009a). Maximal robust feasible sets for constrained linear systems controlled by piecewise affine feedback laws. In *The 7th IEEE International Conference on Control and Automation*.

Scibilia, F., Olaru, S., and Hovd, M. (2009b). Approximate explicit linear MPC via Delaunay tessellation. In *Proceedings of the European Control Conference*, 2833–2838.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3), 489–497.

van den Boom, T.J.J. and Haverkamp, B.R.J. (2000). Towards a state-space polytopic uncertainty description using subspace model identification techniques. In *Proceedings of the American Control Conference*, 1807–1811.

Comments – Remarks

Feasible Sets for MPC and their Approximations [⋆]

F. Scibilia [*] **S. Olaru** [**] **M. Hovd** [*]

[*] *Department of Engineering Cybernetics, NTNU, O.S. Bragstads plass 2D, 7491 Trondheim, Norway (e-mail: francesco.scibilia@itk.ntnu.no, morten.hovd@itk.ntnu.no).*
[**] *SUPELEC Systems Sciences (E3S) - Automatic Control Department, Gif-sur-Yvette, 91192, France, (e-mail: sorin.olaru@ieee.org)*

Abstract: This paper considers the problem of computing inner approximations for the feasible set for linear Model Predictive Control (MPC) techniques. An alternative approach for computing the feasible set is presented, based on set relations instead of the conventional orthogonal projection. The approach can be implemented incrementally on the length of the prediction horizon. This is exploited to design an algorithm to compute suitable inner approximations. Such approximations are characterized by simpler representations and preserve the essential properties of the feasible set such as convexity, positive invariance and inclusion of the set of expected initial states. This is important when in order to avoid the online optimization, the optimal MPC solution is precomputed offline in an explicit form as a piecewise affine state feedback control law over the feasible set. Particularly in the context of finding simpler suboptimal explicit solutions the complexity of the feasible set plays a decisive role.

Keywords: Shape description; Polyhedral sets; Piecewise linear controllers; Predictive control.

1. INTRODUCTION

Within the theoretical framework for MPC, a key role is played by the so-called *feasible set*, i.e. the largest subset of the state space such that there exists a control action satisfying all the constraints. The feasible set is closely related to the prediction horizon considered. Generally longer horizons result in larger feasible sets, but this is at the cost of a larger MPC optimization problem. Provided that the MPC optimization problem is formulated so that closed-loop stability is ensured (Mayne et al. (2000)), an optimal control action is guaranteed to exist at each sampling time, for any initial state chosen in the feasible set. This also means that when explicit MPC formulations (Bemporad et al. (2002), Tøndel et al. (2003)) are employed, the feasible set is the domain where the optimal piecewise affine control function is defined. A well-known problem in the explicit MPC area is that finding and deploying the optimal explicit solution may be impractical in several relevant situations. This problem has been extensively tackled by the research community, which has proposed many approaches to approximate explicit MPC. The availability of the feasible set and furthermore its shape description play key roles in the effectiveness of many of these approaches, particularly for the ones based on feasible set discretizations (Scibilia et al. (2009) and Scibilia et al. (2010a), Nam et al. (2010), Bemporad and Filippi (2006), Johansen and Grancharova (2003), Jones and Morari (2009)).

The feasible set is completely described by the linear constraints involved in the MPC optimization problem, which places it in the specific class of convex sets called polyhedra (more precisely, polytopes). The standard approach to compute the feasible set uses an important operation in polyhedral set theory, the orthogonal projection (Burger et al. (1996), Jones et al. (2004), Mount (2002)). However, the orthogonal projection often turns out to be a computationally demanding operation in high spatial dimensions (Jones et al. (2008)). This is the case, for example, when the feasible set is computed for MPC with long prediction horizons.

Convexity is an important characteristic of the feasible set. Another crucial feasible set property in the MPC context is the *positive invariance* with respect to the closed-loop system, i.e. for any initial state contained in the feasible set, the state evolution of the closed-loop system is also contained in the feasible set for all future times. In general, polyhedral sets represent an important family of candidate positively invariant sets and have been particularly successful in the solution of many control engineering problems thanks to their flexibility (Blanchini (1999), Kerrigan and Maciejowski (2000), Gilbert and Tan (1991)). However, the appurtenant disadvantage of this flexibility is the complexity of representation which may be extremely high since it is not fixed by the space dimension considered (Blanchini and Miani (2008)).

Approximating polytopes by simpler sets is a well-known problem in many research areas related to optimization, system identification and control. With any simpler representation a certain loss of information is associated in prin-

[⋆] This work is a revised version of Scibilia et al. (2010b) (Copyright © 2010 Elsevier Ltd) and is, to a large extent, extracted from Scibilia (2010). This version is intended for members of the NIL project groups.

ciple. Thus, in general, the ideal solution is always a right balance between simplicity and accuracy (Dabbene et al. (2003), Bronstein (2008), Gritzmann and Klee (1994b)). However, when approximation approaches of polytopes are considered for feasible set approximation, while convexity is easily maintained by the family of sets we are dealing with, positive invariance is generally lost. Furthermore, at the design stage, one of the requirements of the controller is that it has to be able to regulate the system for a given set of initial states representing the expected initial operation conditions. Assuming that the MPC fulfills the design specifications, this set, here called the *operating set*, is contained within the feasible set. To maintain the effectiveness of the MPC, an additional issue is then that the approximation does not result in a loss of information which will prevent the MPC from performing acceptably for states in the operating set.

This paper proposes two contributions: first it suggests an alternative approach for computing the feasible set which uses set relations instead of orthogonal projection. Set relations of similar nature have also been used in Kolmanovsky and Gilbert (1995). The proposed approach can be implemented incrementally over the length of the horizon, and proves to be computationally less demanding than the standard approach. Thereafter, the main contribution is proposed. A solution to the problem of finding (inner) approximations of the feasible set which are characterized by simpler representations and which preserves convexity, positive invariance and inclusion of the operating set is presented. The approach is based on the introduction of certain conditions which extend existing approaches for the computation of polytope approximations.

2. PRELIMINARIES

Consider the following discrete-time linear time-invariant system:

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^r$ is the control input, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, and the pair (A, B) is stabilizable. Full state measurement and no disturbances or model uncertainty are assumed.

The system is subject to the following state and input constraints:

$$x(t) \in \mathcal{X} \subset \mathbb{R}^n \tag{2}$$

$$u(t) \in \mathcal{U} \subset \mathbb{R}^r \tag{3}$$

for all future times. The sets $\mathcal{X}$, $\mathcal{U}$ are convex polyhedral sets with the origin being an interior point for both sets. Bounded polyhedral sets, i.e. *polytopes*, are the family of sets principally considered in this work. A polytope $\mathcal{P}$ can be expressed as the intersection of a finite number of half-spaces (which gives its facet lattice) referred as the $\mathcal{H}$-representation of $\mathcal{P}$. Equivalently, $\mathcal{P}$ can be represented as the convex hull of its vertices $V = \{v^{(1)}, ..., v^{(n_V)}\}$ referred as the $\mathcal{V}$-representation of $\mathcal{P}$: $\mathcal{P} = \mathsf{conv}(V)$. A vertex (half-space) of a polytope is said to be *redundant* if its omission from the $\mathcal{V}$-representation ($\mathcal{H}$-representation) does not change the shape of the polytope. A $\mathcal{V}$-representation ($\mathcal{H}$-representation) is *minimal* if there are no redundant vertices (half-spaces). Any polytope has a *unique* minimal $\mathcal{V}$-representation ($\mathcal{H}$-representation) (Blanchini and Miani (2008)).

The *orthogonal projection* of a polytope $\mathcal{P} \subset \mathbb{R}^n \times \mathbb{R}^d$ onto $\mathbb{R}^n$ (the first n coordinates) is defined as

$$\Pi_n(\mathcal{P}) = \left\{x \in \mathbb{R}^n | \exists z \in \mathbb{R}^d,\ \left[x^T\ z^T\right] \in \mathcal{P}\right\} \tag{4}$$

The *Minkowski sum* of two polytopes $\mathcal{P}$ and $\mathcal{Q}$ is defined as

$$\mathcal{P} \oplus \mathcal{Q} = \{x = p + q|\ p \in \mathcal{P},\ q \in \mathcal{Q}\}. \tag{5}$$

The *erosion* (or Pontryagin difference) of two polytopes $\mathcal{P}$ and $\mathcal{Q}$ is defined as

$$\mathcal{P} \ominus \mathcal{Q} = \{x|\ x + q \in \mathcal{P},\ \forall q \in \mathcal{Q}\}. \tag{6}$$

The *set difference* of two polytopes $\mathcal{P}$ and $\mathcal{Q}$ is defined as

$$\mathcal{P} \setminus \mathcal{Q} = \{x|\ x \in \mathcal{P}, x \notin \mathcal{Q}\}. \tag{7}$$

More details and algorithmic implementations can be found for example in Mount (2002), Blanchini and Miani (2008), Gritzmann and Klee (1994a) and Kvasnica et al. (2006).

The problem of regulating the system (1) to the origin, such that constraints like (2-3) are satisfied, is solved by the finite horizon MPC

$$\min_{\mathbf{u}} \left\{ J(\mathbf{u}, x(t)) = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \right\} \tag{8}$$

$$\begin{aligned} \text{s.t. } & x_0 = x(t), && && \text{(a)} \\ & x_{k+1} = Ax_k + Bu_k, && k = 0, 1, ..., N-1, && \text{(b)} \\ & x_k \in \mathcal{X}, && k = 1, 2, ..., N-1, && \text{(c)} \\ & u_k \in \mathcal{U}, && k = 0, 1, ..., N-1, && \text{(d)} \\ & x_N \in \Omega, && && \text{(e)} \end{aligned} \tag{9}$$

where x_k denotes the predicted state vector at time $t + k$ obtained by applying the k first elements of the input sequence $\mathbf{u} \triangleq [u_0, ..., u_{N-1}]$; N is the prediction horizon; $Q \succeq 0$ (positive semidefinite) and $R \succ 0$ (positive definite) are symmetric matrices corresponding to weights on state and input; P is the terminal cost matrix and $x_N \in \Omega$ the terminal constraint, which are defined to guarantee stability. The matrix $P \succ 0$ is the solution of the algebraic Riccati equation resulting from the corresponding unconstrained LQR problem. The terminal set Ω is chosen to be feasible and positively invariant for the closed-loop system with this LQR (Mayne et al. (2000)).

It is assumed that the MPC (8)-(9) is designed to regulate the system for a given set of initial states which represents the expected initial operating conditions (the operating set). Without any particular restriction, the operating set can be considered to be a polytope, which is here indicated as $\mathcal{X}_o$.

3. THE FEASIBLE SET

The MPC regulates the state to the origin for all the initial conditions contained in the feasible set. The feasible set is defined as

$$\mathcal{X}_F = \{x \in \mathbb{R}^n |\ \exists\ \mathbf{u} \text{ satisfying (9)}\} \tag{10}$$

and can be interpreted as the maximal controlled invariant set by means of the MPC with prediction horizon N and terminal set Ω.

When explicit solutions are considered, the feasible set is the domain where the piecewise affine controller is defined.

3.1 Computing the Feasible Set: Standard Approach

The feasible set can be completely characterized by the constraints involved in the optimization problem. The constraints (9) can be expressed in terms of the input sequence $\mathbf{u}$ and the initial state $x(t)$:

$$G\mathbf{u} - Ex(t) \leq w \tag{11}$$

where G and E are matrices and w a vector of suitable dimensions (see for example Scibilia et al. (2009) for details). The linear inequalities (11) define a polytope in the space $\mathbb{R}^{n+rN}$

$$\mathcal{Q} = \left\{ \left[x(t)^T \ \mathbf{u}^T\right]^T \in \mathbb{R}^{n+rN} \mid G\mathbf{u} - Ex(t) \leq w \right\} \tag{12}$$

Then, the feasible set $\mathcal{X}_F$ is given as orthogonal projection of $\mathcal{Q}$ onto the state coordinates

$$\mathcal{X}_F = \Pi_n \left(\mathcal{Q}\right) \tag{13}$$

With this approach the computation of the feasible set relies essentially on the efficiency of projection algorithms. However, the orthogonal projection is intrinsically a computationally demanding operation (NP-hard), becoming increasingly prohibitive as the dimension of the projecting polytope increases (Tiwary (2008a)). This affects the computation of feasible sets for MPC, especially when long prediction horizons are considered.

3.2 Computing the Feasible Set: Alternative Approach

This section considers a different approach for computing the feasible set, which will be also useful for the results in the following sections and provides an alternative for the case of long prediction horizons.
Consider the optimization problem (8) subject to the constraints (9- a, b, d, e), i.e. ignoring the state constraints $x_k \in \mathcal{X}$, $k = 1, 2, ..., N-1$.
For this relaxed optimization problem, indicate with $\tilde{\mathcal{X}}_F$ the corresponding relaxed feasible set.
Note that the only constraints on the state are the equality constraints (9- a, b), and the terminal constraint (9- e): the terminal state must be contained in the terminal set.
Using the equality constraints, the terminal state equation can be written as:

$$x_N = A^N x(t) + \hat{B}\mathbf{u} \tag{14}$$

where A^N is the N-matrix-power of A and $\hat{B} = \left[A^{N-1}B \ A^{N-2}B \ ... \ B\right]$.
Equation (14) suggests the existence of a relation in terms of the sets involved in the relaxed MPC optimization problem considered. Set relations of similar nature have been also used in Kolmanovsky and Gilbert (1995) in the context of finding admissible sets for discrete-time systems subject to bounded input disturbances.
Before formally stating the set relation, we need to introduce the set of admissible input sequences:

$$\mathcal{U}^{(N)} = \left\{ \mathbf{u} \in \mathbb{R}^{rN} \mid u_j \in \mathcal{U}, j = 0, ..., N-1 \right\}. \tag{15}$$

Theorem 1. Consider the optimization problem (8) subject to the constraints (9-a, b, d, e). Then the terminal set, Ω, the corresponding feasible set, $\tilde{\mathcal{X}}_F$, and the set of admissible sequence input, $\mathcal{U}^{(N)}$, satisfy the following set relation

$$\Omega = A^N \tilde{\mathcal{X}}_F \ominus \hat{B}(-\mathcal{U}^{(N)}) \tag{16}$$

where $A^N : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $\hat{B} : \mathbb{R}^{rN} \mapsto \mathbb{R}^n$ represent linear maps applied respectively to $\tilde{\mathcal{X}}_F$ and to $\mathcal{U}^{(N)}$.

Proof. According to (10), the relaxed feasible set can be written as:

$$\tilde{\mathcal{X}}_F = \left\{ x \in \mathbb{R}^n | \exists \mathbf{u} \in \mathcal{U}^{(N)} : \ A^N x + \hat{B}\mathbf{u} \in \Omega \right\} \tag{17}$$

Using the linear maps A^N and $\hat{B}$ we can define the sets:

$$A^N \tilde{\mathcal{X}}_F = \left\{ x_e \in \mathbb{R}^n | x_e = A^N x, \ x \in \tilde{\mathcal{X}}_F \right\} \tag{18}$$

$$\hat{B}\mathcal{U}^{(N)} = \left\{ x_u \in \mathbb{R}^n | x_u = \hat{B}\mathbf{u}, \mathbf{u} \in \mathcal{U}^{(N)} \right\} \tag{19}$$

From (17) we can write the equivalence:

$$A^N \tilde{\mathcal{X}}_F = \left\{ x_e | \exists x_u \in \hat{B}\mathcal{U}^{(N)} : \ x_e + x_u \in \Omega \right\} \tag{20}$$

which subsequently implies that $\tilde{\mathcal{X}}_F$ is the collection of all the states that can be obtained as a combination of points in Ω and $\hat{B}(-\mathcal{U}^{(N)})$. This leads to the equivalence:

$$\Omega = \left\{ x_\Omega | \ x_\Omega - x_u \in A^N \tilde{\mathcal{X}}_F, \ \forall x_u \in \hat{B}\mathcal{U}^{(N)} \right\} \tag{21}$$

By the definition of the erosion operator, (21) corresponds to (16).

In the following we assume that the matrix A of (1) is invertible. Notice that zero eigenvalues of A mean that there are modes which are pure delays of the inputs. This is clear by taking the Jordan form of A, which also gives linearly transformed constraint sets $\mathcal{X}'$ and $\mathcal{U}'$ (the constraints on the state corresponding to delayed inputs must then be compatible with $\mathcal{U}'$). Then, the assumption is motivated by considering that the Jordan blocks of A with zero eigenvalue can then be excluded, meaning that constraints involving linear combinations of past inputs and current states corresponding to the remaining Jordan blocks are not allowed. However, we also note that the assumption is always satisfied by discretized (finite dimensional) continuous-time systems.
Therefore, by (16) the relaxed feasible set can be computed as:

$$\tilde{\mathcal{X}}_F := \left(A^N\right)^{-1} \left[\Omega \oplus \hat{B}(-\mathcal{U}^{(N)})\right] \tag{22}$$

Note that the computation of the feasible set via the formula (22) basically costs a Minkowski sum in $\mathbb{R}^n$, given the polytopes Ω and $\mathcal{U}^{(N)}$, which is an operation that can be done in polynomial time (Gritzmann and Sturmfels (1993)). This is generally more convenient than using (13) which requires handling polytopes in higher dimensions, $\mathbb{R}^{n+rN}$.

Inspecting (16), an incremental approach for computing $\tilde{\mathcal{X}}_F$ can be derived, which with a simple modification is extendable for also computing $\mathcal{X}_F$.
Let us explicitly express the dependence of the feasible set from the length of the horizon as $\mathcal{X}_F^{(k)}$, which indicates

the feasible set for a horizon of length k. According to this notation, the feasible set we are interested in is $\mathcal{X}_F = \mathcal{X}_F^{(N)}$.
For the case $k = 1$, relation (16) becomes

$$\Omega = A\tilde{\mathcal{X}_F}^{(1)} \ominus B(-\mathcal{U}) \tag{23}$$

which leads to the set [1] of all the initial states that in one time-step move inside Ω

$$\tilde{\mathcal{X}}_F^{(1)} = (A)^{-1}\left[\Omega \oplus B(-\mathcal{U})\right] \tag{24}$$

At this point, introducing the constraint on the state is straightforward, and thus the feasible set $\mathcal{X}_F^{(1)}$ is simply computed from $\tilde{\mathcal{X}}_F^{(1)}$ as

$$\mathcal{X}_F^{(1)} = \tilde{\mathcal{X}}_F^{(1)} \cap \mathcal{X} \tag{25}$$

The feasible set $\tilde{\mathcal{X}}_F^{(2)}$ is determined by $\tilde{\mathcal{X}}_F^{(1)}$ considering an analogous relations to (23):

$$\tilde{\mathcal{X}}_F^{(1)} = A\tilde{\mathcal{X}}_F^{(2)} \ominus B(-\mathcal{U}) \tag{26}$$

which gives

$$\tilde{\mathcal{X}}_F^{(2)} = (A)^{-1}\left[\tilde{\mathcal{X}}_F^{(1)} \oplus B(-\mathcal{U})\right] \tag{27}$$

and thus also $\mathcal{X}_F^{(2)}$ can be determined analogously to (25). In general, the feasible set with horizon k can be computed in this incremental fashion from the feasible set with horizon $k-1$. This leads to Algorithm 1 for computing the feasible set for the MPC (8-9).

Algorithm 1. : Feasible set

Input: the system state and input matrices A and B; the terminal set Ω; the state and input constraints sets $\mathcal{X}$ and $\mathcal{U}$; the length of the horizon N.
Output: the feasible set $\mathcal{X}_F$.

```
1. Initialize the set $\mathcal{T} = \Omega$.
2. For $k = 1$ to $N$ do
       Compute $\tilde{\mathcal{X}}_F^{(k)} = A^{-1}[\mathcal{T} \oplus B(-\mathcal{U})]$;
       Compute $\mathcal{X}_F^{(k)} = \tilde{\mathcal{X}}_F^{(k)} \cap \mathcal{X}$;
       Set $\mathcal{T} = \mathcal{X}_F^{(k)}$
3. Set $\mathcal{X}_F = \mathcal{X}_F^{(k)}$.
```

In addition to giving the possibility to include the state constraints, the advantage of the incremental approach is that it avoids the necessity of handling the polytope $\mathcal{U}^{(N)}$ which, especially for long horizons, may be undesirable.

4. APPROXIMATION OF FEASIBLE SETS

The feasible set is represented by a polytope in the state space. The problem of finding polytope approximations by means of simpler convex bodies arises in many research areas related to optimization, system identification and control. In general, the solution is a balance between the *simplicity* of the representation and the *accuracy* of the approximation, where the accuracy can be measured using different metrics, depending on the particular approach used to solve the problem. An example is the work in Dabbene et al. (2003), where the authors provide algorithms for computing inner approximations in terms of the largest ellipsoids inscribed. However, more often it is required that the approximating convex body is itself a polytope. In this direction, different approaches have been proposed in the literature, and reference is made to Bronstein (2008) and Gritzmann and Klee (1994b) (and references therein) for surveys on the subject.
The common representation complexity indexes of a polytope are the number of half-spaces (or facets) for the $\mathcal{H}$-representation, and the number of vertices for the $\mathcal{V}$-representation. Since a polytope is characterized by unique minimal $\mathcal{H}$- and $\mathcal{V}$-representations, any lower complexity representation must correspond either to an inner or to an outer approximation of the polytope.
In this section, the scope is to approximate the feasible set by means of a simpler polytope. Since the feasible set corresponds to the maximal feasible controlled invariant set by means of the MPC, no feasible outer approximations can exist, and therefore attention is restricted only to the search for inner approximations. Note that the task is more involved than finding simpler representations maintaining a prescribed accuracy in the approximation. For control purposes it is of prime importance that the approximating polytope preserves the positive invariance property and contains the operating set.
A natural approach for computing inner approximations is based on the fact that, for any polytope, the omission of any of the vertices from the $\mathcal{V}$-representation changes the polytope by reducing it [2]. Typically (but not necessarily), the approximations thus obtained also result in lower complexity $\mathcal{H}$-representations, as will be discussed later. Furthermore, several situations can be recognized where a simpler feasible set characterized by fewer vertices would provide immediate improvements. This is the case for example in approaches to explicit MPC such as Scibilia et al. (2009), Hovd et al. (2009), Nam et al. (2010) and Jones and Morari (2009), in approaches to multi-parametric convex programming such as Bemporad and Filippi (2006)), or also in control approaches as Gutman and Cwikel (1986), where the solution depends strictly on the complexity of the feasible set in terms of the number of vertices.
Therefore, interest is focused on finding appropriate inner approximations characterized by a reduced number of vertices.

An algorithm for computing inner approximations of polytopes based on the removal of vertices is proposed in Reisner et al. (2001) (in Lopez and Reisner (2002) if only 3D polytopes are considered). The fundamental result is the following.
Proposition 2. Given a polytope $\mathcal{P} \subset \mathbb{R}^n$ characterized by $n_\mathcal{V}$ vertices $V_\mathcal{P} = \left\{v^{(1)}, ..., v^{(n_\mathcal{V})}\right\}$, $\mathcal{P} = \mathtt{conv}(V_\mathcal{P})$, there exists a vertex $v \in V_\mathcal{P}$ such that the polytope $\mathcal{Q} = \mathtt{conv}(V_\mathcal{P} \setminus \{v\})$ satisfies

$$\frac{\mathtt{vol}(\mathcal{P}) - \mathtt{vol}(\mathcal{Q})}{\mathtt{vol}(\mathcal{P})} \leq \alpha(n) n_\mathcal{V}^{-\frac{n+1}{n-1}}. \tag{28}$$

The factor $\alpha(n)$ is a constant depending only on the space dimension (details about how to estimate this constant can

[1] This set is also called the one-step controllability set to Ω (Blanchini (1994)).

[2] Dually, the omission of any of the half-spaces from the $\mathcal{H}$-representation changes the polytope by enlarging it.

be found in Reisner et al. (2001) and Lopez and Reisner (2002)).

This result is the best possible in general, for the dependence on the numbers of vertices of the approximating polytope.

The main idea of the algorithm is thus the consecutive removal of the chosen vertices. Taking an appropriate number $k < n_{\mathcal{V}}$, it can be identified a successive minimizing choice of vertices of $\mathcal{P}$, i.e. a sequence $\left\{v^{(r_1)}, ..., v^{(r_{n_{\mathcal{V}}-k})}\right\}$ of different vertices in $V_{\mathcal{P}}$ such that for all $i = 1, ..., n_{\mathcal{V}} - k$

$$\begin{aligned}&\mathtt{vol}\left(\mathtt{conv}\left(V_{\mathcal{P}} \setminus \left\{v^{(r_1)}, ..., v^{(r_{i-1})}\right\}\right)\right) - \\ &\mathtt{vol}\left(\mathtt{conv}\left(V_{\mathcal{P}} \setminus \left\{v^{(r_1)}, ..., v^{(r_i)}\right\}\right)\right)\end{aligned} \qquad (29)$$

is minimal over all choices of $v^{(r_i)} \in V_{\mathcal{P}} \setminus \left\{v^{(r_1)}, ..., v^{(r_{i-1})}\right\}$.

The polytope

$$\mathcal{Q} = \mathtt{conv}(V_{\mathcal{P}} \setminus \left\{v^{(r_1)}, ..., v^{(r_{n_{\mathcal{V}}-k})}\right\}), \qquad (30)$$

characterized by k vertices, is an inner approximation of $\mathcal{P}$. The accuracy of the approximation obtained is measured by the difference of volume between $\mathcal{P}$ and $\mathcal{Q}$ and, in general, it is the best possible obtainable by any polytope with k vertices (up to the dimension dependent constants involved).

More details about the implementation of the algorithm can be found in Reisner et al. (2001) and Lopez and Reisner (2002).

The following presents an approach to extend algorithms like the one found in Reisner et al. (2001) in order to meet the primary objective of maintaining the fundamental properties of the feasible set.

Given a vertex v of $\mathcal{P}$, indicate with $\mathtt{adj}(v)$ all the vertices adjacent to v, i.e. all the vertices of $\mathcal{P}$ which share a facet with v.

Proposition 3. The region excluded from $\mathcal{P}$ when the vertex v is omitted from its $\mathcal{V}$-representation is given by

$$\mathcal{L}_v = \mathtt{conv}\left(\{v, \mathtt{adj}(v)\}\right) \setminus \mathtt{conv}\left(\{\mathtt{adj}(v)\}\right). \qquad (31)$$

Proof. Naturally, $\mathcal{L}_v$ is characterized only by the facets of $\mathcal{P}$ incident in v. These facets comprise a region given by the convex hull of v and all its adjacent vertices. Since the adjacent vertices of v still remain vertices of $\mathcal{P}$, the prospective region identified solely by these vertices needs to be removed from the description of $\mathcal{L}_v$. The remaining vertices of $\mathcal{P}$, i.e. the vertices of the polytope $\mathcal{P} \setminus \mathtt{conv}\left(\{v, \mathtt{adj}(v)\}\right)$, are not affected by the omission of v.

In general, $\mathcal{L}_v$ is non-convex, but can be represented as a finite collection of polytopes.

Let us now consider the polytope $\mathcal{P}$ as our feasible set $\mathcal{X}_F$, i.e. $\mathcal{P} \equiv \mathcal{X}_F$. Since the convexity property is simply maintained by removing a vertex, attention is turned to the problems of how to preserve positive invariance and how to maintain the states comprising the operating set.

4.1 Preserving Positive Invariance

The difficulty in preserving positive invariance comes from the fact that we have to take into consideration the nonlinear dynamics of the closed-loop system with the MPC.

The *next time-step* feasible set $\mathcal{X}_F^+$ is defined as follows

$$\mathcal{X}_F^+ = \left\{x^+ | x^+ = Ax + Bu_0^*,\ x \in \mathcal{X}_F\right\} \qquad (32)$$

where u_0^* is the first element of the MPC optimal control sequence at x.

The asymptotic (exponential) stability of the MPC guarantees that $\mathcal{X}_F^+ \subset \mathcal{X}_F$ and that $\mathcal{X}_F^+$ is positively invariant for the closed-loop system. We can now define the set $\mathcal{X}_{\mathcal{N}} = \mathcal{X}_F \setminus \mathcal{X}_F^+$, which has the interesting property to contain only points of the feasible set that are exclusively initial states of state evolutions starting inside the feasible set. In other words, considering any possible state evolution in $\mathcal{X}_F$, each state in $\mathcal{X}_{\mathcal{N}}$ can only be a starting point of it.

Theorem 4. Any inner approximation of the feasible set obtained as convex hull of vertices inside $\mathcal{X}_{\mathcal{N}}$, such that also all the facets are inside $\mathcal{X}_{\mathcal{N}}$, preserves the positive invariance.

Proof. Consider a set of points $V_{\mathcal{N}}$ inside $\mathcal{X}_{\mathcal{N}}$, such that $\mathtt{conv}(V_{\mathcal{N}})$ has all the facets within $\mathcal{X}_{\mathcal{N}}$. It follows that $\mathcal{X}_F^+ \subset \mathtt{conv}(V_{\mathcal{N}})$. Therefore, for any starting point inside such a polytope, the state evolution either moves in one step inside $\mathcal{X}_F^+$ or is already inside $\mathcal{X}_F^+$, which shows positively invariance of $\mathtt{conv}(V_{\mathcal{N}})$.

The property of positive invariance could then be preserved if for every vertex $v^{(r)}$ removed from $\mathcal{X}_F$, the condition $\mathcal{L}_{v^{(r)}} \subset \mathcal{X}_{\mathcal{N}}$ is satisfied. In fact, this ensures that the remaining vertices satisfy the requirements of Theorem 4.

However, because of the set $\mathcal{X}_F^+$, computing the set $\mathcal{X}_{\mathcal{N}}$ in general involves a substantial computational effort, which drastically reduces the applicability of the approach. Indeed, note from (32) that the definition of the next time-step feasible set implies the knowledge of the optimal input for the states in the feasible set. In particular, since the feasible set is a convex set, only the knowledge of the optimal control input on the border of $\mathcal{X}_F$ is needed for the computation of the next time-step feasible set. Nevertheless, this still comports a computational burden which may compromise the effectiveness of the overall approach. A further undesirable aspect of using the next time-step feasible set is that $\mathcal{X}_F^+$ is, in general, non-convex (Blanchini (1994))(Fig. 2). This results in more difficulties in the computation of $\mathcal{X}_N$ as the intersection between non-convex sets is more involved than the intersection between polytopes, even if the possible resulting non-convex set can still be expressed as a collection of polytopes.

This issue can be easily overcome considering the following relation

$$\mathcal{X}_F^+ \subseteq \mathcal{X}_F^{(N-1)} \subset \mathcal{X}_F \qquad (33)$$

Note that $\mathcal{X}_F^{(N-1)}$ is easily available using an incremental procedure such as Algorithm 1 for computing the feasible set. Moreover it is always convex and it is positively invariant for the closed-loop system. Then, the proposed solution is to use $\mathcal{X}_F^{(N-1)}$ in place of $\mathcal{X}_F^+$. The conservativeness introduced is not severe for the purpose here considered, $\mathcal{X}_F(N-1)$ being a tight outer approximation of $\mathcal{X}_F^+$ (Blanchini (1994)). Defining the set $\bar{\mathcal{X}}_{\mathcal{N}} = \mathcal{X}_F \setminus \mathcal{X}_F^{(N-1)}$, the property of positive invariance is preserved if for every vertex $v^{(r)}$ removed from $\mathcal{X}_F$, the following condition is satisfied

$$\mathcal{L}_{v^{(r)}} \subset \bar{\mathcal{X}}_{\mathcal{N}}. \tag{34}$$

In fact, condition (34) ensures that the remaining vertices satisfy the requirements of Theorem 4, whose results are valid if $\bar{\mathcal{X}}_{\mathcal{N}}$ is considered instead of $\mathcal{X}_{\mathcal{N}}$. Note that checking condition (34) is equivalent to checking $\mathcal{L}_{v^{(r)}} \cap \mathcal{X}_F^{(N-1)} = \emptyset$.

4.2 The Operating Set Condition

The goal of algorithms like the one developed in Reisner et al. (2001) is to find the polytope $\mathcal{Q}$ characterized by k vertices that best approximate the polytope $\mathcal{P}$ (characterized by $n_{\mathcal{V}} > k$ vertices). The accuracy of the approximation is given by the difference of volume between $\mathcal{P}$ and $\mathcal{Q}$. The introduction of condition (34) (invariance) changes the degrees of freedom in the minimization of the difference of volume. Generally, not all the vertices in the successive minimizing choice of vertices of $\mathcal{P}$ satisfy the necessary condition (34) and, therefore, not all can be removed.

When the focus is on feasible sets, the loss of volume may not necessarily be a critical issue in itself, since practically it would be more of interest that the approximating feasible set still contains the operating set $\mathcal{X}_o$. This objective can be achieved simply by checking that every time a vertex $v^{(r)}$ is removed from $\mathcal{X}_F$, the corresponding excluded region does not comprise any part of the operating set,

$$\mathcal{X}_o \cap \mathcal{L}_{v^{(r)}} = \emptyset. \tag{35}$$

If a certain vertex does not satisfy (35), then it is not removed and the next vertex is considered.

4.3 Removing Vertices

Suppose that the interest is to remove as many vertices as possible as long as conditions (34) (invariance) and (35) (operating set inclusion) are satisfied. This can be done iteratively: at each iteration, among all the current vertices which makes (34) and (35) satisfied, remove the one that results in the lowest loss in terms of volume.

Note that for any vertex $v^{(r)}$, conditions (34) and (35) and the volume loss can be evaluated locally, i.e. only $v^{(r)}$ and $\texttt{adj}(v^{(r)})$ are involved in the computation of $\mathcal{L}_{v^{(r)}}$, as can be seen from Proposition 3.

An efficient way to implement the algorithm is to use structures similar to pointers. Given the list of vertices $V_{\mathcal{X}_F}$ characterizing the feasible set, where each element on the list is identified by the position number, two structures can be defined:

1) `Index`, a list containing numbers referring to vertices in $V_{\mathcal{X}_F}$ (list of pointers).
2) `Adj`, a structure containing the adjacency information. `Adj(Index(`i`))` gives the list of pointers to the vertices in $V_{\mathcal{X}_F}$ adjacent to the vertex with pointer `Index(`i`)`.

Then the operation of removing a vertex $v^{(r)} \in V_{\mathcal{X}_F}$ with pointer, say, `Index(`i`)`, can be done removing the i-th element from `Index`, after having removed the `Index(`i`)`-th element from `Adj` and updated the elements `Adj(`j`)`, for all $j \in$ `Adj(Index(`i`))`. The update is done as follows. Each vertex j is also vertex of the polytope $\mathcal{R} =$ `conv(Adj(Index(`i`)))`, then for each list `Adj(`j`)` the reference `Index(`i`)` is removed and the adjacencies resulting from $\mathcal{R}$ are added.

The advantage of using the pointer structures is to allow each iteration to simply update only the data affected by the current vertex removal.

4.4 Discussion on the Complexity Indexes

In general, the number of half-spaces may be much higher than the number of vertices, and vice versa. Thus, a natural question would be how will the reduction of the complexity in the $\mathcal{V}$-representation affect the complexity in the $\mathcal{H}$-representation. While in 2 and 3 dimensions there exist simple relations between the two complexity indexes, in higher dimension analytical relations are very difficult to define (Matousek (2002)). Thus, giving an exact answer to the question is a hard problem. However, a well-known achievement in the theory of convex polytopes allows giving an answer in terms of upper bounds: a polytope in the n-dimensional space with $n_{\mathcal{V}}$ vertices has at most $2\binom{n_{\mathcal{V}}}{\lfloor n/2 \rfloor}$ half-spaces. Thus, for a fixed space dimension n the number of half-spaces has an order of magnitude of $n_{\mathcal{V}}^{\lfloor n/2 \rfloor}$ ("upper bound" theorem Matousek (2002)). The upper bound theorem refers to worst case scenarios. Certainly, not all polytopes exhibit this extreme behavior, for example it is known that if n_p points are chosen uniformly at random in the unit n-dimensional ball, then the expected number of half-spaces of their convex hull is only of order of magnitude of n_p (Matousek (2002)). Thus, even if there exist cases where the omission of a vertex causes an increase in the number of half-spaces, it is reasonable to expect that typically a certain reduction of complexity in terms of number of vertices also provides a fairly relevant reduction of the complexity in terms of number of half-spaces, in the sense that the upper bound on the number of half-spaces decreases.

5. DISCUSSION ON COMPUTATIONAL COMPLEXITY

Both Algorithm 1 for computing feasible sets and the approach proposed in Section 4 for computing simplifications of feasible sets are based on basic geometric operations on polytopes: Minkowski sum, intersection, convex hull and volume computation. Therefore, it is interesting to discuss some aspects connected with the computational complexity of these operations so to provide with additional insight into the algorithmic behavior of the approaches presented. It should be noted, though, that the scope of this section is not to give a comprehensive discussion on the computational complexity of each operation.

Every polytope admits two equivalent representation forms: the $\mathcal{V}$-representation (using vertices) and the $\mathcal{H}$-representation (using half-spaces). For polytopes, the representation conversion from $\mathcal{H}$- to $\mathcal{V}$-representation (vertex enumeration) and the conversion from $\mathcal{V}$- to $\mathcal{H}$-representation (facet enumeration or convex hull computation) are computationally equivalent and, in general, are difficult operations (NP-hard). The computational complexity increases fast with the number of half-spaces

and/or the number of vertices involved (Khachiyan et al. (2008), Fukuda (2004)).
Often, operations on polytopes that are easy to perform in one representation become difficult if the polytopes are instead in the other representation. The Minkowski sum of two polytopes is a computationally easy operation (polynomial time) when the two polytopes are in $\mathcal{V}$-representation while becomes a difficult operation (NP-hard) when they are in $\mathcal{H}$-representation (Gritzmann and Sturmfels (1993), Tiwary (2008b)). This means that the known Minkowski sum algorithms operating on polytopes in the $\mathcal{H}$-representation show a computational complexity which increases fast with the number of half-spaces of the operands.
On the other hand, the intersection of two polytopes given in the $\mathcal{H}$-representation is an easy operation while becomes a difficult operation (NP-hard) with polytopes in the $\mathcal{V}$-representation (Tiwary (2008b)).
Resorting to a representation conversion is often not a solution to reduce complexity since the operation is itself hard.

Let us consider Algorithm 1. At each iteration the computational complexity is basically determined by a Minkowski sum and an intersection operation. Let us assume that the interest is to obtain a feasible set in the $\mathcal{V}$-representation (which here may be motivated by the procedure for computing simplified feasible sets discussed in Section 4). Therefore, the computational complexity of Algorithm 1 would increase fast with the number of vertices considered due to the intersection operation. Note that if the interest is in a feasible set in the $\mathcal{H}$-representation, then the Minkowski sum would be computationally costly. This suggests that computing the feasible set is intrinsically a hard problem. Given that polytopes, in general, are far more complex in terms of number of vertices and facets in higher dimensions, it is reasonable to expect that the computational complexity for computing the feasible set increases fast with the dimension of the state space, n. The computational advantage of the proposed approach in respect to the traditional one is that at each iteration the operands are polytopes of dimension n. Instead, the traditional approach requires the projection of a polytope of dimension $n+Nr$ which is easily a prohibitive operation even for small n since the polytope dimension depends also from the horizon length N and the input dimension r. Also, when the standard approach is implemented incrementally on the horizon length, at each iteration the projection of a polytope of dimension $n + r$ is required, which may still be prohibitive.

Analogous considerations can be made for the approach proposed in Section 4 for computing simplifications of feasible sets. Assume that the interest is to remove as many vertices as possible as long as conditions (34) (invariance) and (35) (operating set inclusion) are satisfied (cf. Section 4.3). The algorithm requires initially the computation of the volume loss associated with each vertex removal. The polytope volume computation, either in the $\mathcal{V}$- or the $\mathcal{H}$-representation, is not a difficult operation (polynomial time) (Gritzmann and Klee (1994a)). Then, at each iteration the intersection operation is used to check the conditions (34) and (35): the conditions are first inspected on the vertex which currently means the lowest loss of volume, continuing with the vertex causing the second lowest loss if the former does not satisfy the conditions, and so on until a suitable vertex is identified for removal (or none, in which case the algorithm terminates). The use of pointer structures allows easily updating just the volumes affected by the current vertex removal. At each iteration the intersections are the most expensive operations to perform, and although they are done on relatively simple polytopes, these may require relevant computation especially when n increases (since this typically means high number of vertices with a complex map of adjacency). Note, however, that here the operation may be implemented in a more efficient way since it is not needed to actually compute the intersection polytope, but rather to decide whether the two polytope operands intersect or not.

6. NUMERICAL ILLUSTRATIONS

6.1 Feasible Set Computation

The computation time efficiency of the proposed approach based on set relations (SR) has been compared with the standard approach based on projection (P) in Matlab by using the Multi-Parametric Toolbox (MPT) [3] (Kvasnica et al. (2006)). Both the algorithms have been implemented incrementally on the horizon length. The incremental implementation is inherent in the set relation-based approach, while for the traditional projection-based approach it may speed up the calculation in many situations.
Extensive simulations have been carried out on several random systems for different state (n) and input (r) dimensions. The common MPC settings used are: $Q = I$ and $R = I$ where I represents the identity matrix; state constraints $-10 * \mathbb{1} \leq x \leq 10 * \mathbb{1}$, input constraints $-\mathbb{1} \leq u \leq \mathbb{1}$, where $\mathbb{1}$ is a suitably dimensioned vector with all elements equal to 1.
Table 1 reports some of the results obtained during the simulations to give a picture of the typical performance from both approaches. In general the proposed approach has performed significantly more efficiently than the standard approach. Furthermore, the projection approach led several times to unsatisfactory results such as no result after one hour of computation or numerical errors.
It must be noted that the proposed approach also led to numerical errors, though in a considerably lower number of cases than the projection approach. It is reasonable to believe that most of the numerical issues faced could be removed by a careful re-implementation of the approach.

6.2 Feasible Set Approximation

The goal of the algorithm here is to reduce the complexity in terms of number of vertices ($n_{\mathcal{V}}$) as much as possible while satisfying conditions (34) and (35) (as discussed in Section 4.3).
Consider the double integrator system represented by the continuous-time linear system

$$\dot{x} = Ax + Bu \tag{36}$$

[3] MPT for Matlab offers several algorithms for computing the projection of a polytope (vertex enumeration/convex hull-based method, Fourier-Motzkin elimination, iterative hull, block elimination, equality set projection). In the simulations, the MPT projection function has been set to automatically select the best method.

Table 1. Set relation-based approach vs. projection-based approach. Time computation measured in seconds. † indicates no result after 1 hour computation. ‡ indicates simulation terminated by Matlab errors.

Rand. sys.	N=5	N=7	N=10
(n, r)	SR−P	SR−P	SR−P
$(3, 1)$	0.14 − 0.40	0.55 − 2.70	2.01 − 19.33
	0.20 − 0.52	0.26 − 0.30	1.91 − 1.72
$(3, 2)$	0.73 − 5.53	4.92 − 12.10	3.72 − 19.06
	4.18 − 12.72	2.20 − 4.79	9.21 − 96.64
$(4, 1)$	4.76 − 25.63	21.30 − 32.67	59.21 − †
	8.82 − 45.81	6.89 − 15.81	26.61 − 475.24
$(4, 2)$	34.85 − 291.7	121.5 − ‡	47.09 − 1067
	38.59 − 88.10	147.4 − †	998.09 − ‡

where $x \in \mathbb{R}^2$, $u \in \mathbb{R}$,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{37}$$

The problem formulation includes a state constraint $-3 \leq x_2 \leq 3$ and an input constraint $-1 \leq u \leq 1$.
The state components x_1 and x_2 can represent, for example, the position and velocity, respectively, of a body having mass 1.
In Table 2 the discrete counterpart of system (36) is considered for several sampling times. For faster sampling times in general the polyhedral borders of the feasible sets approximate ellipsoidal shapes, which therefore imply complex $\mathcal{V}$-representations. Many of the vertices can be removed with a minimal loss in terms of volume. Moreover, faster sampling time means generally a more complex explicit MPC solution, which more likely may require the use of simpler feasible sets to use in approximate explicit MPC approaches. In the table, $n_{\mathcal{V}}$ indicates the number of vertices of the feasible set, $\tilde{n}_{\mathcal{V}}$ indicates the number of vertices of the approximate feasible set.

Fig. 1, 2 and 3 graphically illustrate the idea for sampling time 0.3 and $N = 10$. As operating set we assume that the system has to operate in a range of positions $-10 \leq x_1 \leq 10$, for any feasible x_2 (Fig. 1). Analogously chosen operating sets have been used for the results in Table 2.
Note that the feasible set, characterized by 24 vertices, is approximated by a less complex feasible set characterized by 10 vertices (Fig. 3). The loss of volume introduced by the approximate feasible set is less than 3%.
To compute the set $\mathcal{X}_F^+$ in Fig. 2, the explicit solution of the MPC is obtained and then each region comprising the feasible set is propagated one step forward.

Table 2. Complexity reduction by approximate feasible sets.

Samp. time	N=7	N=10
seconds	$n_{\mathcal{V}}$ - $\tilde{n}_{\mathcal{V}}$ - loss%	$n_{\mathcal{V}}$ - $\tilde{n}_{\mathcal{V}}$ - loss%
0.3	18 - 6 - 0.04%	24 - 10 - 0.03%
0.1	34 - 12 - 0.01%	28 - 8 - 0.01%
0.01	56 - 30 - 0.00%	66 - 34 - 0.00%

Extensive simulations have been carried out on several random systems for different state (n) and input (r) dimensions. The common MPC settings used are: horizon $N = 5$, $Q = I$ and $R = I$; state constraints $-20 * \mathbb{1} \leq x \leq 20 * \mathbb{1}$, input constraints $-\mathbb{1} \leq u \leq \mathbb{1}$.

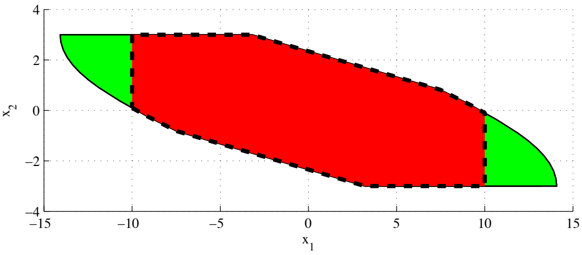


Fig. 1. The largest polytope is the feasible set $\mathcal{X}_F$ (green). The set marked with the wide dashed line represents the operating set (red), contained in the feasible set.

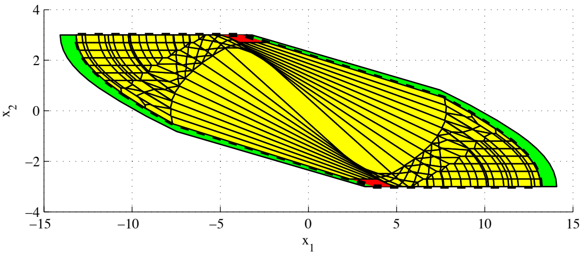


Fig. 2. The largest polytope represents the feasible set $\mathcal{X}_F$ (green). The set marked with wide dashed line represents the set $\mathcal{X}_F^{(N-1)}$ (red and yellow), which corresponds with the feasible set for an horizon length $N - 1$. The internal regions (in yellow) marked with thin lines comprise the next time-step feasible set $\mathcal{X}_F^+$. Note that $\mathcal{X}_F^+ \subset \mathcal{X}_F^{(N-1)} \subset \mathcal{X}_F$ and that $\mathcal{X}_F^+$ is non-convex.

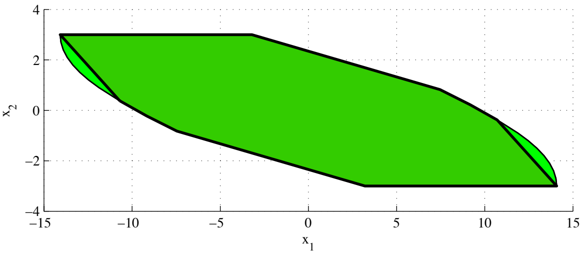


Fig. 3. The largest polytope represents the feasible set $\mathcal{X}_F$. The internal set marked with wide solid line represents the reduced feasible set.

Table 3 lists some of the results obtained in the simulations. As expected from the discussion in Section 4.4, in most of the cases the reduction in the number of vertices also led to a reduction in the number of half-spaces. However, a few cases where this did not happen are reported to illustrate that the upper bound theorem guarantees only that a reduction of vertices will not cause an extreme increase in the number of half-space.
In some cases the algorithmic implementation of the approach faced numerical errors, particularly with high dimensional feasible sets. It must be said that, apart from the use of pointers to make computations more efficient, no particular emphasis has been put in coding an efficient implementation of the approach. The scope here was primarily to provide evidence of its effectiveness. Careful re-implementation of the approach would reasonably remove

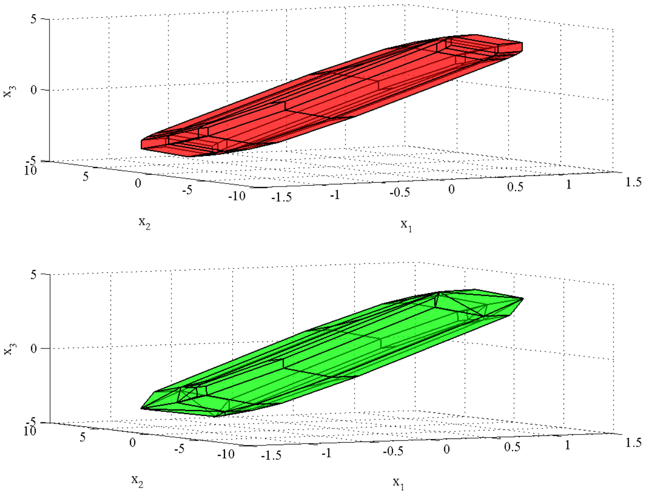


Fig. 4. The upper graph shows the feasible set in the 3-dimensional state space for a system in Table 3. The graph below shows the corresponding approximate feasible set.

most of the numerical issues and improve the computational performance.

Table 3. Feasible set vs approximate feasible set for several random systems. The case marked with (*) is illustrated in Fig. 4.

Rand. sys.		Feasible set	Approximate feasible set
(n, r)		$n_{\mathcal{V}} - n_{\mathcal{H}}$	$\tilde{n}_{\mathcal{V}} - \tilde{n}_{\mathcal{H}}$
$(3, 1)$		$28 - 16$	$16 - 10$
		$58 - 56$	$35 - 46$
	(*)	$82 - 78$	$54 - 64$
$(3, 2)$		$74 - 44$	$41 - 38$
		$64 - 34$	$25 - 28$
		$52 - 28$	$30 - 29$
$(4, 1)$		$124 - 38$	$87 - 57$
		$116 - 58$	$92 - 56$
		$108 - 34$	$84 - 40$

7. CONCLUSIONS

The paper has presented an alternative approach for computing feasible sets when MPC techniques are used. The proposed approach uses set relations instead of the conventional projection, which then unfolds to a procedure based on Minkowski sum and intersection routines. This proves to be computationally more efficient and algorithmically more robust than using projection routines, particularly when high dimensional polytopic sets are involved (i.e. for long prediction horizons, high dimensional state and/or input).
However, some numerical issue suggested the need of future work to improve the algorithmic robustness of the routines for the needed polytopic operations.

When the feasible set is characterized by a critical complexity of representation in terms of number of vertices, an approach to compute a simplified feasible set with a reduced number of vertices has been given. The approach is based on the introduction of certain conditions which extend existing approaches for the computation of polytope approximations, so that the approximating polytope maintains all the fundamental properties of the feasible set required for MPC applications like positive invariance and inclusion of the set of expected operating conditions.
Preserving the positive invariance property in the feasible set approximation is crucial. This issue is inherently difficult to handle since it is concerned with the nonlinear dynamics of the closed-loop system. The proposed approach typically allows a considerable decrease in the $\mathcal{V}$-representation complexity by removing most of the vertices needed to deal with feasible set borders which approximate ellipsoids (according to the operating set considered). However, this approach does not allow to consider possible even simpler feasible set approximations which, while including the operating set, may have borders within $\mathcal{X}_F^{(N-1)}$. A potential future research direction could be to search for different approaches which would give more flexibility. One could for example look at solutions which use level surfaces of Lyapunov functions (Alessio et al. (2006)) to find different vertices than the original ones from the feasible set.
The conditions introduced constrain the goal of minimizing the loss of volume in the approximation. Finding suitable approximating polytopes characterized by the minimum loss of volume is a well known problem. Requiring that the approximation minimizes the loss of volume while satisfying conditions related to system dynamics remains challenge. Here the minimization of the loss of volume was not considered critical, since in the context of the present work the interest often is to preserve given crucial parts of the feasible set, which can be done via the operating set condition. In fact, the algorithm proposed tends to minimize the loss of volume in the sense that at each iteration the suitable vertex which results in the lowest loss of volume in the current approximating polytope is removed. Pointer structures were used to enhance the implementation efficiency, though it may be further improved by a careful re-implementation of the approach. Simulations proved the effectiveness of the results presented.

REFERENCES

Alessio, A., Bemporad, A., Lazar, M., and Heemels, W.P.M.H. (2006). Convex polyhedral invariant sets for closed-loop linear MPC systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 4532–4537.

Bemporad, A. and Filippi, C. (2006). An algorithm for approximate multiparametric convex programming. *Computational Optimization and Applications*, 35(1), 87–108.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Blanchini, F. (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2), 428–433.

Blanchini, F. (1999). Set invariance in control. *Automatica*, 35, 1747–1767.

Blanchini, F. and Miani, S. (2008). *Set-Theoretic Methods in Control*. Birkhauser.

Bronstein, E.M. (2008). Approximation of convex sets by polytopes. *Journal of Mathematical Science*, 153(6), 727–762.

Burger, T., Gritzmann, P., and Klee, V. (1996). Polytope projection and projection of polytopes. *The American Mathematical Monthly*, 103(9), 742–755.

Dabbene, F., Gay, P., and Polyak, B.T. (2003). Recursive algorithms for inner ellipsoidal approximation of convex polytopes. *Automatica*, 39(10), 1773–1781.

Fukuda, K. (2004). Frequently asked questions in polyhedral computation. Technical report, [Online], www.ifor.math.ethz.ch/ fukuda/polyfaq/polyfaq.html.

Gilbert, E.G. and Tan, K.T. (1991). Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9).

Gritzmann, P. and Klee, V. (1994a). On the complexity of some basic problems in computational convexity: 2. volume and mixed volumes. Technical report, DIMACS.

Gritzmann, P. and Klee, V. (1994b). On the complexity of some basic problems in computational convexity: I. containment problems. *Discrete Mathematics*.

Gritzmann, P. and Sturmfels, B. (1993). Minkowski addition of polytopes: computational complexity and applications to Göbner bases. *SIAM J. Disc. Math.*, 6(2), 246–269.

Gutman, P. and Cwikel, M. (1986). Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, AC-31(4).

Hovd, M., Scibilia, F., Maciejowski, J.M., and Olaru, S. (2009). Verifying stability of approximate explicit MPC. In *Proceedings of the 48th IEEE Conference on Decision and Control*, 6345–6350.

Johansen, T.A. and Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48, 810–815.

Jones, C.N., Kerrigan, E.C., and Maciejowski, J.M. (2004). Equality set projection: A new algorithm for the projection of polytopes in halfspace representation. Technical report, Department of Engineering, Cambridge University.

Jones, C.N., Kerrigan, E.C., and Maciejowski, J.M. (2008). On polyhedral projection and parametric programming. *Journal of Optimization Theory and Applications*, 138, 207–220.

Jones, C.N. and Morari, M. (2009). Approximate explicit MPC using bilevel optimization. In *Proceedings of the European Control Conference*, 2396–2401.

Kerrigan, E.C. and Maciejowski, J.M. (2000). Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proceedings of the 39th IEEE Conference on Decision and Control*.

Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., and Gurvich, V. (2008). Generating all vertices of a polyhedron is hard. *Discrete and Computational Geometry*, 39((1-3)), 174–190.

Kolmanovsky, I. and Gilbert, E.G. (1995). Maximal output admissible sets for discrete-time systems with disturbance inputs. In *Proceedings of the American Control Conference*, 1995–1999.

Kvasnica, M., Grieder, P., Baotic, M., and Christophersen, F.J. (2006). *Multi-Parametric Toolbox (MPT) documentation.* Swiss Federal Institute of Technology, http://control.ee.ethz.ch/∼mpt/.

Lopez, M.A. and Reisner, S. (2002). Linear time approximation of 3D convex polytopes. *Computational Geometry*, 23, 291–301.

Matousek, J. (2002). *Lectures on Discrete Geometry*, volume 212 of *GTM*. Springer.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789–814.

Mount, D.M. (2002). Computational geometry. Lecture Notes. Department of Computer Science, University of Maryland.

Nam, N.H., Olaru, S., and Hovd, M. (2010). Patchy approximate explicit model predictive control. In *International Conference on Control, Automation and Systems*.

Reisner, S., Schutt, C., and Werner, E. (2001). Dropping a vertex or a facet from a convex polytope. *Forum Math.*, 13, 359–378.

Scibilia, F. (2010). *Explicit Model Predictive Control: Solutions via Computational Geometry*. Ph.D. thesis, Norwegian University of Science and Technology.

Scibilia, F., Hovd, M., and Olaru, S. (2010a). An algorithm for approximate explicit model predictive control via Delaunay tessellations. *European Journal of Control*, (submitted).

Scibilia, F., Olaru, S., and Hovd, M. (2009). Approximate explicit linear MPC via Delaunay tessellation. In *Proceedings of the European Control Conference*, 2833–2838.

Scibilia, F., Olaru, S., and Hovd, M. (2010b). On feasible sets for MPC and their approximations. *Automatica*, doi:10.1016/j.automatica.2010.10.022.

Tiwary, H.R. (2008a). On computing the shadows and slices of polytopes. *CoRR*, abs/0804.4150.

Tiwary, H.R. (2008b). On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete and Computational Geometry*, 40, 469–479.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3), 489–497.

Comments – Remarks

Frequency domain measures to the diagnosis of model plant mismatch

Selvanathan Sivalingam [*] **Arun K. Tangirala** [**]

[*] *Engineering Cybernetics Department, NTNU, Trondheim 7491, Norway (e-mail: selvanat@itk.ntnu.no)*
[**] *Department of Chemical Engineering, IIT Madras, Chennai 600036 India (e-mail: arunkt@iitm.ac.in)*

Abstract
A new approach is proposed for the diagnosis of poor control loop performance due to model plant mismatch (MPM) in the internal model control framework. A new quantity G_p/G_m, termed as the *Plant Model Ratio* (PMR) in the frequency domain is introduced as a measure of model plant mismatch. It is shown that there exists a unique signature in PMR for each combination of mismatch in model parameters, which is the key step in the proposed method. A method to estimate PMR from routine operating data is provided. Theoretical and practical aspects of the mapping between the type of MPM and the proposed PMR are presented. Simulation studies are carried out to demonstrate the effectiveness of the proposed method.

Keywords: process control, instrumentation, simulation, time-delay estimation, Hilbert transform relation

1. INTRODUCTION

Monitoring and assessment of controller performance has evoked considerable interest to academicians and practitioners in the area of process control and monitoring for two decades now. The incentive in finding solutions to associated problems is immense since poor performance affects product quality, plant economy and safety. In addition, there is a constant drive to improve the performance of existing control schemes and to optimize the overall plant performance. However, one can note that controllers often fail to operate according to their design specifications and, in many cases, they even increase the process variability, as was reported by Ender (1993). Most modern industrial plants have hundreds or even thousands of automatic control loops. These loops can be simple proportional-integral-derivative (PID) or more sophisticated model based linear and non-linear control loops. It has been reported that as many as 60% of all industrial controllers have performance problems Ender (1993). Having an automated means of detecting when a loop is not performing well and then diagnosing the root cause plays a vital role in addressing the problems mentioned at the outset of this paper.

Bialkowski (1993) and Kozub and Garcia (1993) pointed out in their work that the major causes of poor control loop performance are (*i*) improper controller tuning (*ii*) poor hardware (sensors, actuators) maintenance (*iii*) valve stiction (*iv*) model plant mismatch (MPM) (*v*) stochastic disturbances. Hagglund (1999) developed a procedure for the automatic detection of sluggish control loops obtained from conservatively tuned controllers. Sensor fault detection and isolation in the process control community (also known as sensor validation) has also been an active area of research Dunia et al. (1996); Gertler et al. (2000); Tong and Crowe (1995); Deckert et al. (1977). A detailed survey on sensor validation has been given by Crowe (1997). More recent work in this area has been addressed by Qin and Li (1999) and Qin and Li (2001). A common source of oscillation is a limit cycle caused by a control valve with a deadband or excessive static friction. A process variable oscillating for that reason can readily propagate the oscillation to other variables and disturb other control loops, hence causing a plant-wide disturbance. A focus upon non-linear root causes can thus be justified because valve friction causes the majority of cases, according to reported surveys Bialkowski (1993); Ender (1993). Several authors have addressed the detection of oscillatory measurements in process data. There is a wealth of information in the literature to diagnose the poor control loop performance due to valve stiction, improper tuning of controllers and sensor faults.

Jiang and W. Li (2006) proposed a new scheme to detect and isolate MPM for multivariate dynamic systems. In their work, the MPM problem is formulated in the state-space domain, as is widely done in the design and implementation of model-predictive controllers (MPC). The specific issue addressed therein was to identify which among the state-space matrices had to be re-estimated in order to account for significant plant deviations from

[⋆] This paper is based on the article published in Industrial and Engineering Chemistry Research, 2010, 65 (2), pp 660-674. Copyright © 2010 Elsevier Ltd. This version is intended for members of the NIL project groups.

its nominal state. Three MPM detection indices (MDIs) were proposed to detect the MPM for that purpose. A shortcoming of their work is that changes in stae-space matrices cannot be directly translated to changes in gain, time constant or delay of the process. In addition, delay mismatches become difficult to detect with a state-space representation since such mismatches cause either an increase or decrease in the order of the system depending on an increase or decrease in delay of the process. The input-output representation, on the other hand, provides a suitable framework for directly attributing poor loop performance to changes in process characteristics such as gain, time-constant and delay. To the best knowledge of the authors, there is no method in the literature that can diagnose and quantify the effects of gain, time constant and delay mismatches on control loop performance from the closed-loop routine operating data. It may be noted that the effect of MPM on controller design and control loop performance is an evolved area of study and implementation under robust control. However, there is hardly any evidence of an analysis of the direct impact of process charactersitics on loop performance in model-based control loops.

In this article, the diagnosis of poor control loop performance due to model plant mismatch in control loops is addressed. The emphasis of this work is on providing measures to detect changes in process characteristics that lead to performance degradation. The problem setting is in the IMC framework. The diagnosis methodology is developed in the frequency-domain. A new measure, *plant-model ratio* (PMR) is introduced with the purpose of quantifying MPM. Conventionally MPM has been quantified as $\triangle G = G_p - G_m$ where G_p is process transfer function and G_m is model transfer function. While this definition of MPM (and its norms) is useful for robust control design, its utility in the problem of interest in this work is limited. On the other hand, the PMR defined by $\dfrac{G_p(j\omega)}{G_m(j\omega)}$ allows one to obtain a unique mapping between changes in process characteristics and signatures of *PMR*. Based on this fact, it is proposed to detect signatures in *PMR* which can be then uniquely identified with changes in gain, time-constant and/or time-delay. The ratio measure can be calculated from routine operating date. A method to estimate *PMR* from routine data is proposed. The estimation method uses the cross-spectral analysis of set-point, plant output and model output with the assumption that the set-point contains at lesat a pulse change. The foregoing assumption is satisfied to a large extent in industrial loops given that the set-points experience changes due to changing market demands and productivity constraints. The scope of this work is restricted to single-input, single-output (SISO) systems.

The rest of the paper is organized as follows. Section 2 begins with the problem formulation followed by a presentation of the concept of the proposed method to diagnose the model plant mismatch. Section 3 discusses the procedure for estimating the *PMR* which is a key step in the proposed method. Results from simulation studies to demonstrate the proof-of-concept are discussed in Section 4. Concluding remarks appear in Section 5.

2. PROBLEM STATEMENT AND PROPOSED METHODOLOGY

The task of diagnosing poorly performing controllers is a challenging one. The major causes of poor control loop performance in model based control loops are (i) improper controller tuning (ii) actuator/process non-linearities (iii) stochastic disturbances (iv) sensor faults and (v) MPM. Keeping inline with the focus of this work, the last of the aforementioned factors is taken up for study. The question of interest is: *given poor performance of a (model-based) control loop and that it is solely due to MPM, then which among the process characteristics, namely, gain, time-constant and delay have undergone a significant change*? Although the probe begins with the assumption of a unstrctured mismatch, the method developed here can be applied to situations with structure mismatch as well. In the latter case, the question will then translate to: *given poor performance of a (model-based) control loop and that it is solely due to MPM, then which parts of the model, namely, gain, time-constant(s) and delay require an update*? This task is of immense value in industry and arises in the exercise of model updation. The problem setting is in the IMC framework. The idea is that the results here can be extended to the more commonly applied MPC-based control schemes. It may be noted that the scope of this work is restricted to IMC-based SISO loops.

The internal model control (IMC) framework provides an elegant way of re-parametrizing the conventional feedback controller Morari and Zafiriou (1989). An increasing number of practitioners have started using the IMC tuning rules for designing PID controllers. A primary benefit of using IMC-type schemes in the context of performance asssessment is that in situations where the performance degradation is due to changing plant conditions, the presence of the model online enables us to obtain an estimate of such deviations from routine operating data. The key point is that the model predictions are also available along with the plant output, from where one can obtain an estimate of the MPM. It is not to be forgotten that the difference between plant output and the model prediction also contains the effect of disturbances.

Figure 1 shows a schematic of IMC configuration where $G_p(z^{-1})$, $G_m(z^{-1})$, $Q(z^{-1})$ and $G_d(z^{-1})$ denote process, model, controller and disturbance transfer functions respectively. The observed output is denoted by $y[k]$, while the prediction of the model is denoted by $y_m[k]$ respectively. The problem statement and methodology is presented for discrete-time systems. However, these results are equally valid for continuus-time IMC systems as well with some appropriate minor changes. It is reiterated here that any performance degradation is attributed to MPM, *i.e.*, there is no loss in performance due to valve stiction, stochastic disturbances, improper tuning of controllers and sensor faults. Traditionally, MPM has been characterized by $\triangle G = G_p - G_m$, which

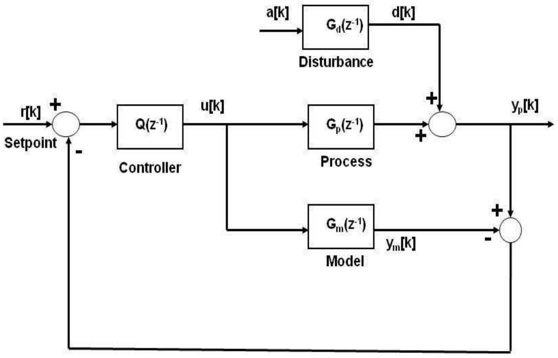


Figure 1. Schematic represenation of internal model control

can never be zero due to lack of accurate process knowledge (modelling uncertainties). The uncertainties can be either in the structure, whence then $\triangle G$ represents a structural mismatch or in the parameters, namely, gain, time constant(s) and delay, in which case $\triangle G$ is said to represent unstructured mismatch. We consider primarily the latter situation in the treatment of the problem under study.

To begin with, it is proposed to replace the conventional definition of mismatch with a measure, known as the plant-model ratio (PMR), based on the frequency response functions (FRF) of the plant and model,

$$G_{PMR}(e^{j\omega}) = \frac{G_p(e^{j\omega})}{G_m(e^{j\omega})} \tag{1}$$

The quantity $G_{PMR}(e^{j\omega})$ is clearly complex-valued. The measure defined in equation (1) has a few advantages over the traditional $\triangle G(e^{j\omega})$. The key advantage is understood by first writing a polar representation for G_{PMR},

$$\begin{aligned} G_{PMR}(e^{j\omega}) &= \frac{|G_p(e^{j\omega})|e^{j\angle\bar{G}_p(e^{j\omega})}e^{-jD_p\omega}}{|G_m(e^{j\omega})|e^{j\angle\bar{G}_m(e^{j\omega})}e^{-jD_m\omega}} \\ &= M(\omega)e^{j(\triangle P(\omega))} \end{aligned} \tag{2}$$

where $\bar{G}_p$ and $\bar{G}_m$ represent the delay-free parts of the plant and model transfer functions respectively. Equation (3) is central to the proposed method from where conditions to diagnose the type(s) of mismatch are formulated in later part of this section. From equation (3), $M(\omega)$ contains the information on mismatch in the magnitude of the FRFs of the plant and model, while $\triangle P(\omega)$ contains the mismatch in the phase of the delay-free parts and the delays. Equivalently, in the absence of any mismatch between the plant and the model, $G_{PMR}(\omega) = 1$, implying $M(\omega) = 1$ and $\triangle P(\omega) = 0$. An immediate utility of the above representation is that delay mismatch, denoted by $\triangle D = D_p - D_m$ only influences $\triangle P(\omega)$, but not $M(\omega)$. Similarly, any mismatch in the gain is contained in $M(\omega)$ only, while time-constant mismatches affect both quantities. Thus, the delay mismatch problem is decoupled from the time-constant mismatch. On the other hand, writing a polar form of representation for the traditional definition of mismatch yields,

$$\begin{aligned} \triangle G(e^{j\omega}) &= |G_p(e^{j\omega})|e^{j\angle\bar{G}_p(e^{j\omega})}e^{-jD_p\omega} - |G_m(e^{j\omega})|e^{j\angle\bar{G}_m(e^{j\omega})}e^{-jD_m\omega} \\ &= A(\omega)e^{jB(\omega)} \end{aligned}$$

It is clear from the above equation that the quantities $A(\omega)$ and $B(\omega)$ are both affected by mismatches in delay and time-constant(s). Moreover, the manner in which they affect the amplitude and phase of $\triangle G(e^{j\omega})$ is not straightforward. Compare this situation with the one arising out of the porposed measure. It is clear that $G_{PMR}(\omega)$ is a better representation of the mismatch than $\triangle G(\omega)$ in the context of diagnosis of loop performance.

In the presentation to follow, we show that a unique mapping exists between changes in process characteristics and the amplitude and phase of $G_{PMR}(\omega)$. The theoretical results are presented for FOPTD processes and models for the sake of simplicity and brevity. The ideas, nevertheless, as shown later, turn out to be valid for all higher-order processes in a straightforward way so long as there is *no mismatch in the structure.*

The discrete domain transfer functions of FOPTD plant and FOPTD model are expressed as

$$G_p(z^{-1}) = \frac{K_p(1 - e^{-\frac{T_s}{\tau_p}})}{1 - z^{-1}e^{-\frac{T_s}{\tau_p}}}z^{-D_p} = \frac{K_p(1 - a_p)}{1 - a_p z^{-1}}z^{-D_p} \tag{3}$$

$$G_m(z^{-1}) = \frac{K_m(1 - e^{-\frac{T_s}{\tau_m}})}{1 - z^{-1}e^{-\frac{T_s}{\tau_m}}}z^{-D_m} = \frac{K_m(1 - a_m)}{1 - a_m z^{-1}}z^{-D_m} \tag{4}$$

where T_s is sampling time ($T_s = 1$ throughout), K_p, τ_p and D_p are process gain, process time constant and process time delay (in *number of samples*) and corresponding model parameters are K_m, τ_m and D_m.

2.1 Gain mismatch

When the loss in performance is only due to gain mismatch, *i.e.,* $K_p \neq K_m$ and $\tau_p = \tau_m$, $D_p = D_m$, equation (1) simplifies to

$$G_{PMR}(\omega) = \frac{K_p}{K_m} \tag{5}$$

From equation (5), we can formulate the following condition for diagnosing the gain mismatch

$$M(\omega) \neq 1 \; \forall\omega \tag{6}$$

$$\triangle P(\omega) = 0 \; \forall\omega \tag{7}$$

The presence of gain mismatch in the control loop is diagnosed if the estimates of the magnitude ratio and phase difference satisfy equations (6)-(7). The right-hand side constants of these equations are constants (independent of frequency) for gain mismatch. It may be noted that this result is true regardless of the order of the process/model and in the absence of any structural mismatch. Thus, this is a general result for all situations with unstructured uncertainties. Also, in order to de-

$M(\omega)$	$\Delta P(\omega)$
$M(\omega)\|_{\omega=0} = 1$	$\Delta P(\omega)\|_{\omega=0,\pi} = 0$
$M(\omega)\|_{\omega=0} \neq M(\omega)\|_{\omega\neq 0}$	$\|\Delta P(\omega)\|_{max} > 0$

Table 1. Diagnostic conditions for time-constant mismatch

termine the nature of gain mismatch, one further notes that

$$M(\omega) \gtreqless 1 \text{ if } K_p \gtreqless K_m$$

2.2 Time constant mismatch

In the presence of time constant mismatch, *i.e.*, $\tau_p \neq \tau_m$ and $K_p = K_m$, $D_p = D_m$, equation (1) reduces to

$$G_{PMR}(\omega) = \frac{(1-a_p)(1-e^{-j\omega}a_m)}{(1-a_m)(1-e^{-j\omega}a_p)} = M(\omega)e^{j\Delta P(\omega)} \tag{8}$$

The expressions for $M(\omega)$ and $\Delta P(\omega)$ then follow

$$M(\omega) = \frac{(1-a_p)}{(1-a_m)}\sqrt{\frac{1-2a_m\cos\omega + a_m^2}{1-2a_p\cos\omega + a_p^2}} \tag{9}$$

$$\Delta P(\omega) = \tan^{-1}\left[\frac{a_m\sin\omega}{1-a_m\cos\omega}\right] - \tan^{-1}\left[\frac{a_p\sin\omega}{1-a_p\cos\omega}\right] \tag{10}$$

The expressions in equations (9) and (10) at a first glance appear complicated; however, they lead to a set of simple necessary and sufficient conditions listed in Table 1 for the time-constant mismatch to be the cause of overall MPM. The presence of time-constant mismatch is diagnosed from either $M(\omega)$ **or** $\Delta P(\omega)$ as provided in Table 1.

Once the source of mismatch has been identified as due to mismatch in time-constants, one can then determine the nature of deviation of τ_p from τ_m using the condition (can be derived from equation (9))

$$M(\omega)|_{\omega=\pi} = \lesseqgtr 1 \quad \text{whenever} \quad \tau_p \gtreqless \tau_m$$

2.3 Delay mismatch

If the only source of mismatch is the mismatch in delay, *i.e.*, $D_p \neq D_m$ and $\tau_p = \tau_m$, $K_p = K_m$, equation (1) simplifies to

$$G_{PMR}(\omega) = e^{-(D_p - D_m)j\omega} \tag{11}$$

leading to the following condition for the diagnosis of delay mismatch

$$M(\omega) = 1 \;\; \forall \;\; \omega \tag{12}$$

$$\triangle P(\omega) = \alpha\omega \;\; \forall \;\; \omega \tag{13}$$

where $\alpha = \Delta D = D_m - D_p$.
Equations (12) and (13) show that delay mismatches in the model based control loops do not cause deviations in the magnitudes of the FRF of the process and model, but introduce a phase difference in the FRFs that varies linearly with respect to the frequency. Since D_m and D_p represent the *sample* delays, the minimum value of $|\alpha|$ is unity for any mismatch in delays to manifest in the sampled data. Therefore, it is appropriate to impose $|\alpha|_{\min} = 1$ in addition to the conditions specified in equations (12) and (13). We now turn to establishing conditions for various other combinations.

2.4 Gain and time constant mismatch

In this case, $\tau_p \neq \tau_m$, $K_p \neq K_m$ and $D_p = D_m$. Equation (1) then reduces to

$$G_{PMR}(\omega) = \frac{K_p}{K_m}\frac{(1-a_p)(1-e^{-j\omega}a_m)}{(1-a_m)(1-e^{-j\omega}a_p)} \tag{14}$$

$M(\omega)$ and $\Delta P(\omega)$ are now written as follows,

$$M(\omega) = \frac{K_p}{K_m}\frac{(1-a_p)}{(1-a_m)}\sqrt{\frac{1-2\cos\omega a_m + a_m^2}{1-2\cos\omega a_p + a_p^2}} \tag{15}$$

$$\Delta P(\omega) = \tan^{-1}\left[\frac{a_m\sin\omega}{1-a_m\cos\omega}\right] - \tan^{-1}\left[\frac{a_p\sin\omega}{1-a_p\cos\omega}\right] \tag{16}$$

From equations (15) and (16), we can formulate the following conditions for diagnosing the presence of gain and time-constant mismatch.

$M(\omega)$	$\Delta P(\omega)$
$M(\omega)\|_{\omega=0} \neq 1$	$\Delta P(\omega)\|_{\omega=0,\pi} = 0$
$M(\omega)\|_{\omega=0} \neq M(\omega)_{\omega\neq 0}$	$\|\Delta P(\omega)\|_{max} > 0$

Table 2. Conditions for the diagnosis of gain and time-constant mismatch

The conditions in this case is a combination of the conditions for mismatches in individual parameters (gain and time-constant). Similar situations arise in other combinations of parametric mismatches as seen below.

2.5 Gain and delay mismatch

This situation arises when, $K_p \neq K_m$, $D_p \neq D_m$ and $\tau_p = \tau_m$, reducing equation (1) to

$$G_{PMR}(\omega) = \frac{K_p}{K_m}e^{-(D_p - D_m)j\omega} \tag{17}$$

From equation (17), we can formulate the following condition for diagnosing the gain-and-delay mismatch

$$\begin{aligned} M(\omega) &\neq 1 \;\; \forall\omega \\ \triangle P(\omega) &= \alpha\omega \;\; \forall\omega \end{aligned} \tag{18}$$

where $\alpha = \Delta D = D_m - D_p$.

In order to determine the nature of gain and delay mismatch, one further notes that

$$M(\omega) \gtreqless 1 \text{ if } K_p \gtreqless K_m$$
$$\alpha \gtreqless 0 \text{ if } D_m \gtreqless D_p$$

2.6 Time constant and delay mismatch

With the above assumption,$\tau_p \neq \tau_m$, $D_p \neq D_m$ and $K_p = K_m$; then equation (1) takes the form

$$G_{PMR}(\omega) = \frac{(1-a_p)(1-e^{-j\omega}a_m)}{(1-a_m)(1-e^{-j\omega}a_p)}e^{-j\Delta D\omega} \tag{19}$$

$M(\omega)$ and $\Delta P(\omega)$ are now written as follows,

$$M(\omega) = \frac{(1-a_p)}{(1-a_m)}\sqrt{\frac{1-2\cos\omega a_m + a_m^2}{1-2\cos\omega a_p + a_p^2}} \tag{20}$$

$$\Delta P(\omega) = \tan^{-1}\left[\frac{a_m \sin\omega}{1-a_m\cos\omega}\right] - \tan^{-1}\left[\frac{a_p\sin\omega}{1-a_p\cos\omega}\right] + \omega(\Delta D) \tag{21}$$

From equations (20) and (21), one arrives at the following diagnostics listed in Table 3 in this case:

2.7 Gain, time constant and delay mismatch

When there exists mismatch in all gain, time constant and delay *i.e.,* $K_p \neq K_m$, $\tau_p \neq \tau_m$ and $D_p \neq D_m$, equation (1) simplifies to

$$G_{PMR}(\omega) = \frac{K_p}{K_m}\frac{(1-a_p)(1-e^{-j\omega}a_m)}{(1-a_m)(1-e^{-j\omega}a_p)}e^{-j\Delta D\omega} \tag{22}$$

$M(\omega)$ and $\Delta P(\omega)$ are now written as follows,

$$M(\omega) = \frac{K_p}{K_m}\frac{(1-a_p)}{(1-a_m)}\sqrt{\frac{1-2\cos\omega a_m + a_m^2}{1-2\cos\omega a_p + a_p^2}} \tag{23}$$

$$\Delta P(\omega) = \tan^{-1}\left[\frac{a_m \sin\omega}{1-a_m\cos\omega}\right] - \tan^{-1}\left[\frac{a_p\sin\omega}{1-a_p\cos\omega}\right] + \omega(\Delta D) \tag{24}$$

From equations (23) and (24), the following conditions are formulated for diagnosing the presence of gain, time constant and delay mismatch together.

The signatures of $M(\omega)$ and $\Delta P(\omega)$ are unique for every type of mismatch. For an easy understanding of the proposed approach, pictoral representations of the theoretical conditions for the diagnosis of different combination of mismatches are given in Figures (2) and (3). It turns out that the diagnostic conditions listed above can be largely simplified to a condensed stepwise procedure, which examines three critical aspects, namely, (i) the value of $M(\omega)|_{\omega=0}$ which provides an estimate of K_p/K_m, (ii) the flatness of $M(\omega)$ indicative of the presence of $\triangle\tau$ and (iii) linearity of $\triangle P(\omega)$ indicative of a delay mismatch. The procedure is presented in Table 6. The condensed procedure can be applied to higher-order processes / models as well with the only difference that the second aspect, *i.e.*, the flatness of $M(\omega)$ will be indicative of mismatch in one or more time-constants of the process. The reader may note with interest that the use of the conventional definition of MPM $\triangle G = G_p - G_m$ would not lead to a formulation of such simple signatures as has been obtained with the proposed PMR.

In the following section, a method to estimate PMR from routine operating data is given.

3. ESTIMATION OF PMR

In this section, we provide a method to estimate PMR from routine operating data in model-based control loops. For this purpose, onsider a linear time invariant (LTI) SISO feedback control system opertaing under an IMC scheme. The setpoint, process input, process output, model output and disturbance are denoted by $r[k]$, $u[k]$, $y_p[k]$, $y_m[k]$ and $d[k]$ respectively. The actuator and sensor dynamics are neglected for the remainder of the presentation. The expressions of process and model outputs can be written for the IMC structure as follows:

$$y_p[k] = G_p(q)u[k] + d[k] = G_p(q)u[k] + G_d(q)e[k] \tag{25}$$
$$y_m[k] = G_m(q)u[k] \tag{26}$$

where

$$G_p(q) = \sum_{k=0}^{\infty} g_p[k]q^{-k}$$

and

$$G_d(q) = \sum_{k=0}^{\infty} g_d[k]q^{-k}$$

In a noise-free environment (*i.e.*, when $d[k] = 0$), the natural estimate of PMR is

$$\frac{G_p(\omega)}{G_m(\omega)} = \frac{Y_p(\omega)}{Y_m(\omega)}$$

where $Y_p(\omega)$ and $Y_m(\omega)$ are the Fourier Transforms of y_p and y_m respectively. In practice, however, due to the presence of disturbances, this will lead to a poor estimate of the PMR. In order to overcome the effect of disturbances, a smoothed estimate is desired. This issue is reminscent of the estimation of the transfer function using the ETFE versus the smoothed estimate involving the cross-spectra (Lennart, 1999). Based on such ideas, equations 25 and 26 multiplied by $r[k-l]$ followed by an expectation operation. Subsequently, taking Fourier Transforms on both sides of the equation leads to equations involving cross-spectra

$$\Phi_{y_p r}(\omega) = G_p(e^{j\omega})\Phi_{ur}(\omega) + G_d(e^{j\omega})\Phi_{ar}(\omega) \tag{27}$$
$$\Phi_{y_m r}(\omega) = G_m(e^{j\omega})\Phi_{ur}(\omega) \tag{28}$$

Noting the fact that the set-point and the disturbance are ideally uncorrelated, we arrive at the expression for the estimate of PMR as

$$\frac{\Phi_{Y_p R}(\omega)}{\Phi_{Y_m R}(\omega)} = \frac{\hat{G}_p(e^{j\omega})}{\hat{G}_m(e^{j\omega})} = \hat{G}_{PMR}(\omega) \tag{29}$$

where the $\hat{}$ is introduced as a natural consequence in estimation problems due to finite sample size and the fact that $\Phi_{rd}(\omega)$ is numerically non-zero (but a small value) in practice.

The estimation method provided above may be misinterpreted as a method for identifying the process model from closed-loop operating data since $G_m(e^{j\omega})$ is a known quantity. However, the estimate is only used to identify signatures relevant to the detection of MPM, but not to identify the process parameters. The key idea in this work is to propose a method to detect changes (and the direction of such changes) in process parameters based on routine operating data without having to explicitly estimate the process parameters. Table 6 is reflective of this fact. The flatness of $M(\omega)$ and linearity of $\triangle P(\omega)$ are the key signatures that are sought, rather than the estimates of τ_p and/or D_p from operating data. The signature-based approach of the proposed method demands only a minimal excitation, *i.e.*, a pulse excitation in the set-point whereas an explicit estimation of process parameters imposes stronger excitation requirements either in the set-point or in a dither signal as is well-known in closed-loop identification. It may be noted, however, that since gain can be estimated merely from step response data. Consequently, one of the conditions listed in Table 6 makes use of the gain estimate by examining the value of $M(\omega)|_{\omega=0}$.

4. SIMULATIONS

A control system consisting of a process characterized by the transfer function $G_p(z^{-1}) = \frac{K_p(1-a_p)}{1-z^{-1}a_p} z^{-D_p}$ and model $G_m(z^{-1}) = \frac{K_m(1-a_m)}{1-z^{-1}a_m} z^{-D_m}$ is simulated with an IMC scheme, with pulse type set-point changes. The tuning parameter, λ and the model parameters K_m, τ_m and D_m are fixed at 2, 1, 10 and 5 respectively. The closed-loop system is simulated under four different situations of MPM, namely, *gain mismatch*, *time constant mismatch, delay mismatch* and a mismatch in all three parameters. The mismatches are expressed in terms of percentages for simplicity and are defined as follows

$$\delta K = \frac{K_p - K_m}{K_m} \times 100 \tag{30}$$

$$\delta \tau = \frac{\tau_p - \tau_m}{\tau_m} \times 100 \tag{31}$$

$$\delta D = \frac{D_p - D_m}{D_m} \times 100 \tag{32}$$

To determine the degradation in performance, the ISE is used as a measure of the performance of the closed-loop system. The ISE value is calculated for the zero MPM case and compared with the value corresponding to each case of mismatch. The extent of degradation, evidently, depends on the magnitude of MPM.

4.1 Gain mismatch

Two different levels of gain mismatch at 10% and 50% are considered. The cross spectra between process output, $y_p[k]$ and setpoint, $r[k]$ and model output, $y_m[k]$ and $r[k]$ are estimated. The magnitude ratio $M(\omega)$ and phase difference $\Delta P(\omega)$ are computed from the estimate of PMR as given in equation (29). Figures 6(a) and 6(b) show the estimates of these measures for the two different levels of gain mismatch. Though visual inspection of Figures (5(a)) and (5(b)) clearly indicates the presence of gain mismatch, non-existence of time-constant mismatch and delay mismatch can not be ruled out. Proceeding with the first step outlined in Table (7), the value of $M(\omega)|_{\omega=0}$ is 1.1009 when $\delta K = 10\%$ and 1.5001 when $\delta K = 50\%$ clearly indicating the presence of gain mismatch. From the values $|\alpha_1| = 0.0009 < \alpha_{c\tau}$, $|\alpha_2| = 0.001 < \alpha_{cD}$ for case 1(a) and $|\alpha_1| = 0.0009 < \alpha_{c\tau}$, $|\alpha_2| = 0.0002 < \alpha_{cD}$ for case 1(b), it can be concluded that there exists only gain mismatch. Further, the extent of mismatch can be calculated from these values by calculating the deviation from the no MPM case, *i.e.*, $M(\omega)|_{\omega=0} = 1$, which yield 10.09% and 50.01% respectively. These values are, as expected, in agreement with the true values.

4.2 Time constant

Two different levels of time mismatch at 10% ($\tau_p = 11$) and 50% ($\tau_p = 15$) are considered. Visual inspection of Figures (7(a)) and (7(b)) does not reveal any information of MPM. The magnitude ratio, $M(\omega)$ and phase difference, $\Delta P(\omega)$ are computed from the estimate of PMR as given in equation (29). Both these quantities are plotted in Figures (8(a)) and (8(b)). The stepwise procedure outlined in Table 7 is followed. The values of $M(\omega)|_{\omega=0} = 0.997$, $|\alpha_1| = 0.018 > \alpha_{c\tau}$ and $|\alpha_2| = 0.007 < \alpha_{cD}$ when $\delta\tau = 10\%$ and $M(\omega)|_{\omega=0} = 0.9983$, $|\alpha_1| = 0.023 < \alpha_{c\tau}$ and $|\alpha_2| = 0.033 < \alpha_{cD}$ when $\delta\tau = 50\%$ clearly indicate the presence of time-constant mismatch. Further, the sign of α_1 in both the cases indicates that τ_p is greater than τ_m in both the cases.

4.3 Delay mismatch

The system is simulated for the delay mismatch of 20% ($D_p = 6$) and 40% ($D_p = 7$). Visual inspection of Figures (9(a)) and (9(b)) does not reveal any information of MPM. The cross spectra, $\Phi_{Y_pR}(\omega)$, $\Phi_{Y_mR}(\omega)$ are estimated which are subsequently used to compute $M(\omega)$ and $\Delta P(\omega)$. Both these quantities are plotted in Figures 10(a) and 10(b). The stepwise procedure outlined in Table 7 is followed. The values of $M(\omega)|_{\omega=0} = 0.9981$, $|\alpha_1| = 0.0009(< \alpha_{c\tau})$ and $|\alpha_2| = 0.9933(< \alpha_{cD})$ when $\delta D = 20\%$ and $M(\omega)|_{\omega=0} = 0.9963$, $|\alpha_1| = 0.0001(< \alpha_{c\tau})$ and $|\alpha_2| = 2.003(< \alpha_{cD})$ when $\delta D = 40\%$ are representative of the presence of only delay mismatch. The negative sign of α_2 clearly indicates that the plant delayD_p is greater than the model delay D_m in both cases.

4.4 Gain, Time constant & Delay mismatch

The system is simulated by setting δK at 10%, $\delta\tau$ at 50% and δD at 20%. $M(\omega)$ and $\Delta P(\omega)$ are estimated and plotted in Figure 11(b). Visual inspection of Figure 11(a) confirms only the presence of gain mismatch. But, the values of $M\omega)|_{\omega=0} = 1.1816$, $|\alpha_1| = 0.018(< \alpha_{c\tau})$ and $|\alpha_2| = 0.9581(< \alpha_{c\tau})$ clearly indicate the presence of gain, time-constant and delay mismatches together. The extent of gain mismatch is calculated from $M\omega)|_{\omega=0} = \dfrac{K_p}{K_m} = 1.1816$, *i.e.*, $\delta K = 18.16\%$.. The sign of $\alpha_1$implies that the process time-constant is greater than that of the model whereas the sign of $\alpha_2$implies that the plant delay is greater than that of the model.

4.5 SOPTD Process/Model

To show the applicability of the proposed method to higher order systems, a control system consisting of a SOPTD process and a SOPTD model is simulated for a pulse type set-point changes in an IMC scheme. The values of the tuning parameter, λ, model parameters, K_m, τ_{m1}, τ_{m2} and D_m are fixed at 0.2, 1, 5, 7, 3 respectively. The simulation is performed by setting $\delta K = 20\%$, $\delta\tau_1 = 20\%$, $\delta\tau_2 = 20\%$, $\delta D = 33\%$. The time-domain trends of plant- and model-outputs are shown in Figure 12(a). The plots of $M(\omega)$ and $\Delta P(\omega)$ are shown in Figure 12(b). The values of $M(\omega)|_{\omega=0} = 1.1952$, $|\alpha_1| = 0.066(< \alpha_{c\tau})$, and $\alpha_2 = 1.053(< \alpha_{cD})$ clearly reveal the presence of mismatches in gain, time-constant and delay together. K_p/K_m is found from the value of $M(\omega)|_{\omega=0} = 1.1952 \Longrightarrow \delta K = 19.5\%$. Further, the signs of $\alpha_1$and α_2 indicate the direction of deviations in time-constant and delay mismatches respectively. To summarize, the findings are $K_p > K_m$,$\tau_p > \tau_m$ and $D_p < D_m$.

5. CONCLUSIONS

In this work, the problem of characterizing the deviations of process parameters from that of the model has been addressed. The major contribution of this work is the formulation of the MPM characterization in the frequency domain for the identification of specific signatures to distinguish among mismatches in gain, time-constant and delay. The current work introduced a quantity $G_p(\omega)/G_m(\omega)$, termed as the plant-model ratio (PMR) as against the conventional $G_p - G_m$ to quantify MPM. The theoretical results show that the advantage of the proposed PMR is the ease of representation in the complex frequency domain. The amplitude part of the PMR contains the effects of gain and time-constant mismatches, while the phase component contains the effects of mismatch in delay and time-constant.

The key outcome of this work is a simple three-step procedure to identify mismatches in gain, time-constant and delay from routine oeprating data through the use of PMR. It is foreseen that the work presented lays the foundations for methods to (i) MPC-based SISO control schemes and to (ii) identify significant deviations in specific process parameters (*i.e.*, of a subsystem) of a multivariable system.

REFERENCES

W.L. Bialkowski. Dreams versus reality: a view from both sides of the gap. *Pulp and Paper Canada*, 94: 19–27, 1993.

C. Crowe. Data reconciliation-progress and challenges. *Process Control*, 6:89–98, 1997.

J. Deckert, M. Desai, J. Deyst, and A. Willsky. F-8 dfbw sensor failure identification using analytical redundancy. *IEEE*, 22:796–803, 1977.

R. Dunia, J. Qin, T. Edgar, and T. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE*, 42:2797–2812, 1996.

D. Ender. Process control performance: not as good as you think. *Control Engineering*, 40:180, 1993.

J. Gertler, W. Li, Y. Huang, and T. McAvoy. Isolation-enhanced principal component analysis. *Process Control*, 10:459–469, 2000.

T. Hagglund. Automatic detection of sluggish control loops. *Control Engineering Practice*, 7:1505–1511, 1999.

H. Jiang and S. Shah W. Li. Detection and isolation of model-plant mismatch for multivariate dynamic systems. *Fault Detection, Supervision and Safety of Technical Processes*, 12:1396–1401, 2006.

D.J. Kozub and C.E. Garcia. Monitoring and diagnosis of automated controllers in the chemical process industries. Technical report, AIChE Annual Meeting, 1993.

L. Lennart. *System Identification Theory for the User*. Prentice Hall, 1999.

M. Morari and E. Zafiriou. *Robust process control*. Prentice Hall, 1989.

S. Qin and W. Li. Detection and identification of faulty sensors in dynamic processes. *AIChE*, 47:1581–1593, 2001.

S. Qin and W. Li. Detection, identification, and reconstruction of faulty sensors with maximized sensitivity. *AIChE*, 45:1963–1976, 1999.

H. Tong and C. Crowe. Detection of gross errors in data reconciliation by principal component analysis. *AIChE*, 41:1721–1722, 1995.

$M(\omega)$	$\Delta P(\omega)$
$M(\omega)\vert_{\omega=0} = 1$ $M(\omega)\vert_{\omega=0} \neq M(\omega)\vert_{\omega\neq0}$	$\Delta P(\omega) = f(\tau,\omega) + \alpha\omega$, $f(\tau,\omega) \to 0$ for $\omega \to \pi$

Table 3. Conditions for the diagnosis of delay and time-constant mismatch

$M(\omega)$	$\Delta P(\omega)$
$M(\omega)\vert_{\omega=0} \neq 1$ $M(\omega)\vert_{\omega=0} \neq M(\omega)_{\omega\neq0}$	$\Delta P(\omega)\vert = f(\tau,\omega) + \alpha\omega$, $f(\tau,\omega) \to 0$ for $\omega \to \pi$

Table 4. Conditions for the diagnosis of gain, time-constant and delay mismatches

	Assessment procedure	Diagnosis of MPM
Step 1	$M(\omega)\vert_{\omega=0} \neq 1$	$K_p \neq K_m$ else $K_p = K_m$
Step 2	Test of flatness (zero slope) of $M(\omega)$	If flat, $\tau_p = \tau_m$, else $\tau_p \neq \tau_m$
Step 3	Linearity check of $\Delta P(\omega)$	Linear: $D_p \neq D_m$, else $D_p = D_m$

Table 6. Theoretical conditions for the diagnosis of MPM in model-based control loops

	Assessment procedure	Diagnosis of MPM
Step 1	$M(\omega)\vert_{\omega=0} \geq 1.05$ or $M(\omega)\vert_{\omega=0} \leq 0.95$	$K_p \neq K_m$, else $K_p = K_m$
Step 2	Test of flatness (zero slope) of $M(\omega)$: $M(\omega) = \alpha_{c\tau}\omega + \beta$	If flat, *i.e.*, $\vert\alpha_{c\tau}\vert \leq 0.001$, $\tau_p \neq \tau_m$, else $\tau_p = \tau_m$
Step 3	Linearity of $\Delta P(\omega)$: $\Delta P(\omega) = \alpha_{cD}\omega$	If $\vert\alpha_{cD}\vert \geq 0.9$ (Linear): $D_p \neq D_m$, else $D_p = D_m$

Table 7. Empirical (data-based) conditions for the diagnosis of MPM in model-based control loops

MPM		Assessment			Diagnosis
		$M(\omega)\vert_{\omega=0}$	α_1	α_2	
Case 1	a	1.1099	−0.0009	−0.0010	$K_p > K_m$, $\delta\tau = \delta D = 0$
	b	1.5091	−0.0009	−0.0003	$K_p > K_m$, $\delta\tau = \delta D = 0$
Case 2	a	0.9970	−0.0184	0.00740	$\tau_p > \tau_m$, $\delta K = \delta D = 0$
	b	0.9853	−0.0235	0.03380	$\tau_p > \tau_m$, $\delta K = \delta D = 0$
Case 3	a	0.9981	0.0009	−0.9930	$D_p > D_m$, $\delta K = \delta\tau = 0$
	b	0.9963	0.0001	−2.0030	$D_p > D_m$, $\delta K = \delta\tau = 0$
Case 4		1.1816	0.0184	−0.9581	$K_p > K_m$, $\tau_p > \tau_m$, $D_p > D_m$
Case 5		1.1952	0.066	1.05300	$K_p > K_m$, $\tau_p > \tau_m$, $D_m > D_p$

Table 5. Simulation results for the diagnosis of different occurences of model plant mismatch

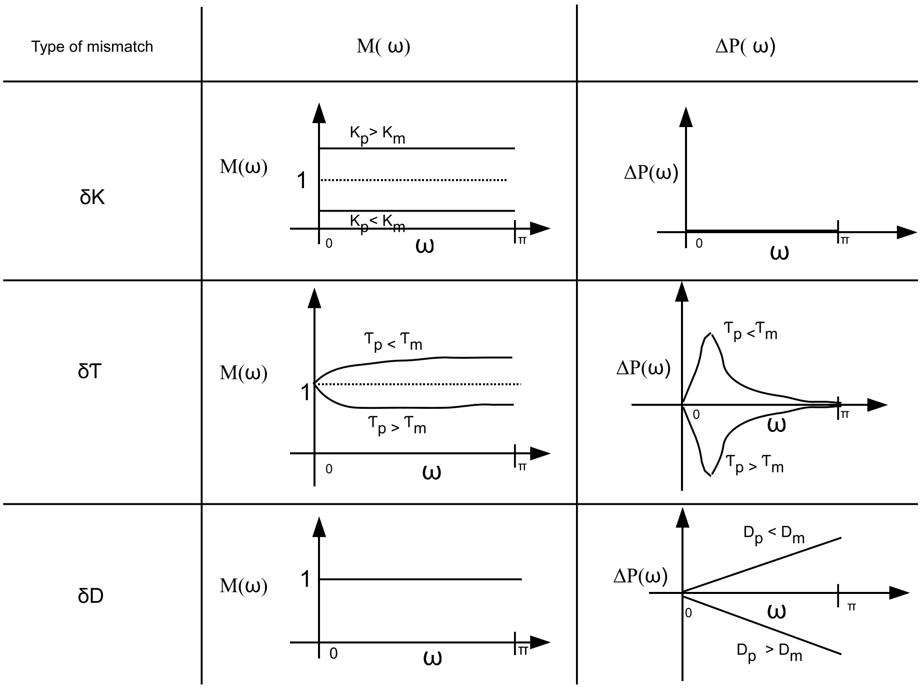


Figure 2. Conditions to diagnose the type of mismatch among gain, time-constant and delay

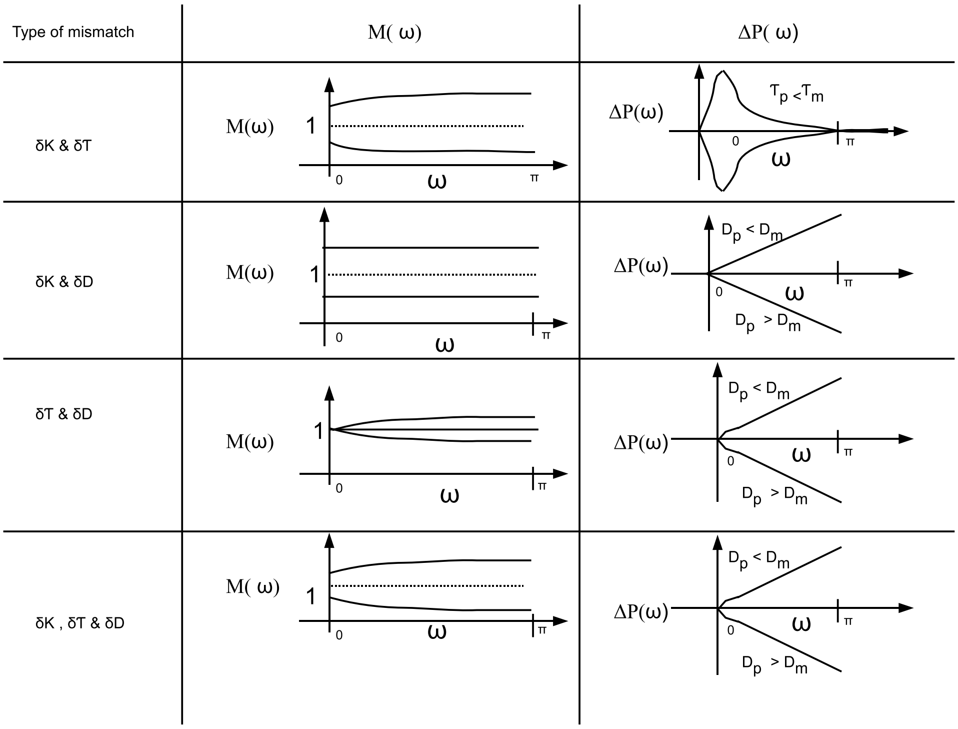


Figure 3. Conditions to the diagnosis of combinations of mismatches

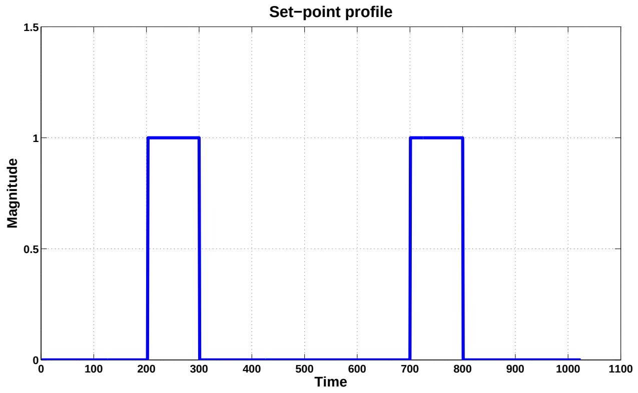
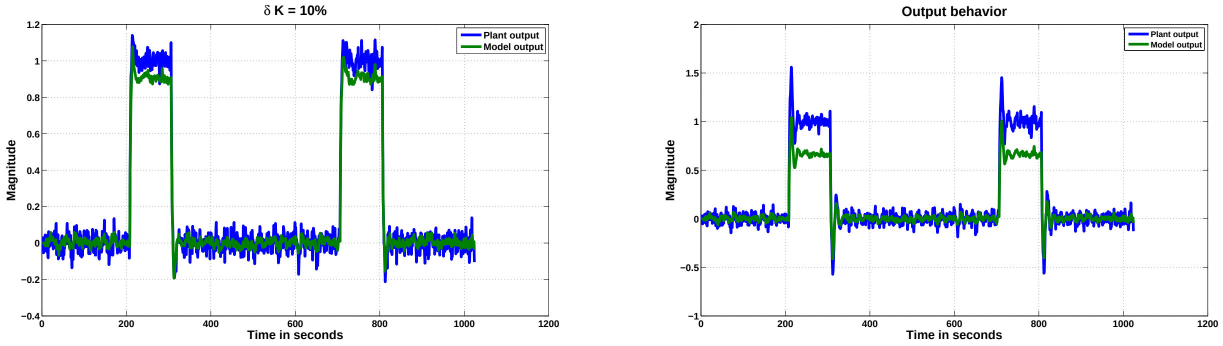
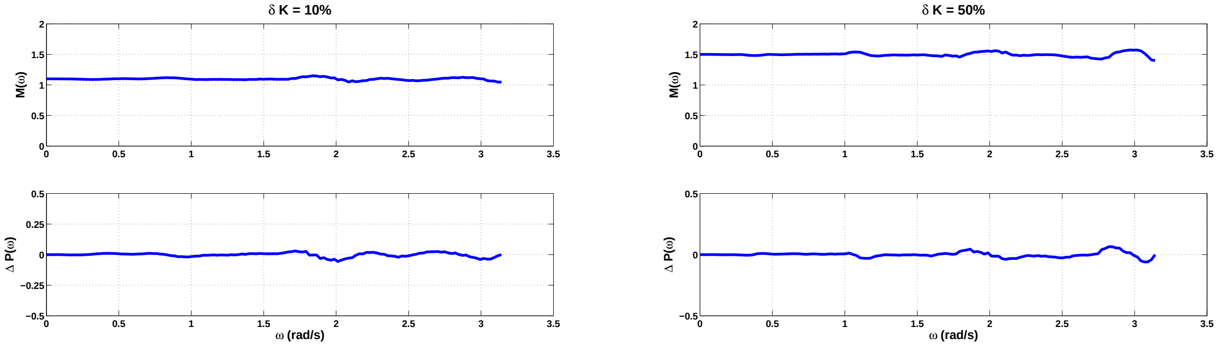


Figure 4. Set-point profile considered for the simulation.



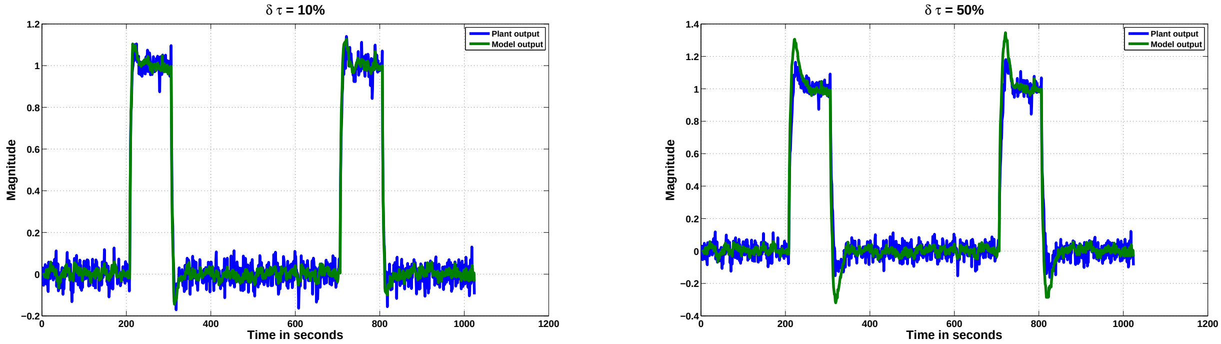
(a) $\delta K = 10\%$ (b) $\delta K = 50\%$

Figure 5. Plant and model ouput behavior for 10% and 50% gain mismatch



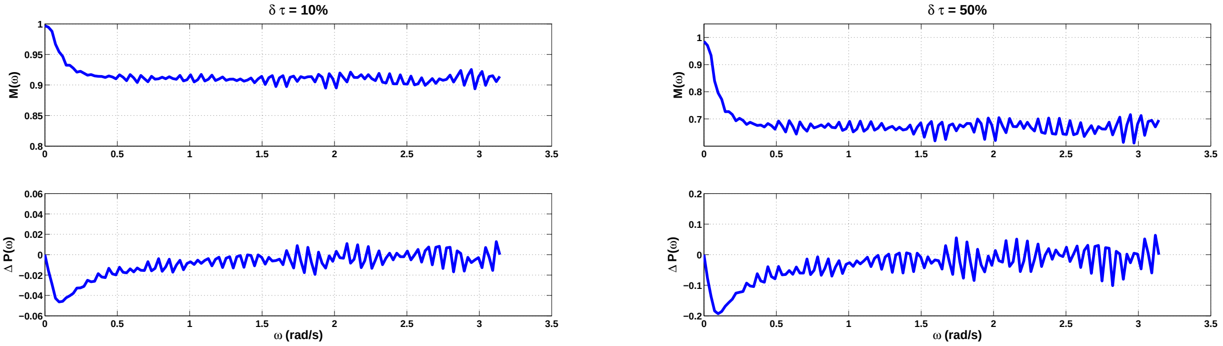
(a) $\delta K = 10\%$ (b) $\delta K = 50\%$

Figure 6. Plots of $M(\omega)$ and $\Delta P(\omega)$ for 10% and 50% gain mismatch



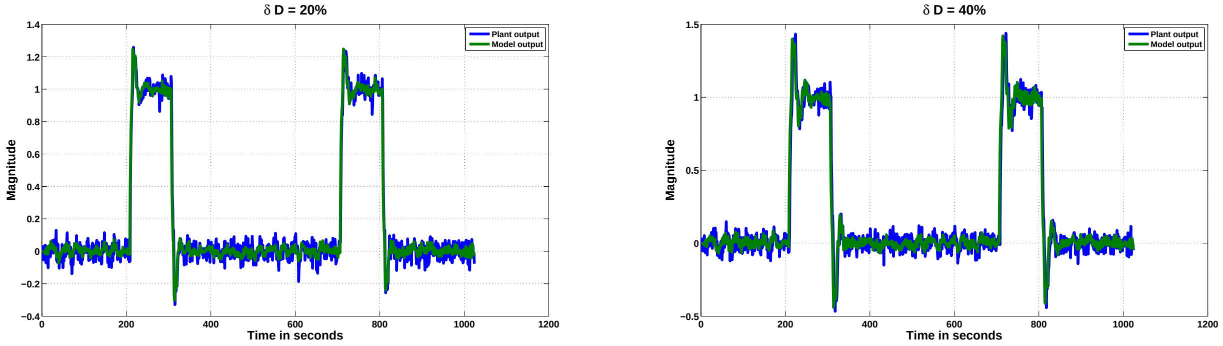
(a) $\delta\tau = 10\%$ (b) $\delta\tau = 50\%$

Figure 7. Plant and model ouput behavior for 10% and 50% time-constant mismatch



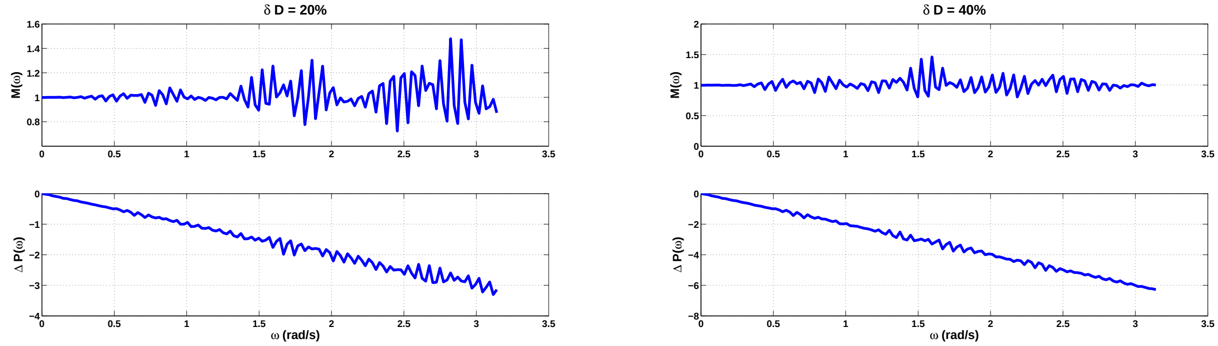
(a) $\Delta\tau = 10\%$ (b) $\Delta\tau = 50\%$

Figure 8. Plots of $M(\omega)$ and $\Delta P(\omega)$ for 10% and 50% time-constant mismatch



(a) $\delta D = 20\%$ (b) $\delta D = 40\%$

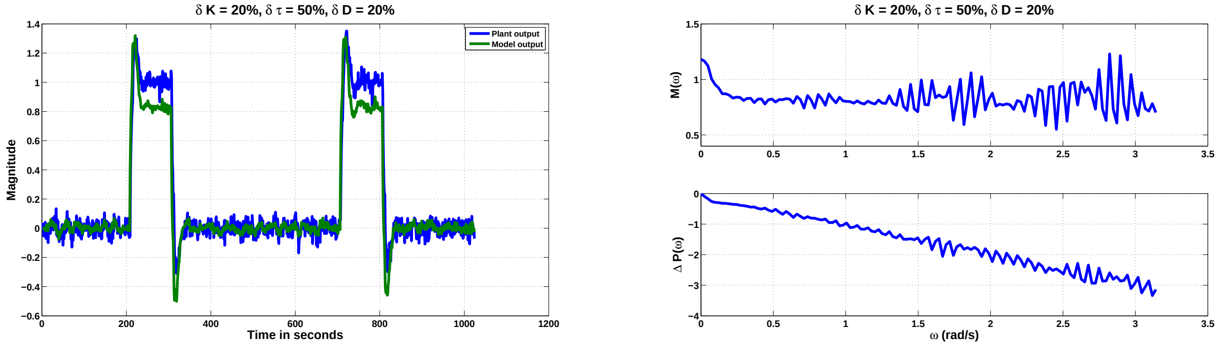
Figure 9. Plant and model ouput behavior for 20% and 40% delay mismatch



(a) $\Delta D = 20\%$

(b) $\Delta D = 40\%$

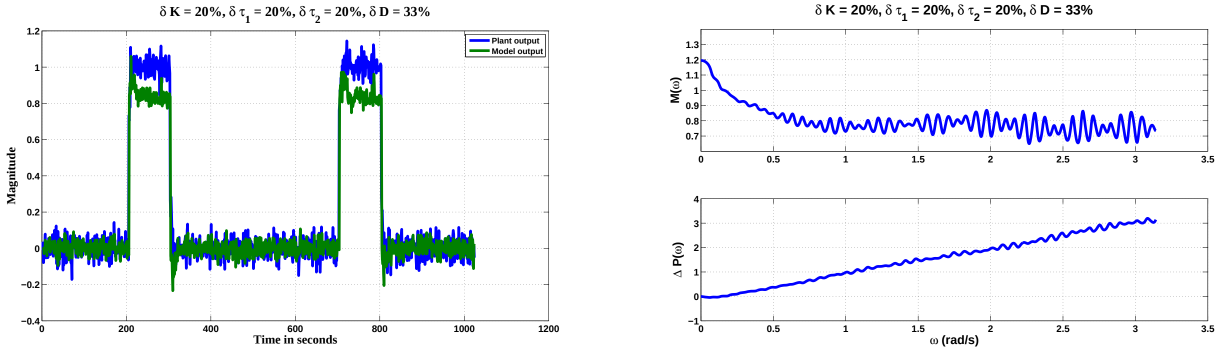
Figure 10. Plots of $M(\omega)$ and $\Delta P(\omega)$ for 20% and 40% delaymismatch



(a) Plant and model outputs behavior

(b) Plots of $M(\omega)$ and $\Delta P(\omega)$

Figure 11. Plots of output behaviors, $M(\omega)$ and $\Delta P(\omega)$ for $\delta K = 20\%,\ \delta\tau = 50\%,\ \delta D = 20\%$



(a) Plant and model outputs behavior

(b) Plots of $M(\omega)$ and $\Delta P(\omega)$

Figure 12. Plots of output behaviors, $M(\omega)$ and $\Delta P(\omega)$ for SOPTD process/ SOPTD model

Comments – Remarks

A spectral method for multivariate time-delay estimation

Selvanathan Sivalingam [*] **Arun K. Tangirala** [**]

[*] *Engineering Cybernetics Department, NTNU, Trondheim 7491, Norway (e-mail: selvanat@itk.ntnu.no)*
[**] *Department of Chemical Engineering, IIT Madras, Chennai 600036 India (e-mail: arunkt@iitm.ac.in)*

Abstract: The delay estimation problem for open-loop single-input single-output (SISO) systems has been widely studied to the effect that well-established methods are available now. On the other hand, the delay estimation for MIMO systems remains a challenge to-date due to the presence of interactions among input and output variables. In this work, a method based on partial coherence functions (PCF) and Hilbert Transform (HT) relation is presented to estimate time-delay for MIMO systems. The key step in the proposed method involves decoupling interactions using the PCF. The method uses frequency-domain analysis which allows one to take into effects of noise in a straightforward way. Under closed-loop conditions, time-delay is estimated by introducing a dither signal at the controller output.

Keywords: process control, instrumentation, simulation, time-delay estimation, Hilbert transform relation

1. INTRODUCTION

Every industrial process has inherent time-delays due to process dynamics and/or instrumentation. The knowledge of this delay can be critical to the identification and control of such processes. Given the complexities underlying the industrial processes and the fact that the measured data are routinely available, it is attractive as well as practical to estimate these delays from the data rather than making an attempt to obtain them from a physical understanding of the process. The idea of time-delay estimation from measurements has been put into practice for decades now. A wealth of literature is available on several methods for a class of linear time-invariant (LTI) SISO systems. However, there are several gray areas in this field, particularly in the context of open-loop/closed-loop multi-input multi-output (MIMO) systems. The aim of this paper is to present a frequency domain method to estimate time-delays in LTI MIMO systems . It is assumed that these time-delays are naturally present in the system and/or the instrumentation rather than artificially arising due to approximations of higher-order systems.

MIMO systems pose specific challenges to the delay estimation problem mainly due to two factors: (*i*) physical interaction (within the system) and (*ii*) correlation between inputs. In the performance assessment of MIMO feedback control systems, the knowledge of interactor matrix is necessary to measure the performance index Harris et al. (1996). The notion of the interactor matrix in a MIMO system is equivalent to the meaning of the time delay in a univariate system. Huang et al. (1997) have shown that the interactor matrix can be estimated from the first few Markov parameters of the process using the algorithm given in Rogozinski et al. (1987). Even though the interactor matrix is meaningful as a multi variable generalization of the time-delay term encountered in single-loop systems, its calculation and the concept itself have been found complicated for use by practicing control engineers (Kozub and Garcia, 1993). Hence, the elimination of the requirement to develop the (unitary) interactor matrix can simplify the calculation of multi variable performance index. Xia et al. (2006) developed a method which can estimate upper and lower bounds of the MIMO MV performance index from routine operating data if the I/O delay matrix is known.

Patwardhan and Shah (2005) proposed a parametric method that can estimate time-delays in a MIMO system by first estimating a parametric model using generalized orthonormal basis filters and subsequently analyzing the simulated step response of the noise-free systems. The step response is analyzed using the traditional point of inflection method. Though the method provides relatively accurate results, it requires the knowledge of model structure and is therefore limited by the classical drawbacks of the parametric methods. A major limitation of the aforementioned methods is that delay parameter is implicitly estimated, *i.e.*, the problem is not specifically formulated to estimate delays. Rather the time-delay is estimated by indirect means either through the search of the first significant correlation coefficient or the point of inflection. There is no closed-form expression relating such quantities to the delay parameter. Consequently, it is not clear how the estimates of those quantities affect the estimate of delay. In view of these facts, it is motivating to seek a non-parametric method that explicitly estimates the delay as well as takes into account the quality of the coefficients of the non-parametric model. The latter is necessary to

[⋆] This paper is based on the article published in Chemical Engineering Science Copyright 2010 American Chemical Society. This version is intended for members of the NIL project groups.

obtain a robust estimate of the input-output time delay matrix for MIMO systems. In this paper, a method that meets these requirements is proposed.

The proposed method also handles the interactions present in the MIMO systems using PCF and accommodates the cause and effect relationship in the presence of noise using HT relation. The ability to handle the causality is due to the fact that the imaginary part of frequency response function of a LTI system is the HT of real part of frequency response function. The use of HT relation to estimate the time-delay in a neuro-physiological SISO system appears in the work by Lindemann et al. (2001). On the other hand, the use of PCF to obtain individual effects of inputs in a MIMO system began about four decades ago (see Priestley (1981) for a detailed list of references). The proposed method fuses these techniques to estimate time-delays in MIMO systems.

2. TIME-DELAY ESTIMATION OF SISO SYSTEMS USING HILBERT TRANSFORM RELATION

The idea involved in the proposed method is explained by considering a following discrete transfer function of an LTI system.

$$G(z) = z^{-D}\bar{G}(z) \tag{1}$$

where $\bar{G}(z)$ is the transfer function of the delay-free process and D is the time-delay.
The phase spectrum of the above system, denoted by $\phi(\omega)$ can be represented as the sum of two terms due to (i) the time-delay, D, and (ii) the argument of the delay-free transfer function respectively.

$$\phi(\omega) = D\omega + \arg\bar{G}(\omega) \tag{2}$$

On the other hand, the gain function of the delay-free system is independent of time-delay and identical to that of the original system.

$$|\bar{G}(\omega)| = |G(e^{j\omega})| \tag{3}$$

It is clear from equation (2) that an estimate of delay can be obtained by knowing the estimates of phase spectrum and the argument of the delay-free transfer function. The argument of the delay-free transfer function is obtained by the Hilbert Transform relation. One can formulate an objective function based on the difference function given below,

$$\epsilon(\omega) = \hat{\phi}(\omega) - \arg\hat{G}(\omega) - D\omega \tag{4}$$

where $\hat{\phi}(\omega)$ is the phase spectrum estimate obtained from input and output data and $\arg\hat{G}(\omega)$ is the estimate the argument spectrum obtained using the HT relation. Hamon and Hannan (1974) in their study on the delay estimation proposed the following objective function represented by $J(D)$

$$J(D) = \sum_{B} W(\omega)\cos\epsilon(\omega) \tag{5}$$

The summation is over all ω (ω is discrete) in a proper band B of frequencies contained in [0, π]. Observe that the objective function carries a weighting function similar to that in the weighted least-squares approach (Lennart, 1999) due to the uncertainties in the estimates of $\phi(\omega)$ arising out of imperfections present in the data. Such imperfections can be due to model-plant mismatch, disturbance and non-linear effects. In fact, the weighting function, $W(\omega)$ at each frequency is the reciprocal of the variance of phase spectrum estimates and given by the expression below Priestley (1981).

$$var\left[\hat{\phi}(\omega)\right] = \frac{1}{\nu}\left(\frac{1}{\beta^2(\omega)} - 1\right) \tag{6}$$

where ν is number of degrees of freedom and $\beta(\omega)$ is coherence function estimated from cross spectrum. The weighting function in equation 5 can be written as $W(\omega) = \dfrac{\beta^2(\omega)}{1-\beta^2(\omega)}$. The objective function is evaluated for different values of the time-delay, D. The value of D at which $J(D)$ is maximum is the true time-delay present in the system.

2.1 Hilbert Transform relation

Hilbert Transform relation(discretized version) between magnitude and phase is given as

$$\arg\hat{G}(\omega_l) = \frac{1}{2M}\sum_{k=1,k\neq l}^{M}\log|\hat{G}(\omega_k)| * \left(\cot\frac{(\omega_l-\omega_k)}{2} + \cot\frac{(\omega_l+\omega_k)}{2}\right) \tag{7}$$

where k and l are dummy variables, M is number of independent spectral estimates determined by the length of the time series N,the sampling frequency f_s and the effective number of degrees of freedom ν using the relation, $M = \frac{Nf_s}{2\nu}$. The equation (8) given above is central to the delay estimation procedure used in this article.

2.2 Spectral estimates

The phase spectrum $\phi(\omega)$ is defined as the argument of the cross-spectrum (Hamon and Hannan, 1974),

$$\phi(\omega) = \arg\left[h_{yu}(\omega)\right] \tag{8}$$

where $h_{yu}(\omega)$ represents the cross-spectrum relating the input to the output. The cross-spectrum itself is defined as the Fourier Transform of the cross covariance function Priestley (1981), *i.e.*,

$$h_{yu}(\omega) = \mathcal{F}[y(t) * u(t)] \tag{9}$$

where * is the convolution operator while $y(t)$ and $u(t)$ are zero-mean output and input sequences respectively. The gain function $|G(\omega)|$ is defined as

$$|G(\omega)| = \frac{|h_{yu}(\omega)|}{h_{uu}(\omega)} \tag{10}$$

where $h_{uu}(\omega)$ denotes the power spectrum of the input to the system, which can be estimated by the Fourier Transform of the auto-covariance function. Thus,

$$h_{uu}(\omega) = \mathcal{F}[u(t) * u(t)] \tag{11}$$

The power spectrum of the output, $h_{yy}(\omega)$, is defined in a similar way. The coherence function, $\beta(\omega)$, is a bounded measure of linear association of the system. For a given frequency range, the unity coherence implies a perfect linear relationship while the value of zero indicates the absence of the same. In practice, the squared coherence is used and computed by the following equation Hamon and Hannan (1974):

$$\beta^2_{yu}(\omega) = \frac{|h_{yu}(\omega)|^2}{h_{uu}(\omega)h_{yy}(\omega)} \tag{12}$$

3. EXTENSION TO MIMO SYSTEMS

In Section 2, the approach to the delay estimation problem has been presented for a SISO system. This approach, however, cannot be directly applied to a MIMO system where, in general, input and output relations involve interactions. A consequence of the interactions is such that the regular coherence function fails to capture the true relationship between an input-output pair. Therefore, the extension of the proposed approach in Section 2 to the MIMO systems involves an additional step of decoupling these interactions. This is carried out by means of the *partial* coherence function in this work.

3.1 Decoupling in frequency domain

The process of decoupling the interactions in time-domain is equivalent to conditioning of two or more signals in the frequency domain. The conditioning refers to removal of linear dependence of a signal from other signals in frequency domain. The partial coherence function, henceforth denoted by $K(\omega)$, is defined as the coherence between the conditioned signals in contrast to $\beta(\omega)$, which is the coherence function between the original signals. Conditioning m input and n output signals of a MIMO system is, in fact, equivalent to decomposing that MIMO system into $m \times n$ non-interacting SISO systems. This leads to the estimation of mn partial coherence functions as explained schematically in Figure 2. This procedure is best understood by means of a simple example as described below. An open-loop MIMO system described by two inputs $\{u_1(t)$ and $u_2(t)\}$ and two outputs $\{y_1(t)$ and $y_2(t)\}$, as shown in Figure 1, is considered for illustration. Conditioning these two inputs and outputs decomposes this MIMO system into four non-interacting SISO systems, which relate the four pairs of conditioned signals (in the frequency domain). The conditioned output and input signals in time-domain are denoted by $\psi(t)$ and $\gamma(t)$, while their frequency-domain counterparts are denoted by $\Psi(\omega)$ and $\Gamma(\omega)$ respectively.

We shall now show the computation involved in conditioning $y_1(t)$ and $u_1(t)$ to ultimately obtain the true coherency between these signals. These signals are conditioned on $u_2(t)$ to obtain $\psi_{11}(t)$ and $\gamma_{11}(t)$ respectively as given in equations (13) and (14) below.

$$\psi_{11}(t) = y_1(t) - \sum_{k=-\infty}^{\infty} b_1(k)u_2(t-k) \tag{13}$$

$$\gamma_{11}(t) = u_1(t) - \sum_{k=-\infty}^{\infty} b_2(k)u_2(t-k) \tag{14}$$

where $\{b_1(k)\}$, $\{b_2(k)\}$, are determined by minimizing $E[\psi^2_{11}(t)]$ and $E[\gamma^2_{11}(t)]$, respectively. Here, E is the *expectation operator*. The corresponding transfer functions are given by

$$B_1(\omega) = \sum_{k=-\infty}^{\infty} b_1(k)e^{-i\omega k} = \frac{h_{y_1u_2}(\omega)}{h_{u_2u_2}(\omega)} \tag{15}$$

$$B_2(\omega) = \sum_{k=-\infty}^{\infty} b_2(k)e^{-i\omega k} = \frac{h_{u_1u_2}(\omega)}{h_{u_2u_2}(\omega)} \tag{16}$$

Refer to Priestley (1981) for a detailed derivation of the above results. Now, the corresponding frequency-domain representations of $\psi_{11}(t)$ and $\gamma_{11}(t)$ can be written as,

$$\Psi_{11}(\omega) = Y_1(\omega) - B_1(\omega)U_2(\omega) \tag{17}$$

$$\Gamma_{11}(\omega) = U_1(\omega) - B_2(\omega)U_2(\omega) \tag{18}$$

The cross spectral density function of $\psi_{11}(t)$ and $\gamma_{11}(t)$ is found by evaluating $E[\Psi_{11}(\omega){\Gamma_{11}}^*(\omega)]$,

$$h_{\Psi_{11}\Gamma_{11}}(\omega) = h_{y_1u_1}(\omega) - B_1(\omega)h_{u_2u_1}(\omega) - B^*_2(\omega)h_{y_1u_2}(\omega) + B_1(\omega)B^*_2(\omega)h_{u_2u_2}(\omega)$$

The function $h_{\Psi_{11}\Gamma_{11}}(\omega)$ is called the partial cross-spectral density function of $y_1(t)$ and $u_1(t)$ {allowing for $u_2(t)$}, and is also denoted by $h_{y_1u_1.u_2}(\omega)$. The partial (complex) coherency $\zeta_{y_1u_1.u_2}(\omega)$, is now defined as the (complex) coherency of $\psi_{11}(t)$ and $\gamma_{11}(t)$ and is given by

$$\zeta_{y_1u_1.u_2}(\omega) = \frac{h_{\Psi_{11}\Gamma_{11}}(\omega)}{\{h_{\Psi_{11}\Psi_{11}}(\omega)h_{\Gamma_{11}\Gamma_{11}}(\omega)\}^{1/2}} \tag{19}$$

where $h_{\Psi_{11}\Psi_{11}}(\omega)$ and $h_{\Gamma_{11}\Gamma_{11}}(\omega)$, the spectral density functions of $\psi_{11}(t)$ and $\gamma_{11}(t)$.
Further simplification gives

$$h_{\Psi_{11}\Psi_{11}}(\omega) = h_{y_1y_1}(\omega)\{1 - |\zeta_{y_1u_2}(\omega)|^{1/2}\} \tag{20}$$

$$h_{\Gamma_{11}\Gamma_{11}}(\omega) = h_{u_1u_1}(\omega)\{1 - |\zeta_{u_2u_1}(\omega)|^{1/2}\} \tag{21}$$

Substituting the equations (20) and (21) in (19), the following expression is derived from which partial coherency can be obtained.

$$\zeta_{y_1u_1.u_2}(\omega) = \frac{\zeta_{y_1u_1}(\omega) - \zeta_{y_1u_2}(\omega)\zeta_{u_2u_1}(\omega)}{\{(1 - |\zeta_{y_1u_2}(\omega)|^{1/2})(1 - |\zeta_{u_2u_1}(\omega)|^{1/2})\}^{1/2}} \tag{22}$$

The function $|\zeta_{y_1u_1.u_2}(\omega)|$ is defined as the partial coherence function, $\kappa_{y_1u_1}(\omega)$, between $y_1(t)$ and $u_1(t)$, after removing the common influence of $u_2(t)$. The conditioned signals for the remaining combination of input-outputs namely, $\{\Gamma_{12}(\omega)$, $\Psi_{12}(\omega)\}$, $\{\Gamma_{21}(\omega)$, $\Psi_{21}(\omega)\}$ and $\{\Gamma_{22}(\omega)$, $\Psi_{22}(\omega)\}$ are estimated in a similar way. Subsequently, $\kappa_{y_1u_2}(\omega)$, $\kappa_{y_2u_1}(\omega)$ and $\kappa_{y_2u_2}(\omega)$ are obtained.

The subscripts on conditioned pair of signals are best understood by noting that a pair $\{\Gamma_{ij}(\omega)$, $\Psi_{ij}(\omega)\}$ allows us to estimate the true coherency, $\kappa_{ij}(\omega)$, between the j^{th} input and the i^{th} output. The generalized partial coherence function matrix for a MIMO system containing m inputs and n outputs can be written as follows.

$$K(\omega) = \begin{bmatrix} \kappa_{11}(\omega) & \kappa_{12}(\omega) & \dots & \kappa_{1m}(\omega) \\ \kappa_{21}(\omega) & \kappa_{22}(\omega) & \dots & \kappa_{2m}(\omega) \\ \vdots & \vdots & & \vdots \\ \kappa_{n1}(\omega) & \kappa_{n2}(\omega) & \dots & \kappa_{nm}(\omega) \end{bmatrix} \quad (23)$$

The quantity $\kappa_{ij}(\omega)$ is now used in place of $\beta(\omega)$ in equation (6) to compute the objective function for estimating the time-delay between j^{th} input and i^{th} output. Under closed-loop conditions, the procedure differs from the above that the effects of outputs are also taken into account in equations (13) and (14) with appropriate changes in the subsequent equations.

4. SIMULATION STUDIES

Delay estimation for multivariate systems are shown by taking up a simulated MIMO system and the standard SHELL control problem.

Transfer function model A system of two inputs and two outputs is simulated with known delays in all four transfer functions under open- and closed-loop conditions.

The transfer function model for the open-loop case is as follows

$$G_(z^{-1}) = \begin{bmatrix} \dfrac{z^{-5}}{1-0.5z^{-1}+0.05z^{-2}} & \dfrac{0.3z^{-4}}{1-0.9z^{-1}+0.2z^{-2}} \\ \dfrac{0.8z^{-6}}{1-0.5z^{-1}} & \dfrac{z^{-3}}{1-0.7z^{-1}} \end{bmatrix} \quad (24)$$

The input-ouput delay matrix for the above system is written as

$$D_{true} = \begin{bmatrix} 5 & 4 \\ 6 & 3 \end{bmatrix} \quad (25)$$

Simulation is carried out by exciting the system with pulse inputs for different output SNR levels (10, 5 and 1). As discussed in Section 3.1, the MIMO system that contains two inputs and two outputs is decomposed into four non-interacting SISO systems with the help of PCF yielding four conditioned pairs (in the frequency domain). The partial coherency, $\kappa_{ij}(\omega)$, at each frequency is now estimated between the conditioned pairs $\{\Gamma_{ij}(\omega),\ \Psi_{ij}(\omega)\}$. Subsequently, the quantity $J(D)$ in the each channel is estimated by using $\kappa_{ij}(\omega)$ in place of the regular coherency, $\beta(\omega)$, in equation (5) for a range of values of D. Correspondingly, the time-delay in each channel is obtained. The objective functions for the four sub-systems at SNR$_{out}$ = 10 and SNR$_{out}$ = 1 are plotted and shown in Figures 3(a) and 3(b) respectively. The estimated time-delay matrix is as follows.

$$\hat{D}_{OL} = \begin{bmatrix} 5 & 4 \\ 6 & 3 \end{bmatrix} \quad (26)$$

which is identical to the true time-delay matrix.

The open-loop system is now controlled using a PI controller with tuning values $K_c = 0.1$ and $K_I = 0.01$. A random type dither signal is introduced at each input such that the ratio of variance of the measurement noise to that of the dither signal is varied from one to 10.

The time-delays in the four channels are estimated in a way similar to that under open-loop conditions. The corresponding four objective functions obtained for output SNR levels of 10 and 1 are plotted in Figures 4(a) and 4(b) respectively. The estimated time-delay matrix for the closed-loop case is provided below.

$$\hat{D}_{CL} = \begin{bmatrix} 5 & 4 \\ 6 & 3 \end{bmatrix} \quad (27)$$

which is again in agreement with the true delay matrix, D_{true}. This result is demonstrative of the ability of the method to correctly estimate the time-delays in MIMO systems in both open- and closed-loop conditions.

Shell control problem The Shell control benchmark problem (Figure 5) was first published by SHELL in 1986 in their first Process Control workshop (Maciejowski, 2002), with the intention to provide a standard and realistic test bed for the evaluation of new control theories and technologies. It captures most of the relevant control issues while staying as realistic as possible.

The problem involves control of a heavy oil fractionator system characterized by large time delays in each input-output pair. The heavy oil fractionator has three product draws, three side circulating loops and a gaseous feed stream. The system consists of seven measured outputs, three manipulated inputs and two unmeasured disturbances. Product specifications for top and side draws are determined by economic considerations. There is no product specification on bottom draw, however, there is an operating constraint on the bottom reflux temperature. Top draw, side draw and bottoms reflux duty can be used as manipulated variables to control the column while heat duties on the two other side loops (upper reflux duty and intermediate reflux duty) act as unmeasured disturbances to the column.

The original problem describes the system with 35 transfer functions relating three manipulated inputs and two disturbances to three controlled outputs and four measured outputs. For the sake of demonstration, we have considered only the transfer functions between manipulated (control) inputs and controlled outputs, which are given in Table 2. The continuous transfer functions are discretized with a sampling interval of two units. The corresponding discrete transfer functions are tabulated in Table 3.

The time-delay estimation problem is taken up under both open- and closed-loop conditions. The delay matrix corresponding to the sub-system of interest is written as

$$D_{true} = \begin{bmatrix} 14 & 15 & 14 \\ 10 & 8 & 8 \\ 11 & 12 & 1 \end{bmatrix} \quad (28)$$

A state-space realization of the transfer functions, as given in Table 2, and the simulation settings used in Patwardhan and Shah (2005) are used to obtain the input-output data. Further, the settings are adjusted such that SNR (at the output) is maintained at 10.

The three inputs, namely, the top draw composition, side draw composition and bottoms reflux duty, and the three outputs, namely, the top end point composition, side end-point composition and bottoms reflux duty, of the 3×3 MIMO system are conditioned using the PCF to obtain nine conditioned pairs, thus resulting in nine non-interacting SISO systems. The objective function $J(D)$ is estimated for each conditioned pair and is plotted against the optimization parameter D. The resulting plots for the open- and closed-loop conditions are shown in Figures 6(a) and 6(b) respectively. The input-output delay matrices estimated from these figures (for open- and closed-loop conditions) are as follows.

$$\hat{D}_{OL} = \begin{bmatrix} 14 & 15 & 14 \\ 10 & 8 & 8 \\ 11 & 12 & 1 \end{bmatrix}, \hat{D}_{CL} = \begin{bmatrix} 15 & 15 & 14 \\ 10 & 8 & 8 \\ 11 & 12 & 1 \end{bmatrix} \qquad (29)$$

It is observed that the delay matrix under the open-loop conditions is identical to the true delay matrix. There is an excess of one delay estimated in G_{11} under the closed-loop conditions. This is due to the presence of non-minimum phase behavior in the channel between u_1 and y_1 (G_{11}). It is to be noted here that the HT relation is derived assuming the system under minimum phase conditions. However, when the HT relation is used to estimate the time-delay for non-minimum phase conditions, the estimate of the is believed to provide an upper bound Lindemann et al. (2001). From the experience with applying this method to various systems it is observed that the deviation in the estimate is of the order of at most a unit sampling period.

REFERENCES

T. Harris, F. Boudreas, J. Macgregor, Performance assessment of multivariable feedback controllers, Automatica 32 (11) (1996) 1508–1518.

B. Huang, S. Shah, H. Fujiii, The unitary interactor matrix and its estimation using closed-loop data, Journal of Process Control.

M. Rogozinski, P. A, G. M, An algorithm for calculation of a nilpotent interactor matrix for linear multivariable systems, IEEE Trans. AC 32 (3) (1987) 234–237.

D. Kozub, C. Garcia, Monitoring and diagnosis of automated controllers in the chemical process industries, Tech. rep., AIChE Annual Meeting (1993).

H. Xia, P. Majecki, A. Ordys, M. Grimble, Performance assessment of mimo systems based on i/o delay information, Journal of Process Control.

S. Patwardhan, S. Shah, From data to diagnosis and control using generalized orthonormal basis filters. part i: Development of state observers, Journal of Process Control 15 (2005) 819–835.

M. Lindemann, J. Raethjen, J. Timmer, G. Deuschl, G. Pfister, Delay estimation for cortico-peripheral relations, Journal of Neuroscience Methods 111 (2001) 127–139.

M. Priestley, Spectral Analysis and Time series, Academic Press, 1981.

B. Hamon, E. Hannan, Spectral estimation of time delay for dispersive and non-dispersive systems, Applied statistics 23 (1974) 134–142.

L. Lennart, System Identification Theory for the User, Prentice Hall, 1999.

A. Oppenheim, R. Schafer, Digital Signal Processing, Prentice Hall of India, 1975.

J. Maciejowski, Predictive Control with Constraints, Pearson Education, 2002.

R. Vadigepalli, E. Gatzke, F. D. III, Robust control of a multivariable experimental four-tank system, Industrial & Engineering Chemistry Research 40 (8) (2001) 1916–1927.

Table 1. *Names, roles and symbols of input and output variables used in the schematic of the Shell control problem*

Variable	Role	Symbol
Top Draw	Control input	u_1
Side Draw	Control input	u_2
Bottoms Reflux Duty	Control input	u_3
Intermediate Reflux Duty	Measured disturbance	d_1
Upper Reflux Duty	Unmeasured disturbance	d_2
Top End Point	Controlled output	y_1
Side End Point	Controlled output	y_2
Bottoms Reflux Temperature	Controlled output	y_3
Top Temperature	Measured output	y_4
Upper Reflux Temperature	Measured output	y_5
Side Draw Temperature	Measured output	y_6
Intermediate Reflux Temperature	Measured output	y_7

Table 2. *Transfer functions from manipulated inputs to controlled outputs for the Shell control problem*

outputs/inputs	u_1	u_2	u_3
y_1	$e^{-27s}\dfrac{4.05}{50s+1}$	$e^{-28s}\dfrac{1.77}{60s+1}$	$e^{-27s}\dfrac{5.88}{50s+1}$
y_2	$e^{-18s}\dfrac{5.39}{50s+1}$	$e^{-14s}\dfrac{5.72}{60s+1}$	$e^{-15s}\dfrac{6.9}{40s+1}$
y_3	$e^{-20s}\dfrac{4.38}{33s+1}$	$e^{-22s}\dfrac{4.42}{44s+1}$	$\dfrac{7.2}{19s+1}$

Table 3. *Discrete transfer functions of the Shell control system with $T_s = 2$ units*

outputs/inputs	u_1	u_2	u_3
y_1	$z^{-13}\dfrac{0.0802z^{-1}+0.07861z^{-2}}{1-0.9608z^{-1}}$	$z^{-14}\dfrac{0.05803z^{-1}}{1-0.9672z^{-1}}$	$z^{-13}\dfrac{0.1164z^{-1}+0.1141z^{-2}}{1-0.9608z^{-1}}$
y_2	$z^{-9}\dfrac{0.2113z^{-1}}{1-0.9608z^{-1}}$	$z^{-7}\dfrac{0.1875z^{-1}}{1-0.9672z^{-1}}$	$z^{-7}\dfrac{0.1875z^{-1}}{1-0.672z^{-1}}$
y_3	$z^{-10}\dfrac{0.2576z^{-1}}{1-0.9412z^{-1}}$	$z^{-11}\dfrac{0.1964z^{-1}}{1-0.9556z^{-1}}$	$\dfrac{0.7194z^{-1}}{1-0.9001z^{-1}}$

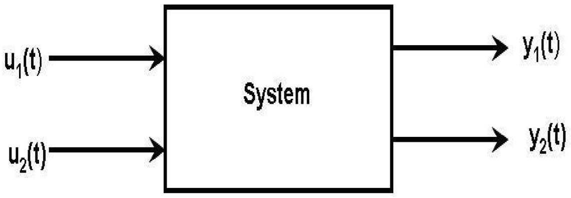


Fig. 1. *Multivariate linear system considered for the illustration of PCF*

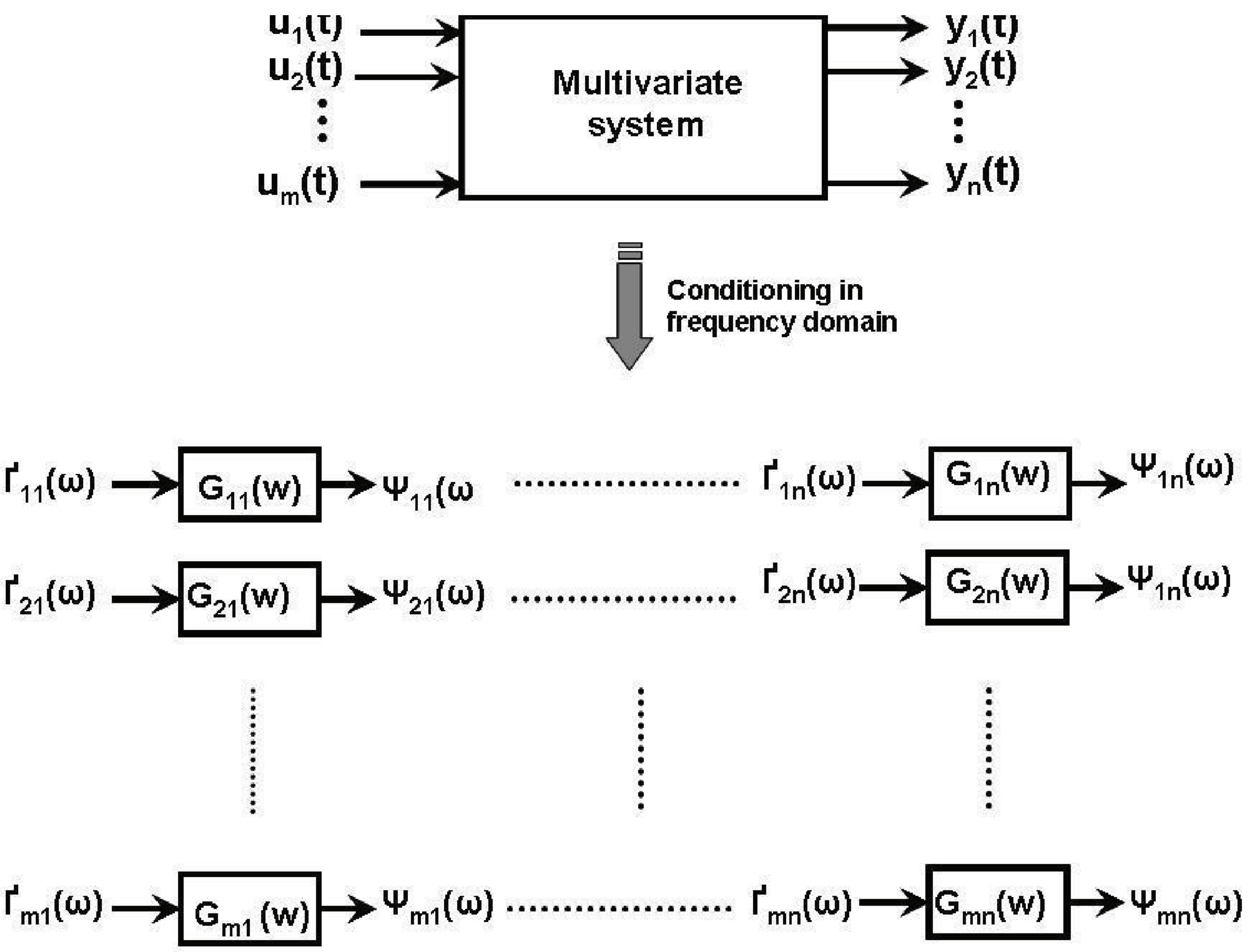
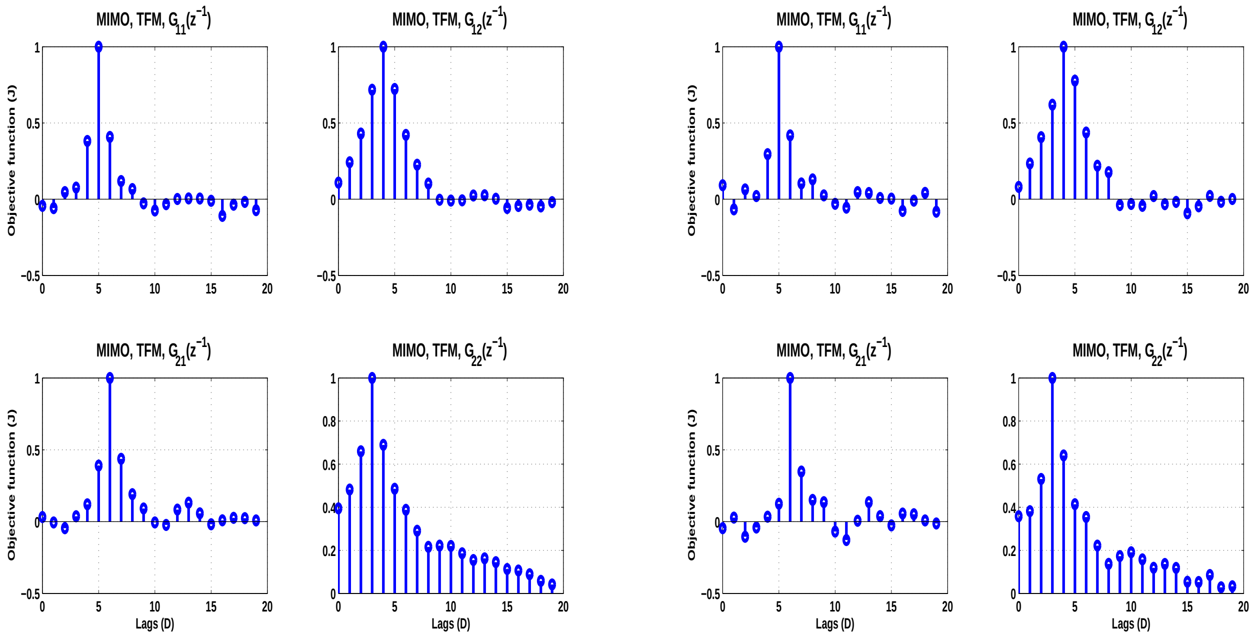


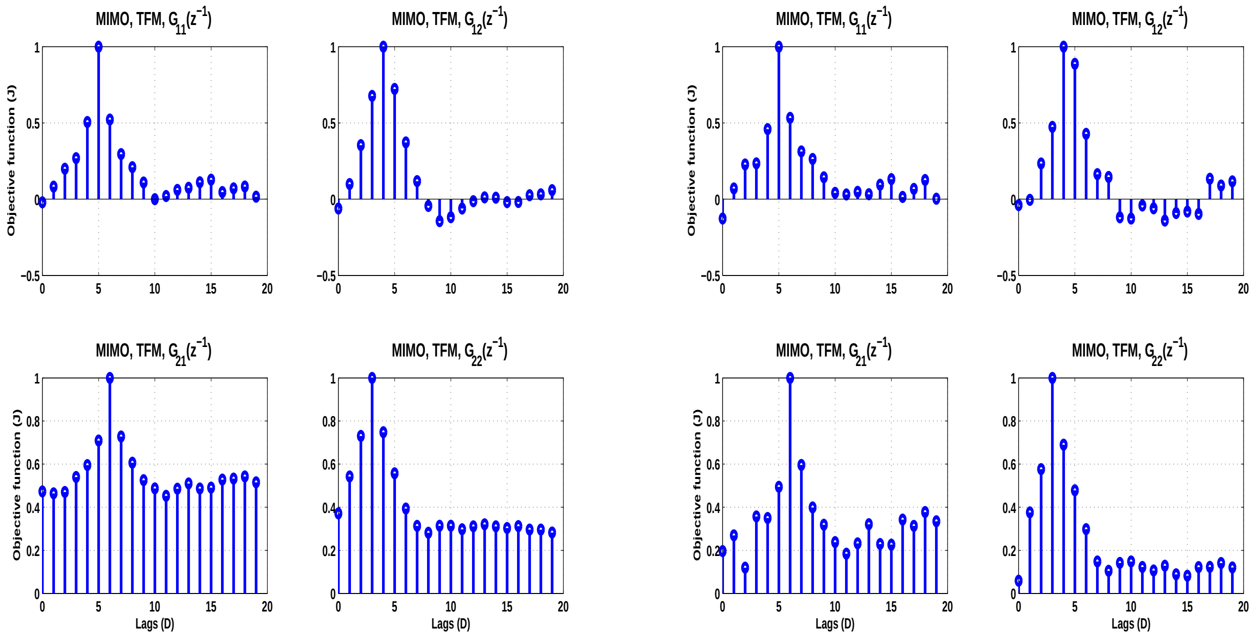
Fig. 2. *Pictorial representation of decoupling interactions*



(a) SNR = 10

(b) SNR = 1

Fig. 3. *Time delay estimation of the multivariate open-loop system (TFM)*



(a) SNR = 10 (b) SNR = 1

Fig. 4. *Time delay estimation of the multivariate closed-loop system (TFM)*

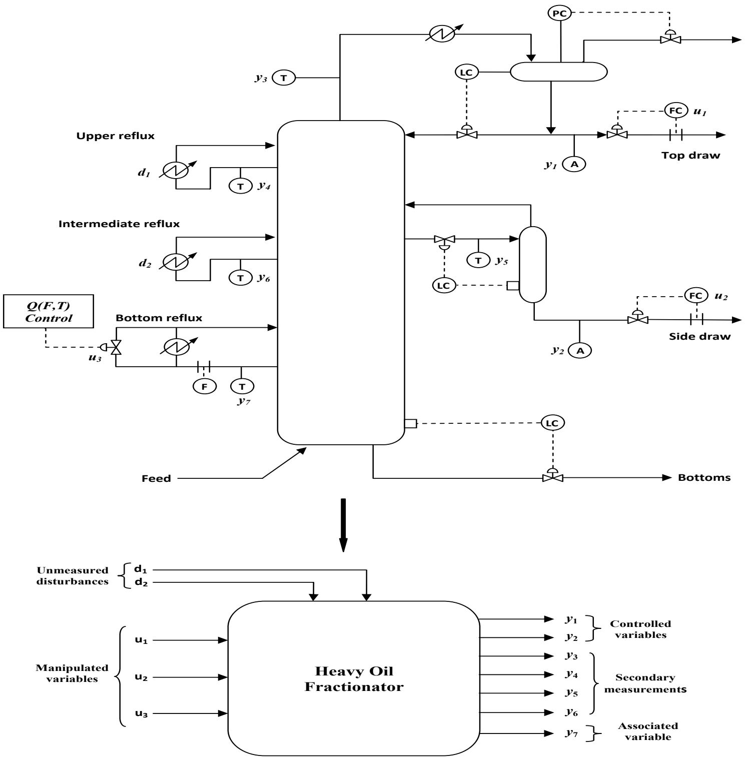
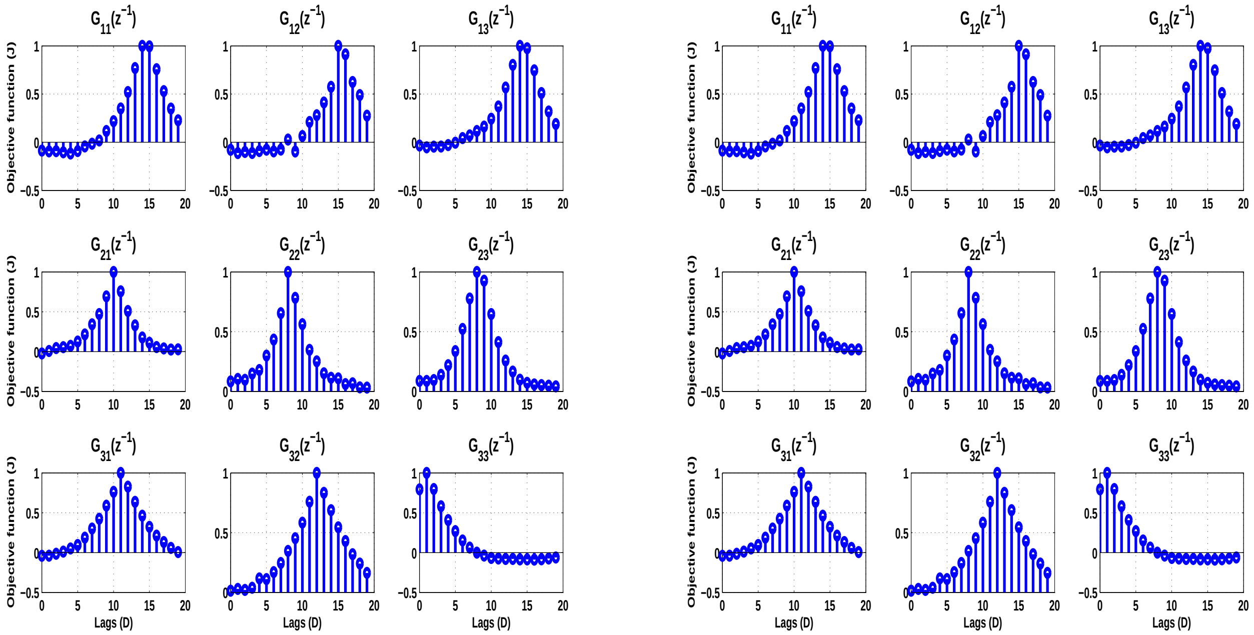


Fig. 5. *Schematic diagram of the heavy oil fractionator and the associated control problem*



(a) Open-loop

(b) Closed-loop

Fig. 6. *Delay estimation for the Shell control problem, top end point composition, side end point composition and bottoms reflux temperature as outputs respectively*

Comments – Remarks

Institute of Automation, Measurement and Applied Informatics, FME STU in Bratislava

PID Control of Laboratory Model of Antilock Braking System

C. Belavý*, M. Smeja

*Institute of automation, measurement and applied informatics, Faculty of Mechanical Engineering,
Slovak University of Technology in Bratislava, Nám. Slobody 17, 812 31 Bratislava,
(Tel., fax +421 2 5249 5315; e-mail: cyril.belavy@stuba.sk)

Abstract: The antilock braking system (ABS) in a car is a feedback control system, where controller has to be able to maintain specified tire slip for each wheel during braking. This paper presents PID control synthesis method for a simple laboratory model of an antilock braking system. Tuning of controller parameters is solved as an optimization problem based on dynamical model of the controlled system. In the end of the paper, simulation results of the antilock braking of the laboratory model are showed.

1. INTRODUCTION

The antilock braking system (ABS) is an important safety component of a car. It prevents the wheels from locking while braking and contribute to shorten braking distances to its minimum possible value. From the standpoint of systems and control theory, it is a feedback control system, where controller has to be able to maintain specified tire slip for each wheel during braking.

In the last decade intensive development of control systems for car ABS has been observed, including also various PID and LQ control strategies, e.g. (Johansen *et al.*, 2003) proposed a robust gain-scheduled LQ controller. For control strategy of ABS in a car it is necessary to estimate the car velocity and parameters of the friction curve in the presence of various effects which are difficult to model (Petersen, 2003).

In this paper, the laboratory model of ABS (LABS) manufactured by Inteco Ltd. has been used. It allows to design control synthesis methods based on mathematical model which are suitable for ABS, (Bania *et al.*, 2005), (Oniz *et al.*, 2007). The paper organized as follows. Section 2 contains a short description of the LABS and its mathematical model. In the next section simple PID control feedback loop for LABS in MATLAB-Simulink is arranged. There is also optimization process of PID controller parameters with respect of response constraints performed. Further, control process for given constant reference wheel sleep with optimal controller parameters is executed. The paper ends with some conclusions remarks.

2. LABORATOTY MODEL OF ABS

The laboratory model of ABS shown in Fig. 1 consists of two wheels rolling on one another. The upper wheel is mounted on a rocker arm and has plastic wheel disk rubber tire. The lower wheel is made of aluminum. The angles of rotation of the wheels are measured by encoders with the resolution $2\pi/4096 = 0.0875^o$. The upper wheel is equipped with a disk brake controlled by a DC motor. Another DC motor, placed on the axle of the lower wheel is used to set the system in motion and accelerate it. The upper wheel represents wheel of a car and lower wheel represents road. During the braking process, the latter motor is switched off and a braking torque is applied to the upper wheel, which causes wheel speed to decrease. Notation of the LABS parameters is in Table 1.

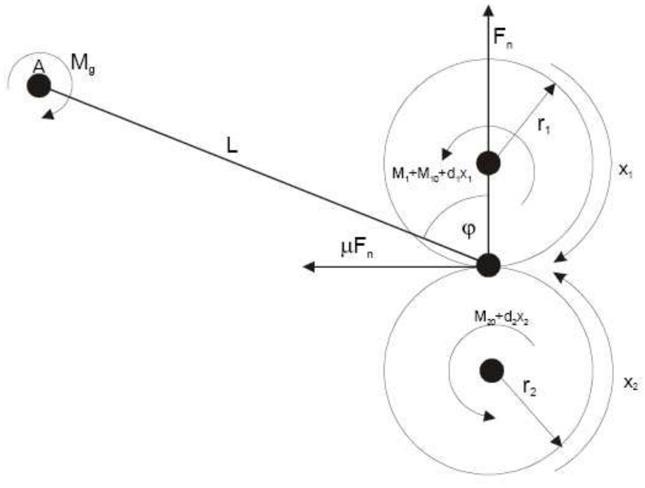


Fig. 1. Scheme of laboratory model of ABS

According to Newton's second law, the equations of the motion of the system are in following form:

$$J_1\dot{x}_1 = \mu(\lambda)F_n r_1 - (d_1x_1 + M_{10} + M_1) \qquad (1)$$

$$J_2\dot{x}_2 = -(\mu(\lambda)F_n r_2 + d_2x_2 + M_{20}) \qquad (2)$$

$\mu(\lambda)F_n$ in (1) and (2) represents road friction force F_t according Coulomb law.

Normal force F_t is obtained from equation of moments for point A:

$$F_n = \frac{M_g + M_1 + M_{10} + d_1x_1}{L(\sin\varphi - \mu(\lambda)\cos\varphi)} \qquad (3)$$

where L is the distance between the contact point of the wheels and the rotational axis of the balance lever and φ is the angle between the normal in the contact point and the line L.

The relative difference of peripheral velocities of the wheels, or the wheel slip λ is defined as:

$$\lambda = \frac{r_2 x_2 - r_1 x_1}{r_2 x_2} \qquad (4)$$

While a wheel slip of 0 indicates that the wheel velocity and the vehicle velocity are the same, a ratio of 1 indicates that the tire is not rotating and the wheels are skidding on the road surface, i.e., the vehicle is in practice no longer steerable.

For simplify equation were created following variables:

$$\begin{aligned} &c_{11} = \frac{r_1 d_1}{J_1},\ c_{12} = \frac{(M_{10} + M_g) r_1}{J_1},\ c_{13} = -\frac{d_1}{J_1}, \\ &c_{14} = -\frac{M_{10}}{J_1},\ c_{15} = \frac{r_1}{J_1}, c_{16} = -\frac{1}{J_1}, \\ &c_{21} = -\frac{r_2 d_1}{J_2}, c_{22} = -\frac{(M_{10} + M_g) r_2}{J_2}, \\ &c_{23} = -\frac{d_2}{J_2}, c_{24} = -\frac{M_{20}}{J_2}, c_{25} = -\frac{r_2}{J_2}. \end{aligned} \qquad (5)$$

Then, dynamics of the system is described by the following state equations:

$$\dot{x}_1 = \mu(\lambda)(c_{11}x_1 + c_{12}) + c_{13}x_1 + c_{14} + (c_{15}\mu(\lambda) + c_{16})x_3 \quad (6)$$

$$\dot{x}_2 = \mu(\lambda)(c_{21}x_1 + c_{22}) + c_{23}x_2 + c_{24} + c_{25}\mu(\lambda)x_3 \quad (7)$$

$$\dot{x}_3 = c_{31}(u - x_3), \quad 0 \le u \le u_{\max} \quad (8)$$

The road adhesion coefficient is a nonlinear function of some physical variables including wheel slip. Here it is approximated by the following formula:

$$\mu(\lambda) = \frac{w_4 \lambda^p}{a + \lambda^p} + w_3\lambda^3 + w_2\lambda^2 + w_1\lambda \qquad (9)$$

Table 1. Notation of LABS parameters

r_1, r_2	Radius of the upper and lower wheel
x_1, x_2	Angular velocity of the upper and lower wheel
J_1, J_2	Moment of inertia of the upper and lower wheel
d_1, d_2	Viscous friction coefficient of the upper and lower wheel
M_{10}, M_{20}	Static friction of the upper and lower wheel
M_1	Braking torque
M_g	Moment of gravity acting on balance lever
F_n	Total normal load
F_t	Road friction force
μ	Road adhesion coefficient
λ	Wheel slip

The numerical values used in this paper are:

r_1 = 0:0925 (m)
r_2 = 0:0920 (m)
φ = 57,79 (°)
L = 0,26 (m)
J_1 = 0.00898 (kgm^2)
J_2 = .02316 (kgm^2)
d_1 = 0.00014314 (kgm^2 s^{-1})
d_2 =0.00020353 (kgm^2 s^{-1})
M_{10} = 0.0032 (Nm)
M_{20} = 0.025 (Nm)
w_1 = 0.117219
w_2 = -0.044978
w_3 = 0.034217
w_4 = 0.460736
a = 0.0000231606 and
p = 2.986554

The dependence of the adhesion coefficient on the slip is presented in Fig. 2.

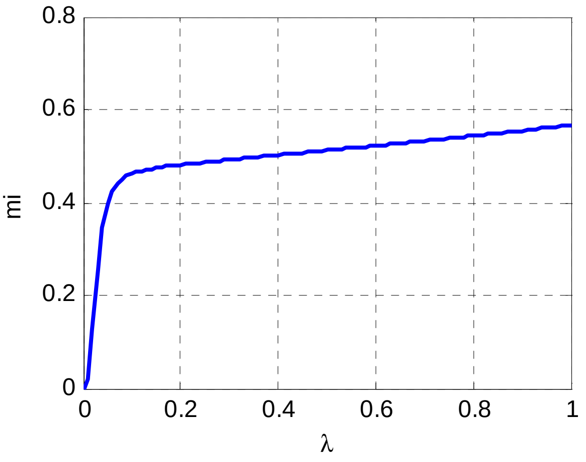


Fig. 2. Adhesion coeficient as a function of slip

3. PID CONTROL SYNTHESIS

The goal of control during antilock braking is to reduce the velocity of wheels in time in such a way that an adequate compromise is ensured between excessive slip, braking distance and accuracy of reaching the target state.

For solution of this task, PID control loop for identified LABS with parameters given above, was arranged in MATLAB – Simulink software environment, see Fig. 3. Controlled variable is wheel sleep with constant reference value 0,2. Tuning of PID controller parameters is solved as an optimization problem in the block *PID Controller Optimization* by means of *Simulink Response Optimization Toolbox*. There for optimization of proportional, integral and derivative parameters of the controller, *Gradient descent* method is used. During of the optimization procedure also constraints for controlled variable λ is used, see Fig. 4. There was found optimal solution which is feasible within the specified tolerances and constraints, see Fig. 5.

With optimized PID controller parameters, simulation of the control process with antilock braking of LABS was executed. Running of wheel sleep λ as a controlled variable is on Fig. 5 and angular speed of both wheels during control process from initial speed is on Fig. 7. Brake torque as actuating variable is on Fig. 8. Performance of control in this case is very good in accordance with given requirements. Time of control is minimal, wheel sleep and velocities of wheels are without oscillations.

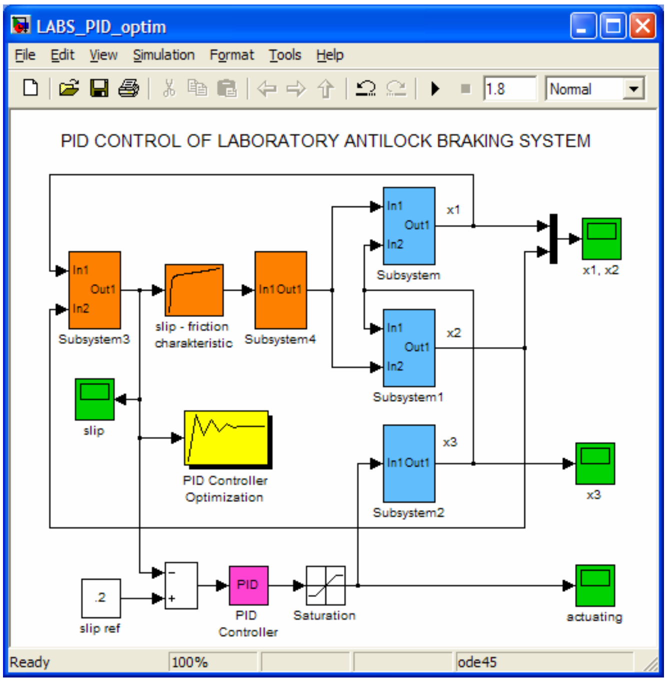


Fig. 3. PID feedback coontrol loop for LABS in MATLAB-Simulink

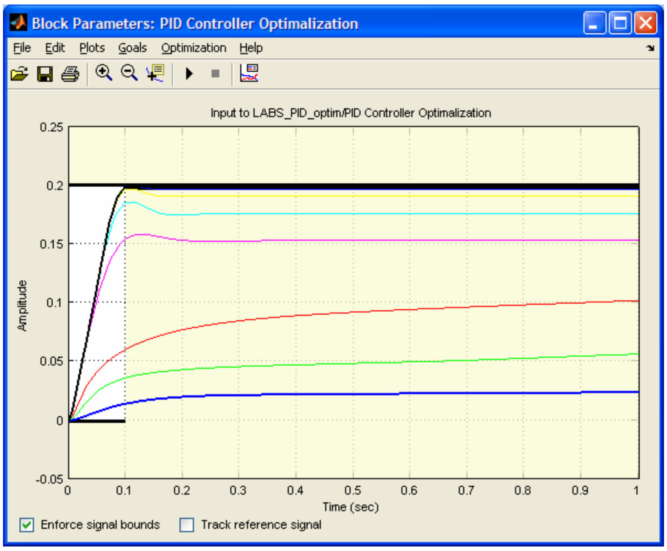
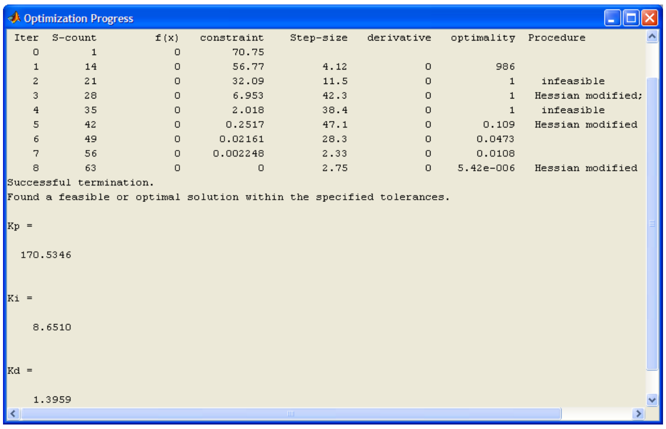

Block Parameters: PID Controller Optimalization
File Edit Plots Goals Optimization Help
Input to LABS_PID_optim/PID Controller Optimalization
Amplitude
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (sec)
☑ Enforce signal bounds
☐ Track reference signal

Fig. 4. PID tuning process



```
 Iter  S-count        f(x)   constraint   Step-size  derivative  optimality  Procedure
    0        1           0        70.75
    1       14           0        56.77        4.12           0         986
    2       21           0        32.09        11.5           0           1   infeasible
    3       28           0        6.953        42.3           0           1  Hessian modified;
    4       35           0        2.018        38.4           0           1   infeasible
    5       42           0       0.2517        47.1           0       0.109  Hessian modified
    6       49           0      0.02161        28.3           0      0.0473
    7       56           0     0.002248        2.33           0      0.0108
    8       63           0            0        2.75           0   5.42e-006  Hessian modified
Successful termination.
Found a feasible or optimal solution within the specified tolerances.

Kp =

  170.5346


Ki =

    8.6510


Kd =

    1.3959
```

Fig. 5. PID Tuning optimization procedure

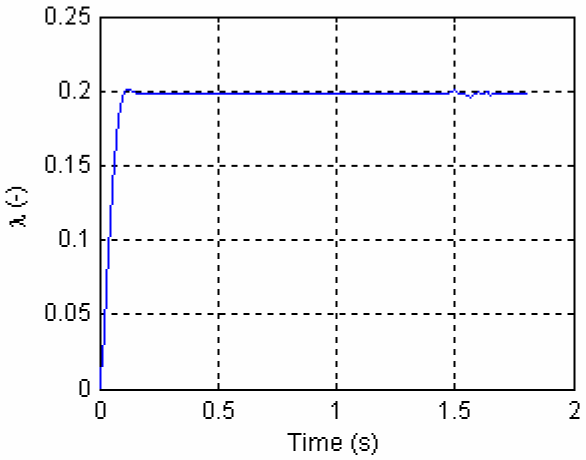


Fig. 6. PID tuning process

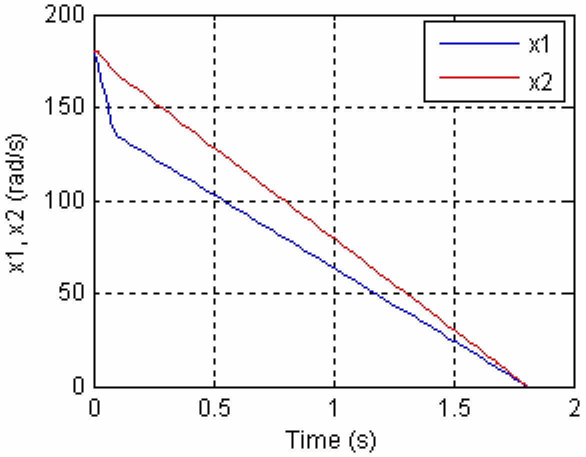


Fig. 7. Speed of wheels

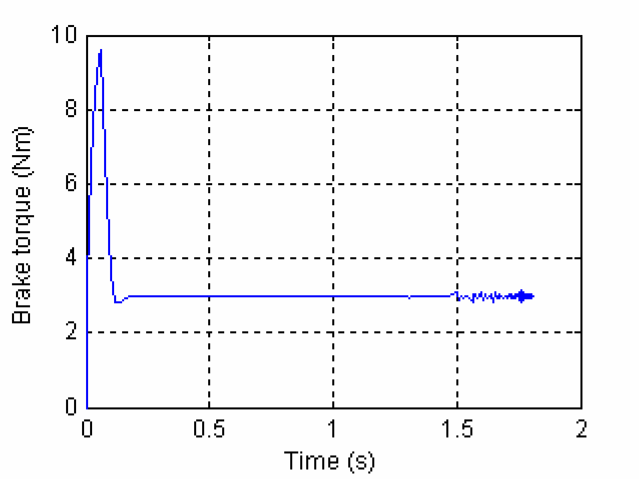


Fig. 8. Brake torque

4. CONCLUSIONS

The Laboratory ABS model provides possibility for development of various antilock brake control methods. PID control as a widely used method in practice is also suitable for this kind of control task. Based on identified dynamical model there is possible to solve PID control synthesis as an optimization problem with respect of various constraints and requirements for performance of control, like reference slip, control time and braking distance under given adhesive conditions. The software environment MATLAB – Simulink with its specialised Toolboxes is convenient tool for supporting of research activities in the framework of development cycle of active safety control systems in cars.

ACKNOWLEDGMENTS

This research is supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Bania, P., A. Korytowski, M. Szymkat and P. Gorczyca (2002). Optimal control of laboratory antilock brake system. *Preprints of the 16th World Congress of IFAC*, Prague, Czech Republic, July 3-8, 2005.

Johansen, T. A., I. Petersen, J. Kalkkuhl and J. Ludemann (2003). Gain-Scheduled Wheel-Sleep Control in Automotive Brake Systems. *IEEE Transactions on Control System Technology*, Vol. 11, No. 6.

Petersen, I. (2003). *Wheel slip control in ABS brakes using gain scheduled optimal control with constraints*. PhD Thesis, Norwegian University of Science and Technology, Trondheim.

Oniz, Y., E. Kayacan and O. Kaynak (2007). Simulated and Experimental Study of Antilock Braking System Using Grey Sliding Mode Control. *Proceedings of the IEEE SMC 2007*, pp. 90-95, Montreal, Canada, 2007.

The Laboratory Antilock Braking System Controlled from PC, User’s Manual, Inteco Ltd., Poland.

Comments – Remarks

Control of a Spark Ignition Combustion Engine Using dSpace Modular System [⋆]

**Jozef Csambál, Slawomir Wojnar, Marek Honek,
Peter Šimončič, Boris Rohal'-Ilkiv**

*Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315 and e-mail : jozef.csambal@stuba.sk*

Abstract: The aim of this paper is to present some research results achieved in the area of spark ignition (SI) internal combustion engines using free programmable devices dSpace. The main control loops (air fuel ratio (AFR) and preignition) of our experimental engine setup are studied and presented. In the case of AFR are two controllers are analyzed and applied: feedforward (base on a look up table) and feedback controller (base on a measurement with the use of lambda probe LSU 4.9). The preignition is controlled with a feedforward controller only. The used sensor's characteristics and main problems connected with measurement of the engine process variables are presented too. Finally, some emission requirements and physical limitations of presented devices are shown. The goal of experiments is to obtain practical knowledge and experience in the area of real-time SI combustion engines control. The resulting knowledge and experience will be further used for future purposes with more advanced experiments using model based predictive control approaches.

Keywords: combustion engine, feedforward control, AFR control, dSpace, discrete PID controller

1. MAIN MEASURED AND CONTROLLED VALUE

There is a couple of important sensors and actuators connected with two main control loops which are typical of an SI engine (air/fuel ratio control and ignition control).

1.1 Air flow measurement

Air flow into the intake manifold is measured at the DSpace's A/D converter with the use of air flow meter. A static nonlinear characteristic of the air flow meter is shown on the Figure 1 (a relationship between the air flow [kg/h] and the measured voltage [V]).

After the measured output signal is converted according to shown characteristic, it is must be recomputed to g/s. If this signal is integrated during the intake stroke, the amount of air sucked into the combustion chamber can be computed.

$$m_{sa} = \sum_{isa}^{iea} \dot{m} \tag{1}$$

where:
m_{sa} - mass of sucked air
$\dot{m}$ - air flow
isa - intake start angle
iea - intake end angle

The accuracy of this computation depends on amount of steps which take place during the integration.

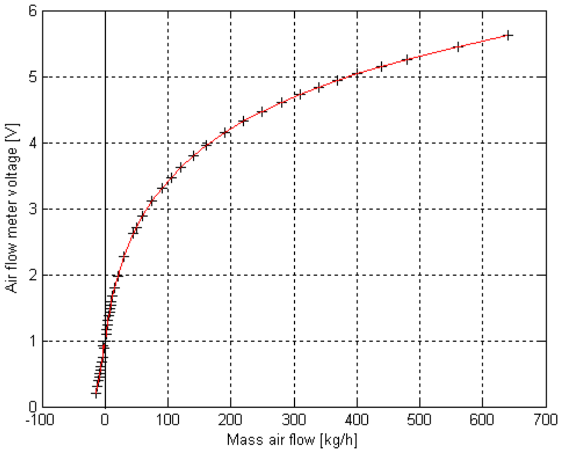


Fig. 1. Air flow meter characteristic

[⋆] The investigation reported in the paper was supported by Slovak Grant Agency APVV, project ID: APVV-0280 - 06, LPP-0096-07 and LPP-0075-09. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. The supports are very gratefully appreciated.

1.2 Injection

An injection is a process in which the fuel is injected to the intake manifold (exactly in front of the intake valves). The air fuel mixture is heated there (walls of the intake manifold and valves are warm) and when the valves are open, it will be sucked into the cylinder. Amount of fuel which should be injected can be precisely computed with the use of air fuel ratio (AFR) and mass of air. If the relationship between the mass of injected fuel and injector's opening time is known it is possible to compute the opening pulse width.

Needed characteristic depends on a differential pressure of injected fuel. This pressure can be express as:

$$\Delta P = P_{IF} - P_{IM} \tag{2}$$

where:

ΔP is a differential pressure
P_{IF} is a pressure of injected fuel
P_{IM} is a pressure in the intake manifold

Pressure of injected fuel is known and preset by the producer to 3bar. Whatever pressure in the intake manifold is changing and pulse width must by recomputed according it. There is two possibility, how to solve this problem.

- Pressure the intake manifold can be measured in a specific time and pulse width can be corrected in each cycle. In this case it is necessary to compute the negative pressure in the manifold. It is equal to:

$$P_{IM} = P_{MWG} - P_{ATM} \tag{3}$$

where:
$P_{IM}P$ is a pressure in the intake manifold
P_{MWG} is a pressure measured with the gauge
P_{ATM} is an atmospheric pressure

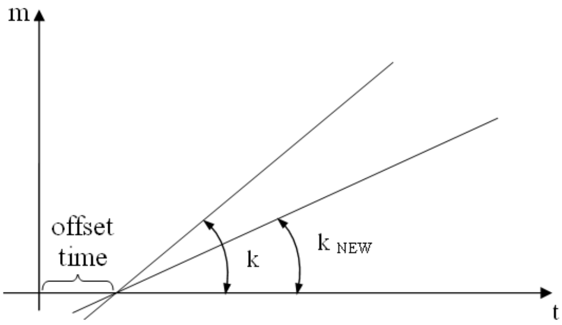


Fig. 2. Injector characteristic and differential pressure relationship

When the differential pressure and injector's characteristic made for preset differential pressure (4kPa) (Figure 2) are known it is possible to compute a new slope-intercept form of the curve which describes injector's opening time for the computed mass of fuel. Finally the pulse width can be computed with the use of new slope.

$$k_{NEW} = k * \frac{\sqrt{\Delta P_{NEW}}}{\sqrt{\Delta P}} \tag{4}$$

where:
$k_{NEW}P$ new slope of injector's opening time
k known slope defined in a preset pressure
ΔP_{NEW} is a differential pressure in the intake manifold
ΔP is a preset differential pressure (4kPa)

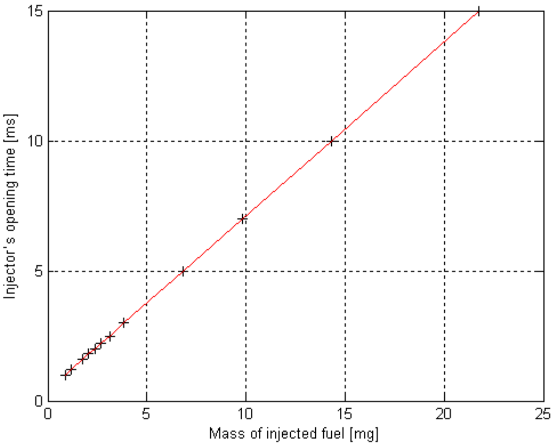


Fig. 3. Injector opening time vs. mass of the injected fuel

It is very important, that the value from pressure gauge is waving significantly (four times for every 720°) and the compensation which utilize the waving is necessary.

- Other one possibility require to use the injection rail with dedicated pressure control devices. In this case difference of pressures (see eq: 2) during the injection is constant and equal to preset value (3bar). This convenience cause, it is only necessary to recompute amount of injected fuel (computed before) into the length of injection pulse.

Fuel amount is directly proportional to the time of injection. Injectors have some offset time (they start to work after some time). This offset is growing with the falling power voltage (Figure 4). That is why it is necessary to control the voltage supply.

1.3 Ignition

An ignition is a process which enables to burn up the air fuel mixture which is pumped to the cylinder. The ignition occurs with predefined angle before the top dead center (TDC). This angle is changing with the engine's load and revolutions. In the case of dSpace the low side drivers and

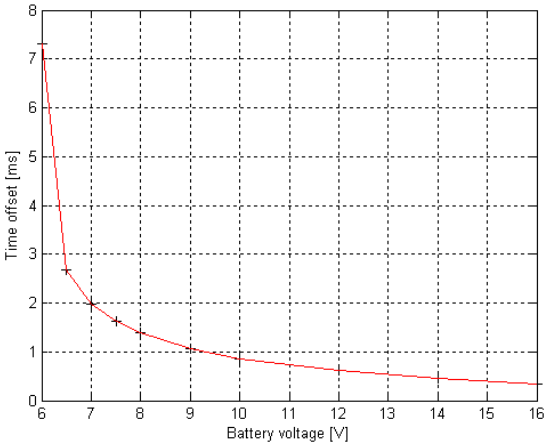


Fig. 4. Offset time vs. battery voltage

Table 1. Ignition timing map

RPM / Load	10	20	30	40	50	60	70	80	90
0	26	22	20	18	16	16	15	15	14
250	26	24	22	19	16	16	16	16	16
500	26	24	22	20	16	14	14	14	14
750	22	22	22	18	16	14	13	15	17
1000	16	15	14	13	12	12	14	16	18
1500	20	20	18	16	16	16	16	16	16
2000	26	26	24	24	22	22	20	20	20
2500	32	30	30	28	26	26	24	24	22
3000	36	34	32	30	30	28	26	26	24
3500	38	36	34	34	32	30	30	28	26
4000	38	38	36	34	34	32	30	30	28
4500	40	38	38	36	34	34	32	30	28
5000	40	40	38	38	36	36	34	32	30
5500	42	42	40	40	40	38	36	34	32
6000	42	42	40	40	40	38	36	34	32

specific ignition device are used. Thus the spark energy is ensure with the ignition device electronic and significant variable which must be controlled is the start ignition angle. Table 1 shows the specific ignition map which is used in this case.

It is very important, that the bigger is the ignition angle, the bigger is the pressure in the cylinder. It can't be too big, because the knocking can occurs.

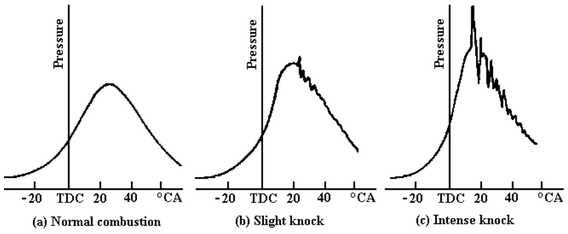


Fig. 5. Normal combustion (a) Slight knock (b) Intense knock (c)

Uncontrolled knocking can be not only unpleasant, but also destructive for the engine. It is possible to see on the Figure 5 that after some critical angle the knocking starts. It is not recommended to increase the angle, otherwise an intense knocking will occur.

1.4 Electronic throttle control

An electronic throttle (ET) is one of the most important parts of SI plant. It enables to change the air flow depending on an accelerator pedal and other variables from the control unit. It improves engine dynamic characteristic, reduces pollution and increases comfort. It is a strongly nonlinear plant, influenced by friction and gearbox backlashes. Detailed description of the ET control is included in "Control of an electronic throttle of spark ignition engine" and it wouldn't be shown here.

1.5 AFR

Mixture based on AFR will be burned in the cylinder. If the mixture is stoichiometric (air/fuel mixture is approximately 14.7 times the mass of air to fuel) all the injected fuel should be burnt. Otherwise the emission will grow, and even if the three-way catalytic converter (TWC) works in the exhaust pipe it is difficult to keep emission in necessary limits. As it is possible to see in Figure 6, the operating range area of TWC is not too wide. Two main circuits in which the AFR is computed and accomplished are defined in this part:

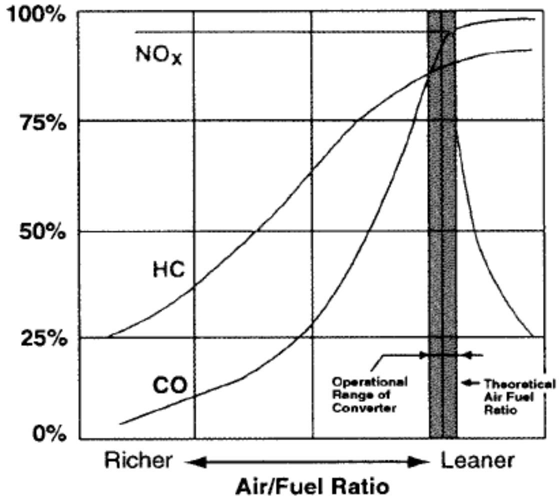


Fig. 6. TWC work area

- `First circuit` – a feedforward part which is based on a look up table. The output AFR value depends on a load and revolutions per minute (RPM). AFR is generally constant and equal to 14.67, but there are two situations when the AFR should be changed for well engine's work: acceleration and deceleration. In these moments the amount of injected fuel is changed (increases or decreases). The engine temperature is changing. When it is cold, wall wetting phenomena can occur. It causes, that some amount of fuel sticks to the cold walls (intake manifold walls, valve walls). Because of this a correction AFR circuit based on the temperature of cooling liquids is applied. If the engine is cold, bigger amount of a fuel will be injected. When the cooling liquids reach a proper temperature AFR is not corrected.

- `Second circuit` – the feedback circuit uses a PI algorithm for the AFR control. The amount of pumped air- fuel mixture and the exhaust gas oxygen measured value strictly depend on each other. But there is time delay between these two variables caused by the engine dynamic. Even for the set AFR value there can be measured lambda value different from the predicted one. That is why it is necessary to control a real AFR depending on the engine dynamics. The PI controller follows the error between the AFR and the measured AFR (signal measured at oxygen sensor and multiplied by 14.67) and influences the injector opening time.

1.6 An oxygen sensor

An oxygen sensor measures plant output (amount of oxygen in exhaust gasses) and allows to modify the current amount of air and fuel in the sucked mixture. The oxygen sensor starts to work after its body will reach 780°C. Till this time the feedback lambda control doesn't work and the emissions are controlled only with the feedforward controller. Water molecules are accumulated in the exhaust pipe. During the condensation water phase, the heating of Lambda sensor must be limited. Otherwise the sensor can be damaged by a thermo shock (Figure 7). In this case 10sec condensation water phase is defined. During this time we supply the lambda sensor with safe voltage (2V). After this time heating starts from this 2V level, not from 8,5V as the manufacturer suggests. It makes the heating longer, but more safety for lambda sensor.

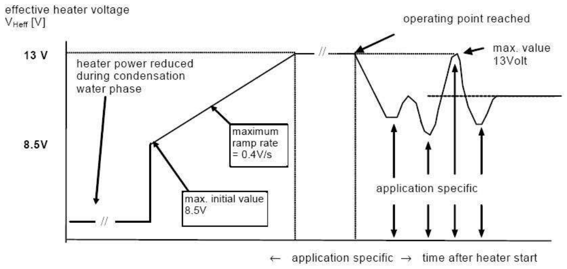


Fig. 7. Heating of oxygen sensor

2. EXPERIMENTAL RESULTS

In this part an experimental results will be shown. Measured data depict the growing engine RPM after the throttle opening. In the same time the set and the measured value of lambda was saved. This experiment was made at engine with no load. Because of this only little throttle opening angle was tested. It is possible to track described values for both the acceleration (Figure 9) and the deceleration (8). It can be seen that the acceleration causes a temporary increase of lambda value, whereas deceleration causes its decrease. As it was expected, the feedforward loop carries a specific value of lambda (applying AFR from the look up table) and the feedback PI controller allows to reduce the influence of the engine acceleration or deceleration on lambda value.

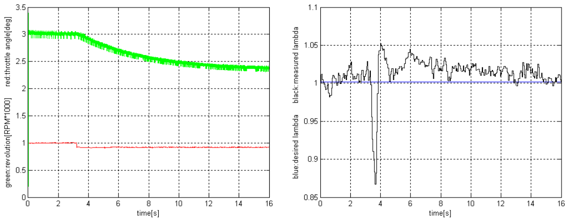


Fig. 8. Engine deceleration

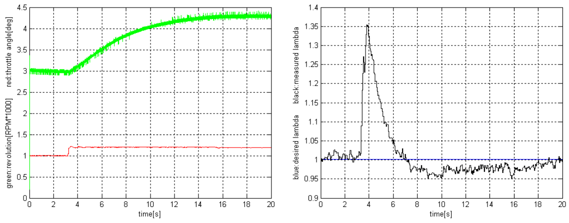


Fig. 9. Engine acceleration

3. CONCLUSION

The SI combustion engine leaves the wide room for improvements. As it was said all made experiments based on measurements at no load engine. It will be repeated soon with loaded engine. Then the predictive control algorithms will be tested and that is why it is necessary to know the mathematical model of the combustion engine. Tomas Poloni in his Ph.D. works describes how to design the predictive controller. The experiment of data acquisition is also described. This data are necessary to the intake air and fuel systems modeling. Then the ARX model which can describe the engine behavior will be built. All necessary systems needed for these experiments are prepared now (described above). A nonlinear model will base on local linear models. It is important to find local models describing the engine dynamics in theirs working points' area. This model must be precisely enough. An interpolation method enables to combine all models and use theirs in a global model which describes the system in its working area. The ARX model is the most used for systems modeling. Main advantage of applying ARX model is the fact that the information from local models are extended by the nonlinear interpolation function which create global NARX model. Parameters of local ARX models will be estimate from data acquired with air flow meter (AFM) and exhaust gas oxygen sensor (EGO). Experiment made in open loop is designed it this way which enable to separate intake air and fuel systems. Models obtained during identification experiment will be used for synthesis of predictive controller.

REFERENCES

dSpace (2007a). *Experiment Guide.* For Control Desk 3.2.

dSpace (2007b). *First Work Step.* For DS1103, DS1104, DS1005, DS1006 and MicroAutoBox.

dSpace (2007c). *Hardware Installation and Configration Guide.* For DS1005 PPC Boards and I/O Boards.

dSpace (2007d). *Hardware Installation and Configration Reference.* For Release 6.2.

dSpace (2008). *New Features and Migration.* For Release 6.2.

Guzzella, L. and Onder, C. (2004). *Introduction to Modeling and Control of IC Engine Systems.* Springer.

Honek, M., Csambál, J., Kopačka, M., and Rohaľ-Ilkiv, B. (2009). Experimental workbench for testing of advanced si engine control approaches. *IWCIT.*

Honek, M., Wojnar, S., Šimončič, P., Csambál, J., and Rohaľ-Ilkiv, B. (2010). Control of an electronic throttle of spark ignition engine. *Proces Control.*

Kiencke, U. and Nielsen, L. (2005). *Automotive Control System. For Engine, Driveline, and Vehicle.* Springer.

Polóni, T. (2008). *Aproximačné Modelovanie a Prediktívne Riadenie Súčiniteľa Prebytku Vzduchu Zážihového Spaľovacieho Motora.* Ph.D. thesis, Slovenská Technická Univerzita, Strojnícka Fakulta.

Comments – Remarks

An Experimental Workbench for Testing of Advanced Techniques for SI Engines control [⋆]

Jozef Csambál, Matúš Kopačka, Marek Honek,
Boris Rohaľ-Ilkiv

Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail : marek.honek@stuba.sk

Abstract: This paper presents an experimental workbench dedicated for implementation of different control approaches on a four-stroke spark ignition combustion engine. It consists of a host PC, target hardware, engine test bench and an engine itself. Target hardware is a high computational power unit able to execute real-time applications for control of a combustion engine. The main point of interest is to have a fully programmable control system, which offers to implement different control approaches and modify control algorithms in a quite easy way. It should also realize following functions: familiarize people with gasoline engine properties, do engine control simulation, design engine control system, and realize engine real-time simulation. It allows to leave the standard look-up table control of today's engines and implement advanced robust control algorithm. Our efforts should ensure better control of engine with increased power and decreased fuel consumption and emissions of CO, CHx and NOx in exhaust gasses.

Keywords: Engine control, Advanced algorithms, Real-time implementation, dSpace system, RapidPro system

1. INTRODUCTION

Development of combustion engines for vehicles of nowadays is mainly focused on the optimization of the combustion process. That concerns the biggest possible turn of the energy hidden in the fuel into the driving power of the engine while keeping the level of emissions as low, as possible. Speaking about spark ignition combustion engines, there are three groups of gases, causing a remarkable harm to the nature and living organisms: carbon hydrates (CH_x), carbon oxide (CO) and nitrogen oxides (NO_x) (L. Guzzella (2004)). Their acceptable (or allowed) amount in the exhaust gases is limited by international standards, which are having narrower tolerance with every release. That is creating an enormous pressure to the engine producers, who are forced to search for more and more sophisticated ways how to decrease the engine consumption and keep with the requirements of international standards.

One of the most significant invention in the field of consumption reduction is for sure introduction of the direct injection of the fuel into the combustion chamber (engine's cylinder bounded from the bottom side by a piston) and a numerical control of the combustion process by Engine Control Unit (ECU) introduced in 80's of the last century. Above listed measures decreased the fuel consumption at 20%.

2. STATE OF THE ART

2.1 Engine Control

Modern ECUs are controlling the combustion process based on the look-up table principle. Amount of the fuel injected into the cylinder depends on the input variables provided by motor sensors, in the simplest case motor revolutions and the motor load (determined from the position of the throttle valve situated in the air intake letting the fresh air in). Having defined a working point of the engine by these two independent variables is the ECU able to look-up into producer-predefined tables and choose the most suitable amount of the injected fuel and a correct position of the piston (in degrees before top dead center) for the ignition start. Value from that basic table can be consequently corrected by other tables taking into the account motor temperature, altitude of a car (and the change in the atmospheric pressure), voltage of a car battery, air humidity and others.

Introduced approach is based on a feedforward control applied for unstable working states of an engine – at its acceleration or deceleration (related to its revolutions).

[⋆] The work has been supported by the Slovak Research and Development Agency under grant LPP-0096-07. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

In the stable work region is activated a feedback control maintaining so called lambda ratio equal to one. The lambda ratio (Eq. 1) is the ratio of the real air mass in the cylinder related to the ideal mass of the air needed for a combustion of a fuel situated in the cylinder (Polóni (2008)):

$$\lambda = \frac{m_a}{L_{th} m_f} \tag{1}$$

where:

m_a is an air mass in a cylinder
m_f is a fuel mass in a cylinder
L_{th} is a stochiometric air/fuel mass ratio

Keeping the lambda ratio equal to one ensures the best possible catalytic reaction in the 3-way vehicle catalytic converter reducing the dangerous emissions to the possible lowest level. The lambda probe is situated in the engine exhaust duct and its working principle is based on sensing the residual oxygen amount in exhaust gases.

2.2 Problems and Solutions

Even though is the presented way of spark ignition engine control used in up-to-date cars it has its limitations and offers new directions for more advanced engine control. First of the problems is the look-up logic itself. The predefined table values, even in big amount, don't continually cover all possible working points of an engine and lets the ECU to interpolate among them, causing an error. The feedback control is also not used continually and a lambda probe situated in the exhaust duct works with an unwanted delay.

Solution of these negative phenomenas is an ECU of a new generation using dynamic algorithms computing a proper fuel (and air) amount for each working point, utilizing advanced robust and predictive control (see e.g. Nicolao et al. (2000), Wang et al. (2006), Manzie et al. (2002), Gorinevsky et al. (2003) and Polóni (2008)).

3. EXPERIMENTAL WORKBENCH

To start research in such a complicated field one needs sophisticated tools allowing to implement complex algorithms needing high computational force to ensure its real-time run. We have decided to implement to our experimental bench a dSpace system, usually used for HIL (hardware-in-loop) applications with attached RapidPro system (Fig. 1) dedicated especially for the engine control, which are going to be connected to a spark ignition engine. Main focus is on the creation of a fully programmable environment allowing us to reach into the motor control and add and/or change parts of a control system logic to be able to follow our target. As a software environment has been used Matlab/Simulink program offering enough flexibility through pre-programmed Simulink blocks and additional Simulink custom-made S-functions.

3.1 dSpace

dSpace Systems is a company providing hardware and software tools for automotive engineering, aerospace, and industrial control in the field of prototyping, model based control, simulation and calibration. As it has been already mentioned, their products are disposing huge computational power providing environment for the run of real time processes. Detailed description of dSpace modules operating in our configuration may enlighten functionality of the workbench from the point of view of the engine control, signal acquisition and generation.

`DS1005 PPC Board`

The DS1005 PPC Board is one of dSPACE processor boards that form the core of dSPACE modular hardware. Processor boards provide the computing power for the real-time system and also function as interfaces to the I/O boards and the host PC. The DS1005 is the board of choice for applications with high sampling rates and a lot of I/O capacity. Great processor power plus fast access to I/O hardware with minimum latencies make dSPACE processor boards considerably faster than solutions based on commonly available PCs. For detailed description of the board functionality, see (dSpace (2007d), dSpace (2007c), dSpace (2007b))

Main features:

- PowerPC 750GX, 1 GHz
- Fully programmable from Simulink
- High-speed connection to all dSPACE I/O boards via PHS bus

`DS2202 HIL I/O Board`

The DS2202 HIL I/O Board has been designed for hardware-in-the-loop simulation in automotive applications, and is tailored to the simulation and measurement of automotive signals. The board contains signal conditioning for typical signal levels of automotive systems and supports 2-voltage systems up to 42 V. Typical use cases are body electronics, transmission, and component tests performed by automotive suppliers, or as an additional HIL I/O board in large powertrain and vehicle dynamics HIL applications.

Main features:

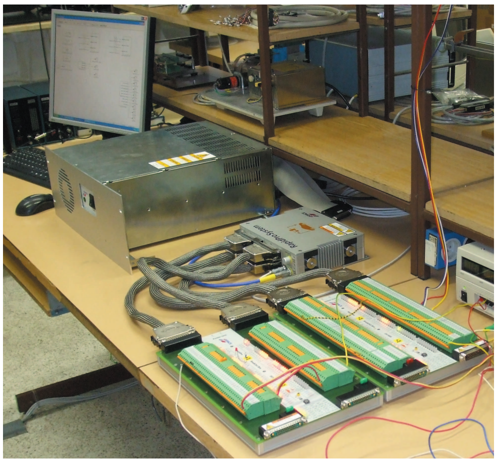


Fig. 1. Hardware assembly of the workbench

- I/O hardware with signal conditioning up to 42V
- 20 D/A channels and 16 A/D channels
- Up to 38 digital inputs Supports 2-voltage systems

`DS4121 ECU Interface Board`
The DS4121 provides the link between electronic control unit (ECU) or RapidPro system and a dSPACE modular system. Bypassing individual ECU algorithms and calculating them on a prototyping system is a typical application example. The connection is made via a dual-port memory plug-on device (DPMEM POD), on-chip debug interfaces like the DCI-GSI1 or an LVDS-Ethernet Link cable. Two independent ECUs can be connected to the DS4121 at the same time. This offers the flexibility to operate such things as powertrain control modules containing the engine and transmission controllers, or valve controls up to 12-cylinder engines.

Main features:

- Real-time interface to ECUs with 8-, 16- and 32-bit microprocessors
- Two LVDS channels for high-speed, low latency communication
- Standard ECU interface blockset fully integrated in MATLAB/Simulink

3.2 RapidPro

It is a sub-module of a dSpace system dedicated for the control of an ignition engine. It consists of two units:

- the Control Unit, able to handle complex I/O tasks
- the Power Unit, supporting high current signals for driving actuators

Particularly, the RapidPro module is used to acquire signals from motor sensors and to drive the motor actuators. Its functionality support the following installed modules:

- in the Control unit
  - · Lambda probe module
  - · Crankshaft/Camshaft signal DAq. module
  - · Knock sensor signal conditioning module
- in the Power unit
  - · Ignition and Lambda probe heating module
  - · Injector control

The following figure shows detailed signal and current flows between the bench and the engine.

3.3 Programming, User Interface, Development Software

The dSpace system is fully programable from Simulink. The Simulink builds a real time application for the target platform ds1005 by Real Time Workshop and uploads it inhere at the same time. Communication with the running application is performed at the host computer through the Control Desk software. This software enables to create a graphic user interface (GUI) for the running application in so called Design mode and to interact with the application in the Animation mode. As an application interface (API) for programming in the Simulink serves the Real-Time-Interface (RTI) included in the Simulink as a library of blocks. During the ECU programming, represented in this case by a modular RapidPro system, it is necessary to include a hardware topology file into the Simulink model. Mentioned file includes information about the hardware topology (its structural and firmware information) available from the Configuration Desk after the hardware scan of a system.

3.4 Engine

For the experiment is used an engine from Skoda Fabia 1.4 16V. This is a 4-cylinder four stroke spark ignition combustion engine.

Features:

- Engine code: AUA
- Engine volume: 1390 ccm
- Number of cylinders: 4
- Number of valves: 16
- Bore x stroke: 76,5 x 75,6 mm
- Compression ratio: 10,5
- Power: 55 kW (75HP) at 5000 rpm
- Torque: 126 Nm at 3800 rpm
- Injection: multipoint, 1 injector per cylinder
- Ignition: electronic, wasted spark
- Camshaft system: DOHC
- Fuel: Natural 95

3.5 Brake

As a brake simulating the load on the motor is used dynamometer based on eddy current principle. The brake slows an object creating eddy currents through electromagnetic induction which creates resistance, and in turn either heat or electricity.

4. PROJECT STATUS

To control a combustion engine means to engage a particular activator (injector or ignition coil) in the right position.

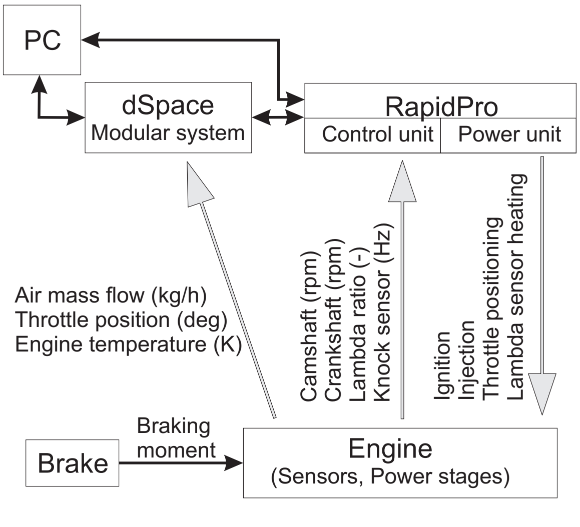


Fig. 2. Signal flow between the engine and the workbench

That means, a combustion engine is a task-triggered process. Tasks are in this case angle positions of the engine, or better, of the crankshaft. Motor works with a period defined through one revolution of the camshaft (360°). The crankshaft executes within this period 2 revolutions (720°) and each piston of the four stroke combustion engine executes all four strokes (intake, compression, combustion and exhaust). That is why is it crucial to know exact position of the crankshaft within the whole range of $\langle 0 \div 719.9^\circ \rangle$. This angle value gives us not only the instantaneous position of each piston, but differentiating its changing value in time determines angular velocity and acceleration/decceleration of the motor, as well. Logically it indicates, that the first stage of the project is a successful determination of the angular position of the engine.

4.1 Determination of an angular Position of the Engine

The RapidPro unit computes the angular position of the motor automatically from acquired signals of the crankshaft and camshaft sensors. Initial condition for unproblematic run is mutual synchronization of those signals, or so called synchronization of the angle measurement. Synchronization of the angle measurement is a process, when an angular position of 0° of the crankshaft is determined according to a specific shape and a mutual position of both - camshaft and crankshaft sensor signals. After the synchronization are all possible positions of the crankshaft known. The angle is determined with the precision of 0,1°. Sample of signals acquired from our experimental device is shown in the following figure:

From the signals can be also seen, how are the situated and distributed the marks of those shafts. Configuration of the used engine is following:

- Crankshaft: 60 marks and one gap created by 2 missing marks
- Camshaft: 3 marks (with coordinates from rising to falling edges related to revolutions of crankshaft (in (°)): 36→102, 258→456, 576→636).

Crankshaft positioned at 0° is defined at the rising edge of the first mark of the crankshaft after the gap. The synchronization starts at the engine start and is related to this position.

It succeeds as follows:

(1) As soon as the angle computational unit (ACU) detects the gap among crankshaft marks, sets the angle counter to zero degrees. From now on starts the measurement of engine revolutions (rpm).
(2) In the step 2 are compared defined camshaft marks with a real signal. If are they identical, the ACU starts to compute the crankshaft angle, if not, to the first mark after the gap is assigned the angle of 360° and the computation starts. Having this operation completed is the synchronization done. From this moment knows the ACU exact value of the angle related to the signal shapes of the shafts. If occurs a mismatch among them, the synchronization is lost.

4.2 Project Status and next Steps

Angle measurement of our engine has been already experimentally tested at the workbench. Crankshaft and camshaft signals were generated by dSpace and by hardware model of an engine (Fig.4). Both are generating identical signals as the real engine.

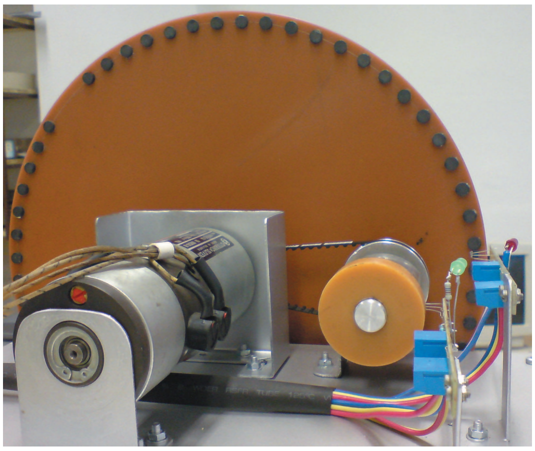


Fig. 4. Hardware simulation of crank- and camshaft signals

The today's effort is to start generation of signals for ignition coils and injectors. When it works, is the next step connection of a system to the real engine.

5. CONCLUSION

In this article the preliminary design of an experimental workbench for testing of advanced control algorithms for SI engines is presented. The first intended algorithms to be verified under the real-time conditions will exploit predictive and multimodel techniques developed and simulated in Polóni et al. (2007) and Polóni et al. (2008) for the air-fuel ratio control.

REFERENCES

dSpace (2007a). *Experiment Guide.* For Control Desk 3.2.

dSpace (2007b). *First Work Step.* For DS1103, DS1104, DS1005, DS1006 and MicroAutoBox.

dSpace (2007c). *Hardware Installation and Configration Guide.* For DS1005 PPC Boards and I/O Boards.

dSpace (2007d). *Hardware Installation and Configration Reference.* For Release 6.2.

Gorinevsky, D., Cook, J., and Vukovich, G. (2003). Nonlinear predictive control of transients in automotive vct engine using nonlinear parametric approximation. *Transaction of the ASME (Journal of Dynamic Systems, Measurement, and Control)*, 125(3), 429–438.

L. Guzzella, C.H.O. (2004). *Introduction to Modeling and Control of Internal Combustion Engine.* Springer.

Manzie, C., Palaniswami, M., Ralph, D., Watson, H., and Yi, X. (2002). Model predictive control of a fuel injection system with a radial basis function network observer. *Transaction of the ASME (Journal of Dynamic Systems, Measurement, and Control)*, 124, 648–658.

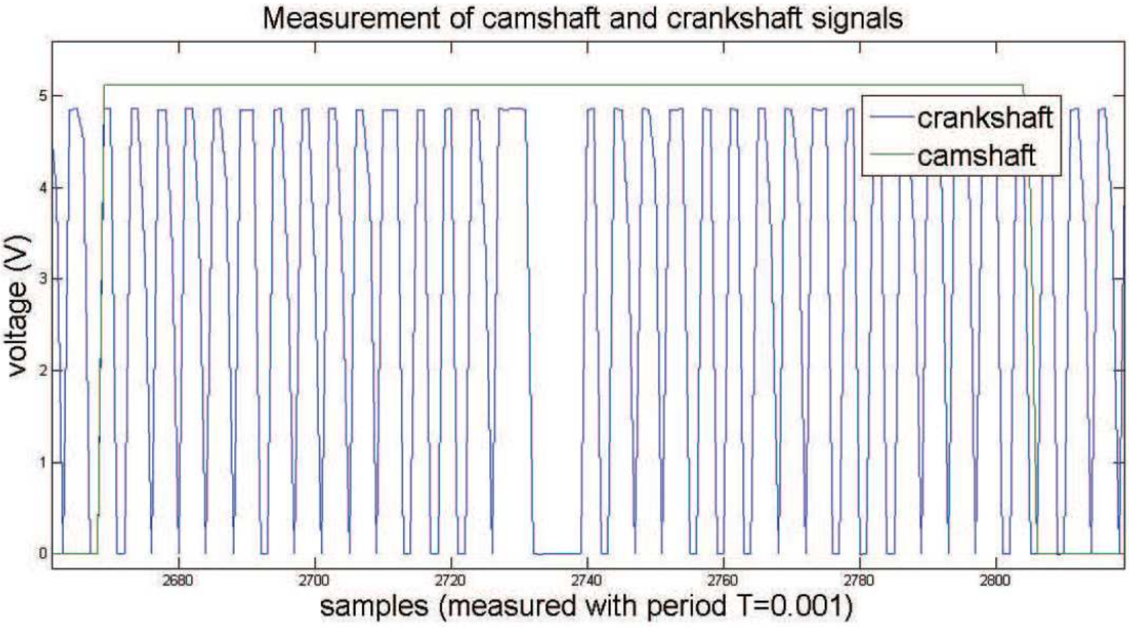


Fig. 3. Sample of camshaft and crankshaft signals

Nicolao, G.D., Magni, L., and Scattolini, R. (2000). *Nonlinear Receding Horizon Control of Internal Combustion Engines*, volume 26 of *Progres in systems and control theory*, 449–459. Birkhäuser Verlag, Basel, Switzerland. Nonlinear Model Predictive Control, F. Algöver, A. Zheng (Ed.).

Polóni, T., Johansen, T.A., and Rohaľ-Ilkiv, B. (2008). Identification and modeling of air-fuel ratio dynamics of a gasoline combustion engine with weighted arx model network. *Transaction of the ASME (Journal of Dynamic Systems, Measurement, and Control)*, 130(6), 061009–0610010.

Polóni, T., Rohaľ-Ilkiv, B., and Johansen, T.A. (2007). Multiple arx model-based air-fuel ratio predictive control for si engines. In *IFAC Workshop on advanced fuzzy and neural control*. Valenciennes, France. Conference paper MO5-3.

Polóni, T. (2008). *Aproximačné modelovanie a prediktívne riadenie súčiniteľa prebytku vzduchu zážihového spaľovacieho motora.* Ph.D. thesis, Strojnícka fakulta STU, Bratislava, Ústav automatizácie, merania a aplikovanej informatiky.

Wang, S.W., Yu, D.L., Gomm, J.B., Page, G.F., and Douglas, S.S. (2006). Adaptive neural network model based predictive control of an internal combustion engine with a new optimization algorithm. *Proc. IMechE Part D: J. Automobile Engineering*, 220, 195–208.

Comments – Remarks

Rapid Control Prototyping System dSpace: Application for an Internal Combustion Engine [⋆]

Marek Honek, Jozef Csambál, Slawomir Wojnar,
Matúš Kopačka, Peter Šimončič, Martin Lauko,
Boris Rohaľ-Ilkiv

Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail : marek.honek@stuba.sk

Abstract: The main goal of this paper is to describe one application of the dSpace Rapid Control Prototyping system (RCP system) to an spark ignition internal combustion engine. The control system is fully programmable from Matlab/Simulink environment. In the role of an application interface for handling of the hardware features there is the Real-Time Interface board library predefined. It provides blocks that implement the I/O capabilities of RapidPro hardware in Matlab/Simulink models. The ControlDesk handle real time applications running on the dSpace system and the ConfigurationDesk enables to change some relevant setting, affecting RapidPro modules. Graphical user interface of ControlDesk offers very simple access to all model variables during the runtime. With the use of the advanced RCP system presented above it was possible to design Model Based Predictive Control (MPC) algorithms for Air Fuel Ratio (AFR) real-time control.

Keywords: Rapid Control Prototyping system, dSpace system, RapidPro, AFR control

1. INTRODUCTION

Before advent of the Electronic Control Unit (ECU) on the field of Combustion Engine Control the two main control loops (for ignition and Air/Fuel ratio control) were realized continuously. Continuously working systems designated for control of these tasks didn't achieve sufficient precision, were difficult to adjust, unreliable and due to solutions of these inadequacies became costly. Nowadays, electronic "digital" control units (ECUs) are used. Such ECUs are systems where the heart of them is microcomputer which serve for data acquisition, computing, communications with other systems and driving of the actuator's power stages. In this book Guzzella and Onder (2004) the basic concept is presented. Additionally these systems must performs different diagnostic services. Design and building of such ECU is not a simple problem. In the phase of a control algorithm implementation, the register level programming for target platform (a specific microcomputer used in ECU) is required. For these reasons, development and testing of designed control algorithms became very difficult. For the reduction of difficulties Rapid Control Prototyping systems (RCP systems) has been developed over time. These RCP system are systems, which are usually hardware modular and have a good software support. Hardware modularity is a feature enabling to compose system, which meet the requirements. At least one module has to be a microprocessor module. This module provides more capabilities (computational power and larger memory) and so it is possible to test advanced control algorithms without the need for optimized code. Good software support means, that the producer of such modular RCP systems supplies the libraries and documentation. These libraries serves as a application interface for handling of devices's features (for example handling of interrupts, reading of the values from analog to digital conversions etc.). In addition, the host service code is running on the RCP system. This host service ensures the data exchange between the real-time hardware and a host computer. The next essential feature of RCP systems is, that they are software configurable. In one word, using of Rapid Control Prototyping systems in development enables to deal with programming of algorithms which are objects of investigation without the requirements to deal with problems like design and realization of hardware and handling of its features by low level programming. In the next chapters of this paper, will be explained the use of dSpace Rapid Prototyping system in a problem of combustion engine processes control.

[⋆] The investigation reported in the paper was supported by Slovak Grant Agency APVV, project ID: LPP-0096-07 and LPP-0075-09. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. The supports are very gratefully appreciated.

2. CONTROL OF SPARK IGNITION COMBUSTION ENGINE

Combustion Engine is naturally discrete-event system which works periodically, each cycle performs the same acts. In the case of four stroke combustion engine this cycle has a period of 720 ° of crank-angle. During this cycle each piston performs each of four strokes. Rapid Control Prototyping system (RCP system) dSpace with configuration as is possible to see at Figure 1 enables to handle every of the events in a crank-angle or time domain. In this paper will be presented results of A/F ratio predictive control based on web of linear models. This approach described in PhD thesis Polóni (2008) is one of the many published, based on computation in time domain. For this approach was crucial to obtain two linear models (for fuel and air path) in each operational point (revolutions × throttle position) of the web. Second typical approaches called Event Based described in Hendricks et al. (1994), Hendricks et al. (1995), Hendricks et al. (1993) and Hendricks et al. (2000) was used earlier. One of the Event Based A/F ratio control approach (based only on feedforward A/F ratio controller) was used, with goal to get engine to steady state for possibility of making the identification experiments and so build the mentioned web of local linear models.

2.1 RCP System dSpace Used for Control of Combustion Engine Processes

As was mentioned above, in Figure 1 is showed the scenario in which we use RCP system. The main real-time program made in Simulink is running on real-time proccesor included in DS1005 processor board. The part of RapidPro is processing of the signals from given sensors (providing them to Master dSpace system), generating of signals for driving of given actuators (look at Table 2) and generating of interrupts requests. For example, computed inputs (by given algorithms running on DS1005), injection start angle, injection time, ignition start angle and stop angle are sent (updating with new values) to RapidPro and then RapidPro generates so defined impulses to actuators. RapidPro can generate angle based interrupts which enables to trigger tasks in specific crank angles (for execution of interrupt service routines by real-time processor). The values of crank angle are necessary to be sent to RapidPro system too. For example these angle based interrupts can be used for capturing of given computed (by control algorithm running on DS1005) input and for consecutive update to RapidPro by invoking of software or time delayed interrupt request. Signals which are processed by RapidPro, for example crank angle, revolutions, temperature of lambda probe (heater element), lambda (temperature and lambda measurement are sampled by RapidPro internaly) is possible to read by sample time of any Timer Task provided by DS1005 processor board. In this sense it is also possible to read values of signals connected and processed by I/O board DS2202 (look at Table 1). Reading of these signals is possible in crank angle domain too.

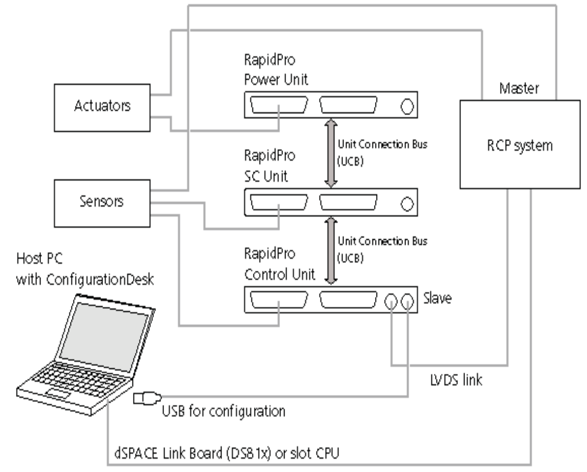


Fig. 1. RCP system configuration

The user has to design this sampling with regard to computational power, with the aim of achieving task scheduling without overrun situations and without overloading of RapidPro. Code generated by Real Time Workshop for target platform rti1005 has a measurement of turnaround time of each task in your model. For detailed information what the turnaround time of any task includes and for detailed specification of the hardware and software support see dSpace documentation dSpace (2007d).

In the next two tables are listed modules used for our application. There is a short description of its exploitabilities at second columns.

Table 1. List of dSpace modules

type	short description
DS1005 module	processor board, based on real-time processor PowerPC 750 is a main processing unit and host interface
DS2202 module	I/O board, analog inputs and outputs for: control of electronic throttle position, MAF sensor, MAP sensor, oil and cooling water temperatures, braking torque and battery voltage
DS4121 module	interface board, makes communication between Master and Slave systems

In this Table 3 are listed specifications about the used combustion engine. The specifications about used sensors and actuators and control algorithms used for specific purpose (lambda probe element heating, electronic throttle position control, control for constant revolutions by loading of engine by eddy current brake etc.) can be found in papers Csambál et al. (2010) and Honek et al. (2010).

2.2 Computation of Fuel Mass, Preignitions and Driving of the Actuators in Crank Angle Domain (Event Based)

The first problem which the user must solve is obtaining of information about shapes of wheel on the crankshaft, on the camshaft and their relation to position of pistons. For this purpose the measurement has been done. The next Figure 2 represent setup of this measurement. Data

Table 2. List of RapidPro modules (RapidPro as a Slave system)

type	short description
MC-MPC565 1/1 mod. (DS1602)	microcontroller module, based on MPC565 microcontroller
SC-CCDI 6/1 mod. (DS1637)	for connecting of crankshaft and camshaft signals
SC-EGOS 2/1 mod. (DS1634)	for connecting of exhaust gas oxygen sensor (lambda probe LSU4.9 Bosch)
SC-KNOCK 4/1 mod. (DS1635)	for connecting of knock sensor
PS-LSD 6/1 mod. (DS1662)	low side driver, for driving of lambda probe heater element and ignition modules
2 x PS-DINJ 2/1 mod. (DS1664)	for driving of electromagnetic injection valves
COM-USB-CI 1/1 mod. (DS1609)	for configuration from Host PC
COM-LVDS 1/1 mod. (DS1606)	module for communication with Master dSpace system

Table 3. Combustion engine type

type	Škoda Fabia 1.4 16V
code	AUA
cubature	1390cm^3
number of cylinders	4
number of inlet valves	16
borehole × throw	76.5 × 75.6 mm
compression ratio	10.5
power	55kW (75HP) by 5000rpm
torque	126Nm by 3800rpm
jetting	multi point, 1 injector / cylinder
ignition	electronic
fuel	Natural 95

Acquisition Toolbox with NI6008 was used for measuring of the mentioned wheel's shapes. Furthermore the signal from hall sensor was measured, which scanned the transfer of the magnet under it. This magnet was glued in such position, that measured impulse carried information about TDC position.

The result gave the possibilities to configure the necessary setup for crankshaft angle and speed measurement together with relation to piston's positions. This can be seen in Figure 3 and in the next list.

Shape of crankshaft wheel:

- has 60 teeth.
- has 1 gap.
- 2 missing teeth per gap.

Shape of camshaft wheel:

- has 3 markers.
- coordinates of rising edges are [36 ° 258 ° 576 °].
- coordinates of falling edges are [102 ° 456 ° 636 °].

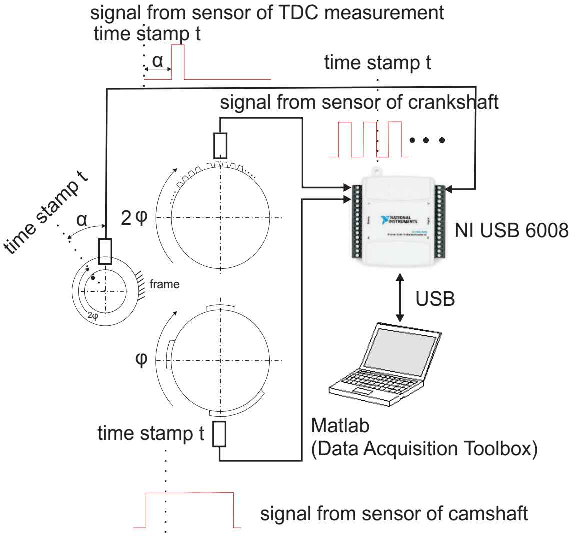

signal from sensor of TDC measurement
time stamp t
α
time stamp t
signal from sensor of crankshaft
time stamp t
α
2φ
frame
2φ
NI USB 6008
USB
φ
time stamp t
Matlab
(Data Acquisition Toolbox)
signal from sensor of camshaft

Fig. 2. Measuring system

With this knowledge the basic feedforward A/F ratio and ignition controls can be explained. In this case we will only talk in detail about task scheduling, which are needed for explaining of feedforward A/F ratio and ignition controls. We will not present achieved quality of these control strategies, because it served only for achievement of desired engine's steady state and after it the A/F ratio controller was turned off for making identification experiments. This feedforward A/F ratio controller meets the requirements on the quality of control in steady state.

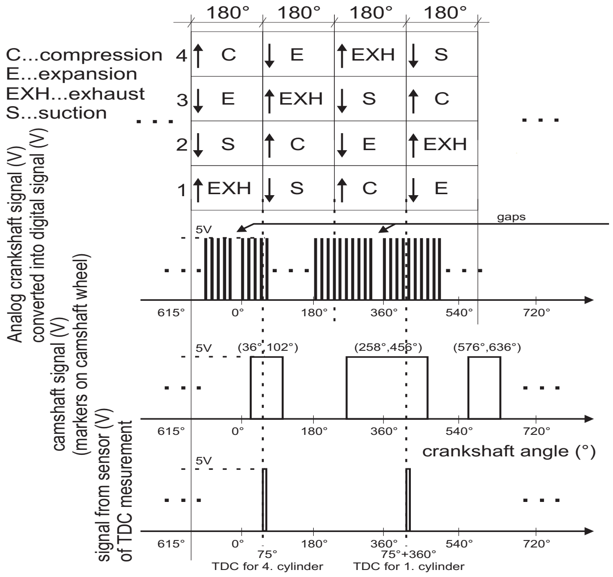


Fig. 3. Engine's events order

In the Figure 4 are the crank angle axis and markers of the moments in which the actions are triggered. Before the explaining of this timing, is the Table 4 which contains the list of tasks arranged by the priority.

First task with sample time 0.0001s serves for sampling of time and MAF signal. Such sampled time and MAF signal is used for computation of air mass sucked to combustion chamber in time of one suction stroke (numerical integration) (by 6000rpm of revolutions the angle of 3.6° takes 0.0001s). This information about air mass sucked per one suction stroke is used for computation of needed fuel mass for achieving of desired fuel mixture (A/F ratio). Consequently this actual fuel mass is recomputed on injection time, from characteristics of injection valve with regard to the fact that controller of fuel pressure is mounted (so a constant difference of pressure is achieved). Then it is captured and updated for driving of the cylinder which is in the actual suction stroke. It is because the injection is started at beginning of compression stroke for the given cylinder. Computation of preignition is performed at the start of each cycle (at crank angle 0°). This datum is then updated separately for all of cylinder in order which is present in Figure 4. In this Figure 4 are showed all moments in which the selected tasks are performed.

Table 4. List of the tasks

priority	type	short description
1	TimerTask1, 0.0001	some measurements, look at the text below
2	Angle based interrupt, 3.6° periodically started from TDC	numerical integration of MAF signal
3	Angle based interrupt (master1), 180° periodically started from TDC	capture and reset of air mass
4	Software interrupt (slave1)	computation of fuel mass => computation of injection time
5	Angle based interrupt (master2), 165°	capture of injection time for 1. cylinder
6	Software interrupt (slave2)	update of injection time for 1. cylinder
7 to 12	Angle based interrupts for next cylinders in order 3,4,2	captures and updates of injection time
13	Angle based interrupt, 0°	computation of preignition
14	Angle based interrupt, 120°	update of preignition for 2. cylinder
15 to 17	Angle based interrupts for next cylinders in order 1,3,4	updates of preignition
18	TimerTask2, 0.0005	for reading of λ measurement
from 19	Others interrupts	for all sample times included in the given Simulink model

3. RESULTS

Engine work-bench with RCP system dSpace ensure the flexible prototyping. It enables to change control strategy often, and quite fast. The RapidPro hardware works as an extension to dSPACE prototyping systems. It provides the hardware and software support. Its modules (mentioned above) and the Real-Time Interface board library made the experiment preparing easier and enabled to focus on the main purpose of the experiment (AFR controller). As it is shown in the Šimončič et al. (2010) model based predictive controller for the AFR was designed and tested. It was undoubtedly easier with the use of the RCP system which provide as high computational power and large memory as described system.

REFERENCES

Csambál, J., Honek, M., Wojnar, S., Šimončič, P., and Rohaľ-Ilkiv, B. (2010). Control of a spark ignition combustion engine using dspace modular system. In *Process Control*.

dSpace (2007a). *Experiment Guide*. For Control Desk 3.2.

dSpace (2007b). *First Work Step*. For DS1103, DS1104, DS1005, DS1006 and MicroAutoBox.

dSpace (2007c). *Hardware Installation and Configration Guide*. For DS1005 PPC Boards and I/O Boards.

dSpace (2007d). *Hardware Installation and Configration Reference*. For Release 6.2.

dSpace (2008). *New Features and Migration*. For Release 6.2.

Guzzella, L. and Onder, C. (2004). *Introduction to Modeling and Control of IC Engine Systems*. Springer.

Hendricks, E., Chevalier, A., and Jensen, M. (1995). Event based engine control: Practical problems and solutions. In *SAE Technical Paper No. 950008*.

Hendricks, E., Jensen, M., Chevalier, A., and Vesterholm, T. (1994). Conventional event based engine control. In *SAE Technical Paper No. 940377*.

Hendricks, E., Jensen, M., Kaidantzis, P., Rasmussen, P., and Vesterholm, T. (1993). Transient a/f ratio errors in conventional si engine controllers. In *SAE Technical Paper No. 930856*.

Hendricks, E., Jensen, M., Kaidantzis, P., Rasmussen, P., and Vesterholm, T. (2000). Avoiding signal aliasing in event based engine control. In *SAE Technical Paper No. 2000-01-0268*.

Honek, M., Csambál, J., Wojnar, S., Šimončič, P., Kopačka, M., and Rohaľ-Ilkiv, B. (2010). Rapid control prototyping system for a combustion engine. In *Mechanical Engineering*.

Polóni, T. (2008). *Aproximačné Modelovanie a Prediktívne Riadenie Súčiniteľa Prebytku Vzduchu Zážihového Spaľovacieho Motora*. Ph.D. thesis, Slovenská Technická Univerzita, Strojnícka Fakulta.

Šimončič, P., Kopačka, M., Honek, M., Csambál, J., Wojnar, S., and Rohaľ-Ilkiv, B. (2010). Real-time air/fuel ratio control of a spark ignition engine using mpc. In *Mechanical Engineering*.

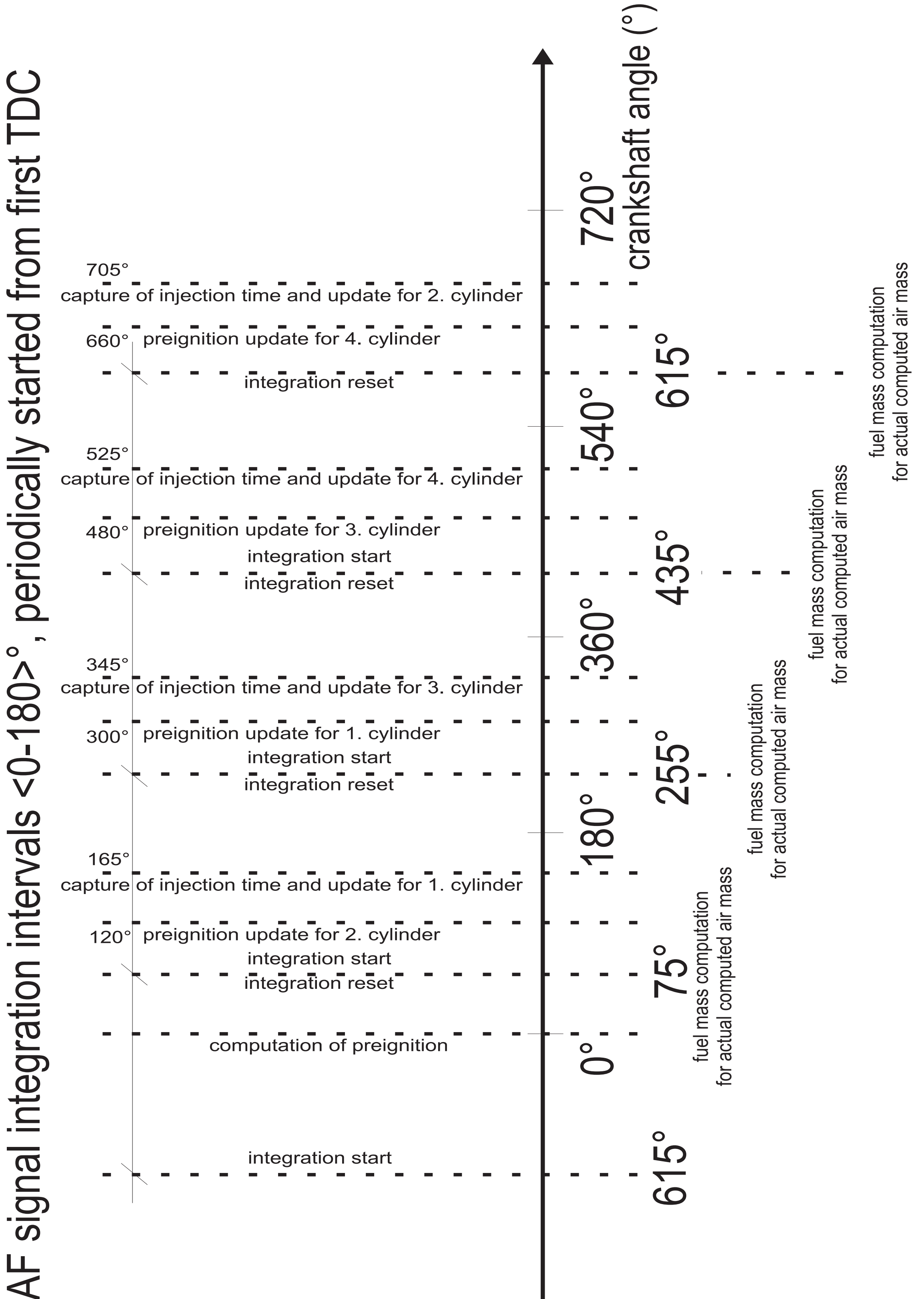

MAF signal integration intervals <0-180>°, periodically started from first TDC
crankshaft angle (°)
0°
180°
360°
540°
720°
615°
75°
255°
435°
615°
integration start
computation of preignition
integration start
integration reset
120° preignition update for 2. cylinder
165°
capture of injection time and update for 1. cylinder
integration start
integration reset
300° preignition update for 1. cylinder
345°
capture of injection time and update for 3. cylinder
integration start
integration reset
480° preignition update for 3. cylinder
525°
capture of injection time and update for 4. cylinder
integration reset
660° preignition update for 4. cylinder
705°
capture of injection time and update for 2. cylinder
fuel mass computation
for actual computed air mass
fuel mass computation
for actual computed air mass
fuel mass computation
for actual computed air mass
fuel mass computation
for actual computed air mass

Fig. 4. Timing of the main tasks

Comments – Remarks

Control of the SI engine's electronic throttle with friction compensation ⋆

**Marek Honek, Jozef Csambál, Sławomir Wojnar,
Peter Šimončič, Boris Rohaľ-Ilkiv**

*Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail: marek.honek@stuba.sk*

Abstract: The paper deals with an automotive electronic throttle control problem. The automotive electronic throttle is a dc-motor-driven valve that regulates air inflow into a spark ignition internal combustion engine. The proper electronic throttle functionality is critical to driveability, comfort, fuel economy and emission performance of present day passenger vehicles. Therefore the control system of electronic throttle should ensure fast and accurate reference tracking of the desired valve plate angles. These demands are hard to accomplish because of strong effects of friction and the limp-home nonlinearity. The proposed method applies a discrete PI controller with the nonlinearity and friction compensation.

Keywords: automotive electronic throttle, nonlinear behaviour compensation, friction compensation, discrete PI controller

1. INTRODUCTION

The electronic throttle (ET) is a valve used in vehicles to control the air flow into the engine combustion system. The ET consists of a DC drive, a gearbox, a valve plate, a dual return spring and a position sensor (potentiometer). The heart of this system is a plate swung by the use of the DC motor. There is also a spring which provides a torque working against the DC motor. This construction has multiple sources of nonlinearity. These sources are:

- usually higher stiffness of return spring near by Limp-Home (LH) position
- significant static friction
- back-lash effect

The use of non-linear control theory is addressed as well. The most used control strategy is PID with compensation of nonlinearities. In work Grepl and Lee (2008) has been used feedback linearization and PID controller tuned experimentally. For compensation of nonlinearities (using feedback linearization) is necessary to obtain estimates of parameters of model which describe behavior of nonlinear part. For this purpose experimental identification has been done in this paper. Very similar approach is more detailed explained in Pavković et al. (2005). In addition in this paper is described the process of auto-tuning control strategy due to process parameters variation. Such control strategy is robust to variation of parameters like resistance of armature, battery voltage and aging of ET at all. The design of PID controller used here (Damping optimum analytical design method) was based on linear model of second order. In this paper is information about requirements on feedback step response. Settling time have to be approximately 70ms and steady-state accuracy better than 0.1 degrees. Author Wilson in his bachelor work Wilson (2007) shows approaches based on MPC and confront it to PID control strategies. That MPC is synthesized by using of MPT toolbox. PWA model was used in this work. For obtaining of this model was used HYSDEL software, which allows to build PWA model described by continuous mathematical equations (time continuous or time discrete dynamical models) combined with logical expressions, which express conditions for operation of concrete dynamical behavior. This hybrid modeling allows to capture nonlinear behavior with good precision. In this work is the derivation of the model by using the first principles. Here is also survey of invented friction models. Modeling of friction is main point for modeling of electronic throttle. A model which encapsulate every essential properties of friction is derived in Canudas de Wit et al. (1995). Also approach for compensation of friction is discussed in this paper. More complex sight on the friction modeling and on the control of machines with friction is done in Armstrong-Hélouvry et al. (1994). In this paper Vašak et al. (2007) is ET modeled as a DTPWA system including behavior of reset-integrator form of friction model. This friction model captures presliding effect (0.3 degrees), what is important because they wanted to design very accurate (up to the measurement resolution 0.1 degrees) tracking system. Pro-

⋆ The investigation reported in the paper was supported by Slovak Grant Agency APVV, project ID:LPP-0096-07 and LPP-0075-09. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. The project is also co-financed from the state budget of the Slovak Republic. The supports are very gratefully appreciated.

posed tracking system is based on explicit (offline) solution of MPC with the purpose to implement it on a low-cost hardware. For estimation of unmeasured states they use UKF. Finally they compare experimental results to results from control with PID controller Pavković et al. (2005) and termination is:

- faster transient
- the absence of an overshoot
- static accuracy within the measurement resolution

In this case very simple control strategy has been chosen. An experimental research allows to apply a PI controller and compensators of nonlinearities. For an implementation of the control algorithm the Real Time Windows target has been used.

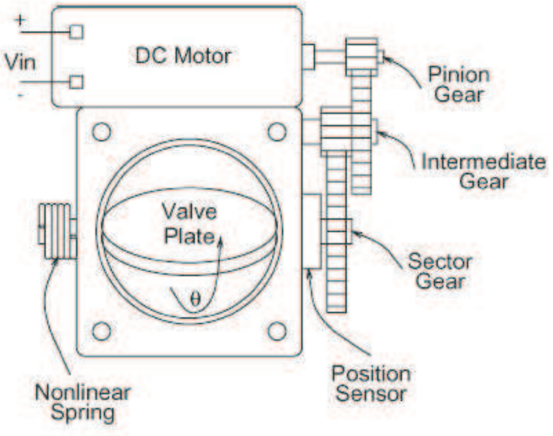


Fig. 1. Electronic Throttle Body

2. HARDWARE DESCRIPTION

At the beginning of this section it is necessary to explain some characteristics. We have a ET from Siemens VDO number 408-238-321-006.

2.1 Input and output ranges

This ET has two potentiometers for measurement of position. These potentiometers have the range from 0V to 5V, but the whole range is not used. As we will see later on static characteristic, ET has range of motion showed in the table 1.

Table 1. Possible positions

	(rad)	(deg)	(V)
closed position	0.1181	6.76	0.500
LH position	0.2745	15.73	0.915
opened position	1.7052	97.70	4.56

For the control system is position computed from the both potentiometer's signals. This position signal is filtered by average filter, which input is sampled with frequency of measurement 1kHz and its output is 10 times slower, 0.1kHz. This filter computes its output from data vector of 8 samples, because there is outliers (minimum and maximum value of data vector) elimination feature. This filtered signal is applicable for computing of rate $(V \cdot s^{-1})$ too.

Input to the plant from control system is first routed to a H-bridge. H-bridge is electrical circuit controlled by microcontroller and it produces PWM signal for control of the DC motor. Range for H-bridge control signal is:

Table 2. Control signal to H-bridge

Range of input to plant	direction of motion	duty cycle
from 0V to 5V	negative	from 0% to 100%
from 5V to 10V	positive	from 0% to 100%

As the real time platform, Real Time Windows Target has been used. As a I/O interface laboratory PC card Advantech PCI1710HG has been used. This laboratory PC card provides required analog inputs and outputs. Finally, developed algorithm will be applied to SI combustion engine, where the real-time platform is based on DS1005 modular system.

2.2 Quasi-static characteristic

Proposed control system in this paper is based on experimental knowledge. In this sense was needed to get knowledge about input-output behavior of the ET. Quasi-static characteristic at figure 2 shows relation between input and output in steady-state. This characteristic is quasi-static, because input signal was ramp with very low frequency. We can see that butterfly valve remain at LH position (for opening phase, input signal from 5V to 7.1V) to the moment when input will reaches the value 7.1V. Approximately for the value of input equal to 7.7V is valve fully opened. For decreasing of input signal (return phase from fully opened position) movement of valve start for 6.8V and valve reaches LH position for 6.4V. From these two traces of opening and returning, can be observed a region in which the action value has to be. The control system was designed according to this. For closing phase (under LH position) is situation identically. In the section Control system will be clarified how this knowledge can to be used with regard to compensation of nonlinearities and PID controller synthesis.

In the paper Pavković et al. (2005) are showed relations between the static characteristic and parameters of nonlinear model of ET.

3. ET MODEL DESCRIPTION

Some indication for modeling of ET will be given in this section, because estimation of parameters is not completely done at this time. Modeling of ET by using first principle can be divided to two main parts. Modeling of DC motor and modeling of mechanical parts. DC motor can be described by linear differential first order equation. Mechanical parts have to be modeled by nonlinear

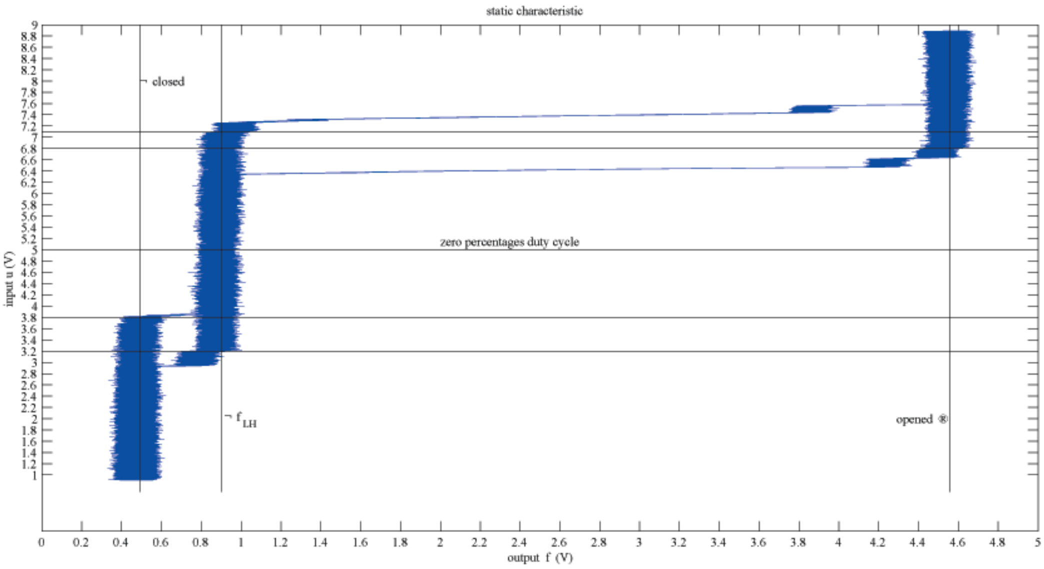


Fig. 2. Static characteristic

equation of motion. Nonlinear behavior of ET is caused by nonconstant stiffness of the return spring, back-lash effect caused by movement through the LH position and the most significant is friction.

Electrical DC motor can be modeled using a linear differential equation. At figure 3 the sum of voltages 1 in the closed circuit has to be equal to zero, as described by second Kirchhoff's law:

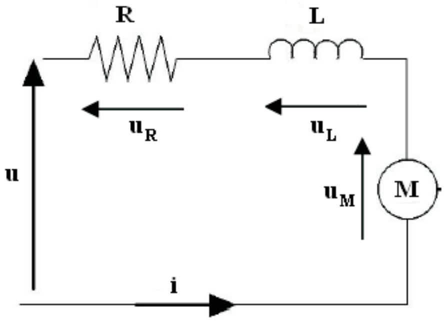


Fig. 3. Electric scheme of DC motor

$$u = Ri + L\frac{di}{dt} + k_e\dot{\varphi} \qquad (1)$$

where:

u is a sum of voltages
k_e is the electromotive force constant
R is the electric resistance
L is the electric inductance

Then equation of motion is in the form:

$$J\ddot{\varphi} = m_G - m_{RS} - m_F \qquad (2)$$

Torque generated by the motor is proportional to the electric current.

$$m_G = k_e i \qquad (3)$$

where:

u is a sum of voltages
$m_{RS}$is the torque acting against
the returning spring
m_G is the torque generated by the motor
m_F is friction torque

m_{RS} is the torque acting against m_G and is caused by the returning spring. Stiffness parameter of this spring is a function of throttle valve position. This is one of the sources of nonlinearity. Spring nonlinearity problem is sufficiently described in referenced work Pavković et al. (2005). Explanation of compensation for this nonlinearity will be in section Control system.

m_F is friction torque and consists of two parts, linear (viscous damping) and nonlinear (static friction). Works Canudas de Wit et al. (1995), Armstrong-Hélouvry et al. (1994) deal with modeling and compensation of the friction.

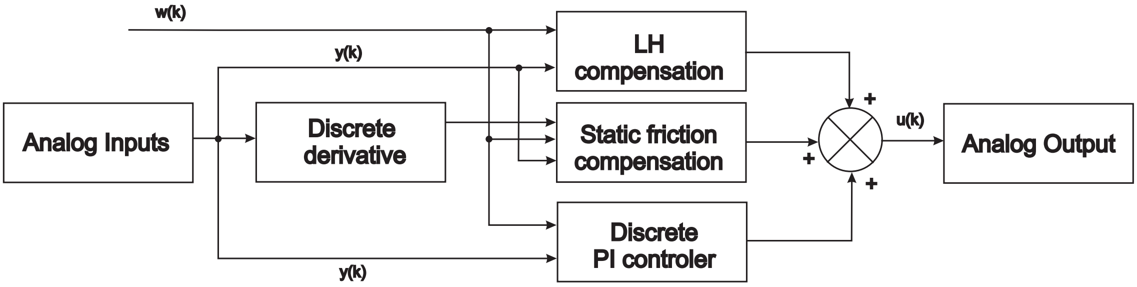


Fig. 4. Structure of the control system

4. CONTROL SYSTEM

Finally at this section will be described proposed control system. Control system consist of compensation of static friction, higher stiffness of return spring and discrete PI controller in velocity form how it is demonstrated at figure 4.

Compensation of friction has to apply breakaway torque to mechanical system by impulse for non-zero error and approximately for zero rate of motion.
LH compensation has to apply three basic levels of input (3.65V, 5V and 6.95V) regarding to fact perceptible at figure 2 of quasi-static characteristic. This block works in sense, that waits (not modify its output) while the output of the plant reaches the LH position.
This features of compensations are perceptible at figures 5 and 6.

Action law of discrete PI controller is in the form at next two equations.

$$u(k) = u(k) + \Delta u(k) \tag{4}$$

$$\Delta u(k) = K_p(e(k) - e(k-1)) + K_i e(k) \tag{5}$$

This PI controller has build in anti-windup feature and scheduling of parameters K_p and K_i. This scheduling is applied, because now is not applied compensation of friction for very low values of error (under 0.05V ≈ 1.1 degrees) and for setting of parameters K_p and K_i to zeros values for obtaining of non-periodic steady-states. Parameters of PI controller were tuned experimentally regarding to quasi-static characteristic at figure 2.

5. EXPERIMENTAL RESULTS AND CONCLUSION

Experimental results you can see at the figures 5 6, feedback responses and input, rate of motion respectively. Achieved feedback response time was less than 0.5s and accuracy in steady state approximately 0.6 degree.

REFERENCES

Armstrong-Hélouvry, B., Dupont, P., and Canudas de Wit, C. (1994). A survey of models, analysis tools and compensation methods for the control of maschines with friction. *Automatica.*

Canudas de Wit, C., Olson, H., Åström, K.J., and Lischinsky, P. (1995). A new model for control of systems with friction. *Transactions on automatic control.*

Grepl, R. and Lee, B. (2008). Modelling, identification and control of electronic throttle using dspace tools. 35. Humusoft ltd. Prague.

Pavković, D., Deur, J., Jansz, M., and Nedjeljko, P. (2005). Adaptive control of automotive electronic throttle. *Control Engineering practice.*

Vašak, M., Baotić, M., Petrović, I., and Nedjeljko, P. (2007). Hybrid theory-based time-optimal control of an electronic throttle. *IEEE Transactions on industrial electronics.*

Wilson, T. (2007). Constrained optimal control of an electronic throttle body. Bachelor work at The University of Queensland.

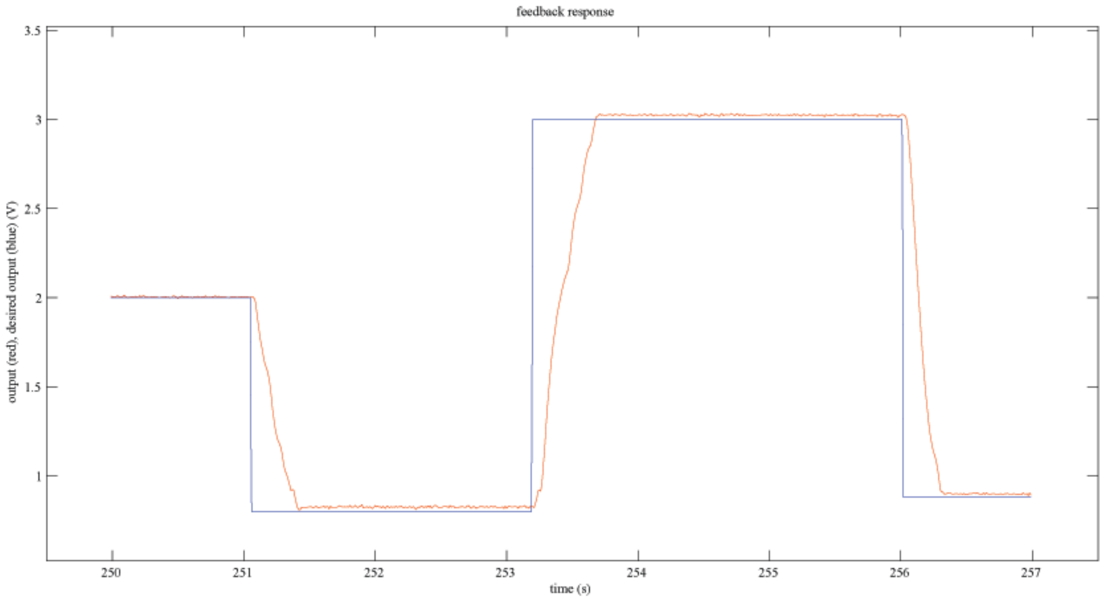


Fig. 5. Feedback response

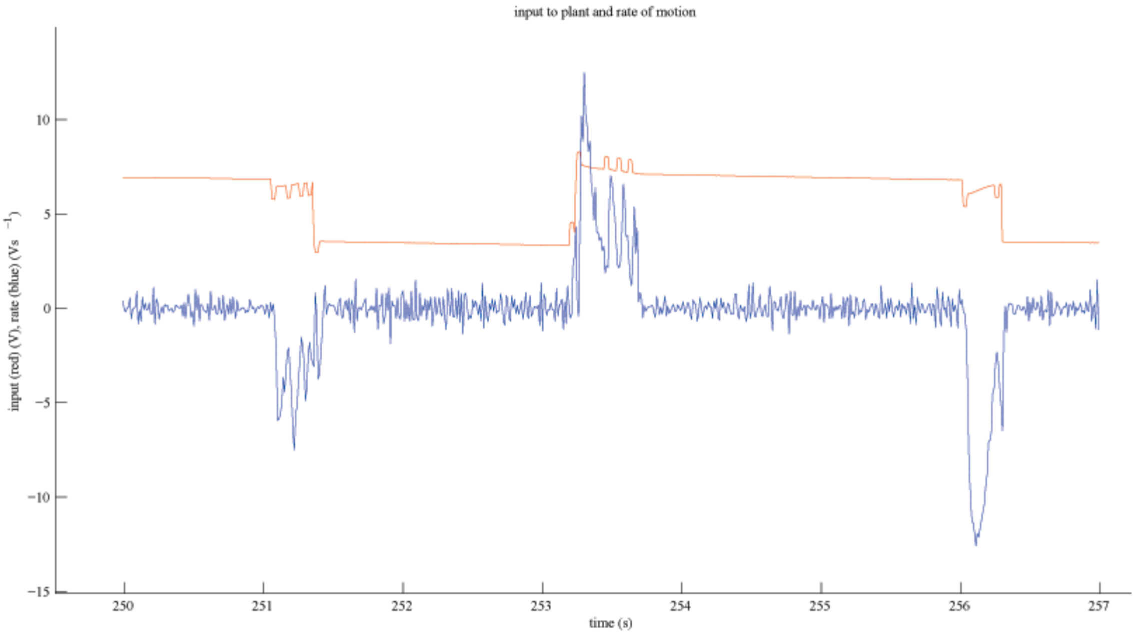


Fig. 6. Input to H-bridge and rate of motion

Comments – Remarks

Design of MPC Controller Using ACADO Toolkit ⋆

Matúš Kopačka *, **Bart Saerens** **, **Hans Joachim Ferreau** **,
Boris Houska **, **Moritz Diehl** **, **Boris Rohaľ-Ilkiv** *

* *Slovak University of Technology in Bratislava*
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
tel: +421 2 5249 5392; e-mail: matus.kopacka@stuba.sk

** *Katholieke Universiteit Leuven*
Electrical Engineering Department (ESAT-SCD)
Kasteelpark Arenberg 10
3001 Leuven-Heverlee, Belgium
tel: +32 16 321884; e-mail: moritz.diehl@esat.kuleuven.be.

Abstract: The objective of this work is to present design of a predictive controller using the ACADO toolkit. It is an open-source framework for automatic control and dynamic optimization dedicated for a real-time control. Toolkit which is presented offers a neat way of a controller build-up. It is possible with the user friendly and intuitive C++ coding environment and computational inexpensiveness. Finally the problem free real-time operation can be achieved. As a study case has been chosen control of a personnel's car velocity which presents necessary steps of a controller build-up including its coding and simulation.

Keywords: model predicitve control, car dynamics, ACADO, dSpace system

1. INTRODUCTION

A control of vehicle's dynamics is a complex task considering different phenomena acting onto the vehicle arising from dynamic nature of car parts; or outer disturbances caused by the road state, or weather conditions. To maintain the vehicle in a prescribed trajectory, or speed, the controller has to have an access to the engine and brake management, in a first case. Another condition of proper functionality is a quick controller able to react on a millisecond level, making it a typical example of a system with fast dynamics.

Such a controller then has to be equipped with a solver, able to solve a minimization problem within that short control period, as well.

In this work is utilized as a solver software package ACADO developed for the MPC controller design disposing of above mentioned features and intuitive coding.

2. VELOCITY CONTROL

2.1 Motivation

Velocity control and its derivatives (torque control, revolutions control) has a variety of applications, e.x. in electronic cruise control, in maintaining the constant idle speed of an engine, or in a holding of a constant power output of a cogeneration unit.

The cruise control maintains desired reference speed, so the driver can relax his foot from the gas pedal, when driving long distances on a sparsely populated highways; or to avoid speed limit violations. Engagement of a cruise control also results in a lower fuel consumption on non-hilly tracks. Disadvantages might be a delayed deceleration of a vehicle, when needed or an improper velocity regarding the traffic conditions or the road state.

Another field, where the tracking of a velocity reference is desired, is a fuel consumption testing. There exist normalized test cycles emitted by international authorities, obligatory for all car producers. They basically consist of predefined time-points, at which has to have a particular car prescribed velocity (Fig. 1). If the following of the mentioned reference velocity is not accurate, vehicle seems to have higher average fuel consumption, as it actually really has. This is a motivation, why to use electronic velocity control, as well.

⋆ The investigation reported in the paper was supported by Slovak Grant Agency APVV, project ID:LPP-0096-07. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. The supports are very gratefully appreciated.

There exist many test cycles, differing in the area of application, country of validity, etc., such as:

EUDC	- Extra Urban Driving Cycle, simulates the highway
ECE	- steady state cycle for heavy duty truck engines
NEDC	- New European Driving Cycle, since 2000; NEDC is ECE plus EUDC
15 MODE	- urban driving cycle, Japan
FTP 75	- a transient test cycle for cars and light duty trucks, US

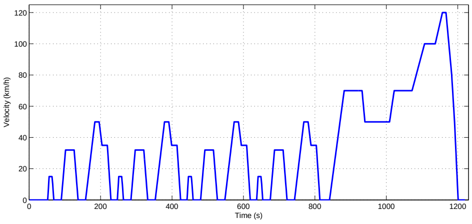


Fig. 1. NEDC test cycle

Detailed list of test cycles can be found at (tes (2010)).

2.2 System Description

In this study case is a task of a car's dynamics control simplified to its longitudinal direction considering the vehicle as a mass point, following a predefined velocity profile.

The considered vehicle-system consist of three, mutually coupled subsystems:

- the car's longitudinal dynamics (Hellström (2005)), including wind resistance, road slope, inertia effects of wheels and engine, transmission ratios and efficiency of a gearbox and the final drive, etc.

$$\dot{v} = \frac{r_w}{J_w + m r_w^2 + \eta_D i_D^2 \eta_G i_G^2 J_E} \big(\eta_D i_D \eta_G i_G T_e - k_b B_b - 1/2 c_w A_a \rho_{air} r_w v^2 - m g r_w (c_r \cos\theta + \sin\theta)\big) \quad (1)$$

- the dynamics of the air intake manifold at the engine (Saerens et al. (2009)) considering the sonic choking

$$\dot{(p_m)} = \frac{r T_m}{V_m} \frac{p_a}{\sqrt{r T_a}} A_{TV}(\alpha) c_d f_n \left(\frac{p_m}{p_a}\right) - \frac{\eta_v V_{tot}}{2 V_m} n p_m \quad (2)$$

- the throttle valve throttling the air sucked into the engine

$$\dot{x} = -43.55x + u \quad (3a)$$

$$\alpha = 27.06x + 0.38u \quad (3b)$$

where the symbols' meanings are:

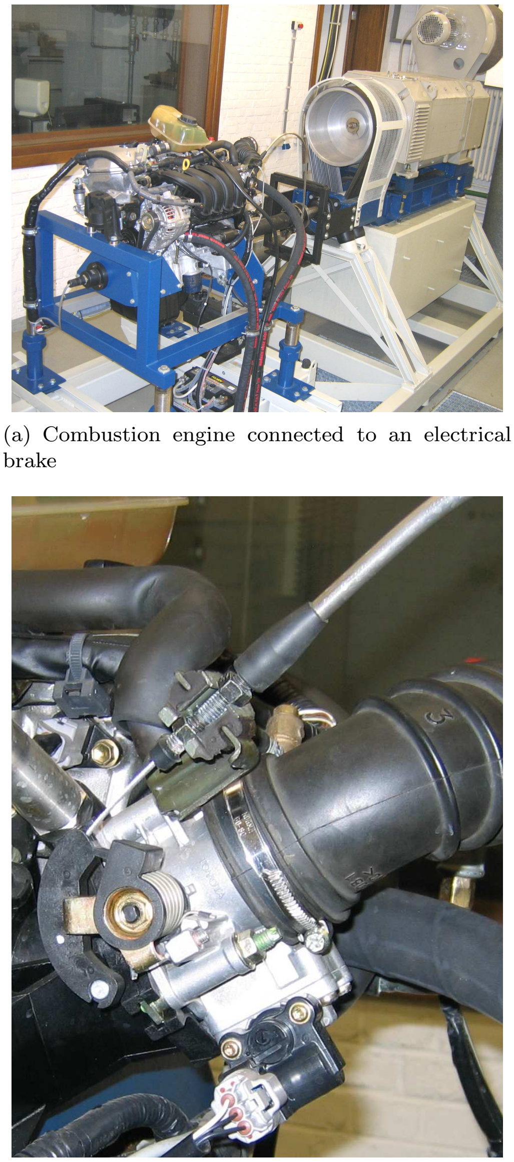
α	throttle valve plate angle
η_D	efficiency of final drive
η_G	efficiency of gearbox
η_v	volumetric efficiency
κ	isentropic exponent
ρ_{air}	air density
θ	road slope
A	frontal cross-sectional area
B_b	maximal brake force
c_d	throttle body discharge coefficient
c_r	rolling friction coefficient
c_w	aerodynamic coefficient
f_n	f. evaluating opened area of a throttle valve
g	Earth gravitational constant
i_D	reduction ratio of final drive (differential)
i_G	reduction ratio of the gearbox
J_e	engine inertia
J_w	wheel inertia
k_b	weighting brake parameter
m	mass of vehicle
m_f	fuel mass flow
n	engine revolutions
p_a	aerostatic pressure
p_m	manifold pressure
r	specific gas constant
r_w	wheel radius
T_a	outer air temperature
T_e	engine torque
T_m	air temperature in a manifold
u	control input
v	velocity
V_{tot}	total cylinder volume
V_m	manifold volume
x	identified system state

The input into the entire system is a control input u (voltage) scaled to $< 0 \dots 1 >$ forcing the throttle valve to change its position between $< 0 \dots \pi/2 >$ radians (also scaled to $< 0 \dots 1 >$) and so to let the air into the engine. The output of the system is a vehicle velocity v. The cost function $\mathcal{J}$ itself is of a simple form:

$$\mathcal{J} = \|v - v_{ref}(t)\|_2^2 \quad (4)$$

where: $v_{ref}(t)$ is a reference velocity of a vehicle, in the simulation set to a constant value.

Due to the technical restrictions, the equations of air intake dynamics and the throttle valve behavior are describing the real systems present in the laboratory, while the vehicle dynamics is simulated by an electric brake connected to the engine, independently controlled from *Matlab/Simulink* (Fig. 2).



(a) Combustion engine connected to an electrical brake

(b) Detail of a throttle valve body

Fig. 2. Electric motor connected to a throttle valve

3. ACADO

3.1 Software description

ACADO Toolkit (aca (2010)) is an open source software environment and algorithm collection for an automatic control and dynamic optimization. It provides a general framework for employing a great variety of algorithms for:

- direct optimal control
- model predictive control
- state and parameter estimation
- robust optimization

It is implemented as a self-contained C++ code and comes along with user-friendly Matlab interfaces. The object-oriented design allows convenient coupling of existing optimization packages and for an extension with user-written optimization routines.
It has been chosen (Table 1) for its speed crucial for the control of fast dynamic systems and other features listed above.

Table 1. Available optimization software sorted by a purpose

Application area	Software
Linear Programming	ILOG CPLEX, SoPlex, lp_solve, LINGO, linprog (MATLAB)
Quadratic Programming	MOSEC, quadprog (MATLAB), qpOASIS
Mixed-Integer Linear Programming	ILOG CPLEX, lp_solve
Mixed-Integer Quadratic Programming	TOMLAB (MATLAB)

3.2 Source code of a controller

The code of the designed controller will be based on all three differential equations and state representation of a car, as described in subsection 2.2; although the explanation of the code structure will be done considering an extra MPC controller taking into account the throttle valve subsystem only (Eq. 3).

Source code is accessible at the end of this paper.

Please note, that the structure of a controller for the full-system case is the same and follows listed logical parts:

(1) Definition of differential states, variables, control inputs, differential equations, constants...
(2) Definition of a cost function in a form:

$$\mathcal{J} = Q\|h - s\|_2^2 \tag{5}$$

where:

Q	is a weighting matrix
h	is a vector of states
s	is a vector of setpoints

(3) Definition of a optimal control problem with corresponding algorithm, constraints, initial values, settings...
(4) Plotting of results
(5) MPC initialization, if the code is ready to be compiled

4. SIMULATION RESULTS

The simulation has been done using the ACADO toolkit, having as a target acceleration of a vehicle from 4 m/s to 10 m/s, while respecting the physical constraints of the system (Fig. 3).

It shows how the system states and quantities are changing, particularly the control input, desired velocity, engine revolutions, instantaneous fuel and air inflow, etc.

5. CONCLUSION

The following paper introduced a field of the vehicle's longitudinal dynamics modeling and a velocity control

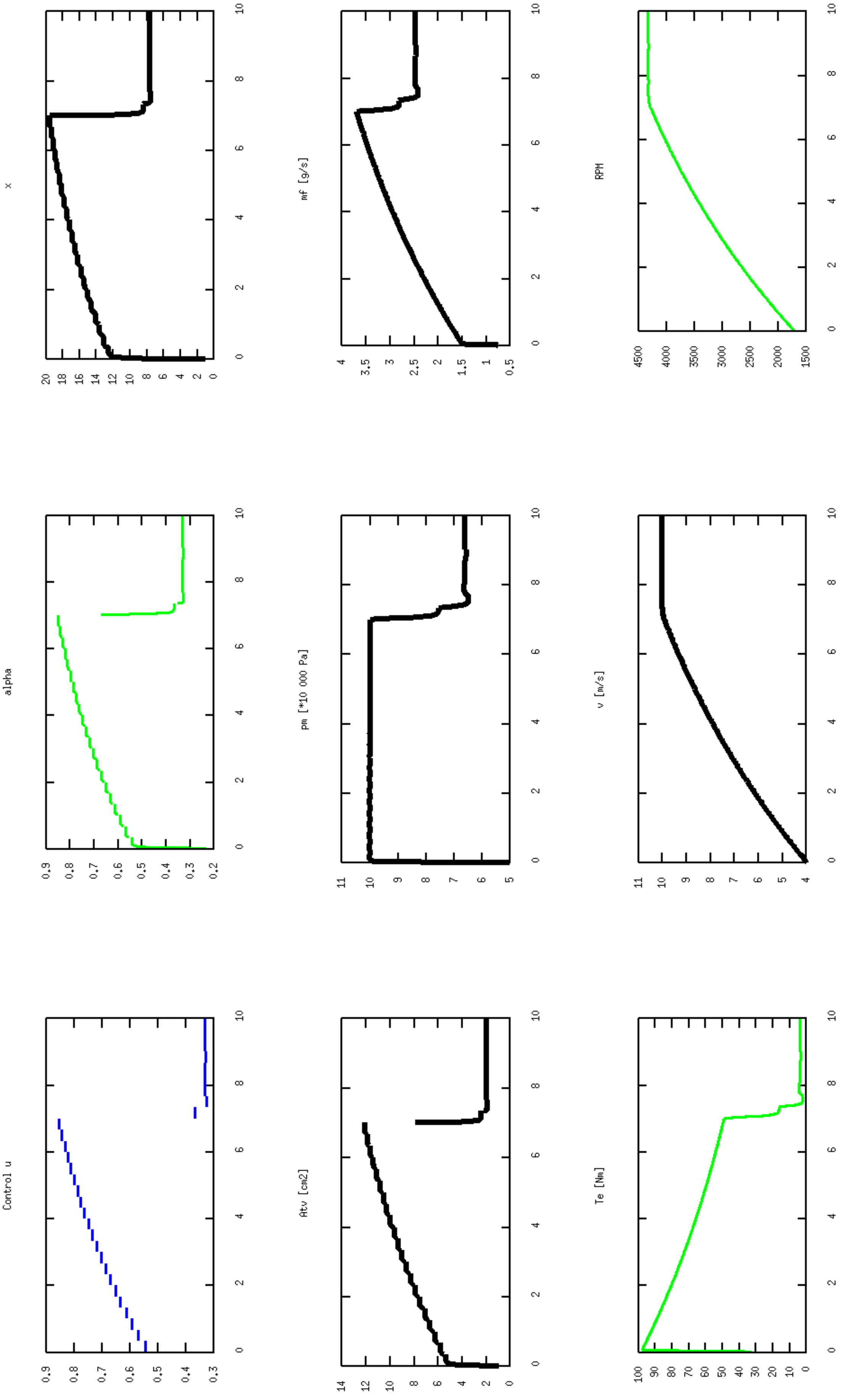


Fig. 3. ACADO simulation of a car MPC (x-axis: Time (s))

using a MPC (model predictive control) controller by using the ACADO toolkit as a solver of a minimization problem within the controller.

A future augmentation of this system shall be a real-time run on a dSpace system, where the electric brake connected to the engine shall simulate the dynamics of a car. Another improvement could be an additional utilization of the brake force described in this case by a constant member $k_b B_b$ in (Eq. 1) and a decentralized multi loop control of the whole plant.

REFERENCES

(2010). Emission test cycles. `http://www.dieselnet.com/standards/cycles`.

(2010). Key features of acado. `http://www.acadotoolkit.org`.

Hellström, E. (2005). *Explicit use of road topography for model predictive cruise control in heavy trucks.* Master's thesis, Dept. of Electrical Engineering, Linköpings universitet.

Ljung, L. (1999). *System identification. Theory for the user.* Prentice Hall PTR.

Saerens, B., Vandersteen, J., Persoons, T., Swevers, J., Diehl, M., and den Bulck, E.V. (2009). Minimization of the fuel consumption of a gasoline engine using dynamic optimization. *Applied energy*, 86, 1582–1588.

— **Beginning of a sample code** —

```
#include <acado_toolkit.hpp>
#include <gnuplot/acado2gnuplot.hpp>

int main( ){

USING_NAMESPACE_ACADO

DifferentialState       x;      // initial definition of differential states,
IntermediateState       alpha;  // variables,
Control                 u;      // control inputs,
DifferentialEquation    f;      // differential equations, constants...

Matrix A(1,1);
A.setZero();

Matrix B(1,1);
B.setZero();

Matrix C(1,1);
C.setZero();

Matrix D(1,1);
D.setZero();

double t_start  = 0.0;
double t_end    = 1.0;
double alphaD   = 1.0;          // alpha Desired - static reference

A(0,0)  = -43.5489;
B(0,0)  =   1.0;
C(0,0)  =  27.0625;
D(0,0)  =   0.3775;

f << dot(x) == A(0,0)*x +   b(0)*u;
     alpha  =  C(0,0)*x + D(0,0)*u;

Function h;                     // h,Q,r are arguments of a cost function
h << alpha;
h << u;

Matrix Q(2,2);
Q.setIdentity();
Q(0,0) = 10.0;
Q(1,1) = 0.01;

Vector r(2);
r(0) = alphaD;
r(1) = 0.0;

OCP ocp( t_start, t_end, 10 );          // definition of Optimal Control Problem (OCP)
ocp.minimizeLSQ( Q,h,r );
ocp.subjectTo( f );

ocp.subjectTo( 0.0 <=   u   <= 1.0  );      // constraints
ocp.subjectTo( 0.0 <= alpha <= 1.0  );
//ocp.subjectTo( AT_START, u==0       );

GnuplotWindow window;                       // results plotting
window.addSubplot( x,          "State x" );
window.addSubplot( alpha, "Output alpha" );
window.addSubplot( u,        "Control u" );

RealTimeAlgorithm algorithm(ocp);           // OCP algorithm and settings
algorithm << window;
algorithm.set( "MaxNumIterations",3 );
algorithm.set( "HessianApproximation", GAUSS_NEWTON );

Vector x0(1);                               //initial conditions of states
x0(0) = 0.0;

algorithm.solve( x0 );

// MPC controller, uncommented during compilation
// DynamicFeedbackLaw feedbackLaw( algorithm,0.05 );
// Controller controller;
// controller.setFeedbackLaw( feedbackLaw );
// controller.init( x0 );
// controller.step( 0.0,x0 );
return 0;
}
```

— **End of code** —

Comments – Remarks

Model Predictive Controller of the Air/Fuel Ratio in ACADO toolkit ⋆

**Matúš Kopačka, Peter Šimončič, Martin Lauko,
Tomáš Polóni, Boris Rohaľ-Ilkiv**

*Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail : matus.kopacka@stuba.sk*

Abstract: The following paper describes a first approach of the control of air/fuel ratio (AFR) of a spark ignition engine utilizing a model predictive controller based on the multi-model approach, prepared for introduction of constraints. The multi-model approach employs the autoregressive model (ARX) network, using the weighting of local models, coming from the sugeno-type fuzzy logic. The weighted ARX models are identified in the particular working points and are creating a global engine model, covering its nonlinearity. Expected improvement of an air/fuel mixture combusted in a cylinder is mostly awaited in the transient working regimes of an engine. In these regimes, the traditional control approach looses its quality, compared to steady state working regimes of an engine. This leads to higher fuel consumption and level of emissions from an engine. Another improvement of the control is awaited using a predictive controller with constraints. As such a controller is not possible to compute the control analytically anymore, the ACADO toolkit has been introduced, as a solver of the minimization task. The effort is focused on a controller preparation, running in the final stage real-time on a dSpace unit, which replaced the engine's original electronic control unit (ECU).

Keywords: Model predictive control, air/fuel ratio, spark ignition engine, ACADO toolkit, constraints

1. INTRODUCTION

Incorrect ratio of the air/fuel mixture in a cylinder of a spark ignition engine may result in the poor engine power, ineffective functionality of the catalytic converter resulting in higher level of emissions polluting the environment and in the extreme case this can lead to the engine stoppage. Due to this reason it is crucial to keep the air/fuel ratio (AFR) at the stoichiometric level, which means, that both, the air and the fuel are completely combusted. Due to above mentioned reasons and all the time tightening emission standards the car producers are improving the control of the air/fuel ratio.

Traditional control of air/fuel ratio is based on a feedforward control using predefined tables determining how much fuel has to be injected into a cylinder, based on the information from the mass air flow meter. This fuel amount is subsequently corrected using the information from the lambda probe, so the stoichiometric mixture can be reached. Due to a lambda probe position (at the engine exhaust) a delay arises, causing an improper feedback correction at the unstable engine regimes, like acceleration, or deceleration. On the other side, this kind of control guarantees stability and robustness at all conditions and therefore is still preferred by car producers, despite its disadvantages in control.

The academic field have started to publish other kinds of air/fuel control, mostly model-based ones. The model-based approaches are bringing good quality of control, but are also more sensitive to the model precision and issues with stability and robustness appear. A survey through popular "mean value engine modeling" is described in Bengtsson et al. (2007). This analytical way of engine modeling is very clear, but requires exact knowledge of the system and the model error has to be taken into account explicitly. Other ways of a model acquisition are based on the experimental identification (black box modeling). Works of Zhai et al. (2010), Zhai and Yu (2009) and Hou (2007) are specialized in employment of neural networks, while Mao et al. (2009) uses for engine modeling CARIMA models.

In the engine control itself became popular fuzzy logic (Hou (2007)), neural network control (Arsie et al. (2008)) and model predictive control (MPC) approaches (Lorini et al. (2006) and Muske and Jones (2006)). General topics on an issue of stability and robustness in MPC can be found in Mayne et al. (2000), or in Zeman and Rohal-Ilkiv (2003).

⋆ The work has been supported by the Slovak Research and Development Agency under grant LPP-0075-09, LPP-0118-09 and LPP-0096-07. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

The method described in this paper is introduced in Polóni et al. (2007) and it is utilizing a model predictive controller using a multi-model approach. The multi-model approach is utilizing a weighted net of autoregressive models (ARX) as a global model. This method continues in the work of Kopačka et al. (2010) and this paper describes preparation of the controller for the introduction of constraints.
As a solver of the minimization problem is used the ACADO toolkit, increasing the functionality of the above mentioned controller, allowing to implement constraints and other features.

2. AIR/FUEL MIXTURE

The model of the air/fuel ratio dynamics λ of a spark ignition engine is based on the mixture, defined as mass ratio of the air and fuel in a time step k. Due to the fact, that the air mass flow is measured as an absolute value, it was necessary to integrate this amount during the particular time and express the air and fuel quantity as relative mass densities in $grams/cylinder$. Hence, the air/fuel ratio is defined, as:

$$\lambda(k) = \frac{m_a(k)}{L_{th} m_f(k)} \tag{1}$$

where $m_a(k)$ and $m_f(k)$ are relative mass amounts of air and fuel in a cylinder and $L_{th} \approx 14.64$ is the theoretical amount of air necessary for the ideal combustion of a unit amount of fuel.
Considering the $\lambda(k)$ modeling, the engine has been divided into two subsystems with independent inputs, namely into:

air path with the air throttle position
as the disturbance input, and
fuel path with the input of fuel injector opening time.

Another disturbance-like acting quantity were engine revolutions, implicitly included in the engine model, particularly for each working point. The output ratio of both paths is the value of λ.

3. THE ANALYTICAL CONTROLLER

The strategy of an "exceeding oxygen amount" control using a predictive controller is based on a prediction of a controlled quantity λ and subsequent minimization of a chosen cost function on the horizon N_p expressed in a standard quadratic form. The value of λ is predicted by utilization of partially linear models of the air and fuel path. Through the independent air path model the proper amount of fuel is predicted and enters the cost function J. Hence, the target of the cost function minimization is to determine such a control law, that the measured system output λ is stoichiometric. The second modeled subsystem, the fuel-path, is an explicit component of the objective function where the amount of the fuel is the function of optimized control action (Polóni et al. (2008)).

3.1 Predictive model

The applied control strategy is based on the knowledge of the internal model (IM) of the air path, predicting the change of air flow through the engine and consequently, determining the set of desired values in the objective function on the control horizon. In this case we will consider the state space (SS) formulation of the system, therefore it is necessary to express linear local ARX models in parameter varying realigned SS model:

$$\begin{aligned} x_{(a,f)}(k+1) &= A_{(a,f)}(\phi)x_{(a,f)}(k) + B_{(a,f)}(\phi)u_{(a,f)}(k) \\ m_{s,(a,f)}(k) &= C_{(a,f)}x_{(a,f)}(k) \end{aligned} \tag{2}$$

The weighted parameters of multi-ARX models are displayed in matrices $A_{a,f}$ and $B_{a,f}$ for each subsystem. This is a non-minimal SS representation whose advantage is, that no state observer is needed. The "fuel pulse width control" is tracking the air mass changing on a prediction horizon from IM of the air path, by changing the amount of injected fuel mass. Due to tracking offset elimination, the SS model of the fuel path (2) (index f), with its state space vector x_f, is written in augmented SS model form to incorporate the integral action

$$\tilde{x}_f(k+1) = \tilde{A}_f(\phi)\tilde{x}_f(k) + \tilde{B}_f(\phi)\Delta u_f(k) \tag{3}$$

in a matrix form:

$$\begin{bmatrix} x_f(k+1) \\ u_f(k) \end{bmatrix} = \begin{bmatrix} A_f(\phi) & B_f(\phi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_f(k) \\ u_f(k-1) \end{bmatrix} + \\ + \begin{bmatrix} B_f(\phi) \\ 1 \end{bmatrix} \Delta u_f(k)$$

$$m_{s,f}(k) = \tilde{C}_f \tilde{x}_f(k) \tag{4}$$

The prediction of the air mass ($\underrightarrow{m}_a$) on the prediction horizon (N) is solely dependent on the throttle position ($\underrightarrow{t}_r$) and is computed, as:

$$\underrightarrow{m}_a(k) = \Gamma_a(\phi)x_a(k) + \Omega_a(\phi)\underrightarrow{t}_r(k-1) \tag{5}$$

where the x_a denotes the state space vector of the air path and $\Gamma_a(\phi)$ and $\Omega_a(\phi)$ are the prediction matrices.

Due to the unprecise modeling (IM strategy) a biased predictions of the air mass and consequently "biased fuel mass prediction" might occur. This error can be compensated by the term $L[\hat{m}_f(k) - m_{s,f}(k)]$ in the fuel mass prediction equation ($\underrightarrow{m}_f$)

$$\begin{aligned} \underrightarrow{m}_f(k) &= \Gamma_f(\phi)\tilde{x}_f(k) + \Omega_f(\phi)\Delta\underrightarrow{u}_f(k-1) + \\ &+ L[\hat{m}_f(k) - m_{s,f}(k)] \end{aligned} \tag{6}$$

The matrices Γ_a, Γ_f, Ω_a, Ω_f are computed from (2) (Maciejowski (2000)). Since there is only $\lambda(k)$ measurable in (1), the value of $m_a(k)$ needs to be substituted using IM of the air-path, then

$$\hat{m}_f(k) = \frac{1}{L_{th}} \frac{m_{s,a}(k)}{\lambda(k)} \tag{7}$$

The estimate $\hat{m}_f(k)$ is used to compensate for possible bias errors of predicted $\underrightarrow{m}_f(k)$ in (6).

3.2 Cost function

The main part of a cost function is a sum of deviations of predicted outputs from a set point, i.e. a weight of a future control deviations. Its another elements are a penalization of control increments r; and a p penalizing a deviation between a predicted and desired end state.
To eliminate a the steady state control error, the criterion (8) is defined through the control increments. That guarantees in a steady state, that $J_\lambda = 0$ also for the system with no integration properties, what is the case of the fuel path.

$$J_\lambda = \left\| \frac{\underrightarrow{m}_a(k)}{L_{th}} - \underrightarrow{m}_f(k) \right\|_2^2 + r\|\Delta \underrightarrow{u}_f(k-1)\|_2^2 \\ +p\|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2 \qquad (8)$$

The chosen MPC approach utilizes the state space representation and its structure uses a control deviation for the correction of the prediction.
Due to a disturbance $d(k)$, the steady state values of u and x have to be adapted so, that the assumption $J = 0$ is valid. This problem solves an explicit inclusion of the disturbance into the model.

The fuel injectors are controlled by a fuel pulse width, what is at the same time the control u_f. The optimal injection time can be computed by minimization of a cost function (8), which has after expansion by the fuel path prediction equation $\tilde{x}_f(k+1) = \Gamma_f(\phi)\tilde{x}_f(k) + \Omega_f(\phi)\Delta\underrightarrow{u}_f(k-1)$ a form of:

$$J_\lambda = \\ \left\| \frac{\underrightarrow{m}_a}{L_{th}} - \Gamma_f\tilde{x}_f(k) + \Omega_f\Delta\underrightarrow{u}_f(k-1) + L[\hat{m}_f(k) - m_{s,f}(k)] \right\|_2^2 \\ +r\left\|\Delta\underrightarrow{u}_f(k-1)\right\|_2^2 + p\|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2 \qquad (9)$$

An analytical solution of $\frac{dJ_\lambda}{\Delta\underrightarrow{u}} = 0$ of (9) without constraints leads to a definition determining the change of fuel injector's opening time on a chosen control horizon, as:

$$\Delta u = \left(\Omega^T\Omega + Ir + p\Omega_{xN}^T\Omega_{xN}\right)^{-1} \\ \cdot\left[\Omega^T[w(k) - \Gamma\tilde{x}(k) - L(y(k) - y_s(k))] \right. \\ \left. -p\Omega_{xN}^T A^N\tilde{x}(k) + p\Omega_{xN}^T\tilde{x}_{f,r}(N)\right] \qquad (10)$$

Hence, the absolute value of the control action in a step k is given by a sum of a newly computed increment in a control and an absolute value of the control in a step $(k-1)$:

$$u_f(k) = u_f(k-1) + \Delta u_f(k) \qquad (11)$$

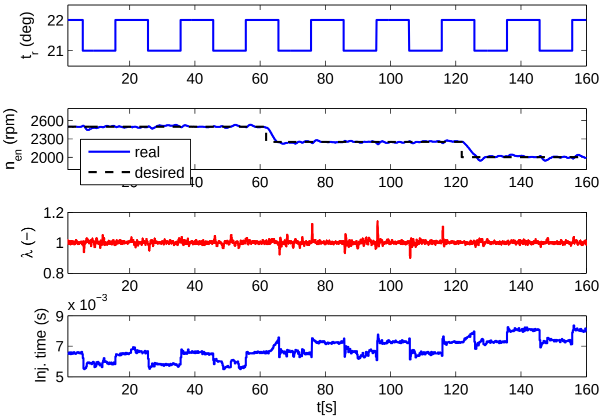


Fig. 1. Results of a real-time control of the SI engine

3.3 Real-time Control Results

The primary target of a control was to hold the air/fuel ratio at a stoichiometric value ($\lambda = 1$), in the worst case to keep the mixture ignitable ($0.7 \leq \lambda \leq 1.2$).
During the experiment, the change in throttle valve opening, between 21 and 22 degrees (Fig. 1, variable t_r) and the change of engine revolutions (Fig. 1, variable n_{en}), has been performed, at the same time. These changes simulate varying working regimes of an engine in a daily traffic. Changes in t_r and n_{en} quantities are determining the engine load; and at the same time they are ensuring, that the engine passes through several working points during its operation. The engine revolutions are not included among explicit variables of local models, but they build together with a delayed throttle valve position a vector of a working point $\phi(k) = |t_r(k)\ \ n_{en}(k)|$.
Looking at the results (Fig. 1) one can say, that the quality of control is sufficient (Fig. 1, variable λ), with exceptional acceptable overshoots in both directions. These overshoots of the controlled variable λ have been caused by smaller model precision, due to its distance from the working point, at which the system identification has been performed. This effect is caused by the approximation of a particular model from the other working points' models.
The corresponding control action computed by the controller is shown in (Fig. 1, variable *Inj.time*).

The initial engine warm-up (to 80 °C) eliminated model-plant mismatch caused by temperature dependent behavior of the engine.

The control has been performed by choosing the penalization $r = 0.1$. Utilizing the member $p\|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2$ of a cost function by setting $p = 2.0$ allowed us to shorten the control horizon to $N_p = 20$ from original $N_p = 30$ what significantly unloaded the computational unit and stabilized the controlled output of the engine on this shortened horizon, as well. The best control has been achieved in the neighborhood of working points, what is logically connected to the most precise engine model at those points. In other working points the control is still good enough, with small deviations from the stoichiometric mixture.

Considering the preliminary results from the real-time experiments at the engine, it can be concluded, that the idea of the AFR model predictive control based on local ARX models is suitable and applicable for the SI engine control. Proposed design of a predictive controller offers easy tuning possibilities and extension of the global engine model to other working regimes of the engine. On the other hand, its analytical solution doesn't allow implementation of constraints, until a numerical solver is employed.

4. CONSTRAINED PROBLEM

The combustion engine, as any technical system in a real world has its limitations, its constraints.
These constraints can have a nature of:

- physical constraints (dimensions, max. allowed revolutions, max. amount of fuel injected per cycle, etc.)
- process constraints (ignitability constraints in the combustion process, ideal air/fuel ratio in the cylinder, etc.)

and it is a smart move, to include the relevant constraints of the problem to the controller, so that the ideal control, respecting the constraints, can be computed. Introduction of constraints at the same time excludes the analytical computation of the optimal control and so the numerical solver of the minimization task has to be used.
For our purposes has been chosen the *ACADO toolkit* software package, meeting the specifications of a free and powerful tool with user friendly interface. More information on the software package can be found in Houska et al. (2010). A model predictive controller setup is illustrated in acadoweb (2010).

Problem Statement

The mathematical formulation of a problem, compared to the work of Kopačka et al. (2010) has been changed slightly. The syntax has been changed from the *Matlab/Simulink* block scheme to a more flexible and easily extendible *C++* code. The general structure utilizing the weighted ARX models as a system model stayed in tact, but the cost function formulation has been changed.
In the cost function does not appear the member considering the final state penalization, as in the original equation (8), anymore, so the final form of the cost function in ACADO is defined, as:

$$J_\lambda = \left\| \frac{\underrightarrow{m}_a(k)}{L_{th}} - \underrightarrow{m}_f(k) \right\|_2^2 + r\|\Delta \underrightarrow{u}_f(k-1)\|_2^2 \qquad (12)$$

Due to a simulation stage of the controller, also the member eliminating the model-plant mismatch $L(y(k) - y_s(k))$has been left out.

5. OUTLOOK

Nowadays is the new controller showing the same functionality as the original one running with the penalization of the terminal state switched off (Fig. 2). The fuel mass output m_f of the fuel path and the applied control du are looking quantitatively very similar. The almost identical results of the air/fuel ratio λ in the steady state region also prove its correct functionality. This facts build a good base for further improvements.

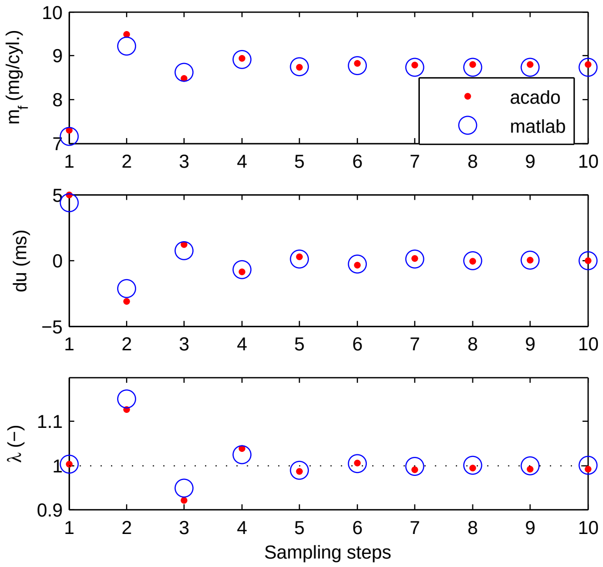


Fig. 2. Comparison of simulations in Simulink and ACADO (Np=10, sampling = 0.2s)

The implementation of the constraints into the controller has to be done at first, namely the constraints on the output value λ and on the change of the control du. The influence of these constraints on the control quality and the minimal length of the necessary prediction horizon has to be investigated.
Another step in development is a controller with a guaranteed stability based on the terminal set method.
All of the improvements have to be tested real-time on the existing motor bench with a VW Polo 1390cm^3 engine, at which the original electronic control unit (ECU) has been replaced by a dSpace system.

ACKNOWLEDGEMENTS

I would like to thank to Moritz Diehl and Hans Joachim Ferreau from the Catholic University in Leuven, Belgium, for their help and support by the implementation of the controller into the ACADO toolkit software.

REFERENCES

acadoweb (2010). Key features of ACADO. `http://www.acadotoolkit.org`.

Arsie, I., Iorio, S.D., Noschese, G., Pianese, C., and Sorrentino, M. (2008). Optimal air-fuel ratio. *dSpace Magazine*, (1), 20–23.

Bengtsson, J., Strandh, P., Johansson, R., Tunestal, P., and Johansson, B. (2007). Hybrid modeling of homogenous charge compression ignition (HCCI) engine dynamics -a survey. *International journal of control*, 80(11), 1814–1847.

Hou, Z. (2007). Air fuel ratio control for gasoline engine using neural network multi-step predictive model. 3rd international conference on intelligent computing, Qingdao, China.

Houska, B., Ferreau, H.J., and Diehl, M. (2010). Acado toolkit - an open-source framework for automatic control and dynamic optimization. *Optimal Control Methods and Application.* (accepted).

Kopačka, M., Šimončič, P., Csambál, J., Honek, M., Wojnar, S., Polóni, T., and Rohaľ-Ilkiv, B. (2010). Real-time air/fuel ratio model predictive control of a spark ignition engine. To appear.

Lorini, G., Miotti, A., and Scattolini, R. (2006). Modeling, simulation and predictive control of a spark ignition engine. In Predimot (ed.), *Predictive control of combustion engines*, 39–55. TRAUNER Druck GmbH & CoKG.

Maciejowski, J.M. (2000). *Predictive control with constraints.* University of Cambridge.

Mao, X., Wang, D., Xiao, W., Liu, Z., Wang, J., and Tang, H. (2009). Lean limit and emissions improvement for a spark-ignited natural gas engine using a generalized predictive control (GPC)-based air/fuel ratio controller. *Energy & Fuels*, (23), 6026–6032.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, (36), 789–814.

Muske, K.R. and Jones, J.C.P. (2006). A model-based SI engine air fuel ratio controller. American Control Conference, Minneapolis, USA.

Polóni, T., Johansen, T.A., and Rohaľ-Ilkiv, B. (2008). Identification and modeling of air-fuel ratio dynamics of a gasoline combustion engine with weighted arx model network. *Transaction of the ASME (Journal of Dynamic Systems, Measurement, and Control)*, 130(6). 061009.

Polóni, T., Rohaľ-Ilkiv, B., and Johansen, T.A. (2007). Multiple ARX model-based air-fuel ratio predictive control for SI engines. In *IFAC Workshop on advanced fuzzy and neural control*. Valenciennes, France. Conference paper MO5-3.

Zeman, J. and Rohal-Ilkiv, B. (2003). Robust minmax model predictive control of linear systems with constraints. 930 – 935. IEEE International Conference on Industrial Technology.

Zhai, Y.J., Ding-WenYu, Hong-YuGuo, and D.L.Yu (2010). Robust air/fuel ratio control with adaptive DRNN model and AD tuning. *Engineering Applications of Artificial Intelligence*, (23), 283–289.

Zhai, Y.J. and Yu, D.L. (2009). Neural network model-based automotive engine air/fuel ratio control and robustness evaluation. *Engineering Applications of Artificial Intelligence*, (22), 171–180.

Appendix A. SOURCE CODE OF THE CONTROLLER (SHORTENED)

A.1 Definition of variables and Optimal Control Problem Statement

```
int main( ){

const double steplength =     0.2;
const double Lth          =   14.67; //
    steichiometric coefficient
int            Np          =     10; //prediction
    horizon

DifferentialState    xf(13,1); // augmented state
    of the fuel path
Matrix              Aa(12,12),Ba(12,1), Ca(1,12);
      //A,B,C matrices of the air path
Matrix              Af(13,13),Bf(13,1); //A,B
    matrices of the fuel path
Control                    du;

update_model_parameters(Aa,Ba,Af,Bf,1800,20);
Ca.setZero();
Ca(0,0) = 1.0;

Aa.printToFile("Aa_ACADO.mat","Aa_ACADO",
    PS_MATLAB_BINARY); // print matrices to
    matlab
Ba.printToFile("Ba_ACADO.mat","Ba_ACADO",
    PS_MATLAB_BINARY);
Af.printToFile("Af_ACADO.mat","Af_ACADO",
    PS_MATLAB_BINARY);
Bf.printToFile("Bf_ACADO.mat","Bf_ACADO",
    PS_MATLAB_BINARY);

const double t_start = 0.0;
const double t_end   = Np*steplength;

DiscretizedDifferentialEquation   f(steplength) ;
f << next(xf) == Af*xf + Bf*du; //mf = xf(0);

DynamicSystem fuelpath (f);

Process fuelpath_simulator (fuelpath,
    INT_DISCRETE);

Vector xa0( 12 );            // initial values of
    xa
xa0(0) = 96.6;        xa0(1) = 96.6;        xa0(2) =
    96.6;       xa0(3) = 96.6;
xa0(4) = 96.6;        xa0(5) = 20.0;        xa0(6) =
    20.0;       xa0(7) = 20.0;
xa0(8) = 20.0;        xa0(9) = 20.0;        xa0(10)=
    20.0;       xa0(11)=  1.0;

Vector map(Np+1);    //() int in the bracket has
    to be identical with the Np in the function
    call (under)
map_predictionANALYTICAL(map, Aa, Ba, Ca, xa0, Np
    +1);

Function J;
J << xf(0);
J << du;

Matrix S(2,2);
S.setIdentity();
S(0,0) = 1000.0;
S(1,1) = 1.0;

// DEFINE AN OPTIMAL CONTROL PROBLEM:
VariablesGrid r( 2,t_start,t_end, Np+1 ); //
    defines reference vector for different time
    points
for( int i=0; i<(Np+1); i++ )
{
r( i,0 ) = map(i)/Lth;
r( i,1 ) = 0.0;
}
OCP ocp( t_start, t_end, Np );
ocp.minimizeLSQ( S, J, r );
ocp.subjectTo( f );

// OptimizationAlgorithm algorithm(ocp);
RealTimeAlgorithm algorithm (ocp, steplength);
```

```
algorithm.set( KKT_TOLERANCE, 1e-6 );
```

A.2 Looping of the OCP Problem to finish the MPC controller

```
// SETUP CONTROLLER AND PERFORM A STEP:
VariablesGrid StatTrajectory (2, t_start ,t_end,
    Np+1);
for (int i=0; i<Np+1; ++i)
{
StatTrajectory(i,0) = map(i)/Lth;
StatTrajectory(i,1) = 0.0;
}
StatTrajectory.print("StatTrajectory");

StaticReferenceTrajectory ST (StatTrajectory);

Controller controller( algorithm, ST );

Vector xf0(13);          //initial conditions of "
    xf" state vector
xf0(0) =  6.56;      xf0(1) =  6.56;      xf0(2) =
     6.56;     xf0(3) =  6.56;
xf0(4) =  6.56;      xf0(5) =  4.98;      xf0(6) =
     4.98;     xf0(7) =  4.98;
xf0(8) =  4.98;      xf0(9) =  4.98;      xf0(10)=
     4.98;     xf0(11)=  1.0;
xf0(12)=  4.98;

Vector uCon;
VariablesGrid ySim;

controller.init( 0.0, xf0 );
controller.step( 0.0, xf0 );
controller.getU( uCon );

fuelpath_simulator.init( t_start, xf0, uCon );
fuelpath_simulator.getY( ySim );

double sim_Tstart, sim_Tend;
sim_Tstart = 0.0;
sim_Tend   = 2.0;
double currentTime = sim_Tstart;
int nSteps = 0;

Vector Sim_mf (round((sim_Tend - sim_Tstart) /
    steplength));
Sim_mf.setAll(3333);

Vector Sim_du (round((sim_Tend - sim_Tstart) /
    steplength));
Sim_du.setAll(3333);

Vector mapi(Np+1);

Vector Lambda (round((sim_Tend - sim_Tstart) /
    steplength));
Lambda.setAll(3333);

while ( currentTime <= sim_Tend - steplength )
{
acadoPrintf( "\n*** Simulation Loop No. %d (
    starting at time %.3f) ***\n",nSteps,
    currentTime );
map_predictionANALYTICALidentical(mapi, Aa, Ba,
    Ca, xa0, Np+1);

VariablesGrid ma_ref (2, t_start, t_end, Np+1);
    // map into VariablesGrid
for (int i=0; i<Np+1; ++i)
{
ma_ref(i,0) = mapi(i)/Lth;
ma_ref(i,1) = 0.0;
}

double t = acadoGetTime();
controller.step( currentTime,ySim.getLastVector()
    , ma_ref );
controller.getU( uCon );
printf( "t = %e\n", acadoGetTime()-t );

fuelpath_simulator.step( currentTime,currentTime+
    steplength,uCon );
fuelpath_simulator.getY( ySim );

Sim_du(nSteps) = uCon(0);
Sim_mf(nSteps) = ySim.getLastVector()(0);
Lambda(nSteps) = (mapi(0)/Lth) / ySim.
    getLastVector()(1);

currentTime += steplength;
++nSteps;
}

Lambda.print("Lambda");
Lambda.printToFile("Lambda_ACADO.mat","
    Lambda_ACADO",PS_MATLAB_BINARY);
printf("\n");
Sim_du.print("Sim_du");
Sim_du.printToFile("Sim_du_ACADO.mat","
    Sim_du_ACADO",PS_MATLAB_BINARY);
Sim_mf.print("Sim_mf");          // simulated
    output of the system
Sim_mf.printToFile("Sim_mf_ACADO.mat","
    Sim_mf_ACADO",PS_MATLAB_BINARY);
return 0;
}
```

Comments – Remarks

Swing up and Balancing Control of Pendubot System ⋆

Martin Lauko, **Pavol Seman**, **Gergely Takács**,
Boris Rohal'-Ilkiv

Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail : martin.lauko@stuba.sk

Abstract: Swing up and balance control are two interesting control problems for the Pendubot system, which is one of the most typical experiment objects in the field of automatic control. It is a two-link under actuated robotic mechanism, presenting the classic inverted pendulum problem; well suited for control theory education as well as for research in the control of nonlinear mechatronic systems with fast dynamics.In this paper the swinging up and balancing control will be presented. The basic balancing control is based on a linear-quadratic (LQ) controller which were implemented and tested on the experimental setup in a real-time. Results received in verifying experiment are evaluated in the presented work. The swing up of the pendulum is based on a method of impulse build up of energy. A fourth order linearized time-invariant state-space system model is identified for the pendubot system. Carried out verification confirms, that the linear quadratic controller actuated system response is excellent. Inclusion of process constraints in a model predictive control (MPC) based balancing scheme could potentially offer numerous benefits as well.

Keywords: pendubot system, inverted pendulum, LQ control, pendulum swing up

1. INTRODUCTION

Stability control is one of the basic problems of control theory. This problem is studied on different models, which include the pendubot system. The pendubot is a two-link planar robot with an actuator on the first arm and no actuator on the second arm. It is underactuated and has fast nonlinear dynamics. The objective is to stabilize the system in one of its unstable equilibrium positions. That means keep the second arm, the free pendulum, upright and the first arm in a desired position. We began with a already build physical model of pendubot. We will show a brief derivation of the motion equations, using the Lagrange energy balance method, and the resulting state space model.

2. PENDUBOT MODEL

There are different ways how to realize the construction of pendubot. Our approach is based on the work of Mates Mates and Seman (2009).The physical model is a combined model of pendubot and furuta pendulum. We will in this paper discus only pendubot control. This design uses a servo motor to rotate the arm. This has the benefit that, we can directly control the torque applied to the arm and also brake at any given time. The brake is useful to apply braking moment in impulse control.

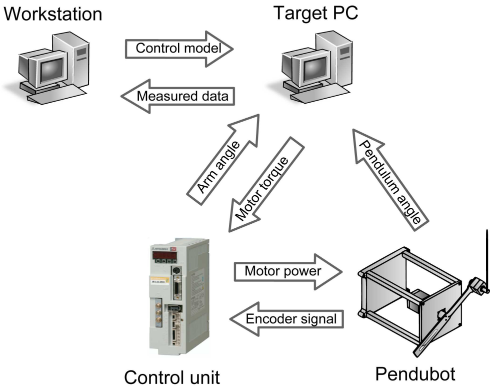


Fig. 1. Hardware connection scheme

2.1 Model hardware

The Figure 1 shows connections between hardware components. As can be seen the control model is designed on the Host PC. It is equipped with Matlab / Simulink software. The model is compiled and transfered to the Target PC, where it is run in a simplified environment in

⋆ The authors gratefully acknowledge the financial support granted by the Slovak Research and Development Agency under the contracts APVV-0280-06 and APVV-0160-07, and reviewers' comments. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is co-financed from the state budget of the Slovak Republic.

real time. Communication with hardware is done through the I/O card in Target PC. As can be seen on Figure 1 only the pendulum angle is read directly by the I/O card. Communication with the servo motor is done through the control unit. The control unit translates the encoder signal to arm angle and controls the motor power to achieve the desired torque.

The physical model is made of these components:

- Servo motor - Mitsubishi HC-KFS43
- Control unit - Mitsubishi MR-J2S-40A
- Incremental rotary sensor - OMRON E6B2
- I/O Card - Humusoft MF-624

2.2 Mathematical model

The pendubot is a second order dynamic system. We will use the energy balance method, Lagrange equation, to derive the equations of dynamics.

In this section mathematical equations use the following symbols:

- mr - Weight of arm
- l1 - Length of arm
- lg1 - Distance from center of gravity of the arm to the axis of rotation
- k1 - Friction coefficient in arm joint
- Ir - Mass moment of inertia of the arm
- mk - Weight of pendulum
- lg2 - Distance from center of gravity of the pendulum to the axis of rotation
- Ik - Mass moment of inertia of the pendulum
- k2 - Friction coefficient in pendulum joint
- φ - Pendulum angle
- θ - Arm angle

The basic form Lagrange equations is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = Q_i \tag{1}$$

where L, the Lagrangian, is the difference of kinetic and potentian energy. For the pendubot system these are simple to derive. Details, how to do this, are in different publications Fantoni and Lozano (2001). The end result from eq.1 are two equations 2 and 3.

$$\begin{aligned}\tau = \ddot{\varphi}\left(I_r + m_k l_1^2\right) + \ddot{\theta} m_k l_1 l_{g2}\cos(\varphi-\theta) + \\ m_k l_1 l_{g2}\sin(\varphi-\theta) + \left(m_r l_{g1} + m_k l_1\right) g\cos\varphi\end{aligned} \tag{2}$$

$$\begin{aligned}0 = \ddot{\varphi} m_k l_1 l_{g2}\cos(\varphi-\theta) + \ddot{\theta}(I_k + m_k l_{g2}^2) - \\ m_k l_1 l_{g2}\dot{\varphi}^2\sin(\varphi-\theta) + m_k g l_{g2}\cos\theta\end{aligned} \tag{3}$$

Linearizing these equations around chosen point and solving the result, we get the state space model $\dot{x} = Ax + Bu$. Where x is the state vector $x = [\varphi \quad \theta \quad \dot{\varphi} \quad \dot{\theta}]^T$. Choosing the point $\varphi = 0$ and $\theta = 0$ as the up-up position, the resulting state space matrices A and B for our model are:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 10,714 & -7,162 & -2,424 & 0 \\ -15,755 & 43,320 & 0 & -0,027 \end{bmatrix} \tag{4a}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 40,033 \\ -58,873 \end{bmatrix} \tag{4b}$$

3. CONTROL DESIGN

To solve the pendubot control problem, it is best to divide it into two parts. The balancing control that will keep the pendulum upright and the swing-up control that will bring the pendulum from complete stop, in the lower position, to the unstable equilibrium position. Switching between these two controls depends on the pendulum and arm position. This is necessary so that the balancing control is active only around the equilibrium point.

3.1 LQ control

The balancing control is realized using a LQ controller, which was derived using the state-space model from eq.4. The control law for LQ is $u = -Kx$ where x is the state vector and K is the LQ gain matrix. The gain was calculated by minimizing the cost function

$$J = \int (x'Qx + u'Ru)dt \tag{5}$$

where the weight matrices Q and R have been chosen as

$$Q = \begin{bmatrix} 19 & 0 & 0 & 0 \\ 0 & 26 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6a}$$

$$R = [3, 5] \tag{6b}$$

The resulting LQ control gain matrix is

$$K = [-1,54 - 7,11 - 0,69 - 1,22] \tag{7}$$

3.2 Swing-up control

There exist different approaches to inverted pendulum swing-up. Predominantly, they are based on adding energy to the pendulum system until it equals the potential energy of the pendulum in the upper position (Albahkali et al. (2009)). Our approach is based on the same idea. Adding of energy is done by swinging the arm around zero position. The arm swings are limited to help smoothen transition to balancing control. The limits should allow the arm to move as widely as possible, to allow more energy build up per swing. For the calculation of torque the the following equation was used:

$$u = sat_n \Big(k(E - E_0) sign(\dot{\theta} \cos\theta) \Big) \qquad (8)$$

where E is the current energy of the system and E_0 is the desired energy. The gain constant K determines speed at which energy is added to the system (Astrom and Furuta (1996)).

4. TEST RESULTS

The functionality of the designed control scheme was tested on our physical model. The measured results are in Fig. 4. The swing-up algorithm was active the first 5 seconds, after which it switched to balancing control, because the pendulum was close to the upper equilibrium position. In Fig. 2(a) can be seen the swinging motion of the pendulum and its stabilization. Speed of the pendulum is in Fig. 2(b). The position and speed of the arm can be seen in Fig. 2(c) and 2(d). The corresponding torque applied to the arm is seen in Fig. 2(e).

5. MPC CONTROL SIMULATION

This section presents the planned model based predictive control (MPC) of the pendubot system through simulations performed using the state-space model of the laboratory hardware. The preliminary considerations introduced here give a basis for further work on the pendubot system and also point out the weaknesses of saturated linear-quadratic control.

All simulations assume identical models for the pendubot system, including an initial condition equivalent to displacing the pendulum 5 degrees away from its upright position. The pendulum is assumed to be placed around its nominal upright position, while the swing-up portion of the control assignment is ignored here.

The constraints placed on the system are applied only to the torque requirement passed onto the stepper motor. The model input is in fact the force moment in N/m applied to the pendulum arm. The MPC model includes this constraint, while the simple LQ version is saturated to these bounds. System sampling is set to $Ts = 0.01$ seconds in all cases.

5.1 Unconstrained system

The MPC method considered in this paper is a dual-mode constrained controller with guaranteed stability (Mayne et al., 2000; Chen and Allgöver, 1998; Rossiter, 2003), providing the maximal admissible region of attraction and target set through high order polyhedral constraints.

Given a five degree initial condition, the MPC controller requires a prediction horizon $n_c = 10$ steps. This places the initial condition just on the edge of the admissible set for all three presented simulation cases.

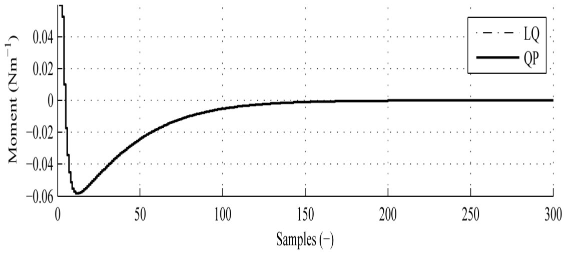


Fig. 3. Torque request to actuator

In the first simulation case, the constraints slightly exceed the expected moments provided to the stepper motor. Here a $0.06N/m$ constraint practically produces an unconstrained response from the system. The torque profile for this case is shown on Figure 3, where the responses produced by the MPC and saturated LQ controllers are identical[1]. Subject to the same input signal, the system behaves identically therefore the individual states are not shown here.

5.2 Constrained system

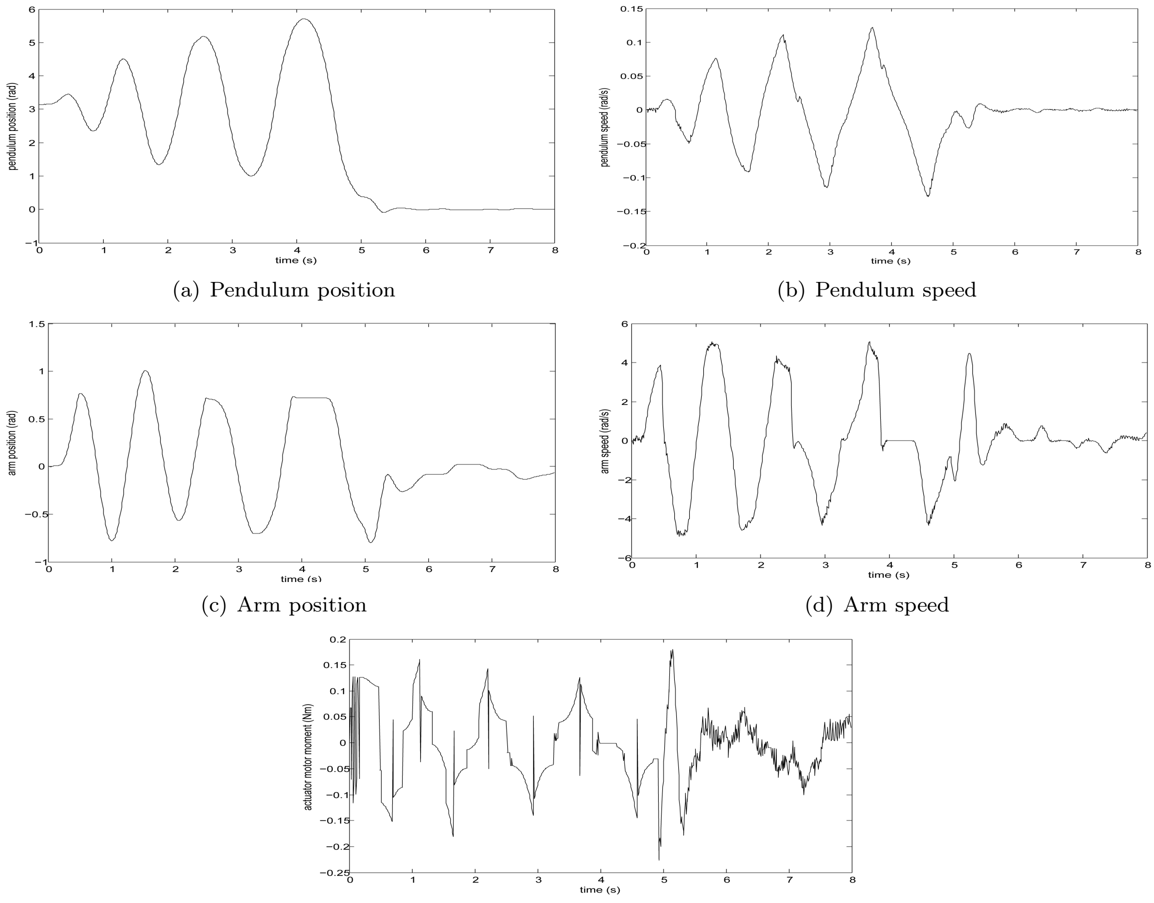
The following two simulation examples involve constraints placed on the torque requirement passed to the actuator. In the first case a stable response is produced, however the benefits of using an MPC controller over LQ are clearly demonstrated. The second simulation involves a more limitive constraint, rendering the saturated LQ controlled system entirely unstable.

Stable Figure 4 shows the individual pendulum states and the torque requirement, when the system is subject to a $\pm 0.05N/m$ moment constraint. As it is visible from the arm and pendulum position and velocity diagrams on Figure 4(a)-(d), the saturated LQ controller stabilizes the pendulum less efficiently. While the control course remains stable, the constrained MPC governed output settles the pendulum arm position much faster. The constraints and the controller outputs may be observed on Figure 4(e), where it is evident that the LQ controller leaves the torque input on its lower saturation limit for a longer period than its MPC counterpart.

Unstable The previous subsection dealt with a constraint control case, where it has been shown that an MPC controlled pendulum arm may bring benefits over simple saturated LQ control. However efficiency is not the only issue here: enforcing more stringent constraints on the stepper motor may render LQ control entirely unstable, while MPC can still function without the hazards of destabilizing the control system.

In this case the constraints were decreased slightly, to $\pm 0.0485N/m$, while the rest of the simulation settings remained the same. As it is clearly visible on the position, velocity and torque output of the system on Figure 5; the system became unstable under LQ control. For the case of saturated LQ control, Figures 5(a),(b) show the arm and pendulum positions increasing indefinitely - causing

[1] Because the two responses are in fact identical, only the MPC controlled torque profile is visible.



(a) Pendulum position

(b) Pendulum speed

(c) Arm position

(d) Arm speed

(e) Motor torque

Fig. 2. Measured results of pendubot swing-up and balancing LQ

the pendulum to fall from its unstable equilibrium position. Similarly Figures 5(c),(d) shows the LQ controlled velocities to grow uncontrollably.

However the MPC controlled responses still remain within reasonable bounds, preserving the stability of the feedback system.

5.3 Implementing the MPC controller

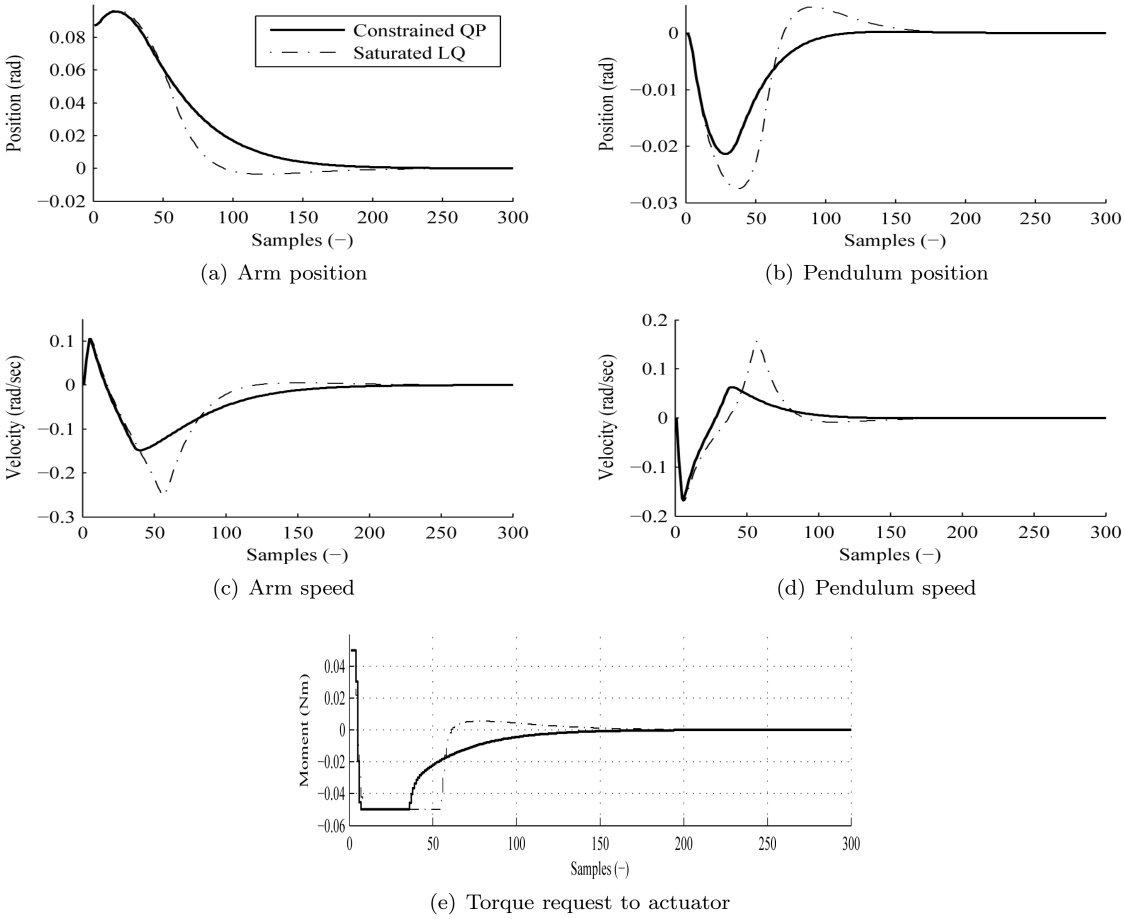
While the previous simulation cases point out numerous advantages of using an MPC controller with guaranteed stability on the pendubot, there are some issues with the practical implementation of such a system on the laboratory device in real time. The expected sampling period is not extremely short and higher order prediction models have been used before with sampling periods of $0.0002s$ (Wills et al., 2008), though without stability guarantees. The MPC control of an unstable equilibrium position requires stability guarantees, which places additional requirements on algorithm efficiency.

On-Line Quadratic Programming It is possible that a controller with a fourth order prediction model and a $n_c = 10$ steps prediction horizon is implementable without significant issues on a real-time rapid software prototyping system. However during simulation stages it has been noted, that if a maximal admissible reachable and target set is considered, the constraints have to be evaluated much further: requiring constraint checking horizons in the excess of 200 steps. Adding more complexity to the problem leaves and open question, whether this system can be implemented using quadratic programming solvers optimized for MPC usage such as presented in Ferreau (2006) and Ferreau et al. (2008).

The given system with the considered settings also caused various numerical problems; both at the stage of searching for the largest admissible set and during the simulation of on-line quadratic optimization. Linear programming used at the initialization stage exited with a warning; stating that the optimization process iteration tolerance limit has been exceeded[2]. At on-line optimization, the quadratic programming solver issued a warning about a non-symmetric optimization Hessian.

Multi-Parametric Programming The pendubot device is a mechatronic system with fast dynamics, as such it requires an efficient MPC implementation. A viable alternative to on-line quadratic programming optimization is the use of multi-parametric programming (MP) based

[2] The solver assumed throughout the simulation was "quadprog", default QP solver in the Matlab suite.



(a) Arm position

(b) Pendulum position

(c) Arm speed

(d) Pendulum speed

(e) Torque request to actuator

Fig. 4. Stable simulation, comparing LQ and MPC based actuator inputs. Arm and pendulum position in radians is visible on (a) and (b), while (c), (d) shows the corresponding arm and pendulum velocities. Figure (e) denotes the torque requirement passed onto the actuator, when its output is limited to ±$0.05 N/m$

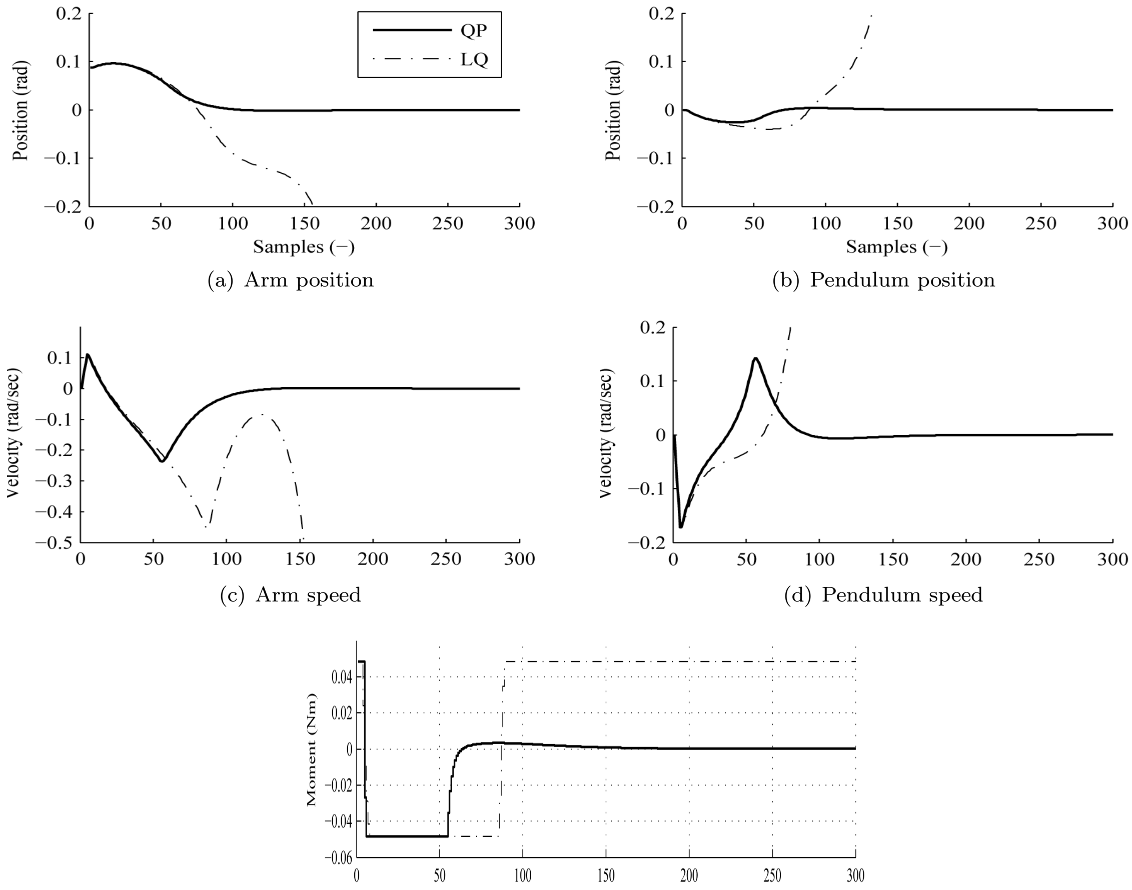
MPC. Assuming a piece-wise affine linear MPC problem, the controller pre-computes controller regions and associates them with polyhedral regions in state-space. This way the computational load is transferred to the off-line mode, while the regions and control laws corresponding to the current states are found directly in a very efficient manner (Kvasnica et al., 2006).

With this approach is very promising, the off-line computational load grows very rapidly when increasing prediction horizons. A multi-parametric controller has been evaluated for the given pendubot system, assuming the same settings as introduced for the QP controller. The explicit controller has been calculated using the Multi-Parametric Toolbox (Kvasnica et al., 2004).

By definition, the controller outputs are the same as the direct QP optimization results up to numerical precision, therefore these results are not indicated in the simulations presented in 5.1 and 5.2. However the resulting controller has been computed in more than 37 hours, using a generic personal computer conforming to current standards [3] . The explicit MPC controller is defined over 132927 region sover 4D, resulting an exported C language header over 97 Mbytes in file size. While this is certainly not prohibitive for personal computers, a digital signal processing board (DSP) or similar hardware with limited amount of RAM could not run such a large application. It remains an open question, whether the large number of regions could cause the search times to exceed sampling during the on-line control of the pendubot arm. While this is unlikely, we have to note that if a larger range of expected deviations from the upper unstable position is required also a larger region of attraction is necessary. This can be ensured by increasing the prediction horizon, which certainly will also exponentially increase the number of regions and controller size (Takács and Rohal'-Ilkiv, 2009).

It is also worth noting, that the multi-parametric controller computation procedure was not without numerical errors. During the lengthy computation time, the algorithm issued warnings involving the linear programming solver. Although the final controller passes the invariance check and behaves as expected during simulations, it is not certain whether these errors could produce erroneous controller outputs.

[3] AMD Athlon X2 DualCore 4400+ at 2.00GHz, 2.93GBytes of RAM.



(a) Arm position

(b) Pendulum position

(c) Arm speed

(d) Pendulum speed

(e) Torque request to actuator

Fig. 5. Unstable controller behavior, comparing LQ and MPC based actuator inputs. Arm and pendulum position in radians is visible on (a) and (b), while (c), (d) shows the corresponding arm and pendulum velocities. Figure (e) denotes the torque requirement passed onto the actuator, when its output is limited to ±0.0485N/m

6. CONCLUSION

Measurements are in correspondence with our expected results. The designed swing-up needed several swings to bring the pendulum to the upright position. This was caused by the limitations to movement of the arm and maximum allowed torque. The balancing LQ controller has oscillations around the equilibrium position, but they were negligibly small.

The simulations performed utilizing the MPC controller clearly show the drawbacks of saturated linear-quadratic control. In the absence of constraints, the LQ and MPC controllers provide identical outputs to the actuator. However a constrained torque request results in a sub-optimal LQ control course when compared to MPC, furthermore a more stringent torque boundary may result in the loss of stability. The preliminary controller implementation analysis presented in this paper points out several difficult aspects of applying MPC on the laboratory device in real time.

REFERENCES

Albahkali, T., Mukherjee, R., and Das, T. (2009). Swing-up control of the pendubot: An impulse–momentum approach. *IEEE Transactions on Robotics*, 25(4), 975–982.

Astrom, K., Block, D., and Spong, M. (2007). *The Reaction Wheel Pendulum.* Morgan Claypool Publishers.

Astrom, K.J. and Furuta, K. (1996). Swinging up a pendulum by energy control. In *IFAC 13th World Congress.* San Francisco, California.

Chen, H. and Allgöver, F. (1998). Ca quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34, 1205–1217.

Fantoni, I. and Lozano, R. (2001). *Non-linear Control for Underactuated Mechanical Systems.* Springer.

Ferreau, H. (2006). *An Online Active Set Strategy for Fast Solution of Parametric Quadratic Programs with Applications to Predictive Engine Control.* Master's thesis, University of Heidelberg.

Ferreau, H., Bock, H., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8), 816–830.

Havlena, V. and Stecha, J. (1996). *Moderni teorie rizeni*. Vydavatelstvo CVUT.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Online. Available: `http://control.ee.ethz.ch/`.

Kvasnica, M., Grieder, P., Baotic, M., and Christophersen, F.J. (2006). *Multi-Parametric Toolbox (MPT)*. Extended documentation.

Mates, M. and Seman, P. (2009). Combined model of pendubot and rotational inverted pendulum. In *IWCIT '09 : Proceedings of 8th International PhD Student's Workshop on Control and Information Technology*. Brno University of Technology, Brno.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789–814.

Rossiter, J.A. (2003). *Model-based Predictive Control: A Practical Approach*. Taylor and Francis (CRC Press).

Seman, P., Mates, M., Lauko, M., and Rohal-Ilkiv, B. (2010). Control of laboratory model of pendubot. In *Cybernetics and Informatics: International Conference SSKI SAV*. Vydavatelstvo STU, Vysna Boca.

Takács, G. and Rohal'-Ilkiv, B. (2009). MPC with guaranteed stability and constraint feasibility on flexible vibrating active structures: a comparative study. In *Proceedings of the eleventh IASTED interantional conference: Control Applications*, 278–285. Cambridge, United Kingdom.

Wills, A.G., Bates, D., Fleming, A.J., Ninness, B., and Moheimani, S.O.R. (2008). Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1), 3–12.

Comments – Remarks

Explicit predictive control of a piezoelectric smart structure [⋆]

Tomáš Polóni [*], **Gergely Takacs** [*], **Michal Kvasnica** [**], **Boris Rohaľ-Ilkiv** [*]

[*] *Institute of Automation, Measurement and Applied Informatics,*
Faculty of Mechanical Engineering, Slovak University of Technology
Nám. Slobody 17, 812 31 Bratislava, Slovakia
fax : +421-2-57294392; e-mail : tomas.poloni@stuba.sk
[**] *Institute of Information Engineering, Automation and Mathematics,*
Slovak University of Technology, Radlinského 9, 81237 Bratislava

Abstract: The results presented in this paper reflect the efficiency of a model explicit predictive control approach in a numerical simulation study of active vibration attenuation of a selected piezoelectric smart structure. Presented active vibration control system consist of the cantilever plate excited whit the use of piezoelectric actuators, the laser sensor and the digital signal processor board. This configuration enables to perform the vibration system experimental identification. Based on the structure responses determined by measurement, an explicit first mode state space model of the equivalent linear system is developed by employing subspace identification approach. Controller design is carried out by the multiparametric programming algorithm. The control law is incorporated into the finite state space partitions to perform as closed loop controller. Laboratory verification experiment evaluate the control law performance.

Keywords: piezoelectric sensor/actuator, explicit predictive controller, subspace system identification

1. INTRODUCTION

Monitoring and control of vibrations has become important for the aims of many engineering systems, as for example automotive industry, airspace industry or precise mechanics. Advances in smart materials have shown an increased interesting applications for passive and active attenuation. The advanced Technologies of smart materials lead to relatively small and light actuators and sensors with good physical integration, i.e. in-building into materials. The vibration control is historically from its beginning highly topical which has led to many methods and approaches to solution.

In general, the efficiency of passive damping materials in suppressing of mechanical vibrations is insufficient for the range of low frequencies. What is more, the passive damping materials considerably add on to the mass of the structure they damp. At the same time, they manipulate with the stiffness of the structure. The resulting strong vibrations can damage or totally destroy the structure. In the case of machining devices, the undesired vibrations lead to decreased precision of products. To avoid the disadvantages of passive damping elements, piezoelectric materials have come to wider use that can be well controlled in a wide range of frequency, without adding great amount of mass to the structure. The actuators that are built in the controlled structure produce force on a given object. The signal that controls the actuator arises from the control system obtaining feedback from sensors that can be also built in the controlled structure. The research of piezoelectric materials is fast gaining attention and it is expected that this technology will upgrade the quality of production in engineering industry. The control of vibration that utilizes piezoelectric materials can be performed passively (with shunt circuits) (Hagood and Flotow, 1991; Wu, 1998; Granier et al., 2001; Niederberger, 2005) or actively. In the shunt circuit techniques the main task of piezoelectric materials is to absorb the energy from the structure.

In active control, an external power is applied to a piezoelectric material to produce a force in opposite direction to that produced by vibrating structure at a particular position. The opposite forces will annul each other and thus reduce the vibration of the structure. In the literature many papers focuss on active vibration techniques. Good comparative study of some control techniques can be found in (Kumar et al., 2006). The predictive control was applied with its on-line computation in Wills et al. (2008); Takács and Rohal'-Ilkiv (2009).

In this paper a single mode explicit predictive controller for vibration attenuation will be studied. The explicit technique is particulary suitable for this fast dynamic application because of its low online computation load.

[⋆] The work has been supported by the Slovak Research and Development Agency under contract APVV 280-06, APVV-0160-07 and LPP 118-09. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

2. SYSTEM DESCRIPTION AND IDENTIFICATION

The vibration of cantilever is from the point of view of mathematical-physical modeling called problem of vibration of continuum, for which it must be considered that the cantilever has theoretically infinite number of its own frequencies (modes). For detailed description of a vibrating cantilever, the vibration of which is exited by external sources, generally speaking one would need to know the differential equation of infinite order for its every single point that would describe the vibration of that particular point. Not every own frequency has the same effect on the overall vibration. That is the reason why on describing the behavior of the vibrating cantilever, it is necessary to take into account the amount of energy (and in consequence the amount of displacement) that is related to the given own frequency. From the point of damping of vibrations, it is very often sufficient to damp the first n modes, where n depends on the actual aim of the control. The first own frequency is related to the greatest vibration of the cantilever in a given point.

In the area of physical mechanics several approaches are known that are based on rigid body mass simplification. They build on the simplified understanding and modeling of only the first n modes of vibration. Generally, these approaches arise from nonlinear differential equations.

On the other hand rather elegant approaches for obtaining system model of vibrating cantilever arise from the principles of system identification, such principles are based mainly on measured input output data. The method of subspace identification (Van Overschee and De Moor, 1996) is an effective method to determine matrices A, B, C of the general dynamic system described by deterministic state space model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= \quad Cx(k) \end{aligned} \tag{1}$$

To obtain the matrices of the state space model, an identification experiment was performed on the laboratory equipment shown in Fig. 1. The cantilever was excited by a piezoelectric actuator fixed to the cantilever. The input variable $u(k)$ was the voltage at amplifier. The output measured variable $y(k)$ was the end displacement of the cantilever (beam) that was measured with laser sensor. The excitation signal was set as swept sine signal with varying frequency, in the range of $0.1 - 15Hz$. The output signal had a significant gain of amplitude at the resonance frequency $8.126Hz$. Both the input and the output signal with resonance are shown in Fig. 2. By applying the FFT (Fast Fourier Transform) algorithm, the data from the time domain were transformed to the frequency domain. With appropriate software toolbox (Ljung, 2008) the matrices of the scaled system of the vibrating cantilever were determined with the following numerical values

$$x(k+1) = \begin{bmatrix} 0.857 & 1.114 \\ -0.212 & 0.874 \end{bmatrix} x(k) + \begin{bmatrix} -1.386 \\ -0.548 \end{bmatrix} u(k) \tag{2}$$

$$y(k) = [\, -0.562 \;\; 0.699 \,]\, x(k) \tag{3}$$

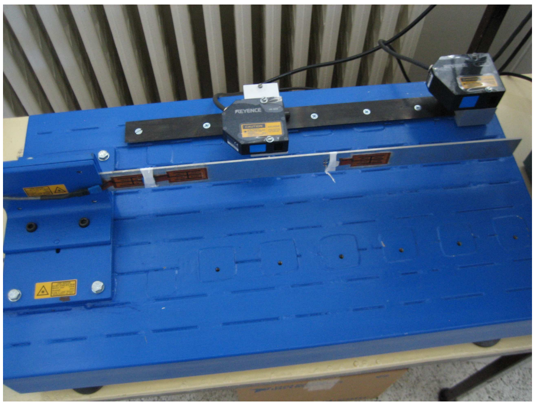


Fig. 1. Cantilever laboratory setup

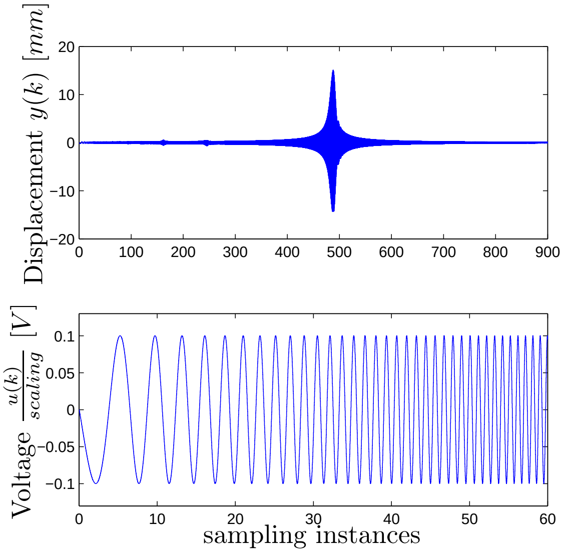


Fig. 2. Input/Output signals for system identification ($scaling = 1000$)

A comparison of agreement between the measured data and the model is in the Fig. 3. The system in the range of its first frequency showed linear behavior, which corresponds to a good agreement with the data measured. The sampling period of the model is $0.01s$

3. EXPLICIT PREDICTIVE CONTROLLER

The aim of the explicit predictive control was to demonstrate the damping effect at the end of a cantilever by minimization of its displacement from its equilibrium state. The task of the control law is to drive piezoelectric actuators in such a way that the control objective is attained in agreement with the physical constraints and the stability of the system. In this case, the objective is to bring the state to its zero position.

To describe the state of the vibrating cantilever, the following equation is to be considered

$$x(k+1) = Ax(k) + Bu(k) \tag{4}$$

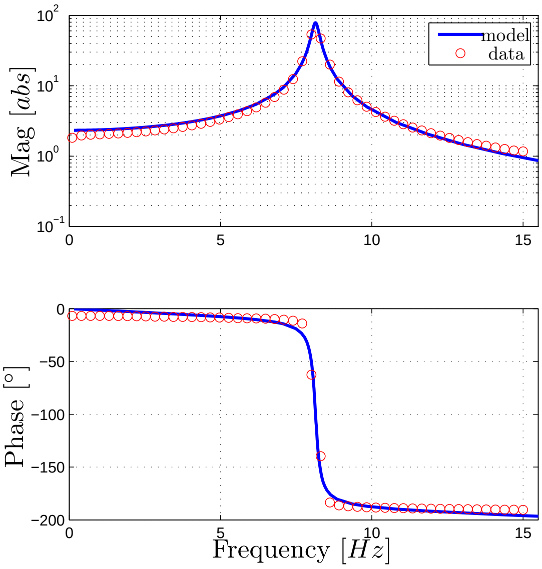


Fig. 3. Plots of measured frequency based data and identified model structure

with the assumption that the state $x(k)$ always range in the polytope given

$$\mathcal{P}_i := \{x \in \mathbb{R}^n | H_i x \le K_i\} \tag{5}$$

thus, $x \in \mathcal{P}_i$. The matrices H_i and K_i are the matrices of suitable dimensions. Defining the task of predictive control in the form

$$\min_u \sum_{k=0}^{N-1} [u^T(k)Ru(k) + x^T(k)Qx(k)] + \\ + x(k+N)^T Px(k+N)) \tag{6}$$

subject to

$$\begin{aligned} y_{min} \le y(k+i) \le y_{max}; i = 1, \dots, N \quad (7)\\ u_{min} \le u(k+i) \le u_{max}; i = 0, 1, \dots, N-1 \\ x(k+1) = f(x(k), u(k)), k \ge 0 \\ y(k) = Cx(k), k \ge 0 \end{aligned}$$

the following solution in the sense of multi-parametric programming (Bemporad et al., 2002; Grieder et al., 2005) can be obtained

$$u(k) = F_i x(k) + G_i \tag{8}$$

The matrices R, Q, P are the weighting matrices. The resultant solution leads to partitioning the state space to particular subspaces. For every subspace different calculated matrices F_i and G_i are valid. The task defined as minimization of objective function (6) with constraints under (7), was solved by Multi-parametric Toolbox (Kvasnica et al., 2006). The optimization task had the following input parameters: $y_{min} = -15$, $y_{max} = 15$, $u_{min} = -0.1$, $u_{max} = 0.1$,

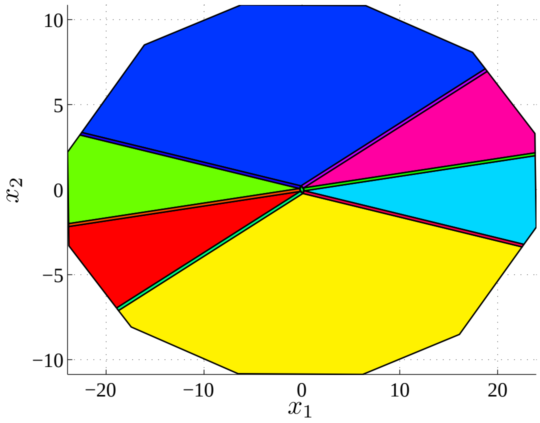


Fig. 4. Partitioning of the state space into 19 regions

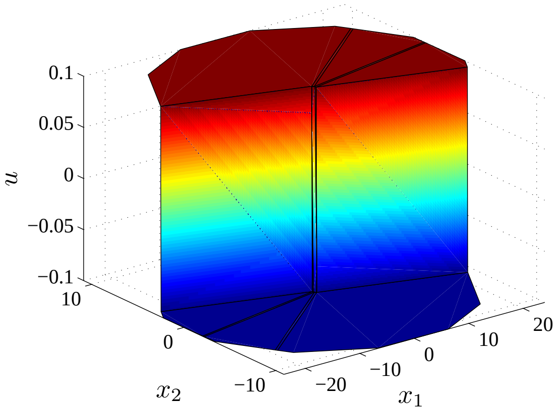


Fig. 5. Look-up table of the control moves

$$Q = \begin{bmatrix} 10^{-12} & 0 \\ 0 & 0 \end{bmatrix},$$

$$R = 10^{-1},$$

$$P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

,with the prediction horizon $N = 3$. The explicit solution consists of 19 regions of partitioned state space, as shown in Fig. 4.

The value of the control action as a function of the state is shown in Fig. 5.

As can be seen from the values of all control actions for all the considered states, this procedure is the on-off type of control, which is logical, as the reshape of the end of the cantilever, caused by the action of the actuator, is negligible, compared to the displacement of the cantilever, caused by the overall vibration dynamics.

4. SIMULATION RESULTS

To simulate the control, the task of state transition from its initial value $x(0) = [-23, 0.5]^T$ to zero equilibrium of

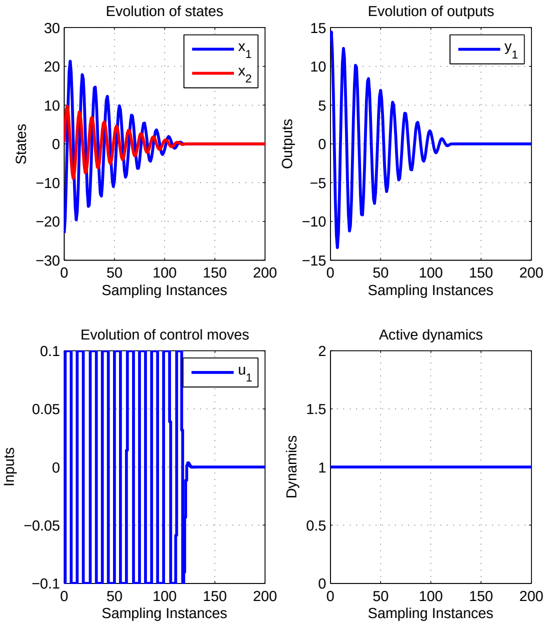


Fig. 6. Closed loop simulation

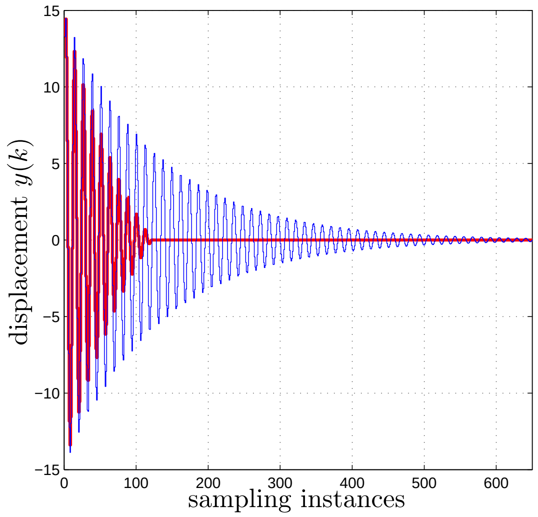


Fig. 7. Free and closed loop response of the system from initial condition $x(0) = [-23, 0.5]^T$

the identified model, given by equations (2) and (3) is to be considered. The closed loop control simulation is demonstrated in Fig. 6. From the course of the states and the output of the model, it can be seen that the control is able to damp the cantilever from a given initial state in $\approx 1s$.

Fig. 7 shows the course of the free response and controlled response of the model of cantilever vibration from its given initial state.

REFERENCES

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. (2002). *The explicit linear quadratic regulator for constrained systems.* Automatica.

Granier, J., Hundhausen, J., and Gaytan, G. (2001). *Passive modal damping with piezoelctric shunts.* Technical report, Los Alamos National Labs.

Grieder, P., Kvasnica, M., Baotic, M., and Morari, M. (2005). *Stabilizing low complexity feedback control of constrained piecewise affine systems.* Automatica.

Hagood, N. and Flotow, A. (1991). *Damping of structural vibrations with piezoelectric materials and passive electrical network.* Journal of Sound and Vibration.

Kumar, R., Singh, S., and Chandrawat, H. (2006). *Adaptive vibration control of smart structures: a comparative study.* Smart Materials and Structures.

Kvasnica, M., Grieder, P., Baotic, M., and Christophersen, F.J. (2006). *Multi-Parametric Toolbox (MPT): User's Manual.* Extended documentation.

Ljung, L. (2008). *System Identification Toolbox 7.*

Niederberger, D. (2005). *Hybrid systems: Computation and Control, volume 3414LNCS, Proceedings of the 8th International Workshop, HSCC 2005, chapter Design of Optimal Autonomous Switching Circuits to Suppress Mechanical Vibration.* Springer Verlag, ETH Zürich, Switzerland.

Takács, G. and Rohal'-Ilkiv, B. (2009). Newton-Raphson based efficient model predictive control applied on active vibrating structures. In *Proceedings of the European Control Conference 2009*, 2845–2850. Budapest, Hungary.

Van Overschee, P. and De Moor, B. (1996). *Subspace Identification for Linear Systems.* Kluwer Academic Publisher.

Wills, A.G., Bates, D., Fleming, A.J., Ninness, B., and Moheimani, S.O.R. (2008). Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1), 3–12.

Wu, S. (1998). *Method for multiple mode shunt damping of structural vibration using pzt transducer.* Proceedings SPIE, Smart Structures and Intelligent System.

Comments – Remarks

Vibration Control of a Cantilever Beam Using Piezoelectric Feedback [⋆]

Gergely Takács, **Boris Rohaľ-Ilkiv**

Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail: gergely.takacs@stuba.sk

Abstract: This paper describes an experimental laboratory device, which utilizes strips of piezoelectric material as structural actuators and sensors, emulating behavior of a class of engineering problems dealing with active vibration attenuation of lightly damped mechanical systems. As an example of such system a cantilever laboratory beam is taken. Detailed account is given on the identification process of the feedback signal provided by the piezoelectric sensors to the device controller. The feedback model is validated both in time and frequency domain, utilizing a linear quadratic controller as a basis of comparison. The cantilever is excited manually and using an electrodynamic shaker. The comparison of the piezoelectric sensor based feedback control and direct high precision industrial laser triangulation feedback are shown. It was verified during the experiment that a mere second order feedback model based control process covering the first dominant vibration frequency is comparable to control utilizing direct distance readings. Moreover the damping effect exceeds the bandwidth of interest as the single-sided cantilever beam tip deflection amplitude spectra shows no substantial difference between the two feedback methods for higher structural vibration modes.

Keywords: vibration control, piezoelectric actuators, sensors, feedback control

1. INTRODUCTION

Employing active vibration attenuation techniques in commercial products is slowly becoming a reality. One might think of aeronautical applications, like the damping of helicopter rotor wing vibrations, active stabilization of large space structures or for example vibration attenuation of antenna masts (Boeing, 2004; Phillips et al., 1990; Blachowski, 2007). Other active and semi-active methods of vibration damping are taken on by the automotive industry for controlling suspension systems. The possibilities of using advanced materials combined with progressive control algorithms to eliminate undesired vibration effects is practically limitless (Preumont, 2002; Inman, 2006).

While the proven technologies are transferred into practice, there is a constant need to investigate further and widen the boundaries of active vibration damping in research laboratories. A rather important branch of research is focused on advancing the field of active materials; like piezoelectrics, electro-active polymers, magneto-rheologic fluids, shape-memory alloys and others. But no active control system is complete without the proper control algorithm, therefore another essential part of the scientific process is to investigate how already existing technologies can benefit from better control methods.

1.1 Motivation

Acquiring a reliable feedback signal is essential for the efficiency and reliability of all control systems. When considering means of sensing vibration levels, one has to balance between precision, price and the physical interaction with the given mechanical system. Mass produced products need to be cost optimized, what naturally involves tradeoffs in the sensing apparatus and thus also the condition of the feedback signal. But not only the sensors must be simple, quite often it is beneficial to keep the computation load on minimum in order to make optimization based control algorithms a viable option.

Laboratory applications often consider LASER Doppler vibrometry to gain feedback signal to the controller. In addition to great precision, contact-free measurement has an advantage of preventing structural interaction with the controlled system. LASER Doppler vibrometers however are out of question for mass-produced applications, as their placement is very problematic, the sensor heads and processing equipment are heavy, large and very expensive. Industrial grade LASER optical sensors based on triangulation methods are a rather good compromise, since they are smaller and less expensive than their laboratory grade counterparts (Takács and Rohaľ-Ilkiv, 2009a).

[⋆] The authors gratefully acknowledge the financial support granted by the Slovak Research and Development Agency under the contracts APVV-0280-06 and APVV-0160-07, and reviewers' comments. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is co-financed from the state budget of the Slovak Republic.

Physical dimensions and inconvenient placement is still an issue, and because of their price range they can be only recommended for high budget products, like the ones encountered in the aviation industry and military.

The use of accelerometers is very common in academic publications, as they are relatively cheap and provide precise feedback to the control system (Qiu et al., 2009; Dong et al., 2006; Petersen and Pota, 2003). Real life product integration is also a feasible possibility. Accelerometer miniaturization has come to a point where these devices can be bonded or mounted to the structural surface without significantly altering its mass-stiffness properties.

Similarly common is the utilization of the direct piezoelectric effect to gain vibration level estimates of the controlled mechanical structure (Wills et al., 2008; Kermani et al., 2004; Lin and Nien, 2005; Sloss et al., 2003). In these feedback control systems a piezoelectric patch is either bonded onto the structural surface, or directly integrated into the structure material. Better mass distribution, very low price and the possibility of direct structural integration is definitely an advantage over accelerometers. Neither accelerometers, nor piezoelectric wafers may sense the D.C. component of vibrations, this is however usually not an issue.

1.2 Problem statement

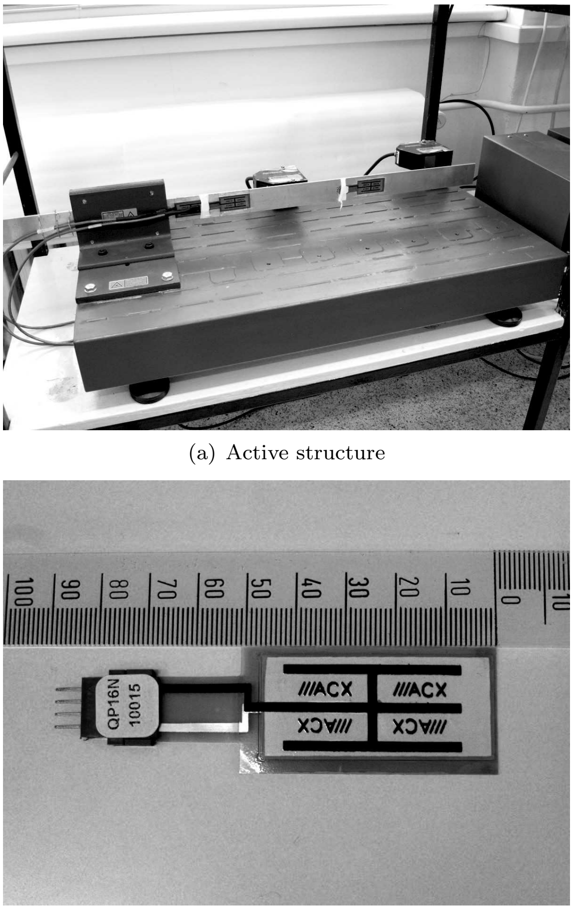
A lightly damped, cantilever-like active mechanical structure is given with bonded piezoelectric actuators. An additional piezoelectric patch is placed close to the clamped end and acts as a vibration sensor. The task is to create a simple second-order linear model to estimate deflections of the mechanical structure at the beam tip.

The measurement estimates shall be compared to conventionally true deflection levels acquired through a LASER triangulation device both in time and frequency domain. In addition to that, possible damping performance degradation is investigated through comparing damping efficiency of the estimate based feedback with direct feedback control.

A second order measurement model includes only the first resonant mode of the structure. However the inclusion of the first dominant mode of the lightly damped structure could be sufficient to control the vibrational behavior even at higher frequency excitations. Damping performance of direct and estimate based feedback shall be thus compared through frequency domain measurements, reaching higher resonant modes. This shall answer the question whether a mere second order model is suitable for position estimates, even when the structure is subject to excitations exceeding the bandwidth of the model. Using the simplest possible solution to assess feedback signal may help to keep additional computational load at minimum, which is essential in optimization based control systems with short sampling periods. The paper thus attempts to find a feedback solution which provides hardware and computation time costs at bare minimum.

2. HARDWARE DESCRIPTION

A clamped cantilever beam is given, which may model the general vibrational response of a class of engineering problems. This lightly damped mechanical system behaves similarly to the helicopter rotor beams in flight, manipulation arms or solar panels in outer space and many other real-life structures.



(a) Active structure

(b) Piezoelectric patch

Fig. 1. A clamped cantilever beam with piezoelectric sensors and actuators is featured on (a), while detail of the piezoelectric actuators and sensor is shown on (b).

The experimental laboratory setup assumed throughout this work is presented on Figure 1(a). The beam is composed of commercially pure aluminum with the dimensions of 550 × 40 × 3 mm. A heavy base is necessary to prevent mechanical interaction with the outside.

A pair of piezoelectric actuators is mounted close to the clamped end. The actuators are identical, manufactured by MIDÉ having the factory designation mark QuickPack QP16n. The outside dimensions of the actuator are 45.9 × 20.7 × 0.25 mm and it is shown on Figure 1(b). The actuators are connected counter phase, receiving the same high voltage signal through a MIDÉ EL-1225 operational power amplifier. The placement of the actuators has been influenced by the goal of maximizing deflection amplitudes at the first resonant frequency.

A third patch identical to the actuators is bonded onto the structural surface. This piezoelectric patch acts as a sensor, making use of the direct piezoelectric effect. Its optimal placement has not been a subject of this research, however we have to note that Finite Element Modeling (FEM) has been used to avoid anti-resonance nodes at higher frequencies. Other placement criteria included the minimization of mechanical interaction with the structure, by placing the minimum amount of lead wires close to the structural surface. Voltage signal acquired from the sensor is directly connected to the analogue input of a laboratory measurement card, without additional amplification. When the beam is in first resonance, the output voltage levels actually exceed the possible input range of the data acquisition device. A 100 kΩ resistor is installed parallel to the piezo patch to match the voltage levels with the A/D input device[1].

Vibration levels are measured directly through an industrial LASER triangulation sensor, placed at the free end of the cantilever beam. A Keyence LK-G82 sensor with an accuracy of ±0.05% and the resolution of 0.2μm provides direct distance readings in the range of 80 ± 15mm. Measurements are forwarded to a Keyence LK-G3001V central processing unit for filtering and finally a scaled analogue voltage signal is passed onto the measurement card. The distance readings provided by this system are considered as reference throughout this work.

Figure 2 features the simplified schematic representation of the laboratory experimental hardware. The computer marked as xPC Target on the figure serves for implementing the controller and data logging software real-time, on the Mathworks xPC Target rapid software prototyping system. This computer contains a National Instruments DAQ-6030 measurement card with 18bit resolution and amongst others two analogue outputs with ±10V range, necessary to drive the amplifiers for the piezoelectric actuators.

Controller design and development is taking place on a separate computer, marked as xPC Host on the figure. The development platform used in this setting is Matlab / Simulink, where the block schemes responsible for control and data logging are transferred to the real-time controller via Ethernet, through the TCP/IP protocol.

3. SYSTEM AND MEASUREMENT DYNAMICS MODELING

Although it would be possible to create a single dynamic model, describing the input - output relationship of the actuator behavior and the piezoelectric patch; creating separate system and measurement models has its advantages. If constrained model based predictive control (MPC) is considered as the choice of control algorithm, a separate structural model is necessary to enforce constraint limits on the vibration output. Therefore separate system and estimation dynamics is assumed in this paper due to the inspiring possibility to use MPC on lightly damped vibrating structure, posing numerous practical issues upon implementation (Wills et al., 2008; Hassan et al., 2007; Takács and Rohal'-Ilkiv, 2009c).

Linear time-invariant (LTI) state-space systems describe the dynamics of the system and measurement process according to:

$$x_{k+1} = Ax_k + Bu_k \qquad\qquad y_k = Cx_k \tag{1}$$

where x is a 2×1 state vector, u is a 1×1 input, y is a 1×1 output. Matrices A,B and C are the transition matrix, input matrix and output matrix. Integer k denotes sampling instances.

3.1 System dynamics

The system dynamics of the experimental device are described by a second-order LTI state-space model, modeling the relationship between voltage input to the piezoelectric actuators and deflections directly measured at the cantilever beam tip in millimeters.

This model has been identified experimentally. A chirp signal in the range of $0-20$Hz, amplified to the polarization limits of the actuators has been supplied, while the tip deflections have been logged with a sampling time of T=0.01 seconds. This sampling period is sufficient for the given model, since it significantly exceeds the first resonant frequency of 8.1 Hz. The filtered and detrended time domain data has been converted into the frequency domain using Fast Fourier Transform (FFT).

The final state-space model has been created utilizing a subspace-iteration method as described in Ljung (1999). The model described by (2) has been already proven in MPC controlled vibration attenuation, using direct tip deflection readings as reference (Takács and Rohal'-Ilkiv, 2009a,c).

$$A = \begin{bmatrix} 0.867 & 1.119 \\ -0.214 & 0.870 \end{bmatrix} \tag{2a}$$

$$B = \begin{bmatrix} 9.336E^{-4} \\ 5.309E^{-4} \end{bmatrix} \tag{2b}$$

$$C = [-0.553 \quad -0.705] \tag{2c}$$

Akaike Final Prediction Error (FPE) criterion for this model has been calculated to be 0.0142 (-). The model validation process proved to yield a satisfactory match, while the transient and frequency response of the model was also adequate for the considered bandwidth.

3.2 Piezoelectric sensor estimation model

The second order LTI state-space model, describing the relationship between the measured output voltage of the piezoelectric sensor and the direct tip deflection readings has been identified experimentally as well.

A pseudo-random manual excitation has been applied to the beam, while the exact tip distance readings using the LASER triangulation system and the voltage output from the sensor has been logged. The resulting data set has been pre-processed to remove trends means and frequencies

[1] The A/D device input levels are set to ±10V.

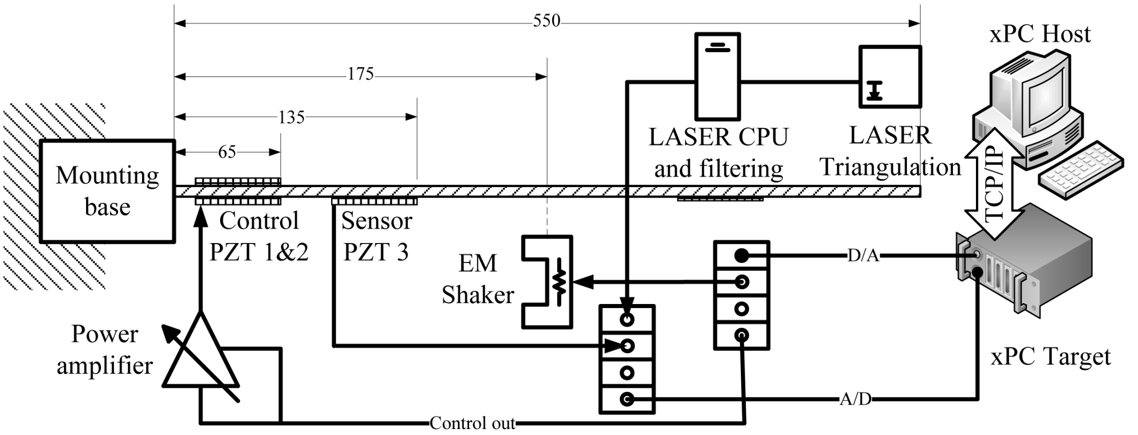


Fig. 2. Simplified schematic representation of the laboratory experimental hardware.

exceeding the bandwidth of interest. The final state-space model has been calculated using the subspace iteration method featured in Ljung (1999).

After comparing the model output with the validation data, the state-space system described by (3) has been selected as the basis for piezo patch based deflection estimation. Examining the transient and frequency response, model residuals, along with a FPE criterion of 0.0091 (-) indicated a model suitable for further use in this work.

$$A = \begin{bmatrix} 0.987 & 0.144 \\ -0.274 & 0.009 \end{bmatrix} \tag{3a}$$

$$B = \begin{bmatrix} 3.959E^{-2} \\ 1.851E^{-1} \end{bmatrix} \tag{3b}$$

$$C = [34.72 \quad -1.359] \tag{3c}$$

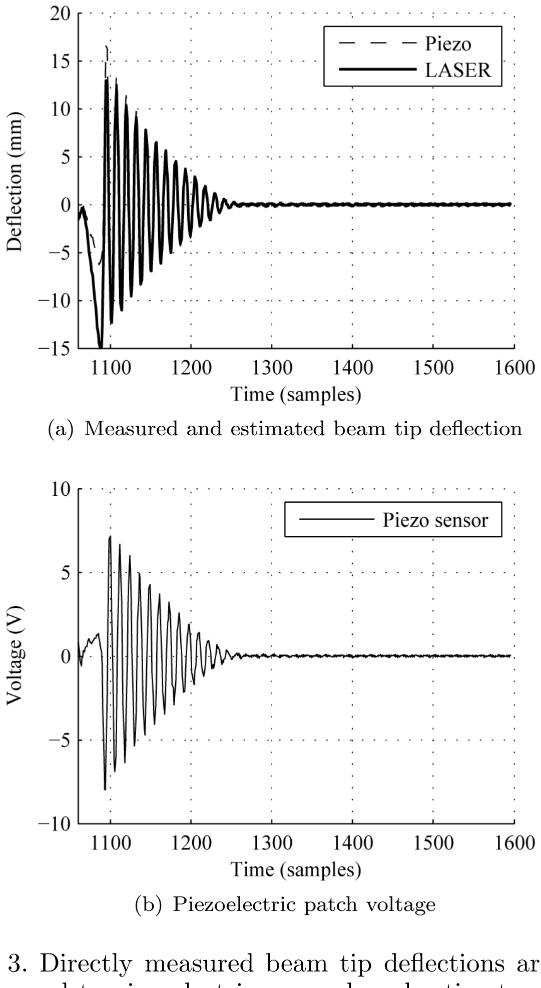
4. FEEDBACK MODEL VALIDATION

Experiments performed to validate the tip deflection estimation model assume a system model described by (2) and a piezoelectric sensor feedback measurement model according to (3). Estimate model sampling has been set to $T = 0.01$ s, however a high frequency excitation test involved a data logging rate of $T = 0.0002$ in order to capture dynamics above the bandwidth of interest, while leaving model sampling at its default value.

4.1 Time domain position estimates

The beam tip has been deflected 10 mm away from its equilibrium state and released to vibrate under LQ control[2]. After the initial deflection, the cantilever beam has not been subjected to other outside excitation. The results of this experiment are featured on Figure 3.

Figure 3(a) shows measured and estimated tip displacements. As it is expected, the piezoelectric patch based tip deflection estimate is incorrect when the beam is subjected to slow changes, since it is not picking up the D.C. component of a changing signal. After the beam starts to vibrate at sample time 1100, the model tends to slightly



(a) Measured and estimated beam tip deflection

(b) Piezoelectric patch voltage

Fig. 3. Directly measured beam tip deflections are compared to piezoelectric sensor based estimates on (a), while (b) shows the corresponding measured voltage output of the piezoelectric patch .

[2] See Section 5 for the controller description.

overestimate positive deflections, however at the following periods it gets nearly indistinguishable from the precise LASER reference measurements. The unprocessed direct voltage output of the piezoelectric sensor patch is featured on Figure 3(a).

4.2 Frequency domain position estimates

The frequency domain experiment involved an outside mechanical excitation provided by a Bruel&Kjær Type 4810 electrodynamic shaker. The shaker has been mechanically connected to the beam surface 175 mm away from the clamped beam base [3]. The excitation signal has been passed onto the laboratory shaker through a Bruel&Kjær Type 2718 amplifier. A 200 seconds long chirp signal swept through frequencies of 0-500Hz, exciting the beam through its first five measurable resonant peaks, up to deflections of approximately ±15mm at the first.

The single-sided amplitude spectra of laser measured and piezoelectric sensor based tip displacement estimates are indicated on Figure 4[4]. The black response is measured directly, while the lighter shade indicates the piezoelectric sensor based estimates. Resonant modes are numbered, where modes (3) and (5) are twisting modes which cannot be controlled or directly measured using this hardware configuration (Takács, 2009).

As it is evident from the response, the beam tip deflections are correctly estimated only around the neighborhood of the first resonant mode. Neither the model order nor the native sampling period makes possible to correctly assess tip position above 15 Hz. Spectral leakage, an artifact of the FFT transformation is visible beyond this frequency range for the piezo estimate. This carries no information and is merely a side-effect of performing FFT on a low frequency content data. On the other side, near D.C. position changes and slow vibrations cannot be detected by the piezoelectric sensor.

If the explicit inclusion of higher order dynamics in the estimation model is necessary, the model order can be increased. However this also leads to a need to increase model sampling rate, which could be an issue for computation intensive control algorithms like MPC (Takács and Rohal'-Ilkiv, 2009b).

5. DAMPING PERFORMANCE

To find out whether the use of estimated beam tip deflections in the feedback control loop cause a significant degradation in control performance, a series of experiments were performed both in the time and frequency domain. Deflections have been measured utilizing the LASER triangulation method, while the system has been subjected to identical excitation with identical control methods. The only difference was the use of direct measurements or piezoelectric sensor based feedback estimates.

The control strategy considered throughout the damping performance comparison test was a simple linear quadratic (LQ) controller. The output of this controller has been saturated to ±120 V in order to prevent depolarization of the piezoelectric material (Spangler, 2007; MIDÉ, 2007). The LQ controller has been calculated using the state space model according to (2), a state penalty matrix $Q = C^T C$ and an input penalty $R = 10^{-4}$. Controller sampling rate has been set to $T = 0.01$s. The input penalty is based on previous experiments, and its value is suitable for the previously mentioned saturation limits (Takács, 2009). The LQ controller gain K can be expressed according to (4).

$$K = [12.97 \qquad -125.50] \tag{4}$$

Figure 5 indicates the simplified block scheme of the control software implemented on the xPC real-time rapid software prototyping system. The analogue voltage output is acquired through the measurement card, this includes exact measurements from the laser and the voltage signal from the piezoelectric sensor. The piezo sensor voltage is passed through the estimation model, while a switch enables the user to select between direct or estimated position feedback signals. The feedback then passes through a Kalman state estimator, and the resulting state estimates are multiplied by the LQ gain vector. Outputs are saturated and adjusted to the amplification level of the power amplifier. The block scheme also includes input to the electrodynamic shaker and means for data logging.

Plots in this section do not indicate free response without control. This is to keep the responses clear and readable. Saturated LQ control with direct feedback provides a very effective damping performance, effectively reducing settling times by an order of magnitude. This is comparable to constrained MPC control featured in (Takács and Rohal'-Ilkiv, 2009b,c; Takács, 2009).

5.1 Time domain test - initial deflection

The beam has been deflected to a position 10 mm away from its equilibrium and then released to vibrate under saturated LQ control. As it has been previously mentioned, the two scenarios differ only in the means of acquiring the feedback signal and all responses are measured directly through the triangulation device.

Beam tip vibrations following the release of the structure are featured on Figure 6(a). As it is evident from this figure, the two responses are indistinguishable, the estimated feedback values do not degrade the damping behavior under saturated LQ control.

A very slight difference between the two responses is observable on Figure 6(b), showing the voltage signal sent to the piezoelectric actuators. The responses are identical up to the time sample 120. After this point the differences are to be attributed to the fact, that unwanted outside excitations occur in both cases and they are compensated by the controller. These effects are only observable when the beam is near its equilibrium state.

5.2 Frequency domain test - electrodynamic shaker

The possible damping performance degradation attributed to the use of estimated tip position feedback has been

[3] This mechanical connection has been only present when the shaker was needed, time domain tests were performed without this addition.

[4] Note that the input force has not been measured, therefore the featured response is not a transfer function - it only analyzes the nature of the output signal in the frequency domain.

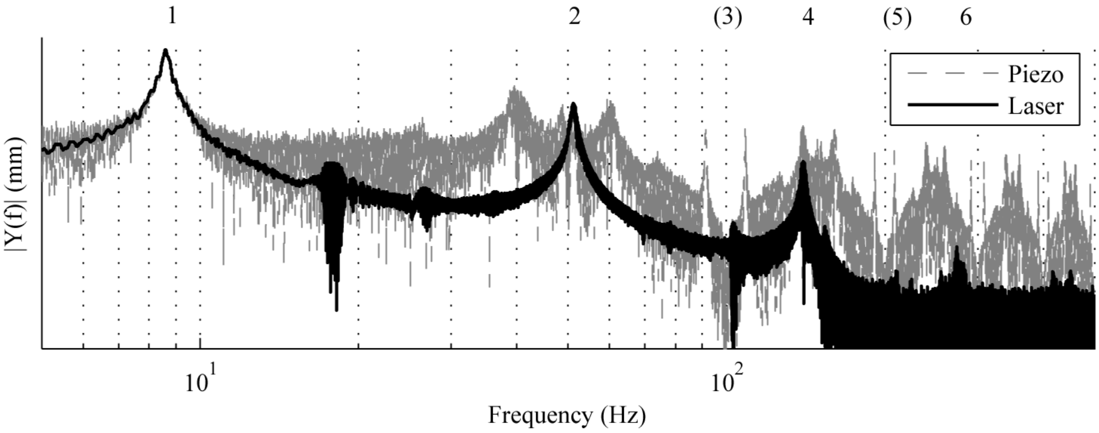


Fig. 4. Measured and piezoelectric sensor feedback estimated beam tip deflections in a single-sided amplitude spectrum. Numbers denote corresponding structural vibration modes.

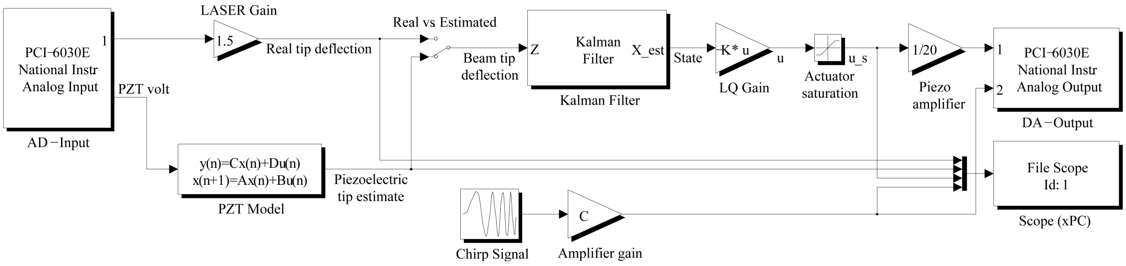


Fig. 5. Simplified block scheme of the controller algorithm.

investigated in the frequency domain as well. The mechanical structure has been excited using the laboratory shaker setup described in 4.2.

Two tests were performed: one for the bandwidth of interest and the other for higher frequencies. The first test involved an excitation through an amplified chirp signal for the bandwidth of $0 - 20$ Hz. All of the discrete sampling periods were set to $T = 0.01$. The second test was aimed to investigate wheter the low order structure and position estimate model provides satisfactory feedback at higher frequencies. For this test the estimation model and controller sampling frequency was left at its original value, however the excitation signal and data logging has been sampled by an increased rate of $T = 0.0002$ seconds.

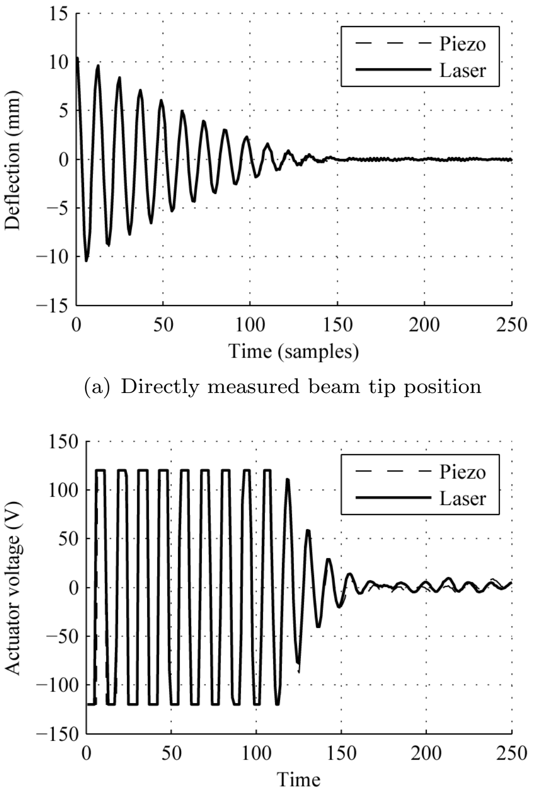
The directly measured single-sided amplitude spectra of beam tip vibrations for the direct and estimated feedback controlled systems is featured on Figure 7. The two control schemes are practically indistinguishable in the bandwidth of interest, just as it is indicated by 7(a). In the region of the first resonant frequency, both control feedback schemes perform equally well.

According to 4.2 the piezoelectric sensor based low-order feedback models are only useful close to the frequency range around the first structural resonant frequency. Despite of this fact the results featured on Figure 7(b) indicate that the estimated control scheme gives comparable results to the one with access to direct feedback readings. This experiment utilized second order system and measurement models, however it has been excited with frequencies high above its normal bandwidth. As it is indicated by Fig. 7(b) the damping performance of the model estimated feedback scheme is very similar to the one with access to direct deflection measurements.

6. CONCLUSION

A low order position estimation model, based on piezoelectric sensor signals for a lightly damped active structure has been introduced in this paper. Along with the experimental validation of the estimator, the control performance has been evaluated for the direct measurement and estimate based controllers.

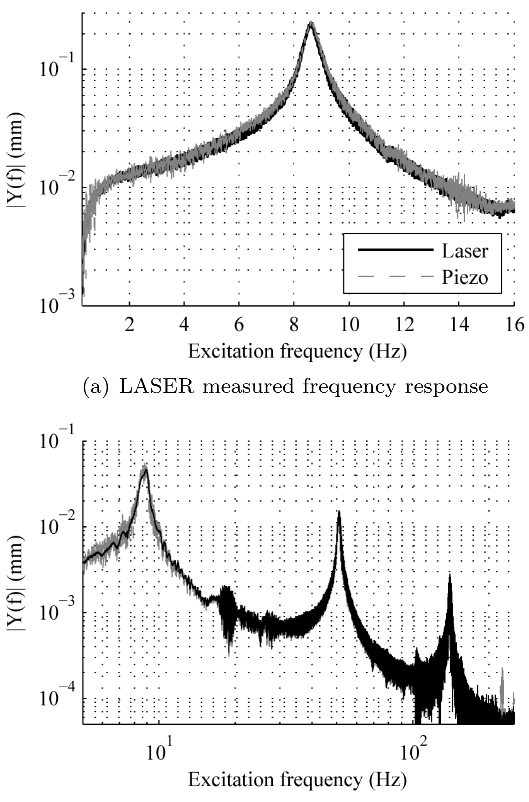
The free and controlled vibration response of the beam tip in the time domain is dominated by the first structural resonant mode, thus a second order tip estimation model is sufficient to generate responses closely matching to reference values. The frequency domain experiment clearly indicates limitations of such low order models, along with the physical properties of the piezoelectric sensors: the estimation model provides the best results in the vicinity of the first structural resonance mode.



(a) Directly measured beam tip position

(b) Voltage signal to actuators

Fig. 6. Direct position feedback based control compared to piezoelectric sensor estimated feedback in an initial deflection test is indicated on (a), while corresponding controller voltage outputs are presented on (b).



(a) LASER measured frequency response

(b) Wide-band frequency response

Fig. 7. Direct LASER triangulation measured response is indicated on the low frequency single sided amplitude spectra on (a), while (b) shows damping performance for higher structural modes.

The controlled vibration verification tests show, that the the controller having access to estimated tip deflections performs nearly identically to the one utilizing direct measurements. This is not only true for the initial deflection test performed in the time domain, but also for the frequency domain. According to these experiments, the damping effect of both control schemes is identical. Additionally, the estimate based feedback controller provides comparable damping performance to the direct measurement feedback counterpart even at excitation frequencies exceeding the original bandwidth of interest.

It may be concluded that a piezoelectric sensor based feedback signal utilized on a lightly damped vibrating structure offers a damping performance comparable to directly measured feedback control, even when using a simple second order feedback estimate model.

6.1 Future works

The inclusion of higher order beam resonance modes in the estimation models may bring a qualitative increase in the precision of tip position estimates. However it still remains a question whether there are any practical advantages to use high order state-space models in model based predictive control of such and similar lightly damped structures. On the other side with larger model orders increased computation times are always to be expected, which may pose practical problems in the implementation of computation heavy optimization based control methods. Future works shall address these questions in more detail.

Further works shall investigate the possibility to utilize other types of sensors in estimating position changes of lightly damped vibrating structures. The use of capacitive proximity sensors in active vibration damping systems is somewhat unusual, however this contact-free measurement method could offer numerous advantages. It is important to study whether the bandwidth limitations of such a sensor permit its use as feedback source in vibration attenuation, or how its dynamic reaction range is influenced by using aluminum instead of standardly assumed steel. An upcoming investigation shall explore this attractive alternative too.

REFERENCES

Blachowski, B. (2007). Model based predictive control of guyed mast vibration. *Journal of Theoretical and Applied Mechanics*, 45, 405–523.

Boeing (2004). Boeing-Led Team Successfully Tests SMART Materials Helicopter Rotor. Online. Available: http://www.boeing.com/news/releases/2004/q2/nr_040518t.html.

Dong, X., Meng, G., and Peng, J. (2006). Vibration control of piezoelectric smart structures based on system identification technique: Numerical simulation and experimental study. *Journal of Sound and Vibration*, 297, 680–693.

Hassan, M., Dubay, R., Li, C., and Wang, R. (2007). Active vibration control of a flexible one-link manipulator using a multivariable predictive controller. *Mechatronics*, 17, 311–323.

Inman, D.J. (2006). *Vibration with control*. Wiley & Sons.

Kermani, M.R., Moallem, M., and Patel, R.V. (2004). Parameter selection and control design for vibration suppression using piezoelectric transducers. *Control Engineering Practice*, 12, 1005–1015.

Lin, J. and Nien, M.H. (2005). Adaptive control of a composite cantilever beam with piezoelectric damping - modal actuators / sensors. *Composite Structures*, 70, 170–176.

Ljung, L. (1999). *System Identification: Theory for the User*. PTR Prentice Hall, Upper Saddle River, NJ., USA.

MIDÉ (2007). *QuickPack Actuator Catalog*. MIDÉ Technology Corporation.

Petersen, I.R. and Pota, H.R. (2003). Minimax LQG optimal control of a flexible beam. *Control Engineering Practice*, 11, 1273–1287.

Phillips, D.J., C.Hyland, D., and C. G. Collins, J. (1990). Experimental demonstration of active vibration control for flexible structures. In *IEEE Conference on Decision and Control*, volume 4., 2024–2029.

Preumont, A. (2002). *Vibration Control of Active Structures*. Kluwer Academic Publishers.

Qiu, Z., Wu, H., and Ye, C. (2009). Acceleration sensors based modal identification and active vibration control of flexible smart cantilever plate. *Aerospace Science and Technology*, 13, 277–290.

Sloss, J.M., Bruch, J.C., Sadek, I.S., and Adali, S. (2003). Piezopatch sensor / actuator control of the vibrations of a cantilever under axial load. *Composite Structures*, 62, 423–428.

Spangler, R. (2007). *Piezo Sensor Techical Note*. MIDÉ Technology Corporation, 2. edition.

Takács, G. (2009). *Efficient Model Predictive Control Applied on Active Vibration Attenuation*. Slovak Technical University. Dissertation thesis.

Takács, G. and Rohal'-Ilkiv, B. (2009a). Implementation of the Newton-Raphson MPC algorithm in active vibration control applications. In *Proceedings of The 3rd International Conference on Noise and Vibration: Emerging Methods*, 145:1–12. Oxford, United Kingdom.

Takács, G. and Rohal'-Ilkiv, B. (2009b). MPC with guaranteed stability and constraint feasibility on flexible vibrating active structures: a comparative study. In *Proceedings of the eleventh IASTED interantional conference: Control Applications*, 278–285. Cambridge, United Kingdom.

Takács, G. and Rohal'-Ilkiv, B. (2009c). Newton-Raphson based efficient model predictive control applied on active vibrating structures. In *Proceedings of the European Control Conference 2009*, 2845–2850. Budapest, Hungary.

Wills, A.G., Bates, D., Fleming, A.J., Ninness, B., and Moheimani, S.O.R. (2008). Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1), 3–12.

Comments – Remarks

Control of a laboratory helicopter model using a guaranteed stability MPC technique *

T. Volenský, **B. Rohaľ-Ilkiv**, **T. Polóni**,

Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17
812 31 Bratislava, Slovakia
fax : +421 2 5249 5315; e-mail : tomas.volensky@stuba.sk

Abstract: The main purpose of this paper is to proposes application of model predictive control and subspace identification techniques for control and estimating of a linearized state-space laboratory helicopter model using input-output data. A Model Predictive Control (MPC) based algorithm equipped with an integral action is presented. The algorithm is defined for physical model of a simplified laboratory helicopter with two rotors. The helicopter model represents nonlinear multi-input multi-output (MIMO) mechanical system (with two-inputs and two-outputs) characterized by strong coupling effects. Based on the discrete time linear model a predictive controller with bounded input signals and guaranteed closed loop stability is designed and implemented. The resulting algorithm is implemented by means of Matlab/Simulink Real-Time workshop and xPC Target prototyping environment to capture the system fast dynamics.

Keywords: predictive control, MIMO systems, guaranteed stability, target sets, laboratory helicopter model

1. INTRODUCTION

The MPC controllers nowadays represent a well-established control technology used in many processes especially if various process constraints must be meet, see (Morari and Lee, 1997), (Qin and Badgwell, 2003), (Maciejowski, 2002), (Rossiter, 2004) and (Wang, 2009). Mostly MPC design assumes Llinear Time Invariant (LTI) models of the controlled plants. These models are usually formulated as ARX or state-space models. The ARX models can by standardly identified from the input/output data collected from experiments with the plants, the state space-space models can by formulated usually using physical principles reflecting the process dynamics. In order to implement the model predictive controller for a nonlinear plant the linearization in a vicinity of a given reference trajectory is usually considered. This paper is focused on design of a MPC algorithms for reliable control of a simplified laboratory helicopter model with two rotors using dual mode approach and closed loop paradigm (Kouvaritakis et al., 2000b) to guarantee the system closed loop stability and utilizing subspace-based methods for the system identification.

It is important to have a proper mathematical model of the system being controlled. In the case of laboratory helicopter setups most of the mathematical description of helicopter dynamics presented in literature are based on classical Newtonian physics. Resulting mathematical model of the system then becomes a set of linear and non-linear differential equations with several non-linear static functions. Such models are often very complicated. In order to apply the models for helicopter control a group of physical parameters and non-linear static functions should be analyzed and determined first, which is not elementary task and may cause significant model uncertainties. Moreover, majority of control techniques expect the resulting helicopter dynamics model in a more simple form of linear/linearized state-space structure with all matrix entries known. Typically the classical physical modelling methods lead to state-space structures in which the parameter values (matrix entries) are not directly available. Therefore, in such cases any system identification techniques which can generate a direct estimate of alternative state-space structures as well as of parameters from the system sampled input/output data sequences may be very valuable.

Subspace-based state-space identification methods have recently been proposed as an alternative to the more tradition identification techniques aimed to obtain state-space models and are ones of the main stream of research in system identification today. These methods are attractive not only because of their numerical simplicity and stability, but also for their ability directly estimate state-space representations without using particular canonical (minimal)

* The work has been supported by the Slovak Research and Development Agency under grant APVV-0160-07. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

parameterizations. Some experiences with application of the methods to identification of helicopter dynamics are reported in (Lovera, 2003). The attractiveness is further increased by small number of parameter (essentially only one) to be selected for determining the model structure without any restriction on model generality. The obtained state-space forms are very convenient for solving problems of optimal state estimation (Kalman filtering) and control. See, for instance, (Overschee and Moor, 1994), (Verhaegen, 1994) and (Overschee and Moor, 1996) for further discussion of subspace identification methods features.

In this paper the subspace identification is applied for obtaining an efficient state-space description of laboratory helicopter setup and used for design of multi-input multi-output (MIMO) MPC equipped with integral action and satisfying given amplitude and rate input constraints. The resulting algorithm is implemented by means of Matlab/Simulink Real-Time workshop and xPC Target to perform sufficiently fast real-time computer control of the helicopter physical model constructed in our lab.

2. DESCRIPTION OF THE LABORATORY HELICOPTER MODEL

The laboratory helicopter setup constructed in our lab basically consists of a beam carrying at its ends two propellers - the main and the tail - driven by DC motors, see Figure 1. The beam is jointed to its base with an articulation point which allows the beam to rotate so that its ends move on spherical surfaces. This connection enables two degree of freedom in the helicopter body movement:

- rotation around the horizontal axis → elevation angle ξ;
- rotation around the vertical axis → azimuth angle θ.

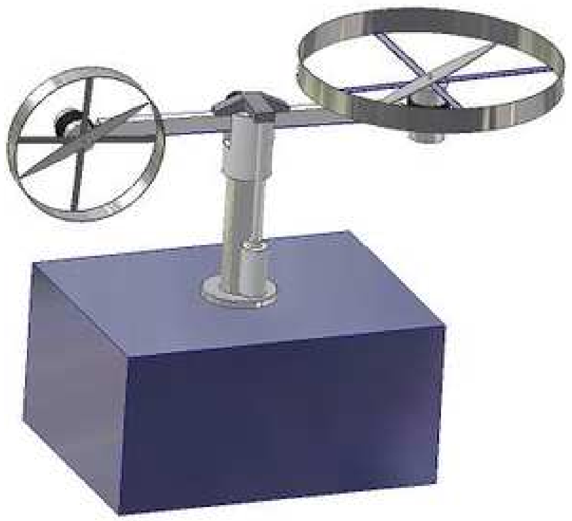


Fig. 1. The laboratory helicopter setup

The axes of the main and the tail propeller and the vertical and the horizontal helicopter axis are perpendicular to each-other so that the movement in the vertical plane and the movement in horizontal plane are each affected by the thrust of only one propeller. A counter-weight fixed to the beam determines a stable equilibrium position. The system is balanced in such a way, that when the motors are switch off, the main propeller end of the beam is lowered. As it is usual in similar small laboratory helicopter setups the control of submitted system is achieved exclusively by controlling speeds of the propellers at fixed angle of attack. The range of the helicopter body rotation (measured by incremental sensors) is ±30 degree in elevation and ±145 degree in azimuth.

The described helicopter physical model can be represented as a non-linear multi-variable system with two inputs (manipulated in the range interface voltage $[0, 10]$ Volts):

- u_1 - voltage driving propeller speed of the main motor;
- u_2 - voltage driving propeller speed of the tail motor;

and two outputs (measured in angular degrees):

- $y_1 = \xi$ - elevation angle;
- $y_2 = \theta$ - azimuth angle.

The interface voltages u_1 and u_2 applied for setting of the helicopter inputs are converted to appropriate voltage values that drive the propeller DC motors. Output ξ denotes the elevation angle, i. e. the angle between the vertical axis and the longitudinal axis of the helicopter body, whereas output θ denotes the azimuth, i. e. the angle in the horizontal plane between the longitudinal axis of the helicopter body and its zero (initial) position.

The voltage driving the main and the voltage driving the tail motor affect both the elevation angle and the azimuth angle, therefore we can say that the interaction make the system multivariable. It is also worth to mention that the used mechanical simplification with fixed angle blades of used propellers does not necessarily translate into simplified dynamics. On the contrary, the input torques and forces are applied via aerodynamical effects, as well as additional coupling effects appearing between the helicopter body and propellers dynamics, due to the reaction forces and torques arising at the acceleration or deceleration of the propellers. These increased couplings effects have important implications on the control dynamics of the helicopter system.

3. THEORETICAL MODELLING

There are two well known approaches to system dynamics modelling: theoretical and experimental. These can be applied to the problem of laboratory helicopter modelling. As mentioned above the helicopter system is a highly non-linear two-inputs two-outputs system, and it is generally a difficult task to develop its exact mathematical model. Different approaches have been used to derive the dynamics of helicopter systems and presented in the literature, see e. g. (Avila-Vilchis et al., 2003), (Huba et al., 2003), (Unneland, 2003) and (Karer and Zupančič, 2006).

Following simplifying modelling assumptions are widely accepted in the literature. First, it is assumed that the dynamics of the propeller subsystems can be described by ordinary differential equations. Further, it is assumed that the friction in the system suspension is of the viscous type. It is also assumed that the propeller - air subsystems can by described in accordance with the postulates of stream flow theory. Exploring the vertical and horizontal plane dynamics, the torques of the propulsion forces of the main and tail propellers, we can arrive to mathematical model of the helicopter set-up in a form of six nonlinear ordinary differential equations, being the state equations, and six algebraic output equations, represented by vector-matrix description:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 \tag{1}$$

$$y = [x_6 x_3]^T \tag{2}$$

with state vector:

$$x = [x_1 x_2 x_3 x_4 x_5 x_6]^T \tag{3}$$

where the matrix and vector functions $f(x), g_1(x)$ and $g_2(x)$ depend on more then 20 various helicopter (design and physical) parameters which have to be determined; x_1 is the angular velocity of the tail propeller, x_2 is the horizontal angular momentum of the beam, x_3 is the azimuth position of the beam, x_4 is the angular velocity of the main propeller, x_5 is the elevation velocity of the beam and x_6 is the elevation angle of the beam. For detail explanation see e.g. (Huba et al., 2003), (Unneland, 2003) and (Gorczyca and Hajduk, 2004).

In order to apply the model structure (1) - (3) for control of a laboratory helicopter all the quantities and involved non-linear static characteristics should be determined first. This is a complicated, time-consuming and iterative process. Therefore it is useful to find some balance between the simplicity and complexity of the modelling procedure according to the purpose of resulting model and operating conditions. Such model has to be clear, concise and flexible for controller design and yet it must consider relevant features of the controlled system. Subspace identification methods appear to have a potential to approach similar goal.

4. EXPERIMENTAL MODELLING

In the past years various time domain methods for identifying dynamic models of the aero-mechanical systems from experimental data have appeared. Much attention have been given to subspace identification methods for identifying state-space models from sampled data records. This section based on the SMI Toolbox of (Haverkamp and Verhaegen, 1997) presents a simple study how to create concise and flexible (linearized) helicopter state space representation suitable for MIMO MPC design.

The identification procedure starts from experiment design with the laboratory helicopter model in order to generate suitable data sets. This incorporates the input signals design for open-loop system excitation near chosen equilibrium point, choice of the sampling rate and experiment duration. Two uncorrelated multi-level pseudo-random signal were applied simultaneously into both inputs for proper excitation of the helicopter outputs with sampling rate 0.1 sec and duration of experiment 400 sec. In Figure 2 and Figure 3 the generated inputs and outputs are shown over 100 sec period on a visual approval.

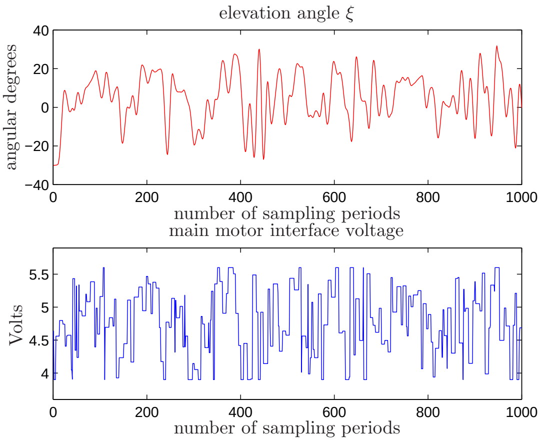


Fig. 2. Experiment design for elevation excitation

After the obtained data pre-processing (detrending, filtering, removing outliers) a model structure has to be chosen. The number of parameters that need to be given to the subspace identification algorithm is reduced to only one, an upper-bound of the expected order of the system. We picked out order 12 for the upper-bound. The state-space model of the system using functions of the SMI Toolbox is estimated within tree steps. In the first step information about the order of the system is extracted from the input-output data using the singular value plot for estimation of the order of the system. In the second step the A and C matrices of the state-space model:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned} \tag{4}$$

are estimated from the compressed data, obtained in the first step. In the third step, the B and D matrices and the initial state are estimated. The singular value plot calculated for our input-output data record is depicted on Figure 4.

From the singular value plot we estimate the system order n to be five, $n = 5$. Next we estimate numerical values of A and C and finally we find B and D using the internal functions of the mentioned toolbox.

At last the obtained model has to be validated. This can by done by comparing the estimated outputs of the model with the true outputs measured on the helicopter system, see Figure 5 for the elevation data and Figure 6 for the azimuth data. In both case we can recognize satisfactory coincidence.

5. MODEL PREDICTIVE CONTROL OF THE HELICOPTER

The field of MPC has been elaborated in detail in the literature and this type of control seems to be very

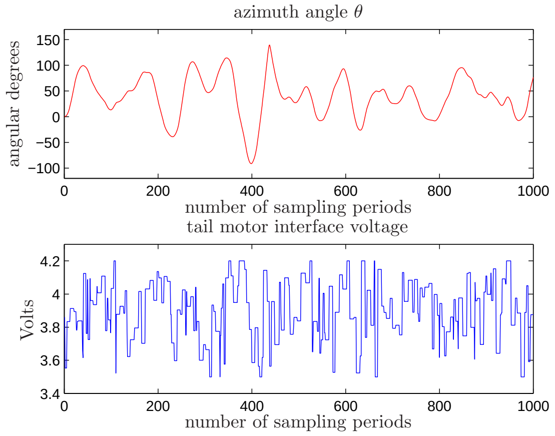


Fig. 3. Experiment design for azimuth excitation

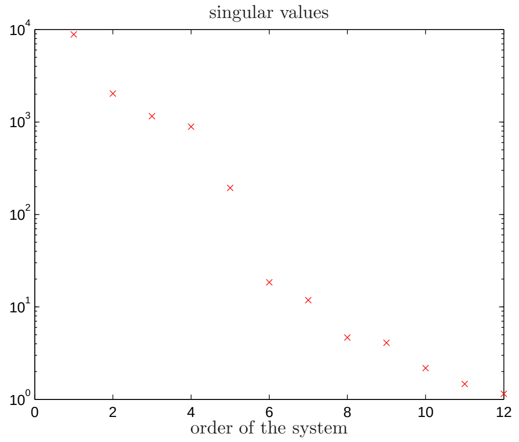


Fig. 4. The singular values plot

attractive for practical control purposes due to ability to meet given process constraints. In this section based on the dual-mode approach and closed-loop paradigm a design procedure of guaranteed stability two-inputs two-outputs MPC controller with internal action intended for the helicopter model stabilization and tracking of reference values is briefly outlined.

We want to regulate the discrete-time linear time invariant system (4) (considering $D = 0$) to the origin. MPC solves such a problem in following way. Given the LTI system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{5}$$

and performance index:

$$J(u) = \sum_{i=1}^{\infty} [x_i^T Q x_i \; + \; u_i^T R u_i] \tag{6}$$

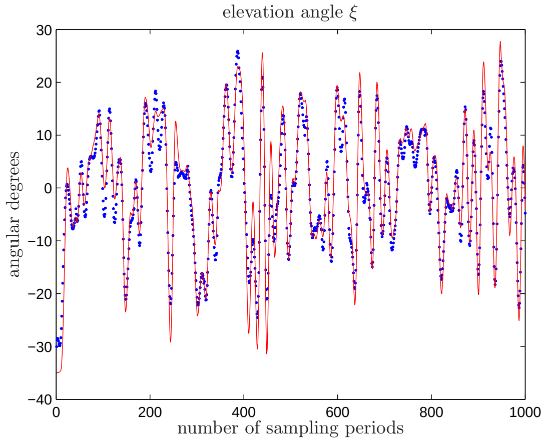


Fig. 5. Validation of the elevation data (full line - true data, dot line - model data)

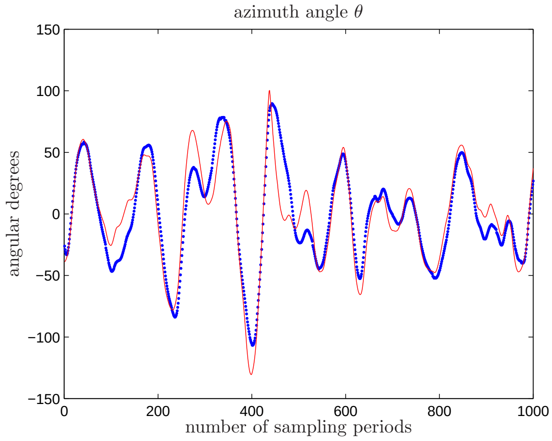


Fig. 6. Validation of the azimuth data (full line - true data, dot line - model data)

where Q and R are the state and input weighting matrices, and

$$Q \succeq 0, R \succ 0,$$

the aim is to find the input trajectory u_k defined as (closed loop paradigm):

$$u_{k+i} = \begin{cases} Kx_{k+i} + c_{k+i}, & i = 0, 1, \ldots, N-1 \\ Kx_{k+i}, & i \geq N \end{cases} \tag{7}$$

(where K is LQ optimal feedback gain and N is finite prediction horizon) in such a way that the performance index (6) reaches its minimum subject to following amplitude and rate input signal constraints:

$$\begin{array}{rcccl} \underline{u} & \leq & u_k & \leq & \overline{u} \\ -\Delta\overline{u} & \leq & \Delta u_k & \leq & \Delta\overline{u} \end{array} \tag{8}$$

After substitution (7) to (5) and applying autonomous augmented state-space description, see (Kouvaritakis et al., 2000a), we can obtain the performance index (6) in the following more efficient form for real-time calculations:

$$J = f_k^T S f_k \tag{9}$$

where:

$$f_k = \left[\, c_k^T \;\; c_{k+1}^T \;\; \ldots \;\; c_{k+N-1}^T \,\right]^T \tag{10}$$

and S is a positive definite matrix. Then using the technique elaborated in (Kouvaritakis et al., 2004) it is a simple way to introduce integral action to resulting control law for improving steady-state performance and elimination of disturbances. All the computational burden reduces to a QP procedure for minimization of (9) subject to constraints (8). The state variables $x(k)$ in (7) usually can not be measured and need to be estimated. For this purpose a discrete-time steady state Kalman filter is further used.

6. IMPLEMENTATION

In this section some details of the implementation of the MPC controller (7) and simulations are presented.

From the hardware and software point of view the MPC controller applied to our laboratory helicopter setup was implemented using Matlab/Simulink Real-Time Workshop and xPC Target. For interaction with the helicopter model an Advantech I/O-cards with AD and DA convertors were used. The target machine was a pentium II with CPU running at 350 MHz. The overall realization of the helicopter control, host and target machines are shown in Figure 7.

The main control task during experiments was tracking and stabilization of the helicopter output $y(k) = [\xi(k)\ \theta(k)]^T$ at in advance given reference values y_r. After proper tuning of control horizon N ($N = 25$) the obtained typical results are depicted at Figure 8 for tracking desired reference values for the elevation and at Figure 9 for tracking desired reference values for the azimuth. The simulations was performed with input constraints $\underline{u} = 3$, $\overline{u} = 8$, $\Delta\overline{u} = 0.15$ and for sampling period 0.1 seconde.

7. CONCLUSIONS

The applicability of the suggested MPC technique for control of the laboratory helicopter set-up based on subspace identification methods was evaluated through real-time experiments. The presented results are only the first outputs from the research initiated in this direction.

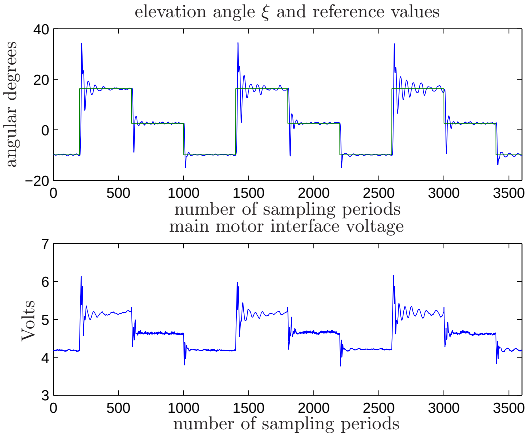


Fig. 8. Tracking of reference values for elevation

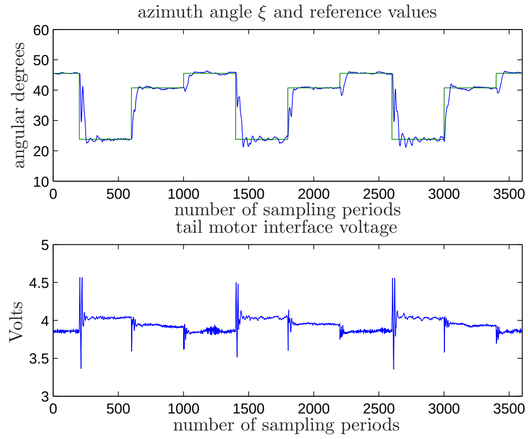


Fig. 9. Tracking of reference values for azimuth

REFERENCES

Avila-Vilchis, J., B.Brogliato, A.Dzul, and R.Lozano (2003). Nonlinear modeling and control of helicopters. *Automatica*, 39(9), 1583–1596.

Gorczyca, P. and Hajduk, K. (2004). Tracking control algorithms for a laboratory aerodynamical system. *Int. J. Appl. Math. Comput. Sci.*, 14(4), 469–475.

Haverkamp, B. and Verhaegen, M. (1997). SMI Toolbox: State space model identification software for multivariable dynamical systems. Technical report, TU Delft, The Netherlands. Version 1.0.

Huba, M., Karas, M., and Veličko, V. (2003). The helicopter rack control. In *The 11th Mediterranean Conference on Control and Automation, MED'03*, paper T7–094. Rhodes, Greece.

Karer, G. and Zupančič, B. (2006). Modelling and identification of a laboratory helicopter. In *5th MATHMOD Vienna*. Austria.

Kouvaritakis, B., Cannon, M., Rohaľ-Ilkiv, B., Karas, A., and Belavý, C. (2004). Asymmetric constraints with polyhedral sets in MPC with application to coupled tanks system. *International Journal of Robust and Nonlinear Control*, 14, 641–353.

Kouvaritakis, B., Lee, Y.I., Tortora, G., and Cannon, M. (2000a). MPC stability constraints and their implementations. In Š. Kozák and M. Huba (eds.), *Preprints from the IFAC Conference Control System Design*, 205–212. Slovak University of Technology, Bratislava, Slovak

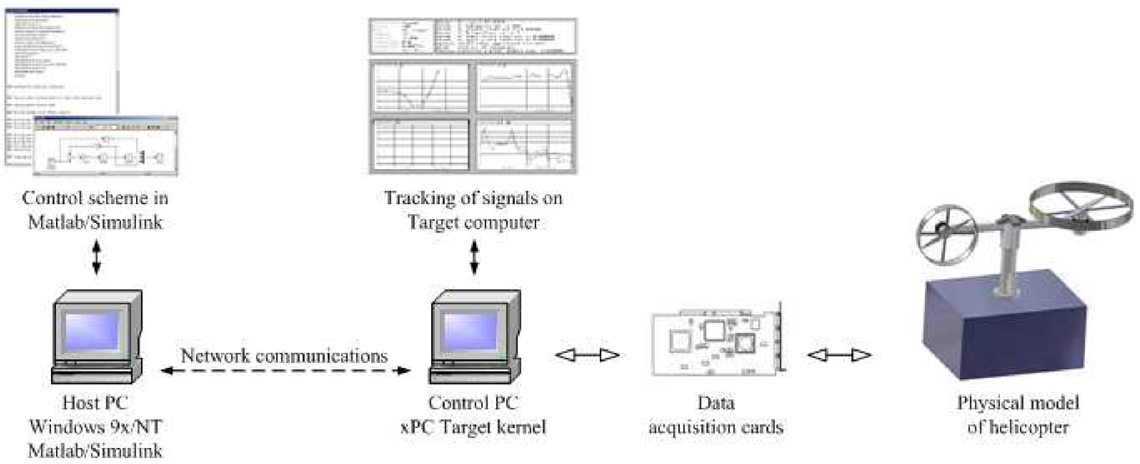


Fig. 7. The helicopter hardware and software implementation

republic.

Kouvaritakis, B., Rossiter, J.A., and Schuurmans, J. (2000b). Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8), 1545–1549.

Lovera, M. (2003). Identification of MIMO state space models for helicopter dynamics. In *Proceedings of the 13th IFAC Symposium on System Identification*, 1387–1392. Rotterdam, Netherlands.

Maciejowski, J.M. (2002). *Predictive Control with Contraints.* Prentice-Hall, Harlow, UK.

Morari, M. and Lee, J.H. (1997). Model predictive control: Past, present and future. In *6th International Symposium on Process Systems Engineering and 30th European Symposium on Computer Aided Process Engineering.* Trondheim, Norway.

Overschee, P.V. and Moor, B.D. (1994). N4SID: Two subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1), 75–93.

Overschee, P.V. and Moor, B.D. (1996). *Subspace Identification for Linear Systems: Theory - Implementation - Applications.* Kluwer Academic Publisher, Dordrecht.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control engineering Practice*, 11, 733–764.

Rossiter, J.A. (2004). *Model-based predictive control: a practical approach.* CRC Press LLC, Boca Raton, FL.

Unneland, K. (2003). *Application of model predictive control to a helicopter model.* Master's thesis, NTNTU and ETH, Automatic Control Laboratory, ETHZ.

Verhaegen, M. (1994). Identification of the deterministic part of mimo state space models given in innovations form from input-output data. *Automatica*, 30(1), 61–74.

Wang, L. (2009). *Model Predictive Control System Design and Implementation Using Matlab.* Springer, New York.

Comments – Remarks

Institute of Control and Industrial Informatics, FEEIT STU in Bratislava

Time Sub-Optimal Control of Three-Level Hydraulic System

Pavol Bisták*

**Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia, (e-mail: pavol.bistak@stuba.sk)*

Abstract: The paper presents the solution of the time sub-optimal control applied for the three-level hydraulic system. The control design has been originally developed for the triple integrator system. It is based on switching surfaces. The high sensitivity of the time optimal controllers is well-known and the proposed sub-optimal controller tries to reduce it by smooth change of control values between the control limits. The changes have an exponential behavior. New parameters introduced during the design correspond in linear cases to the poles of the closed loop system. They enable to tune the control changes. Using the exact linearization method the controller developed for the triple integrator has been applied to the nonlinear three-level hydraulic system.

Keywords: time optimal control, input constraints, smooth switching, exact linearization, nonlinear system.

1. INTRODUCTION

The time optimal control belongs to one of the most important control strategies. It was heavily studied in 50-ties and 60-ties of the previous century (Athans and Falb, 1996) but due to its high sensitivity to unmodelled dynamics, parametric variations, disturbances and noise it was later suppressed by the pole assignment control. Nowadays there exist several strategies how to cope with this problem and so the time optimal control plays an important role in the modern control theory.

Generally, the time optimal problem can be solved by computation of switching surfaces. These can be derived using the Pontryagin's maximum principle. A different approach is offered by dynamic programming based on the Bellman's optimality principle. Another way was presented by Pavlov who solved switching surfaces from phase trajectories (Pavlov, 1966). But switching surfaces can be also expressed by the set of algebraic equations (Walther et al., 2001) that results from the time solution in the phase space. For higher order systems it can be rather complicated to find the exact solution. In this paper we will apply the set of algebraic equations extended by additional parameters to the simple third order system represented by the triple integrator.

The controller designed in this paper tries to combine qualities of both above mentioned approaches (i.e., the time optimal and pole assignment control) (Huba, 2006). Although the time optimal control belongs to the nonlinear class of controllers there is a set of coefficients introduced in this paper that can be in the linear case identified with the set of poles of the closed loop system. In the nonlinear case these coefficients specify exponential changes from one limit control value to the opposite one. Of course, including the additional parameters complicates the set of algebraic equations and resulting switching surfaces. The control law is then derived in relation with the measured position of the current state with respect to the designed switching surfaces.

In this paper we aim to show that the controller that has been originally designed for the linear system is suitable also for control of nonlinear systems. The designed controller will be applied to the tree-level hydraulic system and will be compared with classical approaches as pole assignment control and exact linearization method.

The paper is organized in nine chapters. After introduction and problem statement chapters there is the main chapter where the design of the sub-optimal controller is described in details. This chapter discusses the nonlinear dynamics decomposition and regions of the switching surface. There is a corresponding control law derived for each region. The fourth chapter shows time responses of the designed controller and compares it to the time optimal controller. The fifth chapter describes the three-level hydraulic system. The sixth and seventh chapters deal with linear pole assignment and exact linearization control methods. The eighth chapter shows the application of derived controller to the nonlinear hydraulic system and the paper is finished with short conclusions.

2. PROBLEM STATMENT

Let us consider the linear system given in the state space

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x}+\mathbf{b}u\\ y &= \mathbf{c}^t\mathbf{x}\end{aligned} \qquad \mathbf{A}=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix} \qquad \mathbf{b}=\begin{bmatrix}0\\0\\1\end{bmatrix} \qquad \mathbf{c}^t=\begin{bmatrix}1&0&0\end{bmatrix} \tag{1}$$

that represents the triple integrator. The control input signal is constrained

$$u=\langle U_1 \quad U_2\rangle \tag{2}$$

The task is to design the time sub-optimal controller what means to drive the system from an initial state

$\mathbf{x} = [x \quad y \quad z]^t$ to the desired state $\mathbf{x}_w$ in a minimum time $t_{\min}$ under the additional condition that limits the changes of the control action between two opposite values. When it is required that these changes should have an exponential behavior the additional condition can be expressed by a scalar function $\varsigma_i : R^n \to R$ representing the distance of the current state $\mathbf{x}$ from the switching surface (curve, point) and it holds

$$\frac{d\varsigma_i}{dt} = \alpha_i \varsigma_i \ , \ \alpha_i \in R^- \ , \ i = 1,2,3 \tag{3}$$

For the sake of simplicity we should admit that using a coordinate transformation it is always possible to set the desired state equal to the origin $\mathbf{x}_w = \mathbf{0}$.

The coordinate transformation together with the exact linearization method will be used when we will try to apply the designed controller to the nonlinear system represented by the three-level hydraulic plant.

3. CONTROLLER DESIGN

It is well known that minimum time optimal control with saturated input leads to the control action with at most n intervals switching between limit values where n represents the order of the system. Usually the control algorithm results in deriving switching surfaces as functions of states which signs determine the switching times. It can be very hard task to express these functions exactly and there is no general solution for higher order systems ($n > 3$). Bang-bang control in practice is not desirable because of chattering and noise effects but there are techniques have to cope with them (Pao and Franklin, 1993, Bistak et al. 2005).

The presented sub-optimal controller design belongs to one of them. This time the control action will not be calculated as the sign of the switching surface but will result from (3). If we apply the condition (3) also for the switching curve and switching point this will influence the construction of the switching surface itself. We will explain it with the help of a state vector nonlinear decomposition.

3.1 Nonlinear Decomposition

Let us consider ordered coefficients

$$\alpha_3 < \alpha_2 < \alpha_1 < 0 \tag{4}$$

Then the eigenvectors

$$\mathbf{v}_i = [\alpha_i \mathbf{I} - \mathbf{A}]^{-1}\mathbf{b} = \begin{bmatrix} \frac{1}{\alpha_i^3} & \frac{1}{\alpha_i^2} & \frac{1}{\alpha_i} \end{bmatrix}^t \tag{5}$$

form a base of the state space. In the linear case any point of the state space can be expressed as

$$\mathbf{x} = q_1\mathbf{v}_1 + q_2\mathbf{v}_2 + q_3\mathbf{v}_3 \ , \ q_1, q_2, q_3 \in R \tag{6}$$

Because the control signal is limited only the points where $u = \sum_{i=1}^{3} q_i$ fulfils (2) are covered by (6). In order to express the whole space we have to introduce the nonlinear decomposition of the state

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 \tag{7}$$

where each mode $\mathbf{x}_i$

$$\mathbf{x}_i = e^{\mathbf{A}t_i} q_i \mathbf{v}_i + \int_0^{t_i} e^{\mathbf{A}(t-\tau)} \mathbf{b} q_i(\tau) d\tau \tag{8}$$

consists of a linear part given by the parameter q_i

$$q_i \in \left\langle U_1 - \sum_{k=1}^{i-1} q_k \quad U_2 - \sum_{k=1}^{i-1} q_k \right\rangle \quad when \quad t_i = 0 \tag{9}$$

and a nonlinear part specified by the parameter t_i

$$0 < t_i < t_{i-1} \quad when \quad q_i = U_{3-j} - \sum_{k=1}^{i-1} q_k \ , \ t_0 = \infty \ , \ j = 1,2 \tag{10}$$

After substituting (1) and (5) into (8) one gets

$$\mathbf{x}_i = \begin{bmatrix} \frac{1}{\alpha_i^3} - \frac{t_i}{\alpha_i^2} + \frac{t_i^2}{2\alpha_i} - \frac{t_i^3}{6} \\ \frac{1}{\alpha_i^2} - \frac{t_i}{\alpha_i} + \frac{t_i^2}{2} \\ \frac{1}{\alpha_i} - t_i \end{bmatrix} q_i \tag{11}$$

If we take the subsystem $\mathbf{x}_1$ it represents a one-dimensional variety (Cox et al., 2007) that corresponds to the switching curve. Points from the linear part of $\mathbf{x}_1$ where (9) is fulfilled satisfy (3), i.e. they are decreasing the distance ς_1 from the origin. In this case the system is moving along the line. The other points of the subsystem $\mathbf{x}_1$ given by (10) could not fulfill (3) because of the limited control value (2). They are approaching the linear part of $\mathbf{x}_1$ with the limit control value so they are moving along the trajectory in the form of a curve. In this case (3) is superimposed by (2).

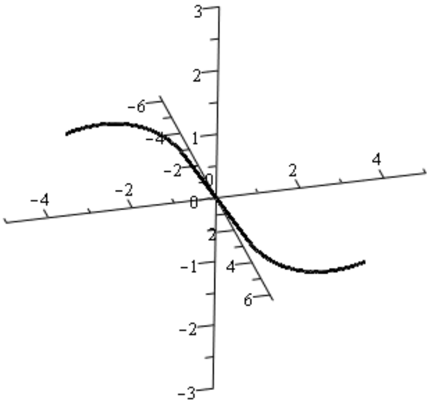


Fig. 1. Subsystem $\mathbf{x}_1$ representing the switching curve

Similarly we can create a two-dimensional variety that will express the switching surface. We simply add to the subsystem $\mathbf{x}_1$ the subsystem $\mathbf{x}_2$. The points of the subsystem $\mathbf{x}_1$ become the target points for the second subsystem $\mathbf{x}_2$.

$$\mathbf{x}_{12} = \mathbf{x}_1 + \mathbf{x}_2 \tag{12}$$

This time we define the distance ς_2 in the direction of the second eigenvector $\mathbf{v}_2$. The points of the $\mathbf{x}_2$ try to reach the $\mathbf{x}_1$ points according to (3) if it does not break (2). Otherwise they are moving with the limit control value q_2 given by (10).

Again there exist a linear and a nonlinear parts of the subsystem $\mathbf{x}_2$. In combination with the previous subsystem $\mathbf{x}_1$ we get four possibilities, i.e. four regions of the switching surface with respect to the limit and nonzero values of q_1, q_2, t_1, t_2. If we take into account the parameter $j = 1,2$ the number of the regions doubles. Later on we will describe these regions in details and derive for each of them the corresponding control value.

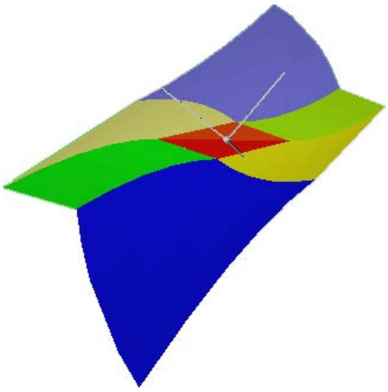


Fig. 2. Regions of the switching surface

To cover the whole space we should realize also the third subsystem $\mathbf{x}_3$ but in the presented control algorithm design it is not necessary. To give reasonable results that can be applied in real time applications we simplified the third subsystem to following one

$$\mathbf{x}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} q_3 \quad , \quad q_3 \in R \tag{13}$$

It represents the unit vector in the direction of the x-axis multiplied by the quotient q_3. Thus the quotient q_3 expresses the distance ς_3 between the current state and the switching surface that is measured in the direction of the x-axis. This simplification enables easier to localize the initial state with respect to the regions of the switching surface because it represents the projection of the switching surface to the (y,z)-plane where the borders between regions are parabolic curves or lines.

In (Ťapák et al., 2006) one can find the solution of the control algorithm when the third subsystem was given by the third eigenvector $\mathbf{v}_3$ multiplied by the quotient q_3

$$\mathbf{x}_3 = q_3 \mathbf{v}_3, q_3 \in R \tag{14}$$

but this was not in the form suitable for real time systems.

After completely decomposing the system to the three subsystems (7) it is necessary to derive the formula of the corresponding region of the switching surface. This comes from the set of equations (12) when the parameters t_i or q_i are evaluated from the last two equations and replaced in the first one. Then one gets the formula for the corresponding switching surface in the form

$$x = f(y,z) \ ; \ f : R^2 \to R \tag{15}$$

Now the resulting control value can be computed from (3) when we realize that the distance $\varsigma_3 = q_3$ can be expressed as the difference between the x-coordinate of the initial point and the x-coordinate of the switching surface given by (15) in the form $f(y,z)$

$$\varsigma_3 = x - f(y,z) \tag{16}$$

After substituting (16) into (3) and taking into account (1) it results in

$$\frac{d\varsigma_3}{dt} = \frac{dx}{dt} - \frac{df(y,z)}{dt} = y - \frac{df(y,z)}{dy} z - \frac{df(y,z)}{dz} u = \\ = \alpha_3 (x - f(y,z)) \tag{17}$$

Finally the control value u can be isolated

$$u = \frac{-\alpha_3 (x - f(y,z)) + y - \dfrac{df(y,z)}{dy} z}{\dfrac{df(y,z)}{dz}} \tag{18}$$

The resulting control value u must be limited by (2).

As one can see from (18) the only one term not evaluated yet is $f(y,z)$ representing the switching trajectory. Because it differs according to the regions of the switching trajectory we will evaluate it individually.

3.2 Control for Region QQ

The region QQ denotes the subset of (12) where both subsystems $\mathbf{x}_1$ and $\mathbf{x}_2$ are in the linear cases, i.e. (9) is fulfilled for q_1 and q_2. The parameters q_1 and q_2 can be evaluated from the last two equations of the set (12). After using (11) and substituting $t_1 = t_2 = 0$ into (12) one gets

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dfrac{q_1}{\alpha_1^3} + \dfrac{q_2}{\alpha_2^3} \\ \dfrac{q_1}{\alpha_1^2} + \dfrac{q_2}{\alpha_2^2} \\ \dfrac{q_1}{\alpha_1} + \dfrac{q_2}{\alpha_2} \end{bmatrix} \tag{19}$$

And for parameters q_1 and q_2 it yields

$$q_1 = -\frac{\alpha_1^2(z - y\alpha_2)}{\alpha_2 - \alpha_1} \tag{20}$$

$$q_2 = \frac{\alpha_2^2(-\alpha_1 y + z)}{\alpha_2 - \alpha_1} \tag{21}$$

By the substitution of (20) and (21) into the first equation of (19) we derive the analytical expression for the region QQ

$$x = f(y,z) = -\frac{z - y\alpha_2}{\alpha_1(\alpha_2 - \alpha_1)} + \frac{-\alpha_1 y + z}{\alpha_2(\alpha_2 - \alpha_1)} \tag{22}$$

According to (18) the control value u results in the form

$$\begin{aligned} u = -y\alpha_1\alpha_2 + \alpha_2 z + \alpha_1 z + \alpha_3 x\alpha_1\alpha_2 \\ - \alpha_3 y\alpha_2 - \alpha_3\alpha_1 y + \alpha_3 z \end{aligned} \tag{23}$$

This is the well-known linear pole assignment controller for the triple integrator.

3.3 Control for Region TQ

By TQ we denote the region of the switching surface when the subsystem $\mathbf{x}_1$ is in the linear cases, i.e. for its states (9) is valid and the subsystem $\mathbf{x}_2$ is in the nonlinear cases, i.e. (10) comes true. The procedure how to derive the control value is very similar to that one performed in the previous region QQ. First we express (12) when $q_1 = U_j$ and $t_2 = 0$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} U_j\left(\frac{1}{\alpha_1^3} - \frac{t_1}{\alpha_1^2} + \frac{1}{2}\frac{t_1^2}{\alpha_1}\right) - \frac{1}{6}U_j t_1^3 + \frac{q_2}{\alpha_2^3} \\ U_j\left(\frac{1}{\alpha_1^2} - \frac{t_1}{\alpha_1}\right) + \frac{1}{2}U_j t_1^2 + \frac{q_2}{\alpha_2^2} \\ \frac{U_j}{\alpha_1} - U_j t_1 + \frac{q_2}{\alpha_2} \end{bmatrix} \tag{24}$$

For this and following calculations we have used the Maple computer algebra system and because of the complexity of several expressions we have used the Maple outputs.

Now it is necessary to solve the last two equations of the set (24). The difference consists in that the second equation of the set (24) is now the quadratic equation. From its two solutions we have chosen such one that assures the positive value of t_2. Then after introducing the notation for the discriminant *DTQ*

$$DQT = -2\,\alpha_1^2\,U_j\,z\,\alpha_2 + \alpha_1^2\,U_j^2 - U_j^2\,\alpha_2^2 + 2\,y\,\alpha_1^2\,\alpha_2^2\,U_j \tag{25}$$

the parameters q_2 and t_1 it can be expressed

$$q_2 = \frac{z\,\alpha_1\,\alpha_2 - \alpha_1\,U_j + sign(U_j)\sqrt{DTQ}}{(U_{3-j} - U_j)\,\alpha_1} \tag{26}$$

$$t_1 = -\frac{-U_j\,\alpha_2 + \alpha_1\,U_j - sign(U_j)\sqrt{DTQ}}{U_j\,\alpha_1\,\alpha_2} \tag{27}$$

Again we substitute (26) and (27) into the first equation of (24) and get the expression for the region TQ

$$\begin{aligned} x = \frac{1}{6}\,\frac{1}{U_j^2\,\alpha_1^3\,\alpha_2^3}\Big(&2\,U_j^3\,\alpha_2^3 + 3\,U_j^3\,\alpha_2^2\,\alpha_1 - 3\,U_j^2 \\ &\alpha_2^2\,sign(U_j)\sqrt{DTQ} - 5\,\alpha_1^3\,U_j^3 + 3\,\alpha_1^2\,U_j^2\,sign(U_j)\sqrt{DTQ} \\ &+ 3\,\alpha_1\,U_j\,sign(U_j)^2 DTQ - sign(U_j)\,DTQ^{3/2} + 6\,\alpha_1^3\,U_j^2\,z\,\alpha_2\Big) \end{aligned} \tag{28}$$

From (18) the control value u is

$$\begin{aligned} u = \frac{1}{3}\,\frac{1}{\alpha_2^2\,\alpha_1^4\,(-y\,\alpha_2 + z)}\Big(&-3\,\alpha_2^4\,\alpha_1^4\,z\,y + 2\,\alpha_3\,U_j^2\,\alpha_2^2\,\alpha_1^2 - \alpha_3\,U_j \\ &\alpha_2^3\,\alpha_1^2\,z + \alpha_3\,U_j\,\alpha_2^4\,y\,\alpha_1^2 + \alpha_3\,\alpha_1^4\,U_j\,z\,\alpha_2 - \alpha_3\,\alpha_1^4\,U_j\,y\,\alpha_2^2 + 2\,\alpha_3 \\ &\alpha_1^4\,z^2\,\alpha_2^2 - \alpha_3\,U_j^2\,\alpha_2^4 - \alpha_3\,\alpha_1^4\,U_j^2 - 3\,y\,sign(U_j)\sqrt{DTQ}\,\alpha_1^3\,\alpha_2^3 \\ &- \alpha_3\,sign(U_j)\sqrt{DTQ}\,U_j\,\alpha_2^3 + \alpha_3\,sign(U_j)\sqrt{DTQ}\,\alpha_1^3\,U_j \\ &- 4\,\alpha_3\,\alpha_1^4\,z\,\alpha_2^3\,y + 2\,\alpha_3\,y^2\,\alpha_1^4\,\alpha_2^4 + 3\,\alpha_2^2\,\alpha_1^3\,z\,sign(U_j)\sqrt{DTQ} \\ &- 3\,\alpha_3\,sign(U_j)\sqrt{DTQ}\,\alpha_1^3\,y\,\alpha_2^2 + 3\,\alpha_3\,sign(U_j)\sqrt{DTQ}\,x\,\alpha_1^3 \\ &\alpha_2^3 + 3\,\alpha_2^3\,\alpha_1^4\,z^2\Big) \end{aligned} \tag{29}$$

3.4 Control for Region TT

The region TT denotes the subset of (12) where both subsystems $\mathbf{x}_1$ and $\mathbf{x}_2$ are in the nonlinear cases, i.e. (10) is fulfilled for t_1 and t_2. Again the parameters t_1 and t_2 can be computed from the last two equations of the (12). This time we substitute $q_1 = U_j$ and $q_2 = U_{3-j} - U_j$ into (12)

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \Bigg[\Bigg[&U_j\left(\frac{1}{\alpha_1^3} - \frac{t_1}{\alpha_1^2} + \frac{1}{2}\frac{t_1^2}{\alpha_1}\right) - \frac{1}{6}U_j t_1^3 + (U_{3-j} \\ &- U_j)\left(\frac{1}{\alpha_2^3} - \frac{t_2}{\alpha_2^2} + \frac{1}{2}\frac{t_2^2}{\alpha_2}\right) - \left(\frac{1}{6}U_{3-j} - \frac{1}{6}U_j\right)t_2^3\Bigg], \\ \Bigg[&U_j\left(\frac{1}{\alpha_1^2} - \frac{t_1}{\alpha_1}\right) + \frac{1}{2}U_j t_1^2 + (U_{3-j} - U_j)\left(\frac{1}{\alpha_2^2} - \frac{t_2}{\alpha_2}\right) \\ &+ \left(\frac{1}{2}U_{3-j} - \frac{1}{2}U_j\right)t_2^2\Bigg], \\ \Bigg[&\frac{U_j}{\alpha_1} - U_j t_1 + \frac{U_{3-j} - U_j}{\alpha_2} - (U_{3-j} - U_j)\,t_2\Bigg]\Bigg] \end{aligned} \tag{30}$$

and when solving the last two equations again the criterion for the choice of the right solution is that the times t_1 and t_2 must be positive.

$$\begin{aligned} DTT = &-\alpha_2^2\,\alpha_1^2\,U_{3-j}\,z^2\,U_j + \alpha_2^2\,U_j^2\,z^2\,\alpha_1^2 - \alpha_1^2\,U_{3-j}^3\,U_j + 2\,U_j^2\,\alpha_1^2 \\ &U_{3-j}^2 + 2\,U_{3-j}^2\,y\,\alpha_1^2\,\alpha_2^2\,U_j - U_{3-j}^2\,U_j^2\,\alpha_2^2 - 2\,U_{3-j}\,U_j^2\,y\,\alpha_1^2 \\ &\alpha_2^2 + U_{3-j}\,U_j^3\,\alpha_2^2 - U_{3-j}\,U_j^3\,\alpha_1^2 \end{aligned} \tag{31}$$

$$t_1 = \frac{-U_j\,U_{3-j}\,\alpha_2 + \alpha_1\,U_j\,z\,\alpha_2 + \sqrt{DTT}}{U_j\,\alpha_1\,U_{3-j}\,\alpha_2} \tag{32}$$

$$\begin{aligned} t_2 \\ = \frac{1}{\alpha_2\,\alpha_1\,U_{3-j}\,(U_{3-j} - U_j)}\Big(&\alpha_1\,U_{3-j}\,z\,\alpha_2 - \alpha_1\,U_j\,z\,\alpha_2 \\ &+ U_j\,\alpha_1\,U_{3-j} - \alpha_1\,U_{3-j}^2 - \sqrt{DTT}\Big) \end{aligned} \tag{33}$$

After using (32) and (33) in the first equation of the set (30) the points of the region TT can be expressed

$$\begin{aligned}
x = \frac{1}{6}\,&\frac{1}{U_{3-j}^2\,\alpha_2^3\,\alpha_1^3\,U_j^2\,(-U_{3-j}+U_j)^2}\Big(-2\,\alpha_2^3\,U_j^3\,z^3\,\alpha_1^3\,U_{3-j}+\\
&\alpha_2^3\,U_j^4\,z^3\,\alpha_1^3+3\,\alpha_2^3\,U_j^3\,U_{3-j}^3\,z\,\alpha_1-6\,\alpha_2^3\,U_j^4\,U_{3-j}^2\,z\,\alpha_1+3\,\alpha_2^3\\
&U_j^5\,U_{3-j}\,z\,\alpha_1-3\,U_j^5\,\alpha_2\,z\,\alpha_1^3\,U_{3-j}+U_j^2\,\alpha_2^3\,\alpha_1^3\,U_{3-j}^2\,z^3+3\\
&U_j^2\,\alpha_2\,\alpha_1^3\,U_{3-j}^4\,z-9\,U_j^3\,\alpha_2\,\alpha_1^3\,U_{3-j}^3\,z+9\,U_j^4\,\alpha_2\,\alpha_1^3\,U_{3-j}^2\,z\\
&+3\,\alpha_2\,U_j\,z\,\alpha_1\,DTT\,U_{3-j}-3\,\alpha_2\,U_j^2\,z\,\alpha_1\,DTT+DTT^{3/2}\,U_{3-j}\\
&-2\,DTT^{3/2}\,U_j+2\,\alpha_2^3\,U_j^3\,U_{3-j}^4-4\,\alpha_2^3\,U_j^4\,U_{3-j}^3+2\,\alpha_2^3\,U_j^5\\
&U_{3-j}^2-6\,U_j^3\,\alpha_1^3\,U_{3-j}^4+6\,U_j^4\,\alpha_1^3\,U_{3-j}^3+2\,U_j^2\,\alpha_1^3\,U_{3-j}^5-2\\
&U_j^5\,\alpha_1^3\,U_{3-j}^2+3\,\alpha_2^2\,U_j^4\,U_{3-j}\sqrt{DTT}-3\,U_j^2\,\alpha_1^2\,U_{3-j}^3\sqrt{DTT}\\
&+6\,U_j^3\,\alpha_1^2\,U_{3-j}^2\sqrt{DTT}-3\,U_j^4\,\alpha_1^2\,U_{3-j}\sqrt{DTT}+3\,\alpha_2^2\,U_j^2\\
&U_{3-j}^3\sqrt{DTT}-6\,\alpha_2^2\,U_j^3\,U_{3-j}^2\sqrt{DTT}\Big)
\end{aligned}\tag{34}$$

In this case the resulting control value resulting from (18) is the most complicated one

$$\begin{aligned}
u = \frac{1}{3}\,&\Big(-6\,U_{3-j}^2\,\alpha_3\sqrt{DTT}\,x\,\alpha_2^3\,\alpha_1^3-4\,U_{3-j}^2\,\alpha_3\,\alpha_2^4\,z^2\,\alpha_1^4\,y+12\\
&U_{3-j}^2\,U_j\,\alpha_1^4\,\alpha_2^4\,z\,y+6\,U_{3-j}^3\,\alpha_3\,U_j\,\alpha_1^4\,y\,\alpha_2^2+2\,U_{3-j}^3\,\alpha_3\,U_j\,\alpha_2^4\,y\\
&\alpha_1^2-U_{3-j}^2\,\alpha_3\,\alpha_2^4\,U_j\,z^2\,\alpha_1^2+2\,U_{3-j}^2\,\alpha_3\,U_j^2\,\alpha_2^4\,y\,\alpha_1^2-8\\
&U_{3-j}^2\,\alpha_3\,U_j\,y^2\,\alpha_1^4\,\alpha_2^4-3\,U_{3-j}^2\,\alpha_3\,U_j\,\alpha_2^2\,z^2\,\alpha_1^4-2\,U_{3-j}^2\,\alpha_3\,U_j^2\\
&\alpha_1^4\,y\,\alpha_2^2+U_{3-j}\,\alpha_3\,U_j^2\,\alpha_2^2\,z^2\,\alpha_1^4-U_{3-j}\,\alpha_3\,U_j^2\,\alpha_2^4\,z^2\,\alpha_1^2+\alpha_3\\
&\alpha_1^4\,U_{3-j}^5-2\,\alpha_3\sqrt{DTT}\,\alpha_2^3\,z^3\,\alpha_1^3-2\,\alpha_3\,U_j\,\alpha_2^4\,z^4\,\alpha_1^4+6\,U_{3-j}^2\\
&\alpha_1^3\,\alpha_2^3\,y\sqrt{DTT}-6\,U_{3-j}\,\alpha_1^3\,\alpha_2^3\,z^2\sqrt{DTT}-6\,U_{3-j}^3\,\alpha_1^4\,\alpha_2^4\,z\,y\\
&-4\,U_{3-j}^4\,\alpha_3\,\alpha_1^4\,y\,\alpha_2^2+2\,U_{3-j}^3\,\alpha_3\,\alpha_2^2\,z^2\,\alpha_1^4+4\,U_{3-j}^3\,\alpha_3\,y^2\,\alpha_1^4\\
&\alpha_2^4+U_{3-j}\,\alpha_3\,\alpha_2^4\,z^4\,\alpha_1^4-6\,U_{3-j}^3\,U_j\,\alpha_1^4\,\alpha_2^2\,z+3\,U_{3-j}^2\,U_j^2\,\alpha_1^4\\
&\alpha_2^2\,z-3\,U_{3-j}^2\,U_j^2\,\alpha_1^2\,\alpha_2^4\,z-6\,U_{3-j}\,U_j\,\alpha_1^4\,\alpha_2^4\,z^3-2\\
&U_{3-j}^2\,\alpha_3\sqrt{DTT}\,U_j\,\alpha_1^3+2\,U_{3-j}^2\,\alpha_3\sqrt{DTT}\,U_j\,\alpha_2^3-\\
&U_{3-j}^4\,\alpha_3\,U_j\,\alpha_1^2\,\alpha_2^2+3\,U_{3-j}^3\,\alpha_3\,U_j^2\,\alpha_1^2\,\alpha_2^2-2\,U_{3-j}^2\,\alpha_3\,U_j^3\,\alpha_1^2\,\alpha_2^2\\
&+2\,U_{3-j}^3\,\alpha_3\sqrt{DTT}\,\alpha_1^3-U_{3-j}^4\,\alpha_3\,U_j\,\alpha_1^4-U_{3-j}^3\,\alpha_3\,U_j^2\,\alpha_1^4+\\
&U_{3-j}^2\,\alpha_3\,U_j^3\,\alpha_1^4-2\,U_{3-j}^3\,\alpha_3\,U_j^2\,\alpha_2^4+U_{3-j}^2\,\alpha_3\,U_j^3\,\alpha_2^4+3\\
&U_{3-j}^4\,\alpha_1^4\,\alpha_2^2\,z+3\,U_{3-j}^2\,\alpha_1^4\,\alpha_2^4\,z^3+6\,U_{3-j}\,\alpha_3\sqrt{DTT}\,\alpha_2^3\,z\,\alpha_1^3\,y\\
&+8\,U_{3-j}\,\alpha_3\,U_j\,\alpha_2^4\,z^2\,\alpha_1^4\,y\Big)\Big/\Big(\Big(z\,\alpha_1^2\,U_{3-j}^3-2\,U_{3-j}^2\,\alpha_1^2\,z\,y\,\alpha_2^2\\
&-2\,U_{3-j}^2\,\alpha_1^2\,U_j\,z+4\,U_{3-j}\,\alpha_1^2\,\alpha_2^2\,U_j\,z\,y+U_{3-j}\,\alpha_1^2\,\alpha_2^2\,z^3\\
&+U_{3-j}\,\alpha_1^2\,U_j^2\,z+2\,U_{3-j}\,y\sqrt{DTT}\,\alpha_2\,\alpha_1-U_{3-j}\,U_j^2\,\alpha_2^2\,z\\
&-2\,U_j\,z^3\,\alpha_2^2\,\alpha_1^2-2\,z^2\sqrt{DTT}\,\alpha_2\,\alpha_1\Big)\,\alpha_1^2\,\alpha_2^2\Big)
\end{aligned}\tag{35}$$

3.5 Control for Region QT

The last region of the switching surface denoted QT is very similar to the second one denoted TQ. As the name says the combination of parameters q_i and t_i values is opposite to the region TQ. Here the first subsystem $\mathbf{x}_1$ is in the linear case, i.e. its states comply with (9) and the second subsystem $\mathbf{x}_2$ fulfils (10) that means t_2 is nonzero. Therefore we substitute $t_1 = t_2$ and $q_2 = U_{3-j} - q_1$ in (12)

$$\begin{aligned}
\begin{bmatrix} x\\ y\\ z\end{bmatrix} = \Bigg[\Bigg[&q_1\left(\frac{1}{\alpha_1^3}-\frac{t_2}{\alpha_1^2}+\frac{1}{2}\,\frac{t_2^2}{\alpha_1}\right)-\frac{1}{6}\,U_j\,t_2^3+\big(U_{3-j}\\
&-q_1\big)\left(\frac{1}{\alpha_2^3}-\frac{t_2}{\alpha_2^2}+\frac{1}{2}\,\frac{t_2^2}{\alpha_2}\right)-\left(\frac{1}{6}\,U_{3-j}-\frac{1}{6}\,U_j\right)t_2^3\Bigg],\\
&\left[q_1\left(\frac{1}{\alpha_1^2}-\frac{t_2}{\alpha_1}\right)+\frac{1}{2}\,U_j\,t_2^2+\big(U_{3-j}-q_1\big)\left(\frac{1}{\alpha_2^2}-\frac{t_2}{\alpha_2}\right)\right.\\
&\left.+\left(\frac{1}{2}\,U_{3-j}-\frac{1}{2}\,U_j\right)t_2^2\right],\\
&\left[\frac{q_1}{\alpha_1}-U_j\,t_2+\frac{U_{3-j}-q_1}{\alpha_2}-\big(U_{3-j}-U_j\big)\,t_2\right]\Bigg]\Bigg]
\end{aligned}\tag{36}$$

First we solve parameters q_1 and t_2 from the last two equations of the set (36). From the solution of the quadratic equation we choose that one that gives the positive solution of t_2. After introducing discriminant DQT

$$DQT=\alpha_2^2\,U_{3-j}^2+z^2\,\alpha_1^2\,\alpha_2^2+\alpha_1^2\,U_{3-j}^2-2\,y\,\alpha_1^2\,\alpha_2^2\,U_{3-j}\tag{37}$$

we get

$$q_1=\frac{-\alpha_2\,U_{3-j}+sign\big(U_j\big)\sqrt{DQT}}{\alpha_1-\alpha_2}\tag{38}$$

$$t_2=-\frac{-\alpha_1\,U_{3-j}+z\,\alpha_1\,\alpha_2-\alpha_2\,U_{3-j}+sign\big(U_j\big)\sqrt{DQT}}{\alpha_1\,\alpha_2\,U_{3-j}}\tag{39}$$

To express the points of the region QT we substitute (38) and (39) into the first equation of the (36)

$$\begin{aligned}
x=\frac{1}{6}\,&\frac{1}{U_{3-j}^2\,\alpha_1^3\,\alpha_2^3}\Big(3\,\alpha_1^3\,U_{3-j}^2\,z\,\alpha_2+z^3\,\alpha_1^3\,\alpha_2^3+2\,\alpha_1^3\,U_{3-j}^3\\
&+3\,z\,\alpha_1\,\alpha_2^3\,U_{3-j}^2-3\,sign\big(U_j\big)^2\,DQT\,z\,\alpha_1\,\alpha_2+2\,\alpha_2^3\,U_{3-j}^3\\
&-2\,sign\big(U_j\big)^3\,DQT^{3/2}\Big)
\end{aligned}\tag{39}$$

The control value results from (18)

$$\begin{aligned}
u=\frac{1}{3}\,&\Big(-3\,y\,\alpha_1^3\,\alpha_2^3\,U_{3-j}^2+3\,\alpha_1^3\,\alpha_2^3\,U_{3-j}\,z^2+3\,\alpha_1^2\\
&\alpha_2^2\,U_{3-j}\,z\,sign\big(U_j\big)\sqrt{DQT}+3\,\alpha_3\,x\,\alpha_1^3\,\alpha_2^3\,U_{3-j}^2+\alpha_3\,z^3\,\alpha_1^3\,\alpha_2^3\\
&-3\,\alpha_3\,\alpha_1^3\,z\,\alpha_2^3\,y\,U_{3-j}-\alpha_3\,\alpha_1^3\,U_{3-j}^3-\alpha_3\,\alpha_2^3\,U_{3-j}^3\\
&+\alpha_3 sign\big(U_j\big)\,DQT^{3/2}\Big)\Big/\Big(\alpha_1^2\,\alpha_2^2\,\big(z^2\,\alpha_1\,\alpha_2-\alpha_1\,\alpha_2\,y\,U_{3-j}\\
&+sign\big(U_j\big)\sqrt{DQT}\,z\big)\Big)
\end{aligned}\tag{40}$$

3.6 Control Algorithm

The control algorithm consists in the localization of the initial state to one of the above mentioned regions and then of the control value calculation. But before we have to specify the parameter j. Then we can calculate the parameters q_i. According their values we can find the region to which the initial point belongs and finally evaluate the control value.

START

1. Evaluate q_1 according (20) and q_2 according (21)
2. IF q_1 fulfils (9) AND q_2 fulfils (9) THEN calculate u according (23) – *Region QQ*

```
3.  IF $q_1$ fulfils (9) AND $q_2$ NOT fulfils (9) THEN
    calculate $j=\frac{3-sign(q_2)}{2}$ AND GOTO 8
4.  Calculate $j=\frac{3+sign(q_1)}{2}$, $q_1=U_j$ and $q_2$
    according (26)
5.  IF $q_2$ fulfils (9) THEN calculate $u$ according (29) – Region TQ
6.  IF $q_2U_j<0$ THEN calculate $u$ according (35) – 1st part of the region TT
7.  Calculate $j=\frac{3-sign(q_1)}{2}$
8.  Evaluate $q_1$ according (38)
9.  IF $q_1$ fulfils (9) THEN calculate $u$ according (40) - Region QT
10. Evaluate $u$ according (35) – 2nd part of the region TT
END
```

It is important to notice that at the end the control value computed according this algorithm must be limited by (2).

4. EVALUATION OF DESIGNED CONTROLLER

To show the performance of the designed controller we have carried out several simulations that differ from the starting point, parameters of the controller, and constraints. In the Fig. 3 one can see the time responses from the initial state $\mathbf{x}=\begin{bmatrix}200 & 0 & 0\end{bmatrix}^t$ under nonsymmetrical control value constraints $u=\langle -1 \quad 2\rangle$.

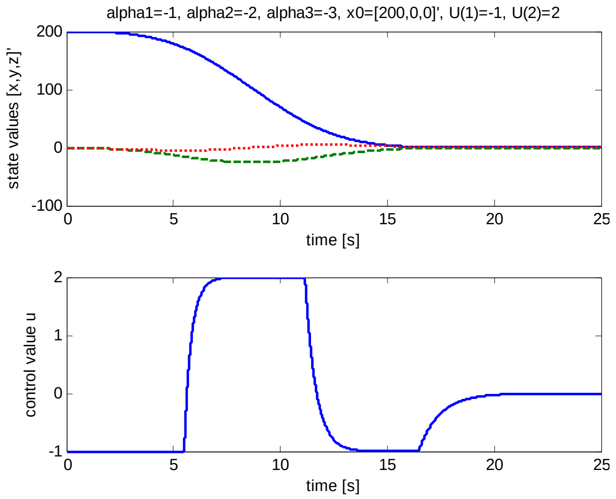


Fig. 3. Time responses of state and control variables. Sub-optimal controller with nonsymmetrical constraints.

All three pulses of the time optimal control can be mentioned but the control value switches from one limit value to the other one smoothly. The change rate is given by the choice of parameters α_i. In this case the values of α_i were $\alpha_1=-1$, $\alpha_2=-2$, $\alpha_3=-3$

The comparison of the time sub-optimal control with the optimal one is shown in the Fig. 4. This time the starting point was $\mathbf{x}=\begin{bmatrix}15.7916 & -4.25 & 1\end{bmatrix}^t$. The other parameters were $\alpha_1=-1.5$, $\alpha_2=-3$, $\alpha_3=-6$ and $u=\langle -1 \quad 1\rangle$.

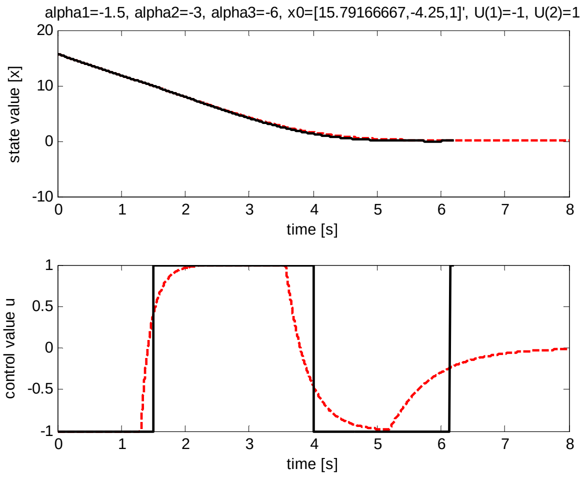


Fig. 4. Comparison of time optimal and sub-optimal control.

There are not big differences in the time responses of the state variable x. But one can see the difference in the behavior of the control variables. The time sub-optimal control variable uses limits for a shorter period because it needs a certain time to switch to the opposite value. The time optimal control variable switches immediately that can cause problems when the dynamics of a controlled system is not precisely identified. The time sub-optimal controller switches in advance and it finishes later but it is not so sensitive to the uncertain parameters or unmodelled dynamics. By moving the negative values of parameters α_i towards a zero we could get behavior similar to the linear pole assignment controller. To get the exact linear behavior we have to higher the control value limits.

5. THREE-LEVEL HYDRAULIC SYSTEM DESCRIPTION

The hydraulic system consists of three tanks (Fig. 5). These are interconnected by valves characterized by c_{12} and c_{23} constants. The last tank has the outlet valve defined by c_3 constant. The pump supplies the water into the first tank. The input value is denoted by q_1. The h_1, h_2 and h_3 describe the height of levels in corresponding tanks. These are the state variables from which h_3 represents the output value that will be controlled to the desired value w.

The system could be described by following differential equations

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{1}{A_1} q_1 - c_{12}\sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= c_{12}\sqrt{h_1 - h_2} - c_{23}\sqrt{h_2 - h_3} \qquad y = h_3 \\ \frac{dh_3}{dt} &= c_{23}\sqrt{h_2 - h_3} - c_3\sqrt{h_3} \end{aligned} \tag{41}$$

The identified parameters are

$$\begin{aligned} A_1 &= 0.001 \\ c_{12} &= 0.015383 \\ c_{23} &= 0.015518 \\ c_3 &= 0.0070655 \end{aligned} \tag{42}$$

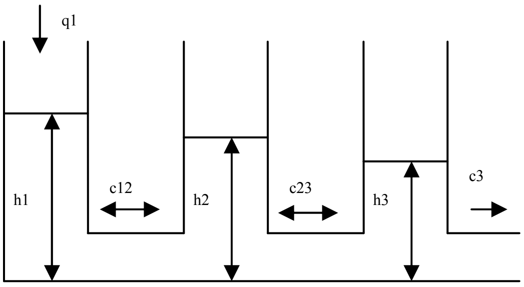


Fig. 5. Three-level hydraulic system.

The Fig. 6 shows the real three-level hydraulic plant. However in this paper only simulation results are presented. The simulations use the parameters identified on the real plant. Results from the real hydraulic system should be available in the future.

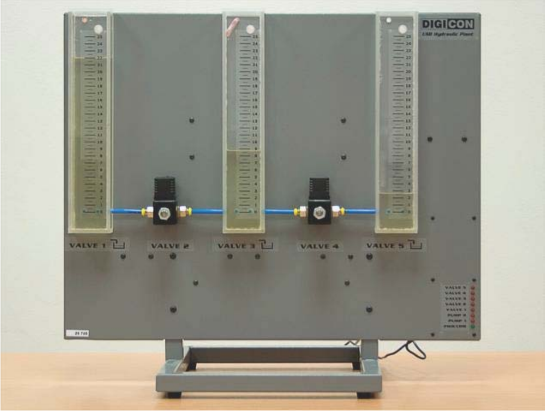


Fig. 6. Real three-level hydraulic plant.

6. LINEAR POLE ASSIGNMENT CONTROLLER APPLIED TO HYDRAULIC SYSTEM

To compare the sub-optimal controller with other types of controllers first we have designed the linear pole assignment controller. Before this it was necessary to linearize the nonlinear hydraulic system in the chosen state. The state of linearization was equal to the desired state

$$\begin{aligned} h_3 &= w \\ h_2 &= 1.207307162w \\ h_1 &= 1.418268913w \\ q_1 &= 0.7065500002 \cdot 10^{-5}\sqrt{w} \end{aligned} \tag{43}$$

The linearization has been made in Matlab using *linmode*, *canon* and *acker* functions. After setting the desired closed loop poles to the values

$$\alpha_1 = -0.1\ ,\ \alpha_2 = -0.2\ ,\ \alpha_3 = -0.3 \tag{44}$$

and choosing the desired value $w = 0.1$ we got for the linear pole assignment controller the feedback gain vector $K = \lfloor 0.3751 \cdot 10^{-3} \quad 0.6641 \cdot 10^{-3} \quad 0.7576 \cdot 10^{-3} \rfloor$ and the feedforward gain $f = 0.0021$. The Fig. 7 shows the corresponding time responses.

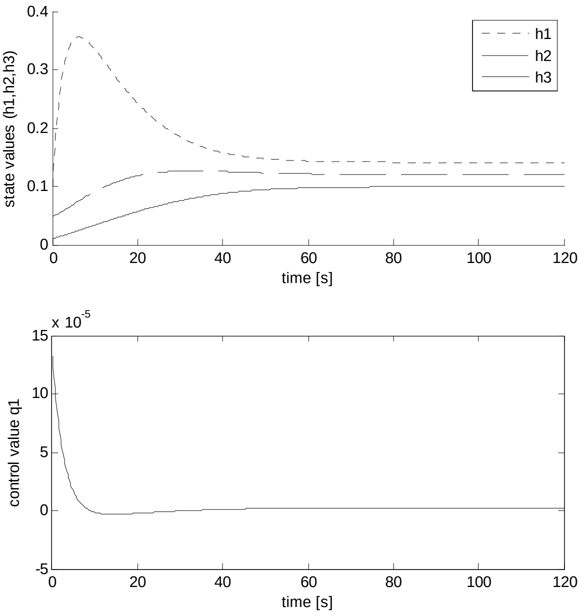


Fig. 7. Time responses of state and control variables. Linear pole assignment controller without constraints.

7. EXACT LINEARIZATION METHOD APPLIED TO HYDRAULIC SYSTEM

The hydraulic system could be linearized not only in the fixed point but also continuously. This is the case of exact linearization method. To apply it first we have to transform the hydraulic system to the nonlinear controllability canonical form. The new coordinates are denoted as *z1*, *z2* and *z3*.

$$\begin{aligned} z1 &:= h3 \\ z2 &:= 0.01551800000\sqrt{h2 - 1.\,h3} - 0.007065500000\sqrt{h3} \\ z3 &:= \ 0.1000000000 10^{-12}(-0.1193566970 10^{10}\sqrt{h3}\,\sqrt{h1 - 1.\,h2} \\ &\quad + 0.2158476789 10^{10}\sqrt{h2 - 1.\,h3}\,\sqrt{h3} - 0.1096424290 10^{10} h3 + \\ &\quad 0.548212145 10^{9} h2) \,/\, (\sqrt{h2 - 1.\,h3}\,\sqrt{h3}\,) \end{aligned} \tag{45}$$

The time derivation of *z3* gives the expression

$$\frac{dz3}{dt} = L_f^3 y + L_g L_f^2 y\, q_1 \tag{46}$$

where $L_f^3 y$ is

$$\begin{aligned} Lf3T1 :=& -0.100000000010^{-15}(0.918032035010^{10}\, h3^{(3/2)} \sqrt{h1-1.\,h2}\; h2 \\ &-0.183606407010^{11}\, h3^{(5/2)} \sqrt{h1-1.\,h2} + 0.421657371310^{10}\, h3\; h2\; h1 \\ &-0.421657371310^{10}\, h3\; h2^2 + 0.918032035010^{10}\, h3^{(3/2)} \sqrt{h1-1.\,h2}\; h1 \\ &+0.926088612010^{10}\, h3^{(5/2)} \sqrt{h2-1.\,h3} \\ &-0.425357803310^{10}\, h3 \sqrt{h2-1.\,h3} \sqrt{h1-1.\,h2}\; h2 \\ &-0.185217722410^{11}\, h3^{(3/2)} \sqrt{h2-1.\,h3}\; h1 \\ &+0.926088612010^{10} \sqrt{h2-1.\,h3}\; h3^{(3/2)}\; h2 \\ &-0.425357803310^{10} \sqrt{h1-1.\,h2} \sqrt{h2-1.\,h3}\; h2^2 + 0.421657371310^{10}\, h3^2\, h1 \\ &-0.421657371310^{10}\, h3^2\, h2 + 0.193669645510^{10} \sqrt{h1-1.\,h2} \sqrt{h3}\; h2^2) \,/\, (\\ &(h2-1.\,h3)^{(3/2)} \sqrt{h1-1.\,h2}\; h3^{(3/2)}) \end{aligned} \tag{47}$$

and $L_g L_f^2 y$ is

$$LgLf2T1 := \frac{0.05967834850}{\sqrt{h1-1.\,h2}\sqrt{h2-1.\,h3}} \tag{48}$$

Then the feedback that linearizes the nonlinear system (41) can be expressed in the form

$$q_1 = -\frac{L_f^3 y}{L_g L_f^2 y} + \frac{v}{L_g L_f^2 y} \tag{49}$$

where *v* represents the desired characteristic polynomial (in terms of *zi*) of the closed loop circuit. If we choose the closed loop poles according to (44) and the desired value will be again $w = 0.1$ we get for *v*

$$\begin{aligned} v :=& -0.006000000000h3 - 0.001706980000\sqrt{h2-1.\,h3} + 0.0007772050000\sqrt{h3} + \\ & 0.600000000010^{-13}(-0.119356697010^{10}\sqrt{h3}\sqrt{h1-1.\,h2} \\ &+0.215847678910^{10}\sqrt{h2-1.\,h3}\sqrt{h3} - 0.109642429010^{10}\, h3 + 0.54821214510^{9}\, h2 \\ &)\,/\,(\sqrt{h2-1.\,h3}\sqrt{h3}) + 0.0005999999999 \end{aligned} \tag{50}$$

After substituting (50), (47) and (48) into (49) one gets the control law

$$\begin{aligned} u :=& \,0.1675649587\ 10^{-14}\,(0.9180320350\ 10^{10}\, \mathrm{h3}(t)^{(3/2)} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)}\, \mathrm{h2}(t) \\ &-0.1836064070\ 10^{11}\, \mathrm{h3}(t)^{(5/2)} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)} \\ &+0.4216573713\ 10^{10}\, \mathrm{h3}(t)\,\mathrm{h2}(t)\,\mathrm{h1}(t) - 0.4216573713\ 10^{10}\, \mathrm{h3}(t)\,\mathrm{h2}(t)^2 \\ &+0.9180320350\ 10^{10}\, \mathrm{h3}(t)^{(3/2)} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)}\, \mathrm{h1}(t) \\ &+0.9260886120\ 10^{10}\, \mathrm{h3}(t)^{(5/2)} \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)} \\ &-0.4253578033\ 10^{10}\, \mathrm{h3}(t) \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)}\, \mathrm{h2}(t) \\ &-0.1852177224\ 10^{11}\, \mathrm{h3}(t)^{(3/2)} \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)}\, \mathrm{h1}(t) \\ &+0.9260886120\ 10^{10} \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)}\, \mathrm{h3}(t)^{(3/2)}\, \mathrm{h2}(t) \\ &-0.4253578033\ 10^{10} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)} \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)}\, \mathrm{h2}(t)^2 \\ &+0.4216573713\ 10^{10}\, \mathrm{h3}(t)^2\, \mathrm{h1}(t) - 0.4216573713\ 10^{10}\, \mathrm{h3}(t)^2\, \mathrm{h2}(t) \\ &+0.1936696455\ 10^{10} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)} \sqrt{\mathrm{h3}(t)}\, \mathrm{h2}(t)^2)\,/\,((\mathrm{h2}(t)-1.\,\mathrm{h3}(t)) \\ &\mathrm{h3}(t)^{(3/2)}) + 16.75649587 \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)} \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)}\,(\\ &-0.006000000000\, \mathrm{h3}(t) - 0.001706980000 \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)} \\ &+0.0007772050000 \sqrt{\mathrm{h3}(t)} + 0.6000000000\ 10^{-13}\,(\\ &-0.1193566970\ 10^{10} \sqrt{\mathrm{h3}(t)} \sqrt{\mathrm{h1}(t)-1.\,\mathrm{h2}(t)} \\ &+0.2158476789\ 10^{10} \sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)} \sqrt{\mathrm{h3}(t)} - 0.1096424290\ 10^{10}\, \mathrm{h3}(t) \\ &+0.548212145\ 10^{9}\, \mathrm{h2}(t))\,/\,(\sqrt{\mathrm{h2}(t)-1.\,\mathrm{h3}(t)} \sqrt{\mathrm{h3}(t)}) + 0.0005999999999\,) \end{aligned} \tag{51}$$

The corresponding time responses are depicted in the Fig. 8.

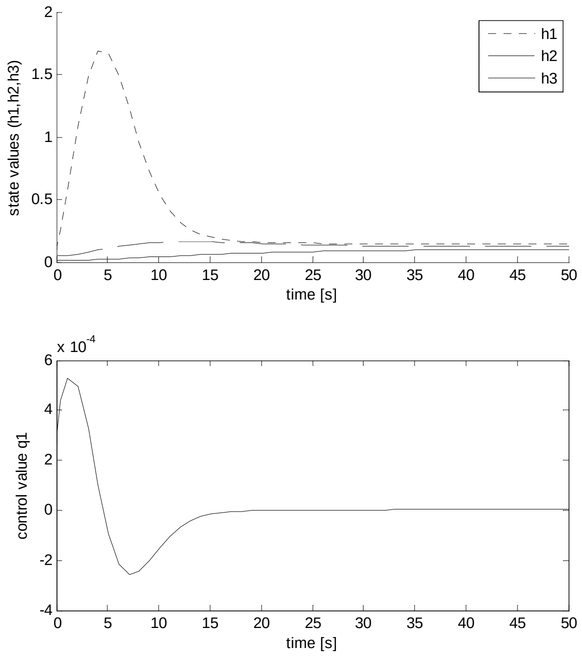


Fig. 8. Time responses of state and control variables. Exact linearization controller without constraints.

8. TIME SUB-OPTIMAL CONTROLLER APPLIED TO HYDRAULIC SYSTEM

In this section we will apply the sub-optimal controller derived in the section 4 to the nonlinear hydraulic system. The nonlinearity is compensated by the control limits transformation. The control value constraints are

$$q_1 \in \langle Q_{\min} \quad Q_{\max} \rangle \ ,\ Q_{\min} = 0,\ Q_{\max} = 10^{-5} \tag{52}$$

Before the control action is computed according to the section 3.6 the control constraints are transformed

$$\begin{aligned} U_1 &= L_g L_f^2 y \cdot Q_{\min} + L_f^3 y \\ U_2 &= L_g L_f^2 y \cdot Q_{\max} + L_f^3 y \end{aligned} \tag{53}$$

where $L_f^3 y$ and $L_g L_f^2 y$ come from (47) and (48) respectively. Then after calculating the control action *u* (section 3.6) it is necessary to carry out the following inverse transformation of it

$$q_1 = \frac{u - L_f^3 y}{L_g L_f^2 y} \tag{54}$$

In the Fig. 9 one can see the time responses with three intervals of control. These results have been simulated for the following chosen parameters that correspond to poles

$$\alpha_1 = -0.1\ ,\ \alpha_2 = -0.5\ ,\ \alpha_3 = -1 \tag{55}$$

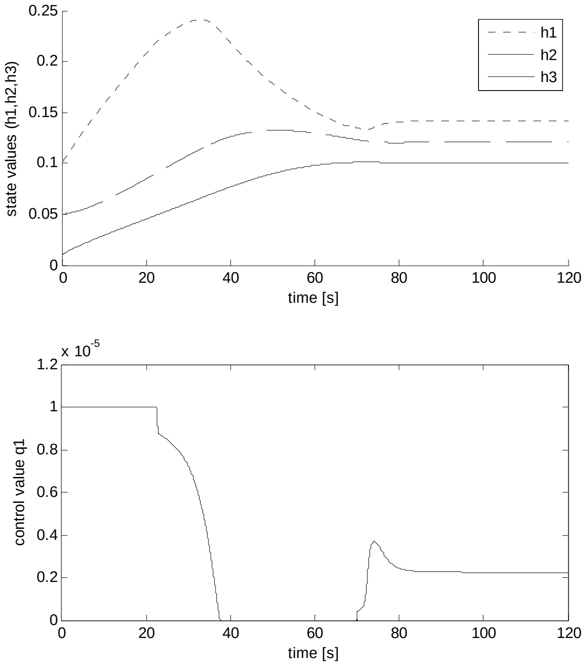


Fig. 9. Time responses of state and control variables using sub-optimal controller with three intervals of control.

According to the control value behavior the last interval of control does not reach the maximal possible value of q_1. There is a small overshoot of the output value. The overshoot can be suppressed by shifting the poles towards the zero value. This will cause that the transients will slow down and there will be only two intervals of control. Fig. 10 shows the responses for the set of poles given by (44).

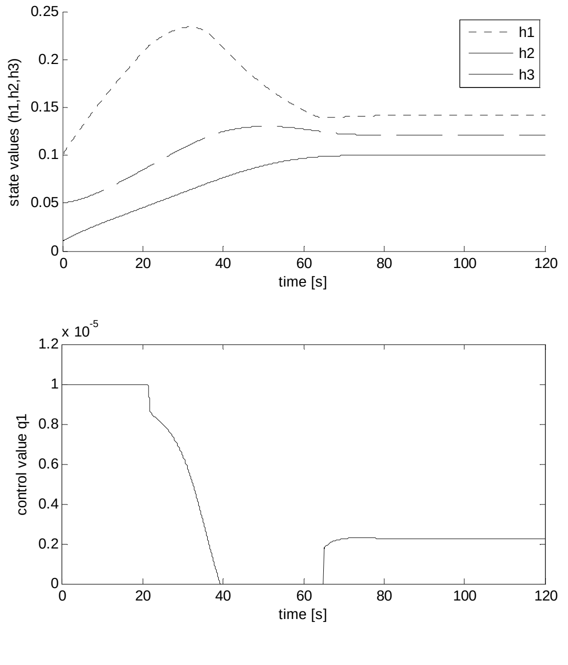


Fig. 10. Time responses of state and control variables using sub-optimal controller with slower poles.

9. CONCLUSIONS

Presented controller design relies on switching surfaces. Because our aim was to decrease the sensitivity of controller we introduced new parameters into the design. In linear cases these parameters are identical with the poles of the closed loop. Of course, the switching surfaces are more complicated in comparison with optimal control that is caused by additional parameters. In this paper we derived the solution with explicit mathematical formulas that is fast enough to be used in real time applications in the near future. Up to now we have applied it for the three-level hydraulic system in Matlab/Simulink environment. Simulations proved that it is possible to use the designed controller also for nonlinear systems. Although it needs more computational power in comparison with a linear pole assignments or the exact linearization method it is able to respect the control value constraints.

ACKNOWLEDGEMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09 and by the Grant VEGA No. 1/0656/09. It was also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Athans, M. and Falb, P. (1996). *Optimal Control: An Introduction to the Theory and its Applications*. McGraw-Hill, New York.

Bisták, P., Ťapák, P., and Huba, M.(2005). Constrained Pole Assignment Control of Double and Triple Integrator. In: CASYS 05, Seveth International Conference on Computing Anticipatory Systems, HEC-Ulg, LIEGE, Belgium.

Cox, D., Little, J., and O'Shea, D. (2007). *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer-Verlag, New York.

Huba, M. (2006). "Constrained pole assignment control," In: *Current Trends in Nonlinear Systems and Control*, Boston: Birkhäuser, 163-183.

Pao, L. Y. and Franklin, G. F. (1993). Proximate Time-Optimal Control of Third-Order Servomechanisms, *IEEE Transactions on Automatic Control*, Vol. 38, No. 4.

Pavlov, A.A. (1966). *Synthesis of relay time-optimal systems*. In Russian, Publishing House "Nauka", Moscow.

Ťapák, P., Bisták, P., and Huba, M. (2006). Control for Triple Integrator with Constrained Input. In: 14th Mediterranean Conference on Control and Automation, Piscataway : IEEE, Ancona, Italy.

Walther, U., Georgiou, T. T., and Tannenbaum, A. (2001). On the Computation of Switching Surfaces in Optimal Control: A Gröbner Basis Approach, *IEEE Transactions on Automatic Control*, Vol. 46, No. 4.

Comments – Remarks

Modified Filtered Smith Predictors for FOPDT Plants

Mikuláš Huba*,**

** Institute of Control and Industrial Informatics, FEI STU, Ilkovičova 3, SK-812 19 Bratislava Slovakia (Tel: +421-2 -60291 771; e-mail: mikulas.huba@ stuba.sk).*
*** MI/PRT, FernUniversität in Hagen, Universitätsstr. 27, D-58084 Hagen Germany (e-mail: mikulas.huba@fernuni-hagen.de)*

Abstract: This paper introduces modifications of the Filtered Smith Predictor for simple linear first order plants with dead time (FOPDT) and constrained input. Constraints may be simply considered due to replacing the PI controller by a two-degree-of-freedom (2DOF) P controller in the primary loop of the classical IMC structure with the dynamical feedforward and parallel plant model. Equivalence of the traditional and the new solutions in the proportional range of control is shown by the setpoint-to-output and disturbance-to-output transfer functions in the nominal case and by the computer simulations for systems with the plant-model mismatch.

Keywords: Robustness analysis; robust controller synthesis; robust time-delay systems.

1. INTRODUCTION

The Smith Predictor (SP) (Smith, 1957) represents one of the oldest structures of the Dead-Time Compensators (DTCs) used for controlling systems with dead time. Due to their infinite dimension, they are still surrounded by not sufficiently explained myths and still in focus of current research (Åström and Hägglund, 2005; Guzmán et al., 2008; Normey-Rico and Camacho, 2007; 2008; 2009; Normey-Rico et al., 2009; Panda, 2009; Tan et al., 2010; Zhang, Rieber and Gu, 2008, Vrecko et al., 2001). One of them is related to the primary controller that is usually chosen as the PI one. But, despite that just few of the known DTCs are interpreted as disturbance observer (DO) based structures, in fact, all of them may be shown to include observers for reconstruction of either input or output disturbances. This has an important impact on choosing the primary controller that need not to include the integral action.

2. PI-FSP CONTROLLERS FOR STABLE FOPDT PLANTS

The Filtered Smith Predictor (FSP) (FSP, Normey-Rico et al., 1997; 2009; Normey-Rico and Camacho, 2007; 2008; 2009) may be interpreted as a structure using parallel plant model (PPM) for reconstruction of the output disturbance. Firstly, it was used in the pioneering work by Smith (1957) and played later a key role within the concept of the Internal Model Control (IMC) (see e.g. Morari and E. Zafiriou, 1989). Here, by controlling the First Order Plus Dead Time (FOPDT) plants it will be shown that the FSP may be further simplified by replacing the primary PI controller by a two-degree-of-freedom (2DOF) P controller, that improves its usability in the constrained control without decreasing its capability in the disturbance rejection.

The filtered Smith Predictor (FSP) was firstly proposed for stable FOPDT processes to improve robustness of the traditional SP and to decouple the reference setpoint and the disturbance response, to stabilize the controller loop in case of unstable and integral plants and to achieve required robustness. It is based on the dynamical feedforward control with the reference plant model (Åström and Hägglund, 2005; Visioli, 2006) used in the disturbance reconstruction and compensation. The approach to designing FSPs for the FOPDT plants introduced in (FSP, Normey-Rico et al., 2009; Normey-Rico and Camacho, 2007; 2008; 2009) considers (similarly as the below explained newly proposed solution in Fig. 2) compensation of an output disturbance by correction of the reference value, whereby the disturbance is reconstructed by using the PPM. For the stable plant $P(s)$ with $P_0(s)$ denoting its “fast“ delay-free nominal dynamics and $P_n(s)$ its nominal model with particular set of parameters considered in controller tuning

$$P(s)=\frac{Ke^{-\Theta s}}{Ts+1} \;;\; P_0(s)=\frac{K_0}{T_0s+1};\; P_n(s)=\frac{K_0e^{-\Theta_0 s}}{T_0s+1} \quad (1)$$

the primary PI-controller

$$C_0(s)=\frac{U(s)}{E(s)}=K_c\frac{1+T_is}{T_is} \;;\; T_i=T_0 \;;\; K_c=\frac{T_0}{T_rK_0} \quad (2)$$

is used, whereby T_r is the time constant of the (fast) primary loop described by the transfer function

$$C(s)=\frac{U(s)}{E(s)}=\frac{C_0}{1+C_0P_0}=\frac{1}{K_0}\frac{1+T_0s}{1+T_rs} \quad (3)$$

The nominal (reference) setpoint-to-output transfer function (considering $P=P_n$) is

$$H_r(s)=\frac{Y(s)}{R(s)}=C(s)P(s)=\frac{e^{-\Theta s}}{1+T_r s} \tag{4}$$

Basic Acronyms	
2DOF	Two Degree of Freedom
DO	Disturbance Observer
DTC	Dead Time Compensator
FOPDT	First Order Plus Dead Time
FSP	Filtered Smith Predictor
MO	monotonic, monotonicity
PP	Performance Portrait
SP	Smith Predictor
ST	stabil, stability

When extending the disturbance compensation loop by the 1st and the 2nd order disturbance filters

$$F_{r1}(s)=\frac{1+\beta_{11}s}{1+T_f s} \tag{6a}$$

$$F_{r2}(s)=\frac{(1+T_r s)(1+\beta_{12}s)}{(1+T_f s)^2} \tag{6b}$$

where T_f represents time constant of the disturbance responses, the equivalent controller may be introduced as

$$C_e(s)=\frac{C}{1-CPF_r} \tag{7}$$

The transfer functions corresponding to the output disturbance do, to the input disturbance di and to the measurement noise n become

$$\begin{aligned}
H_o(s)&=\frac{Y(s)}{D_o(s)}=\frac{1}{1+C_e PF_r}=1-\frac{F_r(s)e^{-\Theta s}}{1+T_r s}\\
H_i(s)&=\frac{Y(s)}{D_i(s)}=\frac{P}{1+C_e PF_r}=P(s)\left[1-F_r(s)\frac{e^{-\Theta s}}{1+T_r s}\right]\\
H_n(s)&=\frac{U(s)}{N(s)}=\frac{C_e F_r}{1+C_e PF_r}=C(s)F_r(s)
\end{aligned} \tag{8}$$

Requirements $H_i(0)=0$ and $H_i(-1/T)=0$ give for (6a) the PI-F1SP controller, when

$$\begin{aligned}
\beta_{11}&=T\left[1-(1-T_r/T)(1-T_f/T)e^{-\Theta/T}\right]\\
H_o(s)&=1-\frac{1+\beta_{11}s}{(1+T_f s)(1+T_r s)}e^{-\Theta s}\\
H_i(s)&=\frac{Ke^{-\theta s}}{Ts+1}\left[1-\frac{1+\beta_{11}s}{(1+T_f s)(1+T_r s)}e^{-\Theta s}\right]\\
H_n(s)&=C(s)\frac{1+\beta_{11}s}{1+T_f s}
\end{aligned} \tag{9}$$

In the case of the filter (6b), this is determined to fulfil $H_i(0)=0$ and $H_i(-1/T)=0$ that yields

$$\begin{aligned}
\beta_{12}&=T\left[1-(1-T_f/T)^2 e^{-\Theta/T}\right]\\
H_o(s)&=\left[1-\frac{1+\beta_{12}s}{(1+T_f s)^2}e^{-\Theta s}\right]\\
H_i(s)&=\frac{Ke^{-\theta s}}{Ts+1}\left[1-\frac{1+\beta_{12}s}{(1+T_f s)^2}e^{-\Theta s}\right]\\
H_n(s)&=C(s)\frac{1+\beta_{12}s}{(1+T_f s)^2}
\end{aligned} \tag{10}$$

The corresponding controller will be denoted as the PI-F2SP.

3 NEW P-F1SP AND P-F2SP CONTROLLERS

Next, we will firstly show that an equivalent solution may also be achieved by considering 2DOF P controller as the primary controller instead of (2). In this way and by considering different disturbance filters (6), two new modifications of the FSP will be introduced denoted with respect to the used disturbance filters as P-F1SP and P-F2SP.

3.1 Primary 2DOF P-controller

Simpler P controllers instead of PI ones were recommended in a slightly modified setting e.g. by Liu et al. (2005), or Lu et al. (2005). The 2DOF controller will be expressed as the P controller with the gain K_P extended by the static feedforward control u_0

$$u=K_P e+u_0\ ;\ u_0=r/K_0\ ;\ K_P=(T_0/T_r-1)/K_0; \tag{11}$$

Thereby, the fast model parameters T_0, K_0 correspond to the estimates of the plant parameters T and K . T_r represents the reference time constant of the fast primary loop. In the nominal case with $P_n=P$ and neglected control signal constraints, the structure in Fig. 1 yields transfer function between the setpoint and the non-delayed output

$$H_{r0}(s)=\frac{X(s)}{R(s)}=\frac{1}{T_r s+1}\ ;\ T_r=\frac{T_0}{K_c K_0+1} \tag{12}$$

As in (3), the dynamics between setpoint and control signals

$$C(s)=\frac{U(s)}{R(s)}=\frac{1}{K_0}\frac{T_0 s+1}{T_r s+1}\ ;\ T_r=\frac{T_0}{K_c K_0+1} \tag{13}$$

represents filtered inversion of the fast plant dynamics.

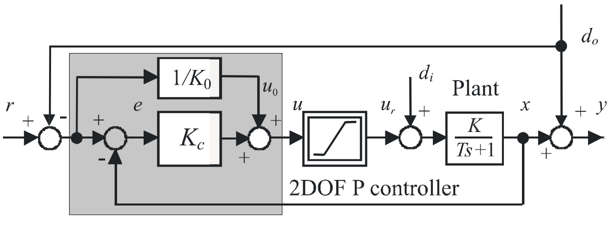


Fig. 1 Idea of compensating the measurable output disturbance d_o by a feedforward to the reference setpoint of the 2DOF P-controller

3.2 Respecting the control signal constraints

Control signal of real plants is always subject to control constraints expressed e.g. in the form of the saturation function

$$u_r = sat(u) = \left\{ \begin{array}{ll} U_{\max}; & u > U_{\max} \\ u; & U_{\min} \le u \le U_{\max} \\ U_{\min} & u < U_{\min} \end{array} \right. \qquad (14)$$

In controlling stable plant P0 (1) with built in constraints (14) by the P-controller (11) the loop remains stable without taking any additional measures for any transients with the final value $y_\infty = r$ satisfying

$$\begin{aligned} & r \in \left(Y_{\min}, Y_{\max}\right) \\ & Y_{\min} = K\left(U_{\min} + d_i\right) + d_o; \\ & Y_{\max} = K\left(U_{\max} + d_i\right) + d_o \end{aligned} \qquad (15)$$

This may be shown e.g. by choosing appropriate Ljapunov function, by the circle criterion, by the Popov criterion, or by the passivity approach (Föllinger, 1993; Glattfelder and Schaufelberger, 2003; Hsu and Meyer, 1968).

In the much more complicated FSP, outputs of the primary controller (11) yields inversion of the fast dynamics. In order to respect constraints imposed on the real plant input, outputs of the primary loop can not be simply generated by their transfer function, but the primary loop must be implemented by including at least so strong constraints as those at the plant input. Furthermore, in order to guarantee relevance of information used in the disturbance reconstruction, also the DO must be supplied with the constrained control signal. The corresponding predicted signal $\hat{x}_p$ (in the structure in Fig. 2, $\hat{x}_p$ plays a role of the output predicted with respect to the delayed output x_d), what again requires to work with constrained primary loop. However, for the sake of simplicity, saturation indicated in Fig. 1 will be omitted from the scheme in Fig. 2.

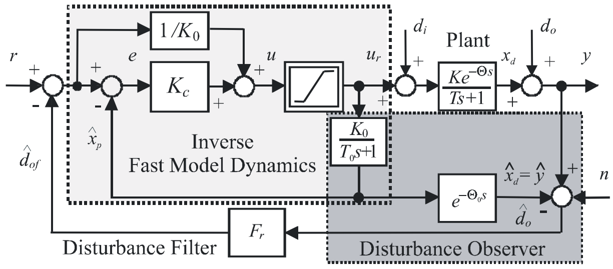


Fig. 2 Modified P-FSP with the primary loop using 2DOF P-controller (11) with the disturbance filters (6)

3.3 Modified P-F1SP and P-F2SP Controllers

Since the primary controllers (3) and (13) are the same, the same also holds for the nominal closed loop transfer functions (8), i.e. both structures are fully equivalent. Also the requirements on filters (6) remain unchanged (i.e. $H_i(-1/T) = 0$, $H_o(0) = 0$ and $H_i(0) = 0$) that finally requires tuning (9-10). This holds, however, just for the nominal tuning ($P = P_n$) and without constraining of the control signal.

COMPARING PI-F1SP WITH P-F1SP

In what follows, both types of controllers are compared by the computer simulation for the plant parameters

$$K = 1; T = 1; \Theta = 1 \qquad (16)$$

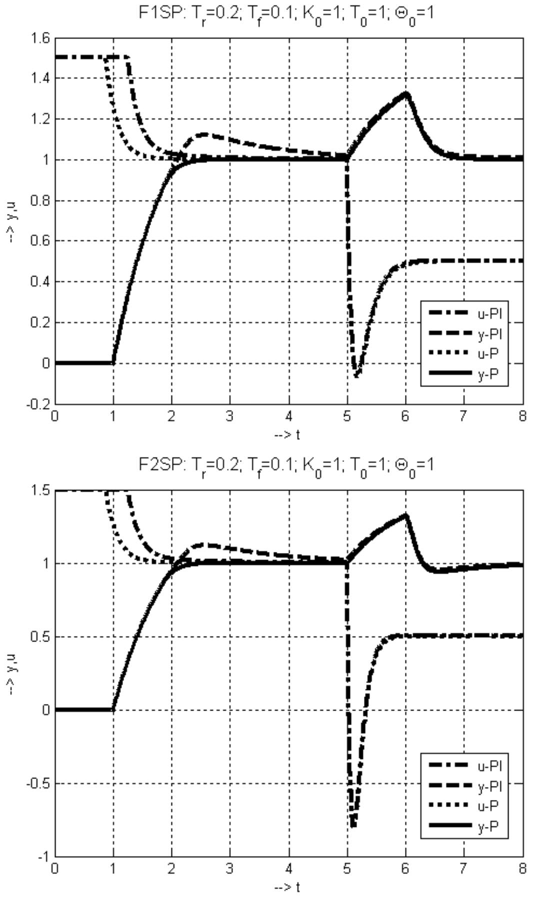


Fig. 3 Nominal responses of constrained PI-F1SP and P-F1SP controllers (above) and PI-F2SP and P-F2SP controllers (below); step response $\Delta d_i = 0.5$ at $t = 4s$

Fig. 3 confirms the main advantage of the P-F1SP structure, when the corresponding setpoint step response remains fully monotonic, whereas the output response corresponding to the PI-F1SP shows already in the nominal case with the exactly known system parameters typical overshooting due to the integrator windup. By decreasing the ratio K/T its value will be increasing and the loop will require appropriate anti-windup measures (Zhang and Jiang, 2008).

Since the remaining transients in Figs 4-6 show practically identical behaviour of both structures also in the non-nominal (perturbed) cases, it is possible to conclude that the traditional PI-F1SP structure may be fully replaced by the P-F1SP one that will guarantee equivalent dynamics in the proportional range of control, but it does not generate the windup effect under constrained control.

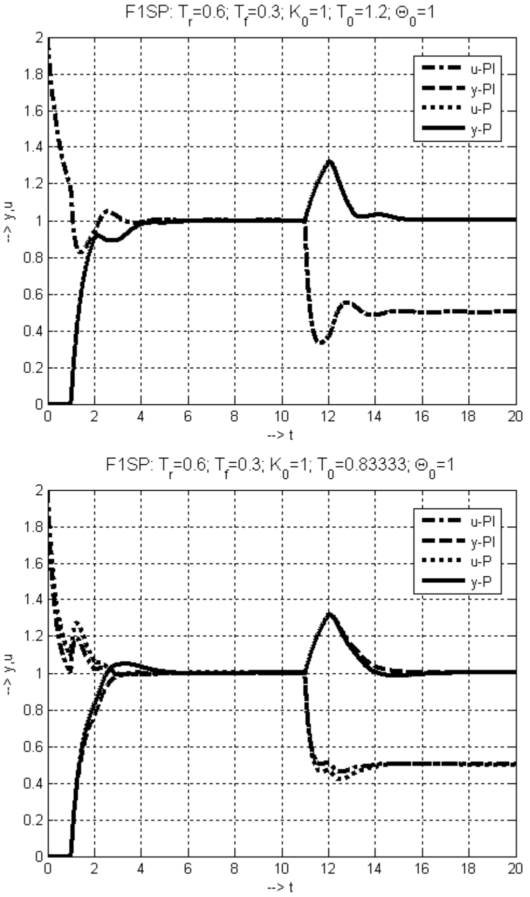


Fig. 4 Comparing P-F1SP and PI-F1SP controllers for the 20% mismatch in the plant time constant T

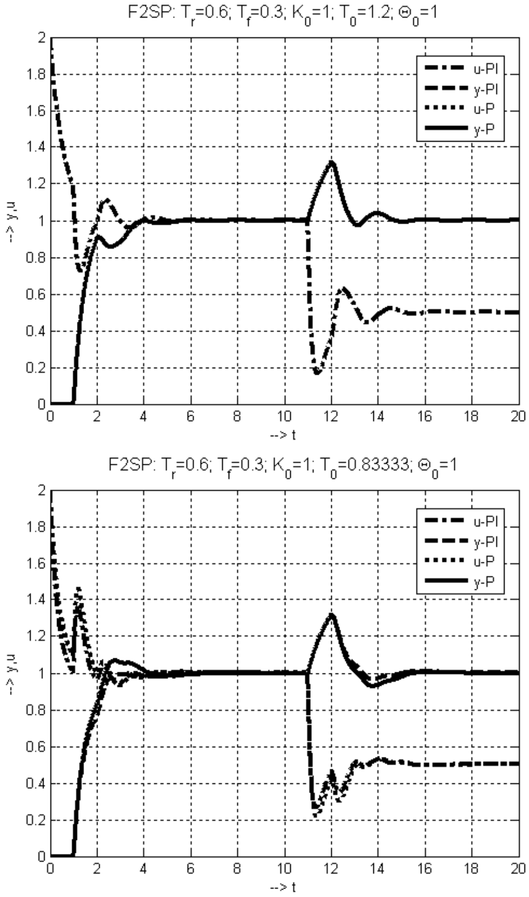


Fig. 5 Comparing P-F2SP and PI-F2SP controllers for the 20% mismatch in the plant time constant T.

COMPARING PI-F2SP WITH P-F2SP

Both types of controllers will again be compared by the computer simulation for the real plant parameters (16) $K=1;T=1;\Theta=1$. Thereby, Fig. 3 (below) again confirms the main advantage of the P-F2SP structure, when the corresponding setpoint step response remains fully monotonic, whereas the output response corresponding to the PI-F2SP shows, due to the integrator windup, typical overshooting already in the nominal case with the exactly known system parameters. By decreasing the ratio K/T its value will increase and the loop will require appropriate anti-windup measures.

Similarly, as in the case of the first order disturbance filters, both equivalent structures give practically the same dynamics, whereby an underestimated time constant value (Fig. 5 below) leads to an output overshooting, whereas an overestimated time constant leads to the slowed-down output response (Fig. 5 above).

In changing model parameters used for simple PID tuning the corresponding transients are either overdamped, or oscillatory. Here, both the underestimated as well as overestimated values T_0 lead to control responses with increased number of peaks. So, in the case of oscillatory control transients an 'intuitive' retuning of the controller is far from being straightforward.

The same may be observed in the case of perturbed model gain values (Fig. 6), just the effects of increased values K_0 are similar to effects of decreased values T_0. But, again, both considered structures give equivalent closed loop dynamics.

The main difference in comparing with the solutions using the first order disturbance filter is illustrated by the extremely large sensitivity to the dead time perturbations in Fig. 7.

The 2nd order disturbance filter (6b) is mainly motivated by the fact that it gives analytical formulas enabling a fully independent tuning of the setpoint and of the disturbance responses. By using the 1st order model (6a), the disturbance response is determined both by the setpoint time constant as well as by the filter time constant T_f. The achieved results, however, show that such a requirement of a fully independent tuning of the setpoint and of the disturbance responses is more of an academic importance than of a practical one.

EVALUATION BY THE PERFORMANCE PORTRAIT

Although the transients corresponding to limit values of the tuning parameters show strong similarities in behaviour of the PI-FSP and P-FSP, still there arise question, if this really holds for all possible working points. The best way to answer such question is to derive Performance Portrait (Huba, 2010;

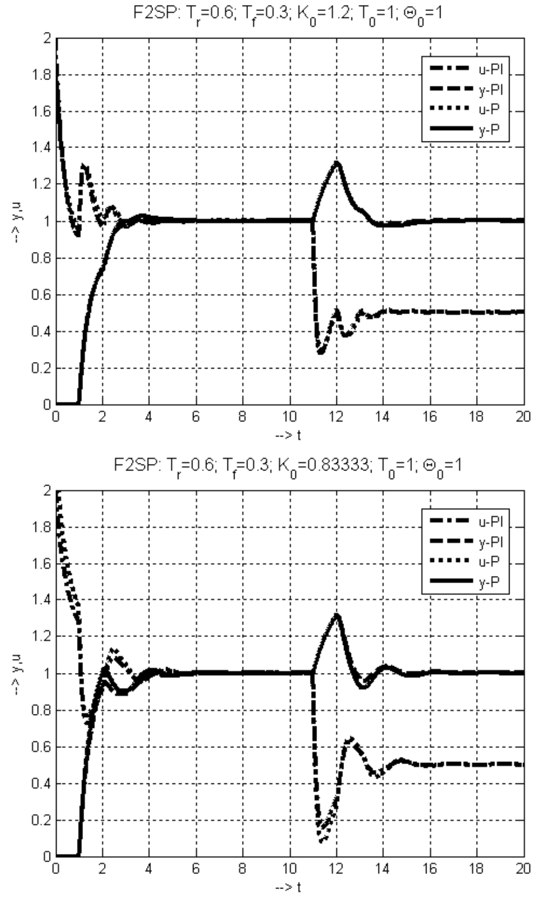


Fig. 6 Comparing P-F2SP and PI-F2SP controllers for the 20% mismatch in the plant gain K.

2011a) showing chosen performance measure over much higher number of operating points.

To characterize speed and duration of transients at the plant output in a simpler way, IAE (Integral of Absolute Error) performance index was used defined as

$$IAE = \int_0^\infty \left|[e(t) - e(\infty)]\right| dt \;\; ; \;\; e(\infty) = \lim_{t\to\infty} e(t) \tag{17}$$

together with the TV$_0$ and TV$_1$ performance indices (Huba, 2010, 2011a) defined by modification of the Total Variance (Skogestad, 2003) according to

$$TV = \int_0^\infty \left|\frac{du}{dt}\right| dt \approx \sum_i \left|u_{i+1} - u_i\right| \tag{18}$$

$$TV_0 = \sum_i \left|u_{i+1} - u_i\right| - \left|u(\infty) - u(0)\right| \tag{19}$$

$$TV_1 = \sum_i \left|u_{i+1} - u_i\right| - \left|2u_m - u(\infty) - u(0)\right| \tag{20}$$

Thereby, TV$_0$ characterizes deviations from ideally monotonic transients corresponding to TV$_0$=0 and here it will be applied both to the plant output *y* and the plant input *u*. Similarly, TV$_1$ performance index characterizes deviations from ideal shape with one extreme point. It was applied to characterizing transients at the controller output.

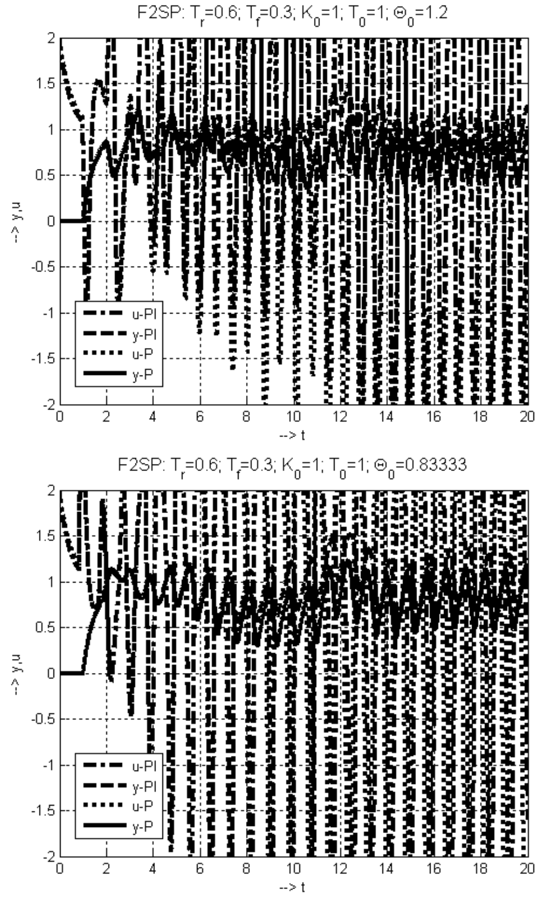


Fig. 7 Comparing P-F2SP and PI-F2SP controllers for the 20% mismatch in the loop dead time Θ.

Performance Portraits calculated for $K = K_0 = 1$ over 20x20 points in the plane of the controller parameters (Θ_0, K_0) in Fig.8 and Fig.9 show practically identical loop properties both for the traditional and modified controllers.

Visualisation of the loop performance in Fig.9 clearly confirms the already mentioned fact that the solutions with the 2nd order filter (6b) are practically not usable over reasonable part of the whole range of the plant-model dismatch, since the deviations from the ideal nominal shapes and the IAE values rapidly increase.

With respect to the u-TV$_0$ and u-TV$_1$ values it is also evident that for this plant and tuning the control signal always has one dominant pulse – it may never be considered to be monotonic.

CONCLUSIONS

New formulations of the Filtered Smith Predictor were proposed based on simplified primary controller for the fast plant dynamics. Thereby, disturbance filters of different complexity may be used that broader spectrum of the available closed loop properties. Two such solutions using primary 2DOF P controller and denoted as the P-F1SP and P-F2SP corresponding to the 1st and the 2nd order filters were considered and compared with the traditional controllers based on the primary PI control (Normey-Rico et al., 2009;

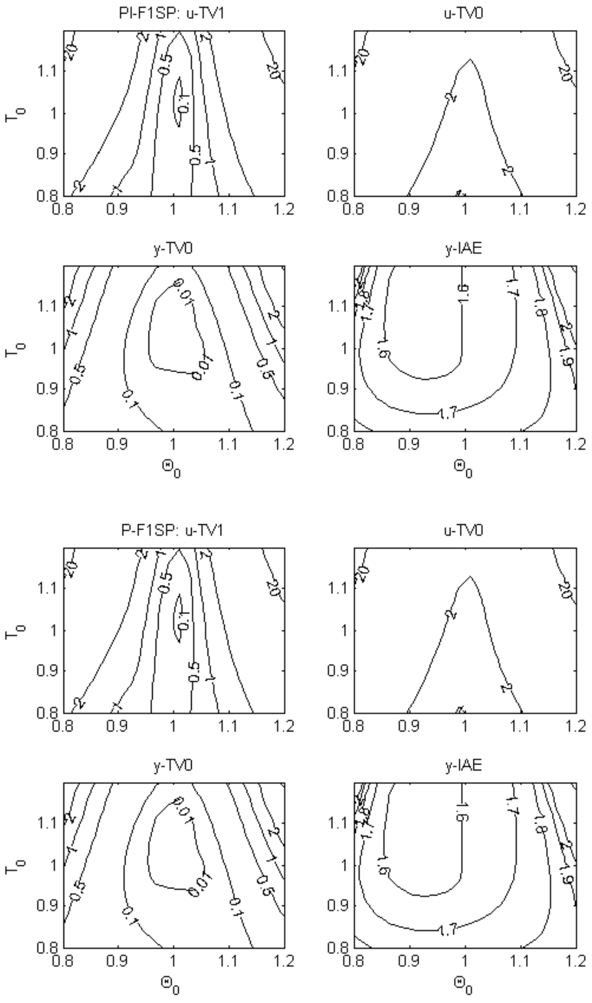


Fig. 8 Performance Portraits of the PI-F1SP and P-F1SP controllers for $K = K_0 = 1; \Theta = 1$ calculated over 20x20 points in the plane of the controller parameters (Θ_0, K_0) are practically identical.

Normey-Rico and Camacho, 2007; 2008; 2009) with equivalent filters denoted as PI-F1SP and PI-F2SP.

To be fair in comparing the alternative solutions, the experiments included both the constrained as well as the unconstrained case, i.e. the case with a sufficiently 'slow' tuning parameters T_r (setpoint response time constant) and T_f (disturbance response time constant). The aim of the simulation experiments was to show, how far the analytical nominal equivalence of the solutions with particular filter does hold in “real” work conditions. From this point of view the experiments showed one unexpected feature - for both alternatives use of the 2nd order filter (6b) considered as ideal decoupling tool is not appropriate in closed loops with a higher dead time uncertainty.

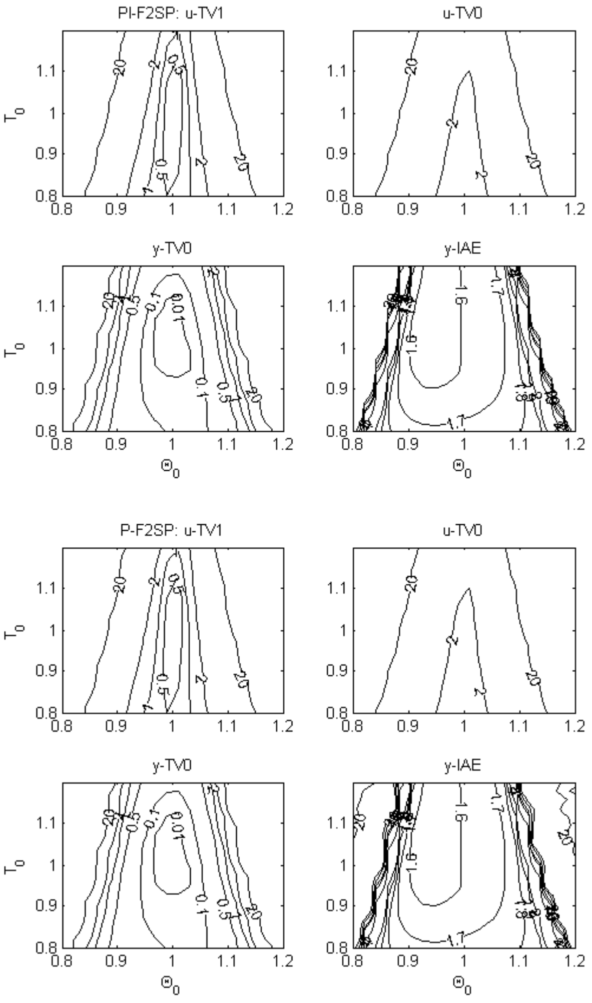


Fig. 9 Performance Portraits of the PI-F2SP and P-F2SP controllers for $K = K_0 = 1; \Theta = 1$ calculated over 20x20 points in the plane of the controller parameters (Θ_0, K_0) are practically identical.

It is just to note that the considered situations do not exhaust whole spectrum of appropriate filters, since also filters with a higher relative degree may be interesting due to higher closed loop robustness and also to higher suppression of the measurement noise that was up to now not focused more deeply. In this context, a more detailed investigation of the role of the disturbance filters in influencing the achievable closed loop robust performance would be desirable that could e.g. be based on the newly developed method of the Performance Portrait Huba (2010, 2011a,b).

The essential advantage of the new solutions is that they do not generate the windup effect - due to the IMC structure and due to replacing the PI controller typically generating windup by the 2DOF P controller. So, in contrast to the PI-FSP controllers that suffer by windup and thus they require appropriate anti-windup measures (Zhang and Jiang, 2008) that make the overall compensator design yet more complex and unsearchable, the P-FSP controller may be used for constrained control without such additional measures.

Consideration of the new structure of the primary loop becomes important not just from the point of view of constrained control. It enables a much easier explanation of similarities with the Model Driven 2DOF PID controllers (MD 2DOF PID, Yukitomo et al. 2004) introduced as a generalization of the Internal Model Control (Morari and Zafiriou, 1989; Skogestad and Postlethwaite, 1996), of the Predictive PI control (PPI) (Hägglund, 1996) and of the PIDτd control (Shinskey, 2000) and covering both systems with short and long dead times.

It represents also one step ahead in direction to considering structures appropriate for nonlinear plants. In such a situation, static feedforward control of the fast loop represented in the linear case by the inverse model gain will be accomplished as the inverse static input-to-output plant characteristic. This will give improved initial values for the generated inversion of the plant dynamics that will finally contribute to increased closed loop performance.

Yet before dealing with nonlinear plants it will, however, be important to deal with controlling unstable and marginally stable plants, where the FSP (despite the proclaimed ability to deal with unstable plant poles) does not represent an acceptable solution.

ACKNOWLEDGEMENTS

The work has been partially supported by the grant VEGA-1/0656/09, KEGA 3/7245/09 and the grant NIL-I-007d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co financed from the state budget of the Slovak Republic.

REFERENCES

Åström, K.J. and T. Hägglund (2005). *Advanced PID Control*. ISA, Research Triangle Park, NC.

Föllinger, O. (1993). *Nichtlineare Regelungen*. 7. Auflage, R. Oldenbourg Verlag München.

Glattfelder, A.H. und Schaufelberger, W. (2003). *Control Systems with Input and Output Constraints*. Springer, London.

Guzmán, J.L. et al. (2008). Interactive tool for analysis of time - delay systems with dead time compensators. *Control Eng. Practice*, 16, 7, 824-835.

Hägglund, T. (1996). An industrial dead-time compensating PI-controller. *Control Engineering Practice* 4, 749-756

Hsu, J.C. and Meyer, A.U. (1968). *Modern Control Principles and Applications*, McGraw-Hill

Huba, M. (2010). Designing Robust Controller Tuning for Dead Time Systems. *IFAC Int. Conf. System Structure and Control*, Ancona, Italy.

Huba, M. (2011a). Basic Notions of Robust Constrained PID Control. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim, 2011.

Huba, M. (2011b). Basic fundamental controllers of DC0. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim, 2011.

Liu, T., Zhang, W. and D. Gu (2005). Analytical design of two-degree-of-freedom control scheme for open-loop unstable processes with delay, *J. Process Control* 15, 559–572.

Lu, X. et al. (2005). A double two-degree-of-freedom control scheme for improved control of unstable delay processes. *J. Process Control* 15, 605–614

Morari, M. and E. Zafiriou (1989). *Robust Process Control*. Prentice Hall, Englewood Cliffs, N.Jersey.

Normey-Rico J.E. et al.(1997). Improving the robustness of dead-time compensating PI controllers, *Control Eng. Practice* 5, 6, , 801–810.

Normey-Rico, J.E.. and Camacho, E.F. (2007). *Control of dead-time processes*. Springer.

Normey-Rico, J.E., Camacho, E.F. (2008). Dead-time compensators: A survey. *Control Eng. Practice*, 16, 4, 407-428.

Normey-Rico,J.E. et al. (2009). An unified approach for DTC design using interactive tools. *Control Eng. Practice*, 17, 1234–1244.

Normey-Rico, J.E., Camacho, E.F. (2009). Unified approach for robust dead-time compensator design. *J. Process Control*, 19 (1), pp. 38-47.

Panda, R.C. (2009). Synthesis of PID controller for unstable and integrating processes. *Chemical Engineering Science*, Vol. 64, 12, 2807-2816.

Shinskey, F.G. (2000). PID-Deadtime Control of Distributed Processes. *Preprints IFAC Workshop on Digital Control: Past, present and future of PID Control PID'2000*, Terassa, Spain, 14-18.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control Analysis and Design*, John Wiley, N.York.

Smith, O.J.M. (1957). Closer Control of Loops with Dead Time. *Chem.Eng.Prog*., 53, 5 217-219.

Tan, K.K. et al. (2010). Deadtime compensation via setpoint variation, *J. Process Control*, 20, 7, 848-859

Visioli, A. (2006). *Practical PID Control*. Springer London.

Vrecko D., Vrancic D., Juricic D., Strmcnik S.: A new modified smith predictor: The concept, design and tuning (2001) *ISA Trans.*, 40 (2), 111-121.

Wang, Q.G., Bi, Q., Zhang, Y: Re-design of Smith predictor systems for performance enhancement *ISA Transactions*, Vol. 39, 1, 2000, 79-92.

Yukitomo, M.; Shigemasa, T.; Baba, Y.; Kojima, F.: A two degrees of freedom PID control system, its features and applications. *5th Asian Control Conf.*, 2004, Vol.1, 456 - 459.

Zhang, B. and W Zhang (2006). Two-degree-of-freedom control scheme for processes with large time delay. *Asian Journal of Control*, Vol. 8, 1, 50-55.

Zhang, M., Jiang, C.: Problem and its solution for actuator saturation of integrating process with dead time. *ISA Transactions*, Vol. 47, 1, 2008, 80-84.

Zhang, W., Rieber, J.M., Gu, D. (2008). Optimal dead-time compensator design for stable and integrating processes with time delay. *J. of Process Control*, Vol. 18, 5, 449-457.

Comments – Remarks

Filtered Smith Predictor Tuning by the Performance Portrait Method

Mikuláš Huba*,**

** Institute of Control and Industrial Informatics, FEI STU, Ilkovičova 3, SK-812 19 Bratislava*
Slovakia (Tel: +421-2 -60291 771; e-mail: mikulas.huba@ stuba.sk).
*** MI/PRT, FernUniversität in Hagen, Universitätsstr. 27, D-58084 Hagen*
Germany (e-mail: mikulas.huba@fernuni-hagen.de)

Abstract: The paper shows tuning of the Filtered Smith Predictor for higher performance requirements expressed in terms of the monotonicity at the plant and the controller output by the Performance Portrait method (Huba, 2010; 2011a).

Keywords: Robustness analysis; robust controller synthesis; robust time-delay systems.

1. INTRODUCTION

The Smith Predictor (SP) may be considered as the best known Dead-Time Compensator (DTC) used for controlling systems with dead time (Smith, 1957). Complexity of its tuning, together with its importance for practice and problems related to its robust tuning lead to the situation that despite its relatively long history it is still in focus of current research (Åström and Hägglund, 2005; Guzmán et al., 2008; Normey-Rico and Camacho, 2007; 2008; 2009; Normey-Rico et al., 2009; Panda, 2009; Tan et al., 2010; Zhang, Rieber and Gu, 2008).

As it was shown in many above mentioned contributions, an increase of the dead-time values with respect to the dominant plant time constant leads in the loops with PID controllers to rapid performance deterioration. DTCs proposed to eliminate dead-time influence are being robustly designed for a nominal plant representing whole family of plants defined by a norm-bounded multiplicative uncertainty error and the required performance is then tuned in the frequency domain. The aim is to keep the modeling error within prescribed limits also for the plant with the largest deviation from the nominal one. In carrying out such a robust control design one may e.g. use the phase, or the gain margin, the maximum sensitivity (Rivera et al., 1986; Skogestad and Postlethwaite, 1996).), or the dead-time margin (Ingimundarson and Hägglund, 2002). However, when aiming to design control loops and having higher requirements on the resulting control quality that should be respected by all plants of a given family, the frequency domain approach shows to be not sufficiently effective.

The filtered Smith Predictor (FSP, Normey-Rico et al., 1997) was proposed for stable First Order Plus Dead Time (FOPDT) processes to improve robustness of the traditional SP and to decouple the reference setpoint and the disturbance response by adding an additional degree of freedom for the loop tuning. It is based (Fig. 1) on the dynamical feedforward control with the reference plant model (Åström and Hägglund, 2005; Visioli, 2006) enabling the disturbance reconstruction and compensation. The approach to designing FSPs for the FOPDT plants introduced in Normey-Rico and Camacho (2009), or Normey-Rico et al. (2009) considers compensation of a disturbance acting at the plant output by correction of the loop reference value, whereby the disturbance is reconstructed by using parallel plant model.

Basic Acronyms	
DO	Disturbance Observer
DTC	Dead Time Compensator
FOPDT	First Order Plus Dead Time
FSP	Filtered Smith Predictor
MO	monotonic, monotonicity
PP	Performance Portrait
SP	Smith Predictor
ST	stabil, stability

In this paper the new computer analysis of control loops based on generating the closed loop Performance Portrait (Huba, 2010; 2011a) is briefly described to be used in:

- designing,
- comparing and
- visualizing

properties of the considered FSP. The key question is, how to robustly design control loop with stable dead-time dominated plant to keep the plant input and output “nearly monotonic” in the case of possible plant-model mismatch. To cope with this problem, it is firstly necessary to introduce measures for evaluating deviations of real responses from ideal monotonic (MO) shapes and then to propose algorithms processing achieved information.

As it is well known, it is not enough to demonstrate the quality of DTCs just by the responses measured in the nominal case with perfect plant-model matching. Their robustness (see e.g. Normey-Rico and Camacho, 2007) is then usually shown for the dead-time uncertainty in the range of ±10-20% of its nominal value. In some papers, also the case with a gain estimation error in the range ±10-20% of the nominal value is shown. So, in total, the DTC performance is being demonstrated in 3-6 different working points. Consideration of several points is usually not possible with respect to the space constraints. This is, however, still far from being enough for fully characterizing such a complex closed loop performance as the DTC have. However, by introducing appropriate performance measures and by using appropriate method for compressing relevant information the loop performance may be characterized by information extracted from hundreds, or even thousands of experiments – each one carried out for a different working point. The interesting shape and time related closed loop properties are mapped, stored, visualized and used for the robust control design based on a compressed form of relevant information representation related to all possible working points of the uncertainty area called here as the closed-loop *Performance Portrait* (PP). The PP method is then used for evaluating and comparing properties of different DTCs.

The paper is structured as follows. Chapter 2 introduces basic information on FSP for FOPDT plants. Chapter 3 recalls basic performance measures for evaluation of nearly monotonic step responses. In Chapter 4 PP corresponding to the tuning achieved by the traditional method in the frequency domain is briefly discussed. Chapter 5 deals with retuning the FSP to yield defined deviations from monotonicity, Chapter 6 yields discussion with results achieved by simple plant approximation and finally Chapter 7 brings conclusions to the paper.

2. FSP FOR STABLE FOPDT PLANTS

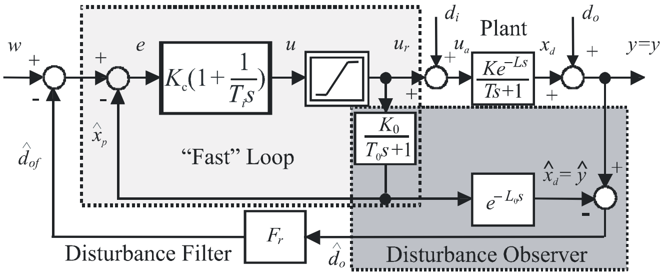


Fig. 1 Filtered Smith Predictor

For the stable plant $P(s)$ with $P_0(s)$ denoting its "fast" delay-free nominal dynamics and $P_n(s)$ its nominal model with particular set of parameters considered in controller tuning

$$P(s)=\frac{Ke^{-Ls}}{Ts+1}\ ;\ P_0(s)=\frac{K_0}{T_0 s+1};\ P_n(s)=\frac{K_0 e^{-L_0 s}}{T_0 s+1} \tag{1}$$

the primary PI-controller

$$C_0(s)=\frac{U(s)}{E(s)}=K_c\frac{1+T_i s}{T_i s}\ ;\ T_i=T_0\ ;\ K_c=\frac{T_0}{T_r K_0} \tag{2}$$

is used, whereby T_r is the time constant of the (fast) primary loop described by the transfer function

$$C(s)=\frac{U(s)}{E(s)}=\frac{C_0}{1+C_0 P_0}=\frac{1}{K_0}\frac{1+T_0 s}{1+T_r s} \tag{3}$$

The nominal (reference) setpoint-to-output transfer function (considering $P=P_n$) is

$$H_w(s)=\frac{Y(s)}{W(s)}=C(s)P(s)=\frac{e^{-Ls}}{1+T_r s} \tag{4}$$

In order to increase the closed loop robustness, it was proposed to extend the disturbance compensation loop by the 1st, order disturbance filter

$$F_r(s)=\frac{1}{1+T_f s} \tag{5}$$

where T_f represents its time constant. The loop may be simplified by introducing equivalent controller for the primary loop

$$C_e(s)=C/(1-CPF_r) \tag{6}$$

In the nominal case with $P=P_n$ the loop transfer functions corresponding to the output disturbance d_o, to the input disturbance d_i become

$$\begin{aligned} H_o(s)&=\frac{Y(s)}{D_o(s)}=\frac{1}{1+C_e P F_r}=1-\frac{F_r(s)e^{-Ls}}{1+T_r s} \\ H_i(s)&=\frac{Y(s)}{D_i(s)}=\frac{P}{1+C_e P F_r}=P(s)\left[1-F_r(s)\frac{e^{-Ls}}{1+T_r s}\right] \end{aligned} \tag{7}$$

In order to get rejection of piecewise stable input and output disturbances at least in steady states, requirements

$$H_o(0)=0\ ;\ H_i(0)=0 \tag{8}$$

have to be fulfilled. Filter (5) enables to hold this requirements and gives nominal transfer functions

$$\begin{aligned} H_o(s)&=\frac{Y(s)}{D_o(s)}=1-\frac{1}{(1+T_f s)(1+T_r s)}e^{-Ls} \\ H_i(s)&=\frac{Y(s)}{D_i(s)}=\frac{Ke^{-\theta s}}{Ts+1}\left[1-\frac{1}{(1+T_f s)(1+T_r s)}e^{-Ls}\right] \end{aligned} \tag{9}$$

The key question of the robust design is, how to choose the tuning parameters K_0, T_0, L_0, T_r and T_f of the FSP in order to guarantee required performance measures for some family of possible plant models given e.g. in form

$$K\in(K_{\min},K_{\max}), T\in(T_{\min},T_{\max}), L\in(L_{\min},L_{\max}) \tag{10}$$

3 PERFORMANCE MEASURES FOR EVALUATING NEARLY MONOTONIC RESPONSES

3.1 Stable (ST) control and indication of instability

In the following analysis, quasi-continuous signals sampled with a sampling frequency enabling to preserve all their important features will be considered.

In the time domain the property of stability (ST) and monotonicity (MO) may be easily tested numerically by evaluating simulated or experimentally measured transients, in our case the setpoint step responses. The Bounded-Input-Bounded-Output (BIBO) and Internal Model Control (IMC) stability require for a bounded plant input $u(t)$ a bounded plant output $y(t)$ that for a given limits $Y_{\max}, U_{\max}$ can be tested e.g. by means of

$$0 \le |y(t)| < Y_{\max} < \infty;\ 0 \le |u(t)| \le U_{\max} < \infty;\ t \in \langle 0, \infty\ ) \tag{11}$$

In this paper we are dealing with constrained continuous signals, as e.g. the plant output $y(t)$ having an initial value $y_0 = y(0)$and a final value $y_\infty = y(\infty)$. From the controller design it is expected to achieve stability for all possible operating points and besides of this trivial requirement simultaneously to achieve required shape of transients at the plant input and output. In doing so, stable control may further be characterized by performance indices such as IAE (Integral of Absolute Error), or TV (Total Variance, Skogestad, 2003). For a setpoint step $w(t)$

$$IAE = \int_0^\infty |e(t) - e(\infty)| dt;\ e(t) = w(t) - y(t); \tag{12}$$

$$TV = \int_0^\infty \left|\frac{du}{dt}\right| dt \approx \sum_i |u_{i+1} - u_i| \tag{13}$$

Stability, or more precisely instability degree can be indicated by values of the parameters Y_{max} and U_{max} in (11), or in many different ways – by achieved IAE, or TV values, by the maximal overshooting, damping ratio, etc. When these measures increase over some value chosen e.g. as a multiple of optimal value, transients may be denoted as unstable.

3.2 Monotonic and Nearly, or ε-Monotonic Control (ε-MO)

Definition 1

A constrained quasi-continuous plant output $y(t)$ having an initial value $y_0 = y(0)$and a final value $y_\infty = y(\infty)$, will be denoted as monotonic when its all samples fulfill condition

$$[y(t_2) - y(t_1)] sign(y_\infty - y_0) \ge 0\ ,\ 0 \le t_1 < t_2 < \infty \tag{14}$$

Output MO may be motivated e.g. by comfort of passengers in traffic control, or at the controller output by precision increase in systems with actuator hystheresis, by energy savings in actuators, by minimizing their wear, generated noise and vibrations, etc. In engineering application dealing with not-perfect systems and finite measurement and processing precision we need measures for enumerating deviations of achieved responses from strict monotonicity.

A simple measure introduced by Åström and Hägglund (2004) for evaluating deviations from strict MO denoted as monotonicity index was defined as

$$\alpha = \int_0^\infty h(t) dt \Big/ \int_0^\infty |h(t)dt|$$

where $h(t)$ is the closed loop impulse response. For a strictly MO response $\alpha = 1$. In this paper for evaluating deviations from strict monotonicity we are going to use a similar and very simple integral measure achieved by modification of the TV criterion that can easily be modified also for evaluating more complex shapes consisting of several monotonic intervals (Huba, 2010; 2011). Since it is frequently used for evaluating the controller output $u(t)$, its definition is shown for this variable.

Definition 2

For a quasi-continuous signal $u(t)$ with the initial value $u(0)$ and the final value $u(\infty)$ the TV$_0$ criterion is given according to

$$TV_0 = \sum_i |u_{i+1} - u_i| - |u(\infty) - u(0)| \tag{15}$$

TV$_0$=0 just for strictly MO response, else it gives positive values. It may be applied also to the plant output evaluation.

Besides of this integral measure for characterizing deviations from strict MO it is important to introduce measure also for the amplitude deviations based on defining an error band specified by the parameters ε_y, or ε_u around the reference MO signal.

Definition 3

A continuous nearly MO signal $y(t)$ with the initial value $y_0 = y(0)$and with the final value $y_\infty = y(\infty)$ will be denoted as ε_y-monotonic when it fulfills condition

$$[y(t) - y(t-T)] sign(y_\infty - y_0) \ge -\varepsilon_y\ ,\ T \le t < \infty, T \in (0, T_{\max}) \tag{16}$$

for any $T_{\max} > 0$.

In order not to prolong time required for testing with any positive $T_{\max}$, this value has to enable capturing sufficient part (e.g. half-period) of the superimposed signal. Number of samples that need to be tested (Huba, 2010; 2011) may be decreased due to:

Theorem 1

Constrained continuous signal $y(t)$ having an initial value $y_0 = y(0)$and a final value $y_\infty = y(\infty)$ with local extreme points $y_{lei} = y(t_{lei});\ i = 1,2,...$ is ε_y-monotonic, if all subsequent local extreme points y_{lei} fulfill condition

$$[y_{le,i+1} - y_{le,i}] sign(y_\infty - y_0) \ge -\varepsilon_y\ ,\ \ ;\ i = 1,2,3,... \tag{17}$$

Proof: Follows from the fact that the maximal signal increase in the direction opposite to $y_\infty - y_0$ in (16) will be constrained by two subsequent extreme points.

By comparing measured amplitude and integral deviations of a signal it is possible to get additional information about its character. So e.g. for signal $y(t)$ having an initial value $y_0 = y(0)$ and a final value $y_\infty = y(\infty)$ with a single overshooting it will hold y-TV$_0$=2ε_y.

3.3 Disturbance response

In evaluating disturbance step responses one can test the same properties as for the setpoint response, but to note that:

1) Immediately after a disturbance step change the plant output starts to rise (fall) and controller needs some time to reconstruct new disturbance value, to balance its effect and to reverse output to move back to the reference value. So, the evaluation of MO output increase (decrease) may start just after its turnover.

2) By a prefilter in the reference signal it is possible to slow down dynamics of the setpoint step responses – in this way it is possible to get tighter disturbance rejection (possibly with overshooting) by simultaneously having MO setpoint step response.

In general, MO areas for disturbance response are different from those corresponding to setpoint response and the controller design has to compromise these differences.

4 TUNING PROPOSED BY NORMEY-RICO AND CAMACHO (2007)

The illustrative example of the FSP tuning is taken from Normey-Rico and Camacho (2007), Example 6.1. The uncertain plant to be controlled is given as

$$F(s) = \frac{Ke^{-Ls}}{(s+1)(0.5s+1)(0.25s+1)(0.125s+1)}\ ; \qquad (18)$$
$$K \in \langle 0.8, 1.2 \rangle\ ;\ L \in \langle 9, 12 \rangle$$

The FSP with the primary PI controller (2) was tuned using standard robust approach in the frequency domain based on a nominal plant and norm bounded multiplicative uncertainty. As the nominal model approximation of the original plant by the FOPDT system (1) with

$$K_0 = 1;\ T_0 = 1.5\ ;\ L_0 = 10.5 \qquad (19)$$

was used. Robust stability was proven for $K_c = 1;\ T_i = T_0$;

$$T_f = L_0 / 2 = 5.25 \qquad (20)$$

By simulating the closed loop step responses for all possible values K and L it is possible to get the PP in Fig. 2. Thereby, output y_1 corresponds to the output of the plant (18), $y_0(t) = Ku(t)$ corresponds to the actual input of the plant multiplied by the plant gain K. By displaying information about these two outputs one gets also information about any other possible output corresponding to different dislocation of dynamical terms involved in (18).

The first not explained point of the original paper by Normey-Rico and Camacho (2007) is, why the effect of the plant time constants in (18) was approximated by $T_0 = 1.5$. As it is obvious from Fig. 2, especially for $K > 1$ this loop does not fulfill higher requirements on MO of transients, expressed e.g. by the amplitude related deviations, or by TV$_0$ values. So, despite giving good initial tuning, the method does not enable finer tuning required by more advanced applications. The loop properties cannot be simply improved even by a radical increase of the filter time constant and require a complete retuning.

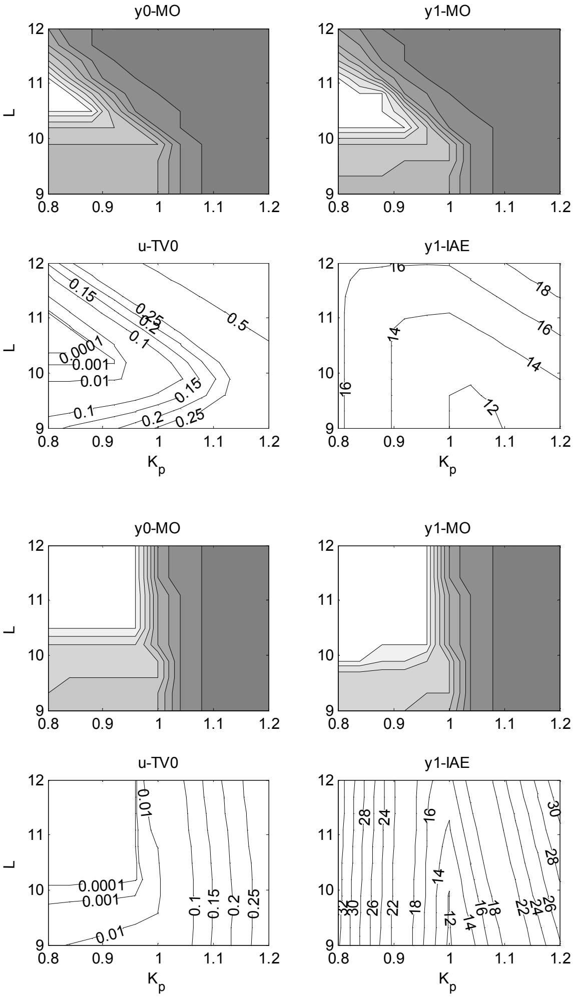


Fig. 2 PP of the plant (18) in the plane of original loop parameters (K_p,L) with the FSP controller based on (19); $T_f = L_n / 2$. ε_y-MO areas identified for the tolerances $\varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ with white denoting the best performance in 11x11 points (above) and $T_f = 10L_n$ (below).

5 RETUNING THE FSP BY THE PP METHOD

Retuning of the FSP controller will be based on the possibility to generate the PP directly for the plant (18) that would suppose possibility to simulate, and or to measure step responses for different FSP tuning and in working points yielding different K and L. Other possibility would be based e.g. on estimating firstly parameters of the FOPDT approximation (1) and on finding appropriate tuning by using the PP corresponding to this model.

When having the possibility to experiment directly on plant (18), by evaluating responses corresponding to different L and K and to controller tuning (19-20) with a variable ratio

$$\kappa = K_0 / K \qquad (21)$$

e.g. for $\kappa \in (0.5, 2.5)$ it is possible to generate the PP in Fig. 3

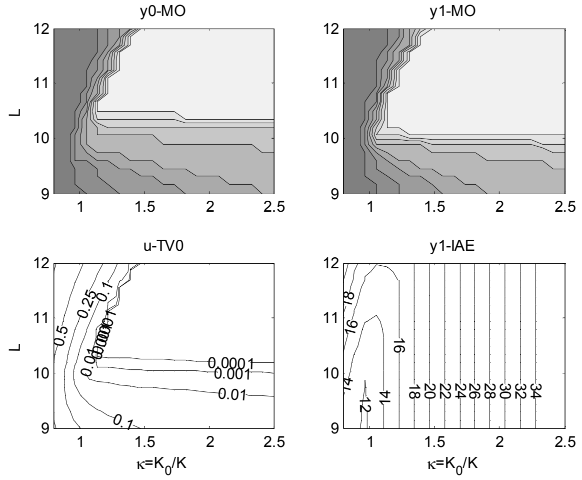


Fig. 3 Example of the PP of the loop consisting of plant (18) and FSP with tuning (19-20). ε_y-MO areas, $\varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ with white denoting the best performance ; 21x21 points.

In a subplane (κ, L) the robust design corresponding to plant (18) and chosen L_0, T_f means to locate the Uncertainty Box

$$UB = \begin{bmatrix} \kappa_{min}, L_{max} & \kappa_{max}, L_{max} \\ \kappa_{min}, L_{min} & \kappa_{max} L_{min} \end{bmatrix};\ \kappa = \frac{K_0}{K};\ \kappa_{max} = \frac{K_{max}}{K_{min}} \kappa_{min} \qquad (22)$$

with vertices corresponding to limit combinations of $\kappa = K_0 / K$ and L to a position in $\varepsilon_y -$MO areas with the given, or better evaluation tolerance and yielding minimal mean IAE values. Solution to the problem over the PP with 21x21 points (Fig. 4) gives for $\varepsilon_y = 0.02$ K_0=1.878>K_{max} with IAE$_{\mathrm{mean}}$=29.08 (output y_1). (It might be shown that by decreasing quantization level K_0 would converge to K_{max}.)

The achieved deviation from MO (including possible overshooting) of all limit step responses in Fig. 5 should not be larger than the tolerable value. By increasing the tolerated deviations from strict MO, the UB may be moved to the areas with lower IAE values (Fig. 6-9). In this point it is to see the main difference of the new method that does not try to choose the nominal point as an internal point of the uncertainty area, what was motivated by the supposed "advantage" of the SP, when for the nominal tuning dead time disappeared from the denominator of (4) and (9). By decreasing ε_y the transients become smoother, the TV$_0$ values decrease, but the IAE values increase.

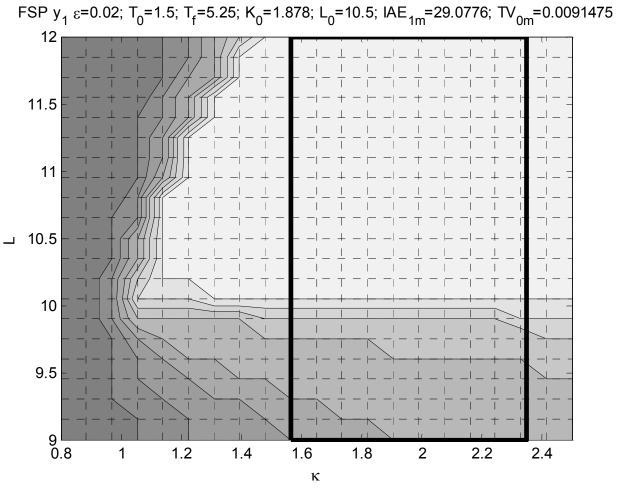


Fig. 4 Example of localizing UB (22) for y_1 into 0.02-MO area to yield minimal mean IAE value. ε_y-MO areas for $\varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$; 21x21 points.

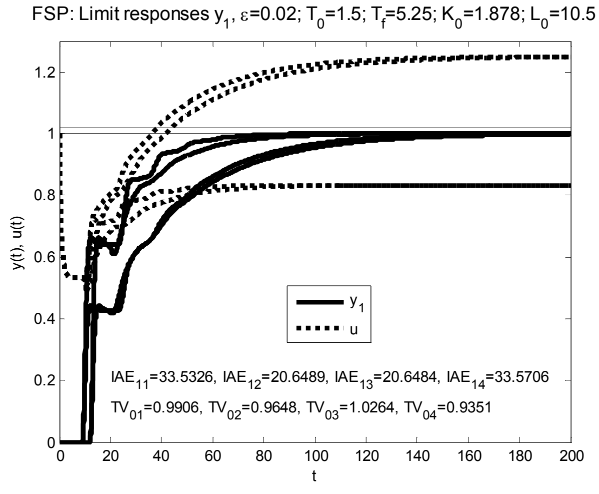


Fig. 5 The limit transients corresponding to UB in Fig. 4

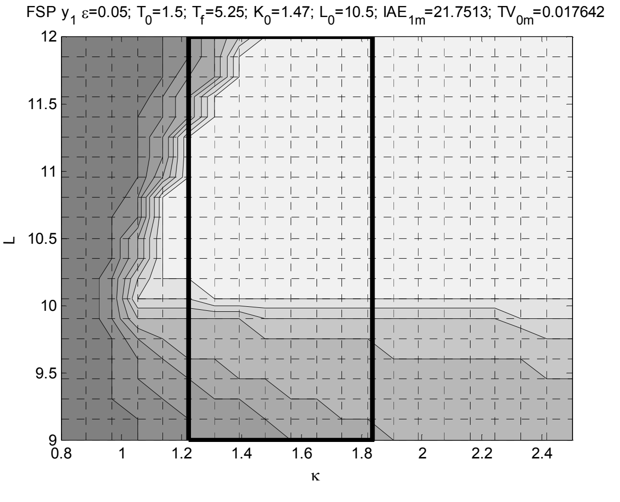


Fig. 6 Example of localizing UB (22) for y_1 into 0.05-MO area with $\varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, white denoting the best performance to yield minimal mean IAE value.

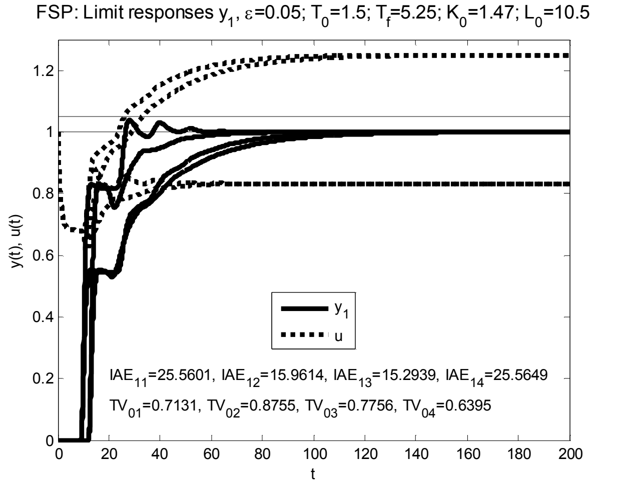


Fig. 7 The limit transients corresponding to UB in Fig. 6

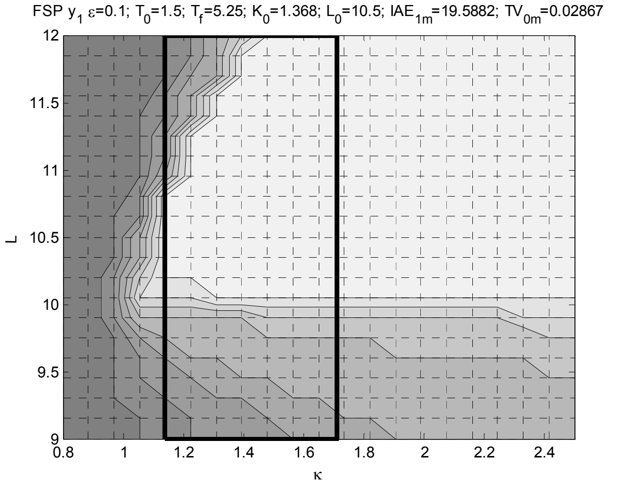


Fig. 8 Example of localizing UB (22) for y_1 into 0.1-MO area in PP with $\varepsilon_y = \left\{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\right\}$, white denoting the best performance to yield minimal mean IAE value.

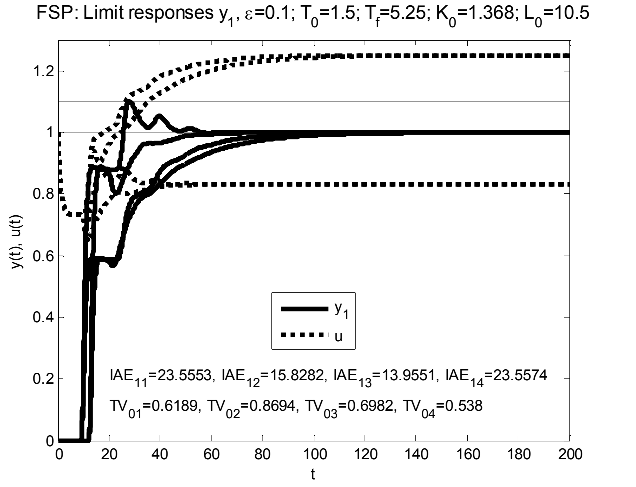


Fig. 9 The limit transients corresponding to UB in Fig. 8

By deriving the PP for $T_0 = 1.0$ it is possible to show that for some deviations from MO this choice yields processes with lower mean IAE values than for $T_0 = 1.5$ (Fig. 12-13). This value was probably determined by the "trial and error" method. The PP method represents a newer and automatized version of this still frequently used method.

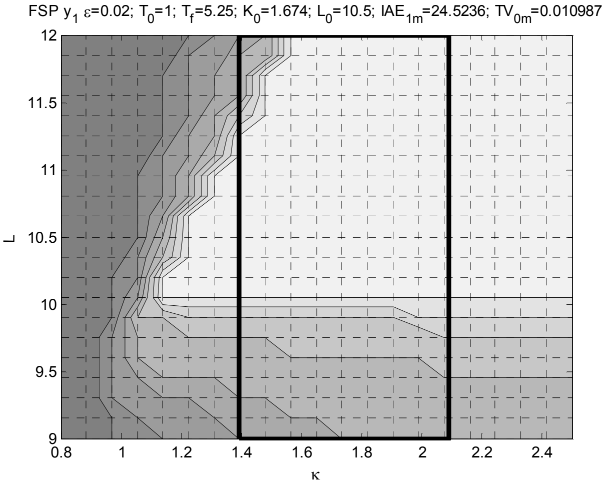


Fig. 10 Example of localizing UB (22) for y_1 into 0.02-MO area in PP, $\varepsilon_y = \left\{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\right\}$, white denoting the best performance, by choosing $K_0 = 1.674$ to yield minimal mean IAE value for $T_f = 5.25; L_0 = 10.5; T_0 = 1$.

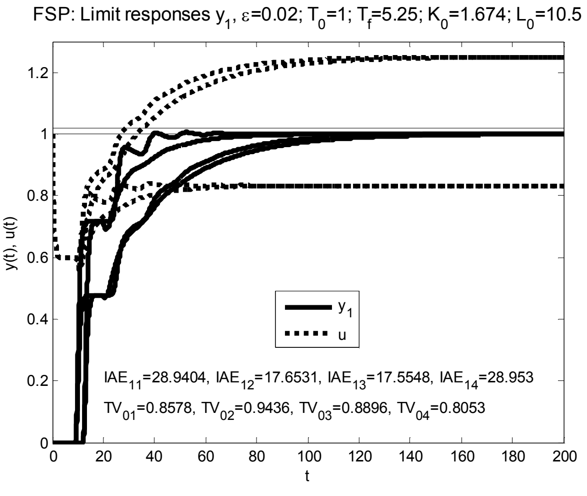


Fig. 11 The limit transients corresponding to UB in Fig. 10

6. DISCUSSION

By being able to retune the FSP to fulfill specified performance requirements, new questions arise with respect to the efficiency of the dominant dynamics approximation (1) used for the DTC design for the plant (18) and with respect to the choice of the primary controller. Here, it will be just briefly mentioned that based on the loop approximation by simple static gain and dead time

$$S(s) = Ke^{-T_d s} \; ; \; K \in \left\langle K_{\min}, K_{\max} \right\rangle ; \; T_d \in \left\langle T_{d\min}, T_{d\max} \right\rangle \qquad (23)$$

whereby information on the plant (18) reduces to sum of its time constants used in deriving limit values of the equivalent dead time

$$T_{d\max} = L_{\max} + S;\, T_{d\min} = L_{\min} + S;\, S = \sum_{1}^{4} T_i = 1.875 \qquad (24)$$

the corresponding Filtered Predictive I controller from the dynamical class 0 (FPrI$_0$) proposed in Huba (2011b) gives for the output y_1 value IAE$_{\text{mean}}$=27.31<29.08, i.e. less that the value corresponding to the first plant approximation with T_0=1.5. Although it was shown that by choosing T_0=1.0 performance of the FSP was improved to IAE$_{\text{mean}}$=24.52, in fact, also the performance of the FPrI$_0$ might be improved by tuning in 2D and using finer quantization and so it is not clear, if the more complex FSP based on more complex plant model will definitely ensure better performance. Although it might seem at the first glance that by increasing order of the model approximating the plant behavior one can only increase performance of the resulting controller, this result shows that also opposite may be true and in the case of dominant uncertain dead time it may be advantageous to reflect just this dominant influence by the plant model.

7. CONCLUSIONS

By analyzing the traditionally difficult task of the FSP tuning we have shown that the methods of robust control design based on expressing the norm-bounded deviations from the nominal model in the frequency domain are strongly limited both in the precision and in spectrum of details they can offer. The Performance Portrait method has firstly shown that the tuning proposed originally in the frequency domain does not fulfill higher quality in terms of amplitude or integral deviations from monotonicity over the whole uncertainty area. Then, it was demonstrated that the monotonicity aspects may be reasonably improved by modifying the tuning parameter K_0 to values lying (due to the quantization) typcally out of the interval $K \in (K_{min}, K_{max})$. It is evident from the shape of identified MO areas that the analysis could be simplified by taking $K_0 = K_{\max}$.

Since it was also shown that the tuning parameter T_0 can be modified to yield lower mean IAE values over the uncertainty area, next the analysis should be repeated for $K_0 = K_{\max}$ in the plane of loop parameters (L, T_0).

By being much more flexible end detailed than the traditional analytical methods and by offering simple solution to the robust controller tuning it is obvious that the newly introduced computer analysis may have strong impact on the control theory and its use in practice. Similarly as the Computer Tomography reasonably influenced procedures used in medicine, strong impact may also be expected in the control context, where the new computer tool allows to visualize any loop property and so dealing with challenging control tasks as e.g. designing Smith predictor, where, the existing interactive tools (Guzmán et al., 2008; Normey-Rico, et al., 2009) bring just limited possibilities to make the traditional “trial and error” controller tuning easier.

The new method directly gives solution matching the chosen performance requirements (if it exists), but it also generates new information that can possibly lead to further improvements and modifications of the traditional analytical methods. Similar development happened also with the method by Ziegler and Nichols (1942), when the proposed formulas achieved originally by evaluating experimental results were later analytically précised and extended.

The presented results could yet be enriched by considering different performance measures, as e.g. those related to more complex control signal shapes consisting of several MO intervals (Huba, 2010; 2011).

Instead of retuning single parameter K_0, for the FSP parameters K_0, L_0, T_0, T_r (denoting the required closed loop time constant for the setpoint response) and T_f have to be tuned in general, with respect to given uncertainty in K, L and T. This is fully possible, but one has to fight with all problems connected with generation and demonstration of the results in 7D space – computation time rapidly increasing with the problem dimension and necessity to illustrate the achieved results by 2D, or 3D projections. Such solutions could, however, give the best answer to the question, which controller tuning gives for a given plant uncertainty and specified performance the best possible tuning parameters.

ACKNOWLEDGEMENTS

The work has been partially supported by the grant VG-1/0656/09 and the grant NIL-I-007d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co financed from the state budget of the Slovak Republic.

REFERENCES

Åström, K.J. and T. Hägglund (2005). *Advanced PID Control*. ISA, Research Triangle Park, NC.

Guzmán, J.L. et al. (2008). Interactive tool for analysis of time - delay systems with dead time compensators. *Control Eng. Practice*, 16, 7, 824-835.

Huba, M. (2010). Designing Robust Controller Tuning for Dead Time Systems. *IFAC Int. Conf. System Structure and Control*, Ancona, Italy.

Huba, M. (2011a). Basic Notions of Robust Constrained PID Control. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim, 2011.

Huba, M. (2011b). Basic fundamental controllers of DC0. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim, 2011.

Liu, T., Zhang, W. and D. Gu (2005). Analytical design of two-degree-of-freedom control scheme for open-loop unstable processes with delay, *J. Process Control* 15, 559–572.

Lu, X. et al. (2005). A double two-degree-of-freedom control scheme for improved control of unstable delay processes. *J. Process Control* 15, 605–614

Morari, M. and E. Zafiriou (1989). *Robust Process Control*. Prentice Hall, Englewood Cliffs, N.Jersey.

Normey-Rico J.E. et al.(1997). Improving the robustness of dead-time compensating PI controllers, *Control Eng. Practice* 5, 6, , 801–810.

Normey-Rico, J.E.. and Camacho, E.F. (2007). *Control of dead-time processes*. Springer.

Normey-Rico, J.E., Camacho, E.F. (2008). Dead-time compensators: A survey. *Control Eng. Practice*, 16, 4, 407-428.

Normey-Rico,J.E. et al. (2009). An unified approach for DTC design using interactive tools. *Control Eng. Practice*, 17, 1234–1244.

Normey-Rico, J.E., Camacho, E.F. (2009). Unified approach for robust dead-time compensator design. *J. Process Control*, 19 (1), pp. 38-47.

Panda, R.C. (2009). Synthesis of PID controller for unstable and integrating processes. *Chemical Engineering Science*, Vol. 64, 12, 2807-2816.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control Analysis and Design*, John Wiley, N.York,.

Smith, O.J.M. (1957). Closer Control of Loops with Dead Time. *Chem.Eng.Prog*., 53, 5 217-219.

Tan, K.K. et al. (2010). Deadtime compensation via setpoint variation, *J. Process Control*, 20, 7, 848-859

Visioli, A. (2006). *Practical PID Control*. Springer London.

Wang, Q.G., Bi, Q., Zhang, Y: Re-design of Smith predictor systems for performance enhancement *ISA Transactions*, Vol. 39, 1, 2000, 79-92.

Zhang, B. and W Zhang (2006). Two-degree-of-freedom control scheme for processes with large time delay. *Asian Journal of Control*, Vol. 8, 1, 50-55.

Zhang, W., Rieber, J.M., Gu, D. (2008). Optimal dead-time compensator design for stable and integrating processes with time delay. *J. of Process Control*, Vol. 18, 5, 449-457.

Comments – Remarks

Robust Controller Tuning for Constrained Double Integrator

Mikuláš Huba*,**

** Institute of Control and Industrial Informatics, FEI STU, Ilkovičova 3, SK-812 19 Bratislava
Slovakia (Tel: +421-2 -60291 771; e-mail: mikulas.huba@ stuba.sk).
** MI/PRT, FernUniversität in Hagen, Universitätsstr. 27, D-58084 Hagen
Germany (e-mail: mikulas.huba@fernuni-hagen.de)*

Abstract: This paper discusses possibilities of the newly developed performance portrait (PP) method in designing as fast as possible transients fulfilling additional shape related performance measures specified at the input and output of the closed loops with the Integrator Plus Dead Time (IPDT) and Second Order Integrator Plus Dead Time (SOIPDT) plants. Broad potential of this new numerical method in arbitrarily shaping the close loop properties of time delayed systems is shown and compared with the traditional analytical controller tuning corresponding to the Multiple Real Dominant Pole (MRDP).

Keywords: Robust Control; Time Delay Systems; Computational Algorithms for Analysis and Design

1. INTRODUCTION

Dead time caused by the energy, mass and information transportation and by signal processing may be met in practically each control application. Plants with significant dead time are not only difficult to control due to intrinsic limitations of the control structure themselves, but many problems are also caused by limitations of the existing methods for their robust and optimal tuning.

Methods available for robust control design of simple control loops based on specification of the closed loop properties in the frequency domain using modifications of the D-decomposition (Neimark, 1973; Hwang and Cheng, 2004), on the phase and amplitude margin, or on the maximum sensitivity (Åström and Hägglund, 1995; Åström et al. 1998; Datta et al., 2000; Keel et al. 2008; Normey-Rico et al. 2009; Skogestad, 2003; 2006; Skogestad and Postlethwaite, 2007) are not primarily focusing on achieving higher control quality expressed in terms of the output monotonicity and overshooting. For achieving smooth, monotonic, but as fast as possible nominal dynamics, analytical controller tuning corresponding to the Multiple Real Dominant Pole (MRDP) may be used (Oldenbourg and Sartorius, 1944; Gorecki, 1971; Viteckova and Vitecek, 2008; 2010). For many applications this tuning is considered to be too conservative and it does not allow respecting robustness issues.

This paper shows simple solution based on the recently developed technique for robust constrained PID control design (Huba, 2010, 2011a,b) enabling mapping of the so called closed-loop performance portrait (PP). It may be considered as a generalization of the well known method for controller tuning used by Ziegler and Nichols (1942) that firstly introduced systematic use of the "trial and error" procedure. Their simple formulas approximate results achieved by carrying out series of experiments on some sample of representative processes under requirement of chosen performance measure (quarter amplitude damping) related to the shape of resulting transient responses. Such an approach of carrying out experiments based on simulation or real time experiments can today be easily performed by using tremendous power of computers for organizing and evaluating such experiments, as well as for processing, visualizing, storing and recalling the achieved results for large number of control loops typical in practice. Thereby, one can easily extend spectrum of different qualitative & quantitative properties that will be evaluated over a grid of normalized loop parameters and then stored in computer database, to be chosen "on demand" and in different combinations by engineer carrying out design requiring particular specifications.

The paper is structured as follows. Chapters 2 and 3 discuss ideal and nearly-ideal shapes of transient responses at the plant input and output and different performance specifications appropriate for evaluating deviations from the ideal shapes and their use in mapping the PP. Chapters 4, 5 and 6 bring illustrative examples with controller design for the Integrator Plus Dead Time (IPDT) IPDT and Second Order Integrator Plus Dead Time (SOIPDT) plants. In Chapter 7, equivalent closed loop poles are introduced important for tuning of constrained pole assignment controllers and the paper is concluded by Chapter 7.

2. PERFORMANCE SPECIFICATIONS

When dealing with the closed loop step responses, in majority of applications, internal plant stability is required. Within the Bounded-Input-Bounded-Output (BIBO) concept, as well as in Internal Model Control (IMC, see e.g. Morari and Zafiriou, 1989) this requires both the bounded plant output $y(t)$ and the bounded plant input $u(t)$, i.e. for some positive $Y_{\max}$ and $U_{\max}$ it must hold

$$|y(t)| < Y_{\max} < \infty\,;\; |u(t)| < U_{\max} < \infty\,;\; t \in \langle 0, \infty\ ) \tag{1}$$

Stable transients (ST) may further be characterized by the *time factor*, i.e. how fast they reach the required state and by *shapes of the transient responses*, for which we may require properties like nonovershooting (NO), monotonicity (MO), maximal overshooting/undershooting, damping ratio (for periodically damped processes), etc.

ACRONYMS	
1P	One-Pulse
2P	Two-Pulse
BIBO	Bounded-Input-Bounded-Output
DRDP	Double Real Dominant Pole
MO, or 0P	Monotonic
MTC	Minimum Time Control
NO	Nonovershooting
nP	*n*-Pulse
IAE	Integral of Absolute Error
IPDT	Integrator Plus Dead Time
ISE	Integral of Squared Error
IMC	Internal Model Control
IPDT	Integrator Plus Dead Time
MRDP	Multiple Real Dominant Pole
P	Proportional
PD	Proportional-Derivative
PP	Performance Portrait
SOIPDT	Second Order Integrator Plus Dead Time
ST	Stable Transients
TRDP	Triple Real Dominant Pole
t_s	Settling Time
TV	Total Variance
ULS	Uncertainty Line Segment
UB, US	Uncertainty Box, Uncertainty Set

2.1 Time related performance measures

For characterizing duration (speed) of transient responses settling time t_s and different integral criteria as e.g. IAE (Integral of Absolute Error), ISE (Integral of Squared Error), or ITAE (Integral of Time multiplied by Absolute Error) criterion are used.

For the output $y(t)$ with the initial value $y_0 = y(0)$and the final value $y_\infty = y(\infty)$, t_s may be defined by the requirement

$$y(t) - y_\infty = 0\,;\; t \in \langle t_s, \infty\ ) \tag{2}$$

However, the transient responses may be theoretically infinitely long and then a finite settling time requires defining a certain error band around the steady state. To decide, if a transient response finished by reaching such error band becomes a delicate problem in a noisy environment. It can e.g. be indicated by fulfilling requirements put both on the plant input $u(t)$and the output $y(t)$

$$|y(t) - y_\infty| \le \varepsilon_y \cap |u(t) - u_\infty| \le \varepsilon_u\,,\; t \in \langle t_s, \infty\ ) \tag{3}$$

Thereby, the parameters $\varepsilon_y, \varepsilon_u$ may depend on particular technology and be chosen in such a way (a) to prevent a premature indication of steady state at flat extreme points of oscillatory transients and (b) to make indication of steady state possibly independent from the measurement noise present in real control. It is to remember that the plant output and the controller output may achieve steady states at different time moments, a fixed output value may be achieved by oscillation at the plant input, or there may exist steady states with permanent nonzero steady state error.

Since the settling time t_s depends on definition of several parameters, in order to characterize the speed and duration of transients (mostly at the plant output), simpler IAE or ISE performance indices may be used defined for a setpoint w as

$$IAE = \int_0^\infty |[e(t) - e(\infty)]| dt\,;\; ISE = \int_0^\infty [e(t) - e(\infty)]^2 dt \tag{4}$$
$$e(t) = w(t) - y(t)$$

Shinskey (1990) argues that IAE is a good performance measure because the size and length of error is proportional to lost revenue, so it is mostly enough to use it instead of ISE that may be convenient for the analytical controller design. Because in optimizing controllers minimal ISE values usually correspond to transients with some overshooting, whereas aiming at transients without overshooting one has to define additional design constraints. Because of increasing weighting by time, the ITAE criterion is not appropriate for dealing with real noisy systems.

2.2 Shape related performance measures

In the minimum time control (MTC) with the integer n denoting the systems full degree, the necessity of n periods of energy accumulation/dissipation during the setpoint step responses was well known as the Felbaum's theorem (Feldbaum, 1965). Instead of the discontinuous transition of the relay control signal typical for the MTC, here we will considered a smooth nearly MO transition from one control signal extreme value to another one and from the last extreme point to the steady state. The only exception of the control continuity may be allowed by considering step transition from an initial actuator value to the first extreme point appearing at $t = 0^+$.

When speaking about shape related performance measures we will start with NO control, MO control and with its generalization to the n-Pulse (nP) control. Then we will introduce weakened versions of these notions and different modifications of the TV criterion used for evaluating integral deviations from the ideal nP shapes. It is to remember that all these properties are considered as child properties of the root loop property ST (1).

2.3 Nonovershooting and Nearly Nonovershooting control

Up to now, specifications of NO control was used in the frequency domain (Keel *et al.*, 2008). This approach did not distinguish between NO and MO control.

In the time domain NO properties are easy to be tested numerically, by evaluating samples of the setpoint step responses. By considering initial and final values $y_0 = y(0)$, $y_\infty = y(\infty)$, and by fulfilling

$$[y(t)-y_\infty]sign(y_0 - y_\infty) \ge -\varepsilon_y\,,\ \forall t>0\ ; \tag{5}$$

the output will be denoted as nearly NO. By introducing several measurement precision levels $\varepsilon_y, \varepsilon_u$ it is possible to replace the true/false information by more detailed quantitative one.

2.4 Monotonic and Nearly Monotonic Control

Final evaluation precision may be introduced also into the MO tests by requiring besides of (1) and (5)

$$[y(t)-y(t-T)]sign(y_\infty - y_0) \ge -\varepsilon_y\,,\ T \le t < \infty, T \in (0, T_h) \tag{6a}$$

T_h has to be chosen long enough to capture amplitudes of higher-harmonics pulses superimposed on the basic MO signal (Fig. 1). One simple solution to speed up computation is to check all subsequent local extreme points y_{lei}, $i = 1,2,...$ of the evaluated signal, if they fulfill condition

$$[y_{le,i+1} - y_{le,i}]sign(y_\infty - y_0) \ge -\varepsilon_y\,,\ \ ;\ i = 1,2,3,... \tag{6b}$$

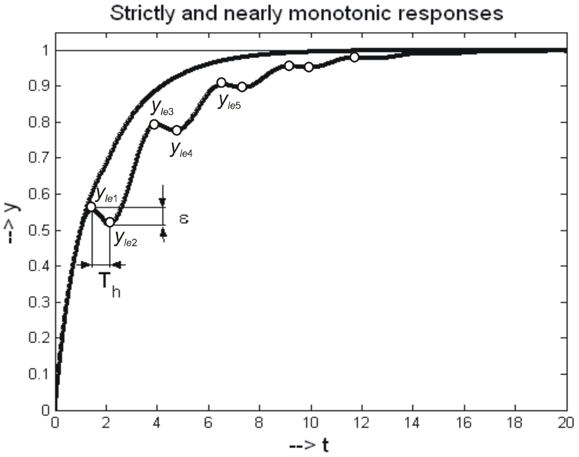


Fig. 1 Strictly MO signal satisfying (6) with $\varepsilon = \varepsilon_y = 0$ and signal that is "nearly monotonic" for $\varepsilon = \varepsilon_y = 0.05$;

2.5 One-Pulse (1P) Control

When considering stable single integrator output changing monotonically from an initial value $y_0 = y(0)$ to a final value $y_\infty = y(\infty)$, it will be increasing (not decreasing) if its derivative is a positive (non negative) function of time, i.e.

$$\dot{y}(t) > 0,\ or\ \ \dot{y}(t) \ge 0,\ \ t \in (0,\infty) \tag{7}$$

Since the single integrator is described by the equation

$$\dot{y}(t) = u_1(t),\ \ t \in (0,\infty) \tag{8}$$

it also means that for $t \in (0,\infty)$ the control $u_1(t)$ must take positive (non negative) values and in the initial and final steady states it holds (Fig. 2)

$$u_1(0^-) = \dot{y}(0) = 0 \text{ and } u_1(\infty) = \dot{y}(\infty) = 0 \tag{9}$$

When considering $u_1(t)$ continuous for $t \ge 0$ and satisfying (9) it means that for some $t_{m1} \in (0,\infty)$ it will have a maximum

$$u_{m1} = u(t_{m1})\ ;\ t_{m1} \in (0,\infty) \tag{10}$$

Under constrained control, when the control signal saturates, the maximum value may also be achieved over an interval $t \in \langle t_{\max 1}, t_{\max 2} \rangle$. It is also obvious that in order to achieve as fast as possible output increase, the maximum u_{m1} should be as large as possible and, in order to keep MO output increase, $u_1(t)$ must remain positive even in the case when it has several extreme points $u_{1e}(t_{ei}); i = 1,2,...$ corresponding to $\dot{u}_1(t_{ei}) = 0$. The simplest control, however, corresponds to situation with $u_1(t)$ having just a single local extreme (10) that separates the overall control into two monotonic intervals: the first one monotonically increasing from $u(0) = 0$ up to $u_{m1} = u(t_{m1})$ and then the second one monotonically decreasing from $u_{m1} = u(t_{m1})$ up to $u(\infty) = 0$. So, we may conclude that in a general case the ideal control guaranteeing MO output transition between two steady state values of single integrator will be characterized by smooth continuous 1P $u_1(t)$ satisfying initial conditions (9) and having one extreme point

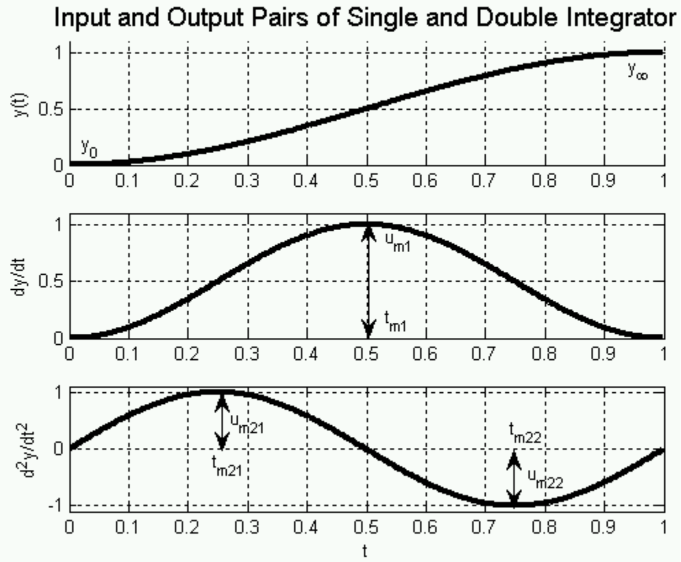


Fig. 2 MO output y satisfying (6) for $\varepsilon = \varepsilon_y = 0$ (above) with the corresponding 1P input signal of single integrator $u_1(t) = \dot{y}(t)$ (middle), or with the corresponding 2P input signal of the double integrator $u_2(t) = \ddot{y}(t)$ (below)

$$u_{m1} = u(t_{m1});\ \dot{u}(t_{m1}) = 0;\ t_{m1} \in (0,\infty) \qquad (11)$$

that is monotonic before and after this extreme point. By accepting possible control discontinuity at $t = 0^+$ the MO output increase finishing by reaching steady state cannot be achieved by simpler (e.g. step) control signal, just the extreme point may move to $t_{m1} = 0^+$.

When denoting the unit step function as $\mathbf{1}(t)$, examples of the 1P functions may be represented by single exponential $f(t) = e^{-t}\mathbf{1}(t)$ that has extreme point $f(0^+) = 1$ and discontinuity at the origin, or by the fully continuous difference of two exponentials $f(t) = (e^{-t} - e^{-2t})\mathbf{1}(t)$ having extreme $f_m = 1/4$ at $t_m = \ln 2$.

2.6 Two-Pulse (2P) Control

Similarly, the 2P function (Fig. 2 below) may be defined as a function continuous for $t > 0$ with two extreme points that is MO on each interval not including one of these extremes.

In order to control the double integrator, one has to put additional integrator in front of the previous one and to consider that for achieving a MO increase of $\dot{y}(t)$ (the earlier input, now output of the added integrator) for $t \in (0, t_{m1})$, the new continuous input $u_2(t)$ must be described by a function having one maximum $u_{m21} > 0$ at an interior point $t_{21} \in (0, t_{m1})$ that divides the whole interval $(0, t_{m1})$ into two MO subintervals $(0, t_{21})$ and (t_{21}, t_{m1}).

During the earlier second phase of control with $t \in (t_{m1}, \infty)$, in order to achieve a monotonic decrease of $\dot{y}(t)$, input of the new integrator $u_2(t)$ must firstly decrease to its minimal value $u_{m22} < 0$ at some $t_{m22} \in (t_{m1}, \infty)$ and then monotonically increase to its final value $u(\infty) = 0$. So, instead of the originally two control intervals, now one has to consider three monotonic control intervals.

By requiring continuous control with the only possible control discontinuity at $t = 0^+$ the MO output increase of the double integrator finishing by reaching steady state cannot be achieved by simpler control, e.g. by a step, or by a 1P signal.

Example of the 2P function with discontinuity at the origin is e.g. given as $f(t) = (2e^{-2t} - e^{-t})\mathbf{1}(t)$ with $t_{m1} = 0^+$; $t_{m2} = 2\ln(2) = 1.3863$. An example of a fully continuous 2P function is $f(t) = 4((1+2t)e^{-2t} - e^{-t})\mathbf{1}(t)$ with extreme points (Fig. 3 below curve) $t_{m1} = -LambertW(-1/4) = 0.357$ and $t_{m2} = -LambertW(-1,-1/4) = 2.153$.

2.7 n-Pulse (nP) Control

Function of time $f(t)$ that is continuous for $t > 0$ (with possible discontinuity at $t = 0^+$) with the initial value $f(0^-) = \lim_{t \to 0^-} f(t)$, having for $t > 0$ n extreme points with respect to the finite final value $f(\infty) = \lim_{t \to \infty} f(t)$, whereby

$$f_{mni} = f(t_{mni}),\ i = 1,\dots,n \text{ at } 0 < t_{mn1} < \dots < t_{mnn} \qquad (12)$$

$$[f_{mn,i} - f(\infty)][f_{mn,i+1} - f(\infty)] < 0;\ i = 1,\dots,n-1 \qquad (13)$$

that is MO on each interval not including one of these extreme points will be denoted as the *n*-Pulse (nP) function. By allowing discontinuity of $f(t)$ at $t = 0^+$, the first extreme point may also move to $t_{mn1} = 0^+$, whereby the first MO interval $(0, t_{mn1})$ before this extreme point shrinks to zero.

By introducing notion of the nP function it is possible to denote the MO transients also as the 0P one.

To cover whole spectrum of typical control transients we might yet complete the above list by definition of periodic function interpreted as nP function with $n \to \infty$. Then, after specifying e.g. the damping ratio (as done by Ziegler and Nichols, 1942) we could treat also oscillatory loop behavior.

2.8 Nearly n-Pulse Control

Next, we will pay attention also to situation with higher harmonics superimposed on the ideally shaped signals what results in signals with several local extremes y_{lei}, $i = 1, 2, \dots$.

In order to be recognized as significant extreme points, the local extremes must fulfill special condition – they must change their position with respect to $f(\infty)$, i.e. it must hold (13). In other words, two subsequent significant maxima must always be separated by a significant minimum, else they still corresponds to the same pulse.

After specifying significant extreme points from the available local extremes, each interval between two such extremes, between the origin and the first extreme point and between the last extreme point and the steady state will be examined regarding the admissible deviation from monotonicity (6).

By considering the weakened definition of MO transients (6), it is possible to introduce nearly nP transients, for which each interval among the significant extreme points (12) satisfies to (6). It is to expect that by considering final evaluation precision $\varepsilon > 0$ some extreme points not violating conditions on overshooting, or on allowed deviations from the monotonicity may be omitted and the control signal be characterized as mP with $m < n$, whereby *n* denotes number of identified extreme points for $\varepsilon \to 0$.

2.9 Integral measures for deviations from ideal nP shapes

Whereas (6) enables to evaluate the maximal amplitude deviations from the strict monotonicity, by modifications of the Total Variance (TV) proposed by Skogestad (2003) as

$$TV = \int_0^\infty \left|\frac{du}{dt}\right| dt \approx \sum_i |u_{i+1} - u_i| > 0 \qquad (14)$$

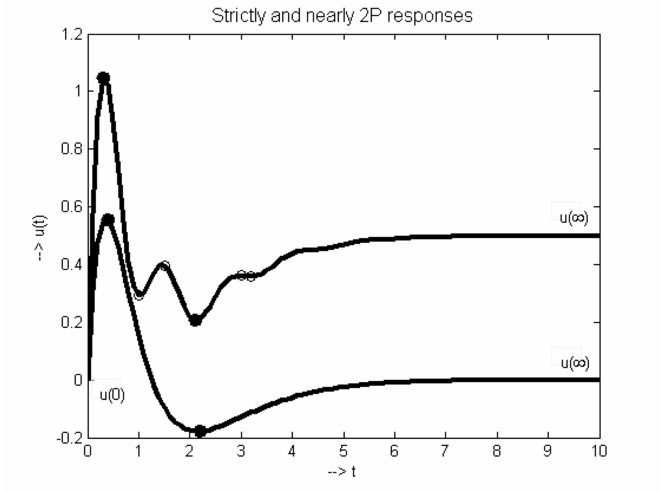


Fig. 3 Nearly and strictly 2P responses; local extreme points denoted by "o" and significant extreme points denoted by "●"

it is possible to introduce integral measures for such deviations. The TV_0 criterion defined as

$$TV_0 = \sum_i |u_{i+1} - u_i| - |u(\infty) - u(0)| \geq 0 \tag{15}$$

takes zero values just for strictly MO control signal transients. TV_0 gives total contribution of superimposed high-frequency signals to the overall control effort that is proportional not just to the amplitude of deviations, but also to the number of peaks. Therefore, it may be applied both to the plant output and to the plant input analysis.

Requirement on the smooth shapes of transients with piecewise MO segments may be motivated by energy savings in actuators, by minimizing actuator wear, generated noise and vibrations, by comfort of passengers in flight control, by precision increase in controlling systems with actuator hysteresis, etc.

Whereas in controlling stable plants it is possible to decrease the number of control pulses up to zero by keeping MO controller output, in controlling unstable and integral plants the number of significant control pulses cannot decrease below the number of unstable poles. To stress contribution of the superimposed oscillation in systems with 1P dominant control it is then appropriate to work with the TV_1 criterion defined as

$$TV_1 = \sum_i |u_{i+1} - u_i| - |2u_m - u(\infty) - u(0)| \geq 0 \tag{16}$$

This gives zero values just for strictly 1P control signal, whereby it may be applied also to constrained control signal. For control signals with superimposed higher harmonics it takes positive values.

Similarly, for systems with dominant 2P control the contribution of higher harmonics may be characterized by

$$TV_2 = \sum_i |u_{i+1} - u_i| - |2u_{m1} - 2u_{m2} + u(\infty) - u(0)| \geq 0 \tag{17}$$

For ideal 2P functions it yields TV_2=0.

3. PERFORMANCE PORTRAIT (PP)

The closed loop PP represents information about the closed loop performance expressed over a grid of normalized loop parameters corresponding to setpoint and disturbance step responses generated by simulation, or by real time experiments.

PP containing information about required loop properties may be used both for optimally localizing a nominal operating point, or for optimally localizing an uncertainty set of all possible operating points corresponding to specified intervals of loop parameters. Although its generation is connected with numerical problems related to the nature of grid computations, it gives very promising results especially when dealing with dead time systems. In the following, use of PP will be illustrated by two simple examples.

4. P CONTROLLER FOR THE IPDT PLANT

Let us start with a simple task requiring tuning the loop with the P controller for the IPDT plant

$$S(s) = \frac{Y_1(s)}{U(s)} = \frac{K_s}{s} e^{-T_d s} \tag{18}$$

to achieve the fastest possible y_1-MO and u-1P transients

a) for the nominal plant;

b) for the interval loop parameters with

$$K_s \in \langle 1,2 \rangle \text{ ; } T_d \in \langle 1,1.5 \rangle \tag{19}$$

The closed loop setpoint response is given as

$$F_w(s) = \frac{Y_1(s)}{W(s)} = \frac{K_P K_s e^{-T_d s}}{s + K_P K_s e^{-T_d s}} = \frac{B(s)}{A(s)} \tag{20}$$

After introducing the normalized variable

$$p = T_d s \tag{21}$$

and the normalized loop parameters

$$\kappa = K_{s0} / K_s \text{ ; } \Omega = K_p K_{s0} T_d \tag{22}$$

(14) can be transformed to

$$F_w(p) = \frac{\Omega}{\kappa e^p p + \Omega} = \frac{B(p)}{A(p)} \tag{23}$$

It means that the PP (Fig. 4) can be mapped in the 2D space of the loop parameters (κ, Ω).

In the case with $\varepsilon = 10^{-5}$ the minimal IAE value corresponding to the MO plant output and 1P plant input (Fig. 5) corresponds to the line specified by

$$\kappa / \Omega = 2.7071 \text{ ; } \kappa = K_{s0} / K_s \text{ ; } \Omega = K_p K_{s0} T_d = 0.3694 \tag{24}$$

This is close to the line corresponding to the double real dominant pole (DRDP) of the characteristic polynomial $A(p)$ (Huba et al. 1997; 1998; Huba, 2006) given by

$$\kappa / \Omega = \exp(1) = 2.7183 \text{ ; } \Omega = K_p K_{s0} T_d = 0.3679 \tag{25}$$

What is causing this difference? Concept of the DRDP comes from the root locus analyses, when by changing the closed loop gain one of the closed loop poles starting at the origin of the complex plane influences the dominantly slow response mode, whereas the dynamics of the fast mode that is moving along real axis from the minus infinity towards the origin may be neglected. Optimal loop properties correspond then to the equally fast (slow) modes. The difference observed in the PP may be caused by the final computation precision (in Matlab the smallest distance from 1.0 to the next larger double precision number is given as 2^(-52)= 2.2204e-016 and when taking into account the u-TV$_1$ values in Fig. 5 it is to see that the differences between these two operating points may already be below resolving power of computations), but they might also be expressing contribution of the infinitely many closed loop poles of the dead time system.

So, from this analysis it is not possible to conclude that the fastest possible MO transients corresponding to an infinitely high measurement & computation precision may be faster than those given by the DRDP, but it is evident that for any finite precision the transients may be reasonably speeded up by a modified tuning.

Robust tuning that would guarantee y_1-MO and u-1P transients for all possible loop parameters means to locate the uncertainty box (UB) (Fig. 4) with vertices

$$\begin{aligned}&UB=\begin{bmatrix}(\kappa_{\min},\Omega_{\max}) & (\kappa_{\max},\Omega_{\max})\\ (\kappa_{\min},\Omega_{\min}) & (\kappa_{\max},\Omega_{\min})\end{bmatrix};\\ &\kappa_{\max}=K_{s0}/K_{s\min}\ ;\ \Omega_{\max}=K_pK_{s0}T_{d\max}\\ &\kappa_{\min}=K_{s0}/K_{s\max}\ ;\ \Omega_{\min}=K_pK_{s0}T_{d\min}\end{aligned} \tag{26}$$

containing all possible operating points below the critical line (24) given e.g. by the pair $\kappa=1;\ \Omega=1/2.7071=0.3694$. From the radial shape of the border of y_1-MO&u-1P control in Fig. 4 it is obvious that the critical role is played by the upper left vertex

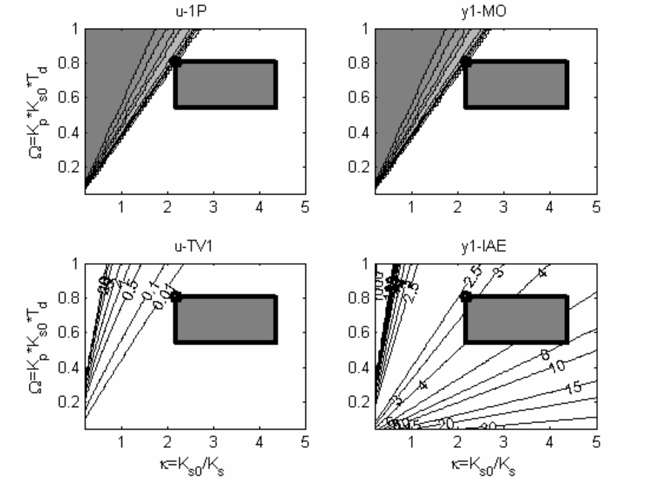


Fig. 4 PP of linear loop with the P-controller and IPTD plant; $\varepsilon=\varepsilon_u=\varepsilon_y=\{0.1,0.05,0.02,0.01,10^{-3},10^{-4},10^{-5},10^{-6},10^{-16}\}$ (from gray to white); Uncertainty Box (20) corresponding to plan parameters (13) (gray)

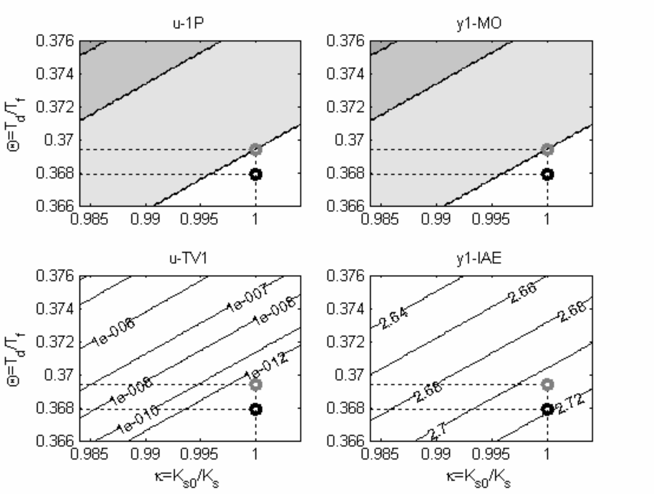


Fig. 4 Detail of the PP with operating point corresponding to the DRDP (19) (black) and to the identified MO border (18) (gray) $\varepsilon_u=\varepsilon_y=\{10^{-4},10^{-5},10^{-6},10^{-16}\}$ (from gray to white);

$$B(1,1)=(\kappa_{\min},\Omega_{\max})=(K_{s0}/K_{s\max},K_PK_{s0}T_{d\max}) \tag{27}$$

By considering just the uncertainty in T_d, the UB (26) reduces to an Uncertainty Line Segment (ULS) with vertices

$$\begin{aligned}&ULS=[\kappa,\Omega_{\min}\quad \kappa,\Omega_{\max}];\\ &\Omega_{\min}=K_pK_{s0}T_{d\min}\ ;\ \Omega_{\max}=K_pK_{s0}T_{d\max}\ ;\ \kappa=const\end{aligned} \tag{28}$$

5. PD CONTROLLER FOR THE SOIPDT PLANT

Let us tune closed loop with the PD controller

$$R(s)=\frac{U(s)}{E(s)}=r_0+r_1s\ ;\ E(s)=W(s)-Y_2(s) \tag{29}$$

and with the SOIPDT plant

$$S(s)=\frac{Y_2(s)}{U(s)}=\frac{K_s}{s^2}e^{-T_ds} \tag{30}$$

to achieve the fastest possible y_2-MO and u-2P transients. By considering (21) and by introducing normalized loop gains

$$R_0=r_0K_sT_d^2\ ;\ R_1=r_1K_sT_d \tag{31}$$

the set point to output transfer function becomes

$$F_w(p)=\frac{Y_2(p)}{W(p)}=\frac{(R_0+R_1p)e^{-p}}{p^2+(R_0+R_1p)e^{-p}}=\frac{B(p)}{A(p)} \tag{32}$$

Requirement of the triple real dominant closed loop pole (TRDP) p_0 corresponding to $A(p_0)=\dot{A}(p_0)=\ddot{A}(p_0)=0$ gives $p_0=-LambertW\left(\left[2e^{(2-R_0/R_1)}\right]/R_1\right)/R_1$ and

$$R_0=0.07912234014\ ;\ R_1=0.4611587914 \tag{33}$$

The PP in Fig. 6 with tuning corresponding to the TRDP (33) and to point giving minimal y_2 IAE values for different $\varepsilon=\varepsilon_y=\varepsilon_u=10^{-5}$ with

$$\varepsilon = 10^{-5} : R_0 = 0.1235 \; ; \; R_1 = 0.5660 \; ; \; \text{IAE} = 4.5833 \qquad (34a)$$

shows that it is possible to get operating point that has practically the same TV$_2$ value as the tuning (33) corresponding to the TRDP, but a reasonably reduced IAE value.

IAE can further be reduced by choosing $\varepsilon = 0.01$, or $\varepsilon = 0.02$ (Fig. 7), however, on costs of a slight TV$_2$ increase, when

$$\begin{aligned} &\varepsilon = 0.01 : R_0 = 0.1660 \; ; \; R_1 = 0.6440 \; ; \; \text{IAE} = 3.8800 \\ &\varepsilon = 0.02 : R_0 = 0.1839 \; ; \; R_1 = 0.6890 \; ; \; \text{IAE} = 3.7494 \end{aligned} \qquad (34b)$$

Step responses in Fig. 8 fully confirm information derived from the performance portrait.

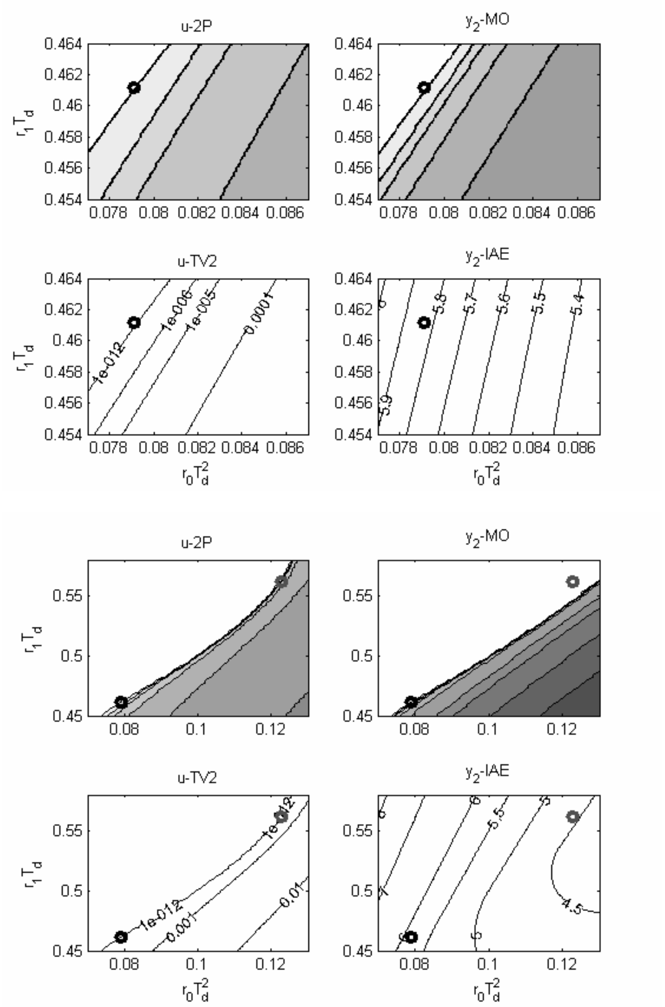


Fig. 6 Above: Detail of the performance portrait of the PD controller for the SOIPDT plant with the nominal tuning corresponding to the TRDP (33) and below: detail of the PP with the nominal tuning corresponding to the TRDP (33) (black) and to the point (34) corresponding to min IAE and $\varepsilon = 10^{-5}$ (gray); 200x200 points

$$\varepsilon = \varepsilon_u = \varepsilon_y = \left\{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-16}\right\}$$

(from gray to white).

When changing T_d for fixed r_0, r_1, operating point (31) will trace out a parabolic curve segment in the PP. Similarly, when changing K_s, it will trace out a line segment. So, the uncertainty sets corresponding to plant uncertainty (19) will have a more complex shape than (26) and their localization in a required region of the PP needs to be checked in all its internal points, what can be easily implemented numerically.

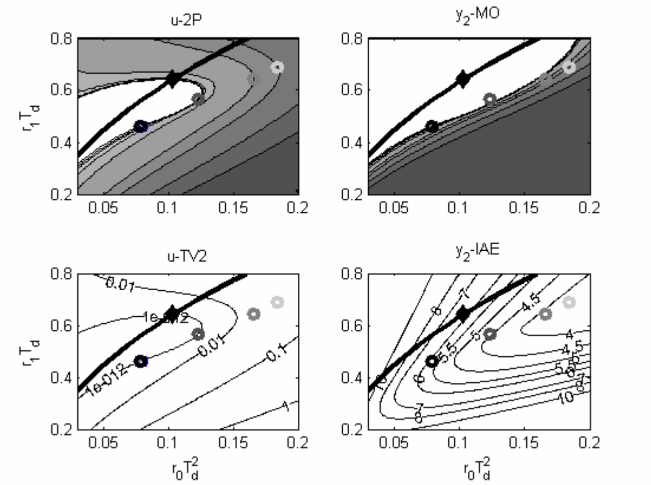


Fig. 7 PP with the nominal tuning corresponding to the TRDP (33) and to points (34) corresponding to min IAE and $\varepsilon = 10^{-5}$, $\varepsilon = 0.01$ and $\varepsilon = 0.02$ (from (circles from black to light gray); bold curve = apperiodicity border; diamond = tuning with real dominant equivalent poles and $\varepsilon = 10^{-5}$ gving minimal IAE; amplitude deviations from gray to white

$$\varepsilon = \varepsilon_u = \varepsilon_y = \left\{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-16}\right\}$$

200x200 points;

6. TUNING MODIFICATIONS FOR CONSTRAINED CONTROLLER

It is to note that the identified amplitude deviations from NO & MO shapes do note directly give amplitudes of deviations at the plant input and output. For this purpose a normalization of the considered signals with respect to their maximal values (similarly as in Skogestad and Postlethwaite, 2007) is required.

The second important point is that tuning of constrained pole assignment controllers requires working with the so called equivalent poles (Huba, 1997; 2003; 2006) that correspond to figures that after substituting for closed loop poles into the pole assignment control algorithm derived for the delay free plant

$$u = \frac{\alpha_1\alpha_2}{K_s}e - \frac{\alpha_1 + \alpha_2}{K_s}\dot{e} \; ; \; e = w - y \qquad (35)$$

give controller parameters of the delayed system corresponding to derived values (31). The equivalent poles may be expressed as

$$\alpha_{1,2} = -\frac{1}{2T_d}\left[R_1 \pm \sqrt{R_1^2 - 4R_0}\right] \qquad (36)$$

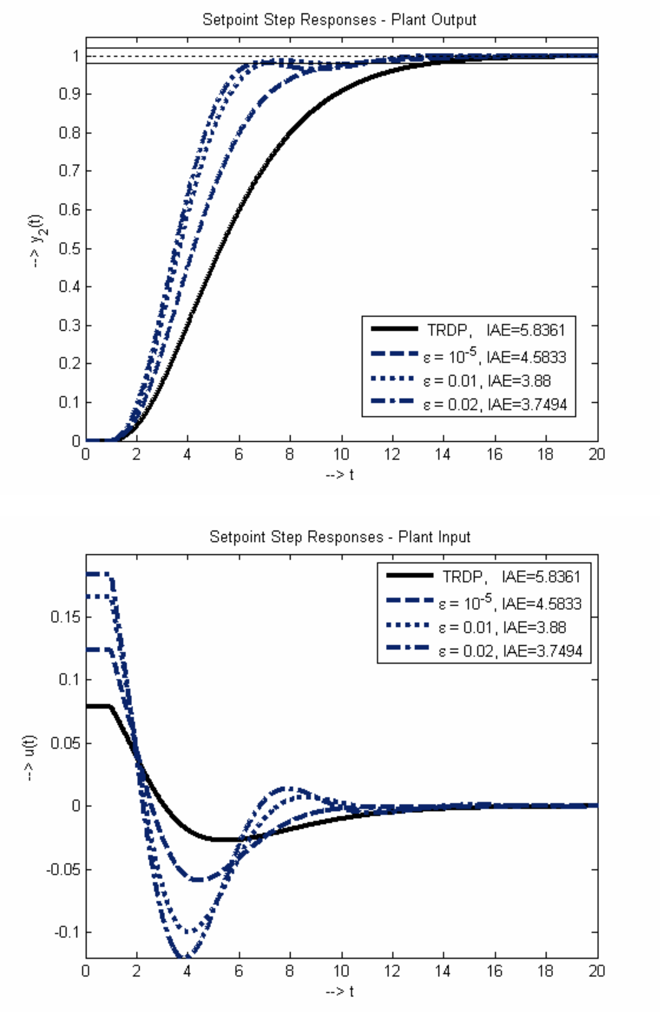


Fig. 8 Setpoint step responses corresponding to tuning (33) and (34) with error band corresponding to different values $\varepsilon\ T_d = 1; K_s = 1$.

They are real just for

$$R_1^2 - 4R_0 \geq 0 \tag{37}$$

else they are complex. By $R_1^2 - 4R_0 = 0$ the so called periodicity border (bold curve in Fig. 7) is given. As it is evident from Fig. 7, equivalent poles corresponding to the TRDP (33), as well as to tuning (34a) are always complex.

Optimal real dominant pair corresponding to $\varepsilon = \varepsilon_y = \varepsilon_u = 10^{-5}$ with the minimal y_2 IAE value (Fig. 7) is given as

$$\varepsilon = 10^{-5} : R_0 = 0.1028 \; ; \; R_1 = 0.6410 \tag{38}$$

The IAE value is, however, even larger than in the case of the TRDP.

By choosing the controller tuning at the upper margin of Fig. 7, when

$$R_0 = 0.16 \; ; \; R_1 = 0.8 \tag{39}$$

the corresponding IAE value will drop to approximately 5, however, on cost of increase TV$_2$ values and tendency to oscillations.

Equivalent poles corresponding to (33), (34) and (38-39) may be calculated as

$$\begin{aligned}
&\varepsilon = 0: \quad \alpha_{1,2}T_d = -0.231 \pm j0.161 \; ; \; \left|\alpha_{1,2}\right| = 0.281 e^{\pm j2.5317} \\
&\varepsilon = 10^{-5} : \alpha_{1,2}T_d = -0.283 \pm j0.208 \; ; \; \left|\alpha_{1,2}\right| = 0.351 e^{\pm j2.5070} \\
&\varepsilon = 10^{-5} : \alpha_{1,2}T_d = -0.321 \\
&\varepsilon = 0.01 : \alpha_{1,2}T_d = -0.322 \pm j0.250 \; ; \; \left|\alpha_{1,2}\right| = 0.407 e^{\pm j2.4821} \\
&\varepsilon = 0.02 : \alpha_{1,2}T_d = -0.345 \pm j0.255 \; ; \; \left|\alpha_{1,2}\right| = 0.429 e^{\pm j2.5038} \\
&\varepsilon \approx 0.03 : \alpha_{1,2}T_d = -0.4
\end{aligned} \tag{40}$$

In comparing these new results with tuning recommended e.g. by (Huba, 2003; 2006) that was derived by approximation of optimal complex pair given by the TRDP by its real part (modul) as $\alpha_{1,2} = -0.231/T_d$, it is obvious that the Performance Portrait method gives much broader choice for specifying the required performance as the previous approximation.

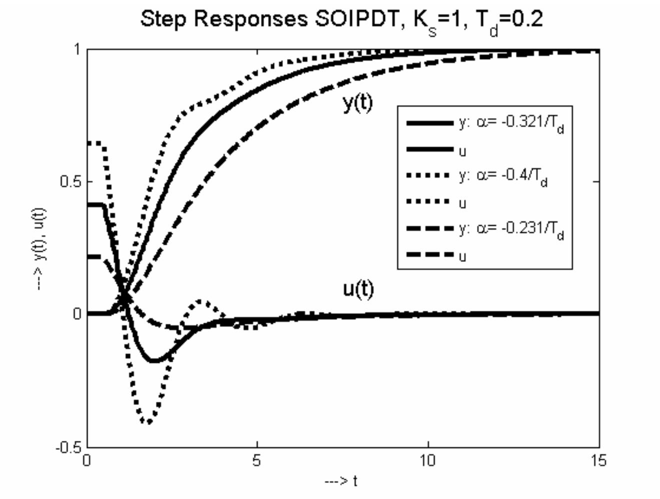


Fig. 9 Setpoint step responses corresponding to the tuning (38) with $\alpha_{1,2} = -0.321/T_d$, to tuning (39) with $\alpha_{1,2} = -0.4/T_d$ that already leads to slightly oscillatory control at the controller output and with tuning recommended e.g. in (Huba, 2003; 2006) corresponding to approximation of complex pair given by the TRDP by its real part (modul) as $\alpha_{1,2} = -0.231/T_d$

7. CONCLUSIONS

By considering conditions on MO output transients corresponding to the setpoint step responses of systems with the n-tuple integrator the paper shows that the corresponding nP-functions represent smooth version to the Felbaum's theorem on n-pulses of the relay minimum time control.

The PP based controller tuning has brought several advantages: it enables both the nominal and robust controller tuning respecting different combinations of performance measures and so also achieving of a much broader spectrum of the closed loop properties than the traditional analytical methods.

The method may be especially useful in dealing with robust tuning of different modifications of the Smith predictor and of the disturbance observer based dead time compensators under consideration of control constraints.

Similarly as in this paper, it is possible to tune controllers for chains with a higher number of integrators plus dead time that can be frequently met in controlling broad class of linear and nonlinear systems by the exact linearization method (Isidori, 1995), what is currently hardly to accomplish analytically, even when using computer algebra tools like Maple, since the LambertW function (see e.g. Hwang and Cheng, 2005) supports just derivation up to the SOIPDT plant.

Giving just an isolated point in the space of controller parameters under $\varepsilon_y = \varepsilon_u \to 0$, importance of the MRDP method may now be seen from the point that it actually proves to some degree the numerical computations underpinning the PP method but is not able to cover broader range of situations occurring in practice.

The PP method also represents tool enabling constrained control design based on the dynamical classes of control introduced e.g. by Huba et al. (1997; 1998); Huba and Bisták (1999), or Huba (2003; 2006).

ACKNOWLEDGEMENTS

This work was partially supported by the Project VEGA 1/0656/09: Integration and development of nonlinear and robust control methods and their application in controlling flying vehicles, by the project KEGA 3/7245/09 Building virtual and remote experiments for network of online laboratories. It was also supported by the grant NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic

REFERENCES

Åström, K. J., Hägglund, T. (1995). *PID controllers: Theory, design, and tuning* – 2nd ed., Instrument Society of America, Research Triangle Park, NC.

Åström, K. J., Panagopoulos, H. and Hägglund, T. (1998). Design of PI Controllers based on Non-Convex Optimization. *Automatica*, 34, 585–601.

Datta, A., Ho, M.T. and S.P. Bhattacharyya (2000). *Structure and synthesis of PID controllers*. Springer London.

Feldbaum, A. A. (1965). *Optimal control systems*. Academic Press, N.York.

Górecki, H. (1971). *Analiza i synteza ukladów regulacji z opóznieniem.* Wydawnictwo Naukowo Techniczne Warszawa.

Huba,M., Skachová,Z., Bisták,P. (1997). Minimum Time Pole Assignment Controllers Based on I1 and I2 Models with Dead Time. *Prepr. Int. Summer School on Computer Aided Education in Automation and Control*, Bratislava, 253-276

Huba,M., et al. (1998). P- and PD-Controllers for I1 and I2 Models with Dead Time. *6th IEEE Mediterranean Conference on Control and Automation, Alghero*.

Huba, M. and Bisták, P. (1999). Dynamic Classes in the PID Control. In: *Proceedings of the 1999 American Control Conference*. San Diego: AACC.

Huba, M. (2003). *Constrained systems design*. *Vol.1 Basic controllers. Vol.2 Basic structures.* STU Bratislava (in Slovak).

Huba, M. (2006). Constrained pole assignment control, *In: Current Trends in Nonlinear Systems and Control*, L. Menini, L. Zaccarian, Ch. T. Abdallah, Edts., Boston: Birkhäuser, 163-183.

Huba, M., Šimunek, M. (2007). Modular Approach to Teaching PID Control. *IEEE Trans. Ind. Electr.*, 54, 6, 3112-3121.

Huba, M. (2009). Robust Design of Integrating Controllers for IPDT Plant. Edits: Fikar, M., Kvasnica, M., In *Proc. 17th Int. Conf. Process Control '09, Š. Pleso*, 353–357, http://www.kirp.chtf.stuba.sk/pc09/data/papers/109.pdf.

Huba, M. et al. (2009) Numerical Issues in Designing PI Controller for IPDT Plant. *ibid*, 57-64, /110.pdf.

Huba, M. (2011a). Basic Notions of Robust Constrained PID Control. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohal'-Ilkiv Editors, STU Bratislava - NTNU Trondheim, 2011.

Huba, M. (2011b). Basic fundamental controllers of DC0. In: *Selected topics on constrained and nonlinear control*, M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohal'-Ilkiv Editors, STU Bratislava - NTNU Trondheim, 2011.

Hwang, Ch. and Y.C. Cheng (2005). Use of Lambert W Function to Stability Analysis of Time-Delay Systems, 2005 *American Control Conference* June 8-10,. Portland, OR , USA, FrB09.6, 4283- 428.

Isidori, A. (1995) *Nonlinear Control Systems*. 3rd edition, Springer Verlag, New York.

Keel, L. H., Kim, Y. C., Bhattacharyya, S. P.: *Advances in Three Term Control.* Pre-Congress Tutorials, 17th IFAC World Congress Seoul, Korea, 2008.

Morari,M. and E.Zafiriou (1989). *Robust Process Control*. Prentice Hall, Englewood Cliffs, N.Jersey.

Neimark, Ju. (1973). I.: D-decomposition of the space of quasi-polynomials (on the stability of linearized distributive systems). *American Mathematical Society Translations*, Series 2. Vol. 102, 1973: *Ten papers in analysis.* American Mathematical Society, Providence , R.I. , pp. 95-131.

Normey-Rico, J.E. et al. (2009). An unified approach for DTC design using interactive tools. *Control Eng. Practice* 17, 1234-1244.

Oldenbourg, R.C., H. Sartorius (1944, 1951) *Dynamik selbsttätiger Regelungen*. Oldenbourg, München.

Shinskey, G. (1990). How good are Our Controllers in Absolute Performance and Robustness. *Measurement and Control*, Vol. 23, 114-121.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* Volume 13, Issue 4, Pages 291-309

Skogestad, S. (2006). Tuning for Smooth PID Control with Acceptable Disturbance Rejection. *Ind. Eng. Chem. Res.*, *45,* 7817-7822.

Skogestad, S. and I. Postlethwaite (2007). *Multivariable Feedback Control Analysis and Design*, John Wiley, N.York,

Vítečková, M., Víteček, A. (2008). Two-degree of Freedom Controller Tuning for Integral Plus Time Delay Plants. *ICIC Express Letters. An International Journal of Research and Surveys*. Volume 2, Number 3, Japan , pp. 225-229.

Vítecková, M., Vítecek, A. (2010) metoda násobného dominantního pólu pro regulátory se dvema stupni volnosti a proporcionální soustavy s dopravním zpoždením. *Int. Conference Cybernetics and Informatics* February 10 - 13, VYŠNÁ BOCA, Slovak Republic

Ziegler,J.G. and Nichols,N.B. (1942). Optimum settings for automatic controllers. *Trans. ASME,* 759-768.

Comments – Remarks

Analysing Constrained Controllers for SOPDT Plant and Complex Assigned Poles by the Performance Portrait Method

Mikuláš Huba*,**

** Institute of Control and Industrial Informatics, FEI STU, Ilkovičova 3, SK-812 19 Bratislava
Slovakia (Tel: +421-2 -60291 771; e-mail: mikulas.huba@ stuba.sk).
** MI/PRT, FernUniversität in Hagen, Universitätsstr. 27, D-58084 Hagen
Germany (e-mail: mikulas.huba@fernuni-hagen.de)*

Abstract: This paper verifies properties of previously proposed simple constrained gain-scheduled PD controllers for the Second Order Integrator Plus Dead Time (SOIPDT) plant (Huba, 2004a,b,c; 2005) based on imposing dynamics characterized by complex closed loop pole and compensating control constraints effect by reference signal scheduling. Simultaneously, this paper broadens tuning possibilities of these solutions by the new method for poles specification based on the Performance Portrait (Huba, 2010; 2011a,b). Due to analyzing broad range of different operating points information of potential user about this solutions was reasonably enlarged

Keywords: robust control, simulation, animation, integral control, interactive systems, visualization.

1. INTRODUCTION

After some decades of dominancy of linear systems theory, the last decade is characterized by a revival of constrained systems design. It is, however, interesting to note that while a high attention is given to the development of complex controllers applicable to multi input – multi output systems (see e.g. Bemporad et al., 2002; Kvasnica et al., 2011), there are still lacking simple solutions and the corresponding dynamics understanding for simpler single-input-single-output systems. One of the results of this situation can be well characterized by the note given in Åström & Hägglund (1995a)"…derivative action is frequently switched off for the simple reason that it is difficult to tune properly"... It is easy to show that using the linear controller design, the PD controller cannot be properly tuned in real (constrained) situations! This problem of balancing simplicity versus performance is neither solved by the most frequently used anti-windup approaches developed originally for dealing with the superfluous controller integration (see e.g. Kothare et al., 1994).

Several simple modifications of the pole assignment controllers for constrained systems were presented (Huba et al., 1998; 1999; Huba, 1999; 2003; 2004; 2006; Huba and Bisták, 1999) for the case of closed loop dynamics specified by a couple of real poles. Simultaneously, a method for the step response based controller tuning was presented (Huba et al., 1998; 1999; Huba, 2003) that can be considered as a generalization of the well-known method by Ziegler and Nichols (1942). This method was generalized also for controller tuning based on the "ultimate sensitivity" method and the relay controller tuning (Huba, 2003). While this approach is based on splitting the 2nd order system dynamics into two 1st order ones, which are robust against the saturation, for a long time it was not clear, how to interpret and apply the case of a complex pole pair. The up to now used solution based on approximation of complex poles by their real part or by their module leads mostly to a performance decrease (Huba et al. 1998; Huba, 2003; Huba, 2006). By introducing the constrained pole assignment approach to the research community, one of the basic questions was, how is it able to prove the proclaimed properties and hot it is able to deal with control of uncertain plants.

In this paper the newly introduced Performance Portrait method (Huba, 2010; Huba, 2011a, b) enables to proceed in solving these problems in two points:

a) Tuning guaranteeing faster output-monotonic and input-2P responses than the analytically derived tuning corresponding to the Tripple Real Dominant Pole is introduced;

b) Performance of the gain-scheduling approach applied in deriving constrained pole assignment controller for the 2nd order integrator + dead time is tested for a broad range of the loop parameters (control signal constraints related to initial conditions).

The paper is structured as follows. Chapter 2 recalls basic properties of the phase plane trajectories of 2nd order systems with complex poles. In Chapter 3 effect of control constraints on the phase-plane trajectories is briefly analyzed. Chapter 4 deals with simple approaches to nonlinear reference signal shaping enabling to keep required performance at the plant input and output. Simple compensation of time delays is proposed in Chapter 5. Chapter 6 introduces briefly basic measures for evaluating performance of considered control loops, Chapter 7 discusses possibilities for optimal controller tuning and Chapter 8 evaluates by the performance portrait method properties of the constrained controller for different control signal constraints and dead time values. Chapter 9 brings finally short conclusions to the paper.

2. PROBLEM SPECIFICATION

Let us consider a closed loop system defined by its characteristic polynomial $A(s)$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \mathbf{A}_R\mathbf{x} \qquad (1)$$

$$A(s) = s^2 + a_1 s + a_0$$

with origin as the reference state. The system is composed of a plant

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \qquad (2)$$

and of the linear controller

$$u = \mathbf{r}^t\mathbf{x} \quad ; \quad \mathbf{r}^t = [r_0 \quad r_1] \qquad (3)$$

Its main task is usually to guarantee stable equilibrium at the origin, what corresponds to the closed loop poles

$$\mathrm{Re}\{\alpha_{1,2}\} = \mathrm{Re}\left\{\frac{-a_1 \pm \sqrt{a_1^2 - 4a_0}}{2a_0}\right\} < 0 \qquad (4)$$

These can, however, specify the close loop dynamics also more precisely. For $a_1^2 < 4a_0$, the closed loop poles are complex

$$\alpha_{1,2} = -\frac{a_1}{2} \pm j\sqrt{a_0 - \frac{a_1^2}{4}} = \sqrt{a_0}\left[-\zeta \pm j\sqrt{1-\zeta^2}\right] =$$
$$= \omega_0[-\zeta \pm j\omega]; \ \omega_0 = \sqrt{a_0} > 0; \zeta = \frac{a_1}{2\sqrt{a_0}} \in (0,1) \qquad (5)$$

Let us introduce state and time transformation

$$\frac{d\mathbf{x}_0}{d\tau} = \begin{bmatrix} 0 & 1 \\ -1 & -2\zeta \end{bmatrix}\mathbf{x}_0 \, ; \, \tau = t\sqrt{a_0} = t\omega_0$$
$$\mathbf{x}_0 = \begin{bmatrix} x \\ \dfrac{dx}{d\tau} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}; \ y = \frac{dx}{d\tau} = \frac{1}{\omega_0}\frac{dx}{dt} \qquad (6)$$

This system has trajectories

$$\omega^2 x^2 + [y + \zeta x]^2 = C^2 e^{2\frac{\zeta}{\omega} arctg \frac{y+\zeta x}{\omega x}}$$
$$y = \frac{dx}{d\tau} = \frac{1}{\omega_0}\frac{dx}{dt}; \ \omega = \sqrt{1-\zeta^2} \qquad (7)$$

In polar coordinates one gets

$$\bar{x}_1 = \omega x \, ; \, \bar{x}_2 = y + \zeta x$$
$$R^2 = \bar{x}_1{}^2 + \bar{x}_2{}^2 \, ; \bar{x}_1 = R\cos\theta \, ; \, \bar{x}_2 = R\sin\theta \qquad (8)$$

whereby

$$R = Ce^{\frac{\zeta}{\omega}\theta} \ ; \ \theta = arct\frac{\bar{x}_2}{\bar{x}_1} = arctg\frac{y+\zeta x}{\omega x} \qquad (9)$$

or

$$x = Ae^{-\zeta\tau}\cos(\omega\tau + \alpha)$$
$$y = -\zeta Ae^{-\zeta\tau}\cos(\omega\tau + \alpha) - \omega Ae^{-\zeta\tau}\sin(\omega\tau + \alpha) \qquad (10)$$

The values C, A and α are given by the initial conditions. After substituting (10) into (8)

$$\theta = -\alpha - \omega\tau \qquad (11)$$

Working just with the modified system (6), the poles are

$$p_{1,2} = -\zeta \pm j\omega \qquad (12)$$

Obviously, the solution (9) satisfies equations

$$\frac{dR}{d\tau} = C\frac{\zeta}{\omega}e^{\frac{\zeta}{\omega}\theta}(-\omega) = -\zeta Ce^{\frac{\zeta}{\omega}\theta} = -\zeta R$$
$$\frac{d\theta}{d\tau} = -\omega \qquad (13)$$

It means that the module of the state vector decreases with the velocity proportional to its actual value by the coefficient ζ and, simultaneously, it rotates clockwise with the angular velocity ω (Fig. 1). In these new variables, the module and argument can be expressed as

$$R = \sqrt{\bar{\mathbf{x}}^t\bar{\mathbf{x}}} \ ; \ \theta = arctg\frac{\bar{x}_2}{\bar{x}_1} \qquad (14)$$

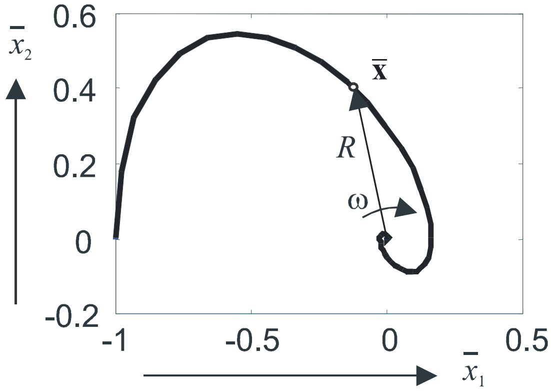


Fig. 1 Spiral trajectory corresponding to complex poles

Basic Acronyms	
2P-function	Two Pulse Function
CPAC	Constrained Pole Assignment Control
MO	monotonic, monotonicity
PP	Performance Portrait
PD	Proportional Derivative
RBC	Reference Braking Trajectory

The first of Eqs. (13) can also be expressed as

$$\frac{dR}{d\tau} = \frac{1}{2\sqrt{\bar{\mathbf{x}}^t\bar{\mathbf{x}}}}\frac{d(\bar{\mathbf{x}}^t\bar{\mathbf{x}})}{d\tau} = -\zeta\sqrt{\bar{\mathbf{x}}^t\bar{\mathbf{x}}} \qquad (15)$$

or

$$\frac{d(\bar{\mathbf{x}}^t\bar{\mathbf{x}})}{d\tau} = -2\zeta(\bar{\mathbf{x}}^t\bar{\mathbf{x}}) \qquad (16)$$

For the double integrator, with respect to $[r_0 \quad r_1] = [a_0 \quad a_1]$ the controller guaranteeing closed loop poles (5) is given as

$$\mathbf{r}^t = -[\omega_0^2 \quad 2\zeta\omega_0] \qquad (17)$$

The main difference to the real poles case is given by the fact that the overall transient corresponds just to a single phase of a systematic decrease of the state vector module: It is no more possible to identify two phases of control as in the case of real poles. The corresponding trajectory has shape of a logarithmic spiral that does not depend on initial conditions. In designing controller for the real pole pair a requirement on the distance decrease is applied twice, while in the case of complex poles just once to the associated quadratic form (16) that shows a first order dynamics. It is well known that the design based on minimizing quadratic criteria leads to slightly underdamped system behaviour – state vector $\overline{\mathbf{x}}$ rotates during transients.

3. CONSTRAINTS IN CONTROL

Let us consider the 2nd order system (2) with origin as the reference state and the control signal

$$u \in \langle U_1, U_2 \rangle \ ; \ U_1 < 0 < U_2 \tag{18}$$

being constrained with U_1 and U_2 as the limit values, which can be simply expressed as

$$u = sat(u) = sat(\mathbf{r}^t \mathbf{x}) \tag{19}$$

The aim is to bring the system from an initial state $[x_0, 0]$ to the origin in the minimal time by respecting the dynamics of the state and control signal changes specified by the closed loop poles (5).

Putting such limiter (19) into the 2nd order loop, its behaviour can become useless or unstable. The dynamics specified by the closed loop poles can be guaranteed just over the invariant set of linear control that may become to be negligible with respect to the required range of variable changes.

3.1 Zone of Proportional Control P_b

Two lines B_j; $j = 1,2$ parallel to vector $\mathbf{z} \perp \mathbf{r}$ satisfying to $\mathbf{r}^t \mathbf{z} = 0$ limit the strip-like band of proportional control P_b (Fig. 2). Just a segment of the spiral linear trajectory (13) limited by vertices

$$P_0^j = \mathbf{v} U_j \ ; \ j = 1,2 \tag{20}$$

may be considered as the reference braking trajectory leading to required state and as the target for the 1st phase of control.

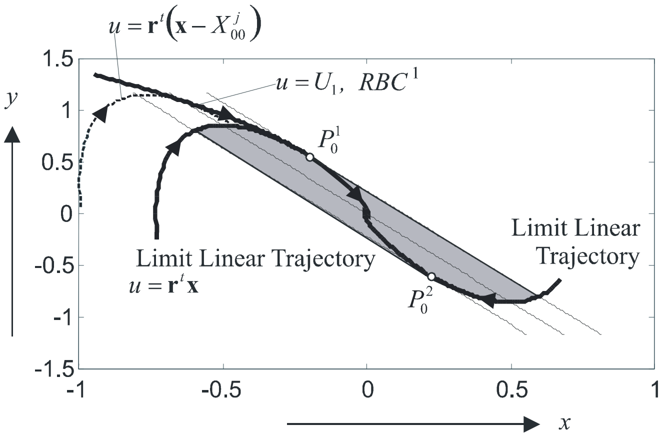


Fig. 2 Zone of Proportional Control corresponding to (19) and chosen Reference Braking Curve (RBC)

In defining shape of invariant set of linear control, it is important to identify vertices P_0^j ; $j = 1,2$, in which the trajectories of the closed loop system have tangent parallel to vector $\mathbf{z}$ defining border of P_b. According to requirement $\mathbf{r}^t \mathbf{z} = 0$, these can be defined by

$$\mathbf{r}^t P_0^j = U_j \ ; \ \mathbf{r}^t \frac{d\mathbf{x}}{d\tau} = \mathbf{r}^t (\mathbf{A} + \mathbf{b}\mathbf{r}^t) P_0^j = 0 \tag{21a}$$

that gives

$$P_0^j = \begin{bmatrix} \mathbf{r}^t \\ \mathbf{r}^t (\mathbf{A} + \mathbf{b}\mathbf{r}^t) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} U_j \qquad (21b|$$

3.2 Reference Braking Curve

The invariant set of linear control will surely be included between lines B_j; $j = 1,2$ and limited by trajectories of the closed loop (1) crossing the vertices P_0^j . For the complex poles in the phase-plane $(x, \dot{x})$ it holds

$$B_j : \ x + \frac{2\zeta}{\omega_0} \dot{x} + \frac{U_j}{\omega_0^2} = 0 \tag{22}$$

$$P_0^j = \begin{bmatrix} p_0^j \\ \dot{p}_0^J \end{bmatrix} = \begin{bmatrix} (4\zeta^2 - 1)/\omega_0^2 \\ -2\zeta/\omega_0 \end{bmatrix} U_j \ ; \ j = 1,2 \tag{23}$$

Next we will introduce an invariant set called Reference Braking Curve (RBC). It will be constructed from the limit linear trajectory that touches boundaries of P_b in the point P_0^j (23) and of the segment of limit braking with $u = U_j$ constructed from P_0^j with time running in negative direction according to

$$\mathbf{x}_b^j(\tau) = e^{-\mathbf{A}\tau} P_0^j + U_j \int_0^{-\tau} e^{-\mathbf{A}\vartheta} \mathbf{b} \, d\vartheta \ ; \ j = 1,2 \tag{24}$$

4. NONLINEAR REFERENCE SIGNAL SHAPING

Pole assignment dynamics of the acceleration phase can be constructed in two ways. Both are characteristic with a shift of the reference point $X_{00} = [x_{00} \quad 0]^t$ (centre of the actual spiral trajectory in the acceleration phase) that is equivalent to nonlinear reference shaping.

4.1 Reference Braking Curve

The 1st solution (Huba, 2003; 2005) is based on a shift of the proportional band (equivalent to the reference signal shaping) in such a way that the line traced out through the actual state **x** in a direction defined by a chosen vector **w** (in Fig.3 parallel to the horizontal axis) crosses the boundary $B_j{}'$ of the controller with shifted origin at a point of RBC

$$U_j = \mathbf{r}^t\left(\mathbf{x}_b - X_{00}^j\right) \tag{25}$$

Then, according to the chosen vector **w**, this controller will be used just for

$$\dot{x} / p_0^j > 1 \tag{26}$$

Else, the linear controller (19) is used.

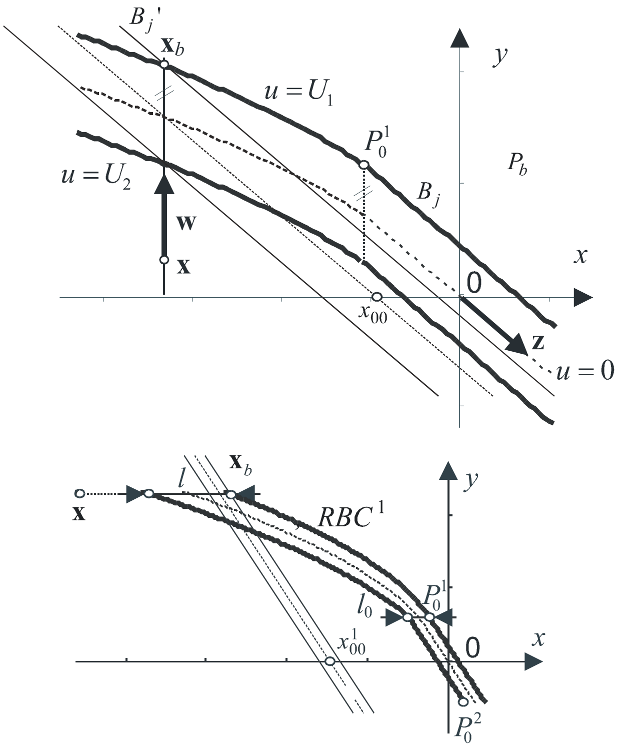


Fig.3. Scheduling controller parameters by projecting the representative point with the boundary of P_b to RBC vertically (above), or horizonally (below).

Strictly speaking, this modification does not fully preserve the required pole assignment dynamics in the transition from full acceleration to braking (due to the interaction with the dynamics of the point X_{00}^j). The advantage is its simplicity.

4.2 Control trajectory toughing border of P_b

The 2nd solution could be constructed in such a way that a spiral trajectory crossing the initial point **x** with a centre of rotation shifted to $X_{00} = [x_{00} \quad 0]^t$ would touch RBC in a point $\mathbf{x}_b$, which corresponds to

$$u = \mathbf{r}^t\left(\mathbf{x}_b - X_{00}^j\right) = U_j \tag{27}$$

For this problem (Huba, 2004c), an analytical solution was not found. This is, however, not substantial problem, since such a solution would not be appropriate for dealing effectively with the time delayed systems that represent the primary motivation for our treatment.

In such situation such solution fully preserving dynamics of the pole assignment control would need additional dead time compensation in crossing the proportional band by higher velocities with diameters of rotation exceeding many time that one corresponding to P_0^j. Due to this, the transients of the 2nd order system with a time delay have dynamics depending on the initial states and disturbances (Fig.4).

5. GAIN-SCHEDULING TO COMPENSATE TIME DELAYS

When the representative point crosses P_b at higher velocities, the time available for changing the control signal from one limit value U_j to another one $U_{3\text{-}j}$ decreases. Due to the time delays introduced by actuators and other inertial components of the control loop, this usually leads to overshooting (in Fig.4 better visible in the control signal), or, for larger steps, to instability. Therefore, for higher velocities the width l of P_b will be broaden by decreasing ω (scheduling of the closed loop poles) in order to achieve approximately constant time of crossing P_b as it corresponds to the points P_0^j with the P_b width l_0. This requires to keep

$$l_0 / \dot{p}_0^j = l / \dot{x} \Rightarrow l = \dot{x} l_0 / \dot{p}_0^j \tag{28}$$

Because of

$$l_0 = (U_1 - U_2)/r_0 = -(U_1 - U_2)/\omega_0^2 \tag{29}$$

$$l = \frac{U_1 - U_2}{2U_j \zeta \omega_0} \dot{x} \tag{30}$$

$$\omega = \sqrt{-2\omega_0 U_j \zeta / \dot{x}} \tag{31}$$

This angular velocity will be substituted into the original linear controller instead of ω_0, when

$$\mathbf{r}_1^t = \left[2\omega_0 U_j \zeta / \dot{x} \quad -2\zeta\sqrt{-2\omega_0 U_j \zeta / \dot{x}}\right] \tag{32}$$

Using this controller, the required new origin for the reference shaping can be derived as

$$x_{00} = \frac{-2\zeta U_j^2 + 4\zeta^3 U_j^2 + \zeta\dot{x}^2\omega_0^2 - 2\zeta\dot{x}^2\omega_0\sqrt{-2\frac{\omega_0 U_j \zeta}{\dot{x}}} - U_j \dot{x}\omega_0}{\omega_0^2 U_j \zeta} \tag{33}$$

when

$$u = \frac{2\zeta U_j^2 - 4\zeta^3 U_j^2 - \zeta\dot{x}^2\omega_0^2 + U_j\dot{x}\omega_0 + 2x\omega_0^2}{\omega_0 \dot{x}} \tag{34}$$

Step responses achieved with the modified controller are shown in Fig.5. It is to see that the transients from the acceleration to braking have approximately the same dynamics both for larger and smaller steps and no overshoot.

6. EVALUATING CONTROLLER PERFORMANCE

In evaluating constrained controller we wish to show how the control performance depends on the amplitude of the setpoint changes with respect to the given control signal limits. In evaluating the loop performance experimentally we are dealing with quasi-continuous plant output $y(t)$ having an initial value $y_0 = y(0)$ and a final value $y_\infty = y(\infty)$, whereby it will be most frequently required to be nearly, or ε_y-monotonic and with constrained control signal $u(t)$ having an initial value $u_0 = u(0)$, a final value $u_\infty = u(\infty)$ and close to the ideal shape of the 2P function (i.e. ε_u-2P, see Huba, 2010; 2011a). The signals will be sampled with a sampling frequency enabling to preserve all their important features.

Stable control will be preferably characterized by performance indices such as IAE (Integral of Absolute Error, Shinskey, 2000), or TV (Total Variance, Skogestad, 2003). For a setpoint step $w(t)$

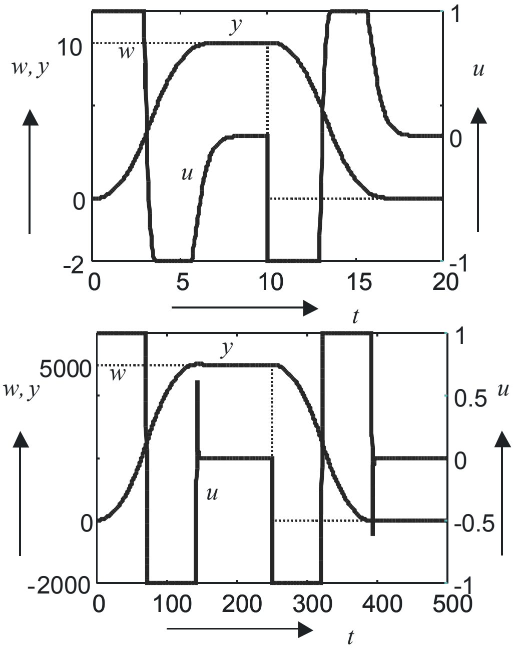


Fig.4 Transients for the double integrator with dead time controlled by controller corresponding to Fig.1 with shifted origin x_{00} (equivalent to reference shaping) $T_d = 0.2s$, $\zeta = 0.82; \omega_0 = 0.28/T_d$.

$$IAE = \int_0^\infty |e(t) - e(\infty)| dt;\ e(t) = w(t) - y(t); \tag{35}$$

$$TV = \int_0^\infty \left|\frac{du}{dt}\right| dt \approx \sum_i |u_{i+1} - u_i| \tag{36}$$

In general, we are looking for minimal values of these indices. When they increase over some value chosen e.g. as a multiple of optimal value, from the practical point of view transients may be denoted as unstable.

Output monotonicity may e.g. be motivated by comfort of passengers in traffic control, by minimizing generated noise and vibrations, by safety requirements, etc.

In engineering application dealing with not-perfect systems and finite measurement and processing precision we need measures for enumerating deviations of achieved responses from strict monotonicity.

A simple measure introduced by Åström and Hägglund (2005) for evaluating deviations from strict MO denoted as monotonicity index was defined as

$$\alpha = \int_0^\infty h(t)dt \Big/ \int_0^\infty |h(t)dt| \tag{37}$$

where $h(t)$ is the closed loop impulse response. For a strictly MO response $\alpha = 1$. A similar and very simple integral measure for evaluating deviations from strict monotonicity we may define e.g. for the plant output samples $y(t_i) = y_i$ by modification of the TV criterion and denote it as the TV$_0$ criterion (Huba, 2010, 2011a)

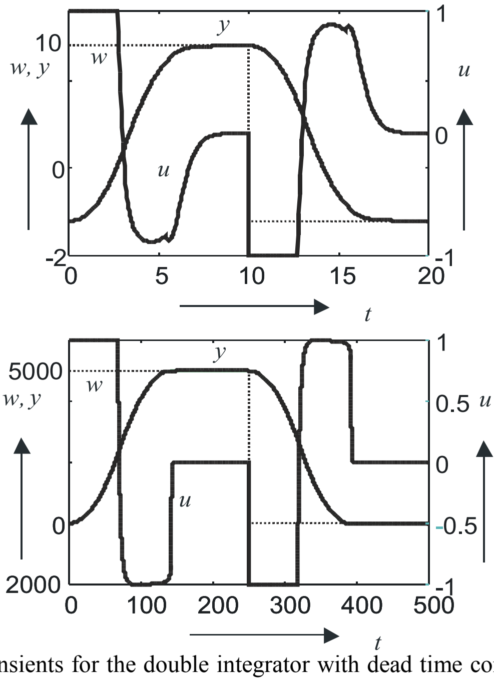


Fig.5 Transients for the double integrator with dead time controlled by controller corresponding to Fig.1 with shifted origin x_{00} and parameters scheduled with the velocity $\dot{x}$ $T_d = 0.2s$, $\zeta = 0.82$; $\omega_0 = 0.28/T_d$

$$y - TV_0 = \sum_i |y_{i+1} - y_i| - |y(\infty) - y(0)| \tag{38}$$

TV$_0$=0 just for strictly MO response, else it gives positive values.

Besides of this integral measure for characterizing deviations from strict MO it is important to introduce also amplitude deviations measure based on defining an error band specified by the parameters ε_y, or ε_u around the reference MO signal (Huba, 2010, 2011a). In general, ε – MO signal $f(t)$ has to fulfil requirements

$$[f(t_2) - f(t_1)] sign\{f(\infty) - f(0)\} \geq -\varepsilon\ ;\ \forall t_2 > t_1 \geq 0\ ;\ \varepsilon > 0 \tag{39}$$

In dealing with the SOIPDT plant and other plants with two unstable poles the ideal control signal has always shape of 2P functions with two extreme points u_{m1} and u_{m2} that separate three monotonic intervals. When allowing discontinuous control signal changes after a setpoint step the MO interval before the first maximum may shrink to zero.

Deviations from ideal 2P shapes may be characterized by the integral TV_2 criterion defined for $u(t)$ as

$$u-TV_2=\sum_i |u_{i+1}-u_i|-|2u_{m1}-2u_{m2}+u(\infty)-u(0)| \tag{40}$$

Ideal 2P control functions yield TV_2=0. The amplitude measure for characterizing deviations from ideal 2P shapes may be based on the amplitude measure for deviations from monotonicity working with tolerance ε (39) that will be subsequently applied over each of the three (two) MO intervals.

7. CONTROLLER TUNING

In general, we have interest to work with tuning guaranteeing admissible deviations from the output monotonicity (ε_y-MO output with limited y-TV_0 values) together with the admissible (amplitude or integral) deviations from ideal 2P shapes at the plant input and giving the fastest possible transients characterized e.g. by the minimal IAE values.

One traditional possibility of the analytical controller tuning was based on conditions of the Multiple Real Dominant Poles. Using requirement of the triple real dominant pole (TRDP) for the loop with the SOIPDT and PD controllers (Huba, 1999), it was possible to derive controller coefficients

$$r_0=-0.079/K_sT_d^{\,2}\ ;\ r_1=-0.461/K_sT_d \tag{41}$$

with the corresponding closed loop poles

$$s_{1,2}=-(0.231\pm j0.161)/T_d \tag{42}$$

that correspond to parameters appropriate for the constrained controller tuning

$$\zeta=0.82\ ;\ \omega_0=0.28/T_d \tag{43}$$

In (Huba, 2011b), based on amplitude deviations from ideal monotonic and 2P shapes at the plant output and input $\varepsilon_y=\varepsilon_u=\varepsilon$, alternative controller tuning parameters may be derived by the performance portrait method that e.g. gives

$$\begin{aligned}&\varepsilon=10^{-5}:\alpha_{1,2}T_d=-0.283\pm j0.208\ ;\ |\alpha_{1,2}|=0.351e^{\pm j2.5070}\\&\varepsilon=0.01:\alpha_{1,2}T_d=-0.322\pm j0.250\ ;\ |\alpha_{1,2}|=0.407e^{\pm j2.4821}\\&\varepsilon=0.02:\alpha_{1,2}T_d=-0.345\pm j0.255\ ;\ |\alpha_{1,2}|=0.429e^{\pm j2.5038}\end{aligned} \tag{44}$$

When transforming these figures into parameters used for the controller tuning one gets

$$\begin{aligned}&\varepsilon=10^{-5}:\xi=0.805\ ;\ \omega_0=0.351\\&\varepsilon=0.01:\xi=0.790\ ;\ \omega_0=0.407\\&\varepsilon=0.02:\xi=.803\ ;\ \omega_0=0.429\end{aligned} \tag{45}$$

In this new interpretation, tuning (41)-(43) corresponds to ideal y-MO and u-2P shapes with $\varepsilon\to 0$.

Fig. 6 shows step responses corresponding to tuning (42-43) width $\varepsilon=0$ and to tuning (44-45) for $\varepsilon=10^{-5}$, $\varepsilon=0.01$ and $\varepsilon=0.02$. While the new tuning corresponding to $\varepsilon=10^{-5}$ shows reasonably improved dynamics with respect to that one corresponding to the triple real pole ($\varepsilon=0$), responses corresponding to the values $\varepsilon=0.01$ and $\varepsilon=0.02$ taken from the performance portrait of the unconstrained system have obviously larger amplitude than the tolerance band drawn for $\varepsilon=0.02$ – tuning corresponding to specific overshooting in linear case cannot be simply adopted for constrained control without deeper analysis of the actual amplitude relations and without corresponding norming of all variables as it was e.g. recommended by Skogestad and Postlethwaite (2007).

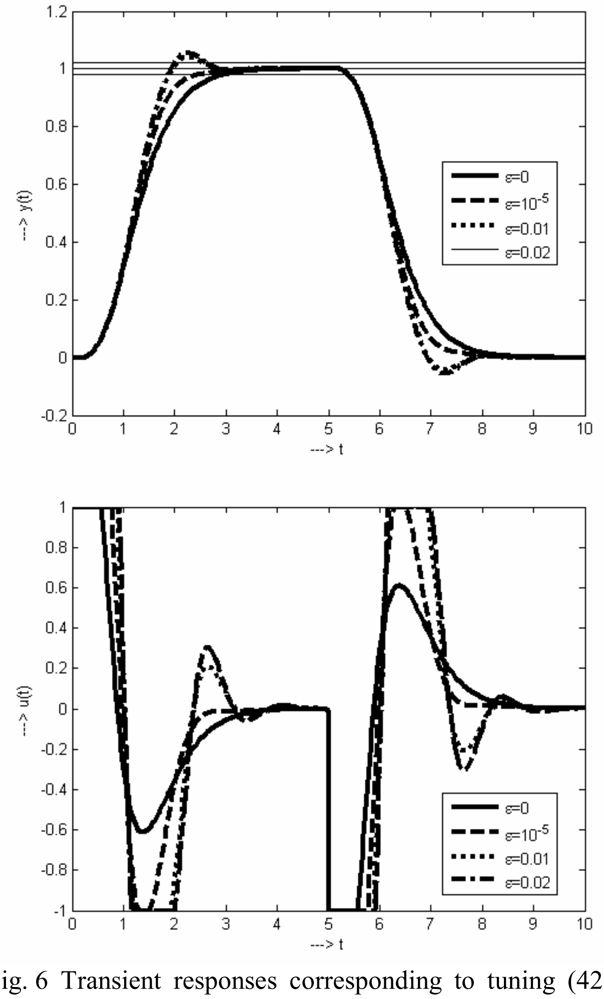


Fig. 6 Transient responses corresponding to tuning (42-43) (TRDP with $\varepsilon=0$) and to tuning (44-45) for $\varepsilon=10^{-5}$, $\varepsilon=0.01$ and $\varepsilon=0.02$; $K_s=1;|u|\le 1;T_d=0.2$

8. EVALUATING CONSTRAINED CONTROLLER BY THE PERFORMANCE PORTRAIT METHOD

Now we are going to deal with the question, how is the loop performance in controlling SOIPDT influenced by the control signal constraints.

Fig. 7 shows closed loop Performance Portrait consisting of ε_y-MO and u-TV_2 values is mapped over grid of control limit values $U_1 = U_{\min}; U_2 = U_{\max}$ that will change by keeping constant amplitude of the setpoint step.

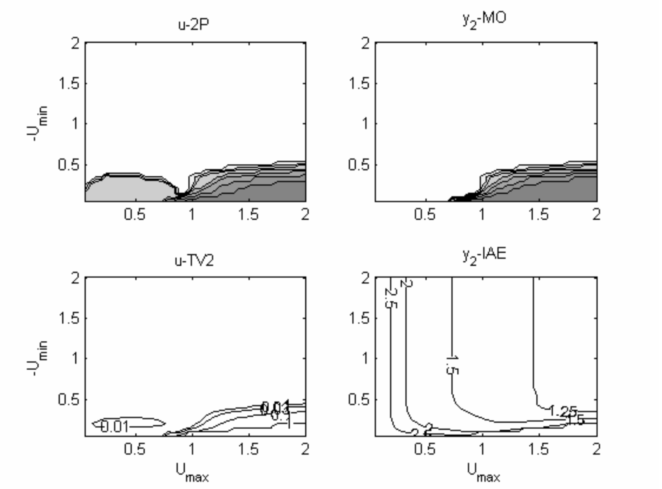


Fig. 7 Performance Portrait of the proposed gain-scheduling constrained pole assignment controller (34) for complex poles (42-43); 41x41 points (from gray to white); $K_s = 1;\ \varepsilon = \varepsilon_u = \varepsilon_y = \left\{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\right\};$ $w = 1;\ T_d = 0.2$

The PP shows that the algorithm cannot be used for situations with relatively strong limitation of the value $U_{\min}$ when it is possible to observe rapid increase of both the TV$_2$ and y_2-IAE values and also increasing amplitude deviations from ideal monotonic shapes at the plant output and 2P shapes at the plant input.

Results achieved by the performance portrait in Fig. 7 achieved for $K_s = 1; w = 1; T_d = 0.2$ over 41x41 points for $-U_{\min} \in \langle 0.05, 2 \rangle; U_{\max} \in \langle 0.05, 2 \rangle$ are illustrated by transient responses corresponding to $U_{\max} = 1$ and several values of $U_{\min}$ in Fig. 8.

For $U_{\min} = -0.5$ and $U_{\min} = -0.25$ the transients are *y*-MO and *u*-2P.

For $U_{\min} = -0.1$ some overshooting in *y* occurs, together with the third pulse of the control signal *u*.

For $U_{\min} = -0.05$ the output overshooting increases up to the moment when the control collapses and starts to diverge.

Next, we are going to check properties for different dead time values.

Performance Portrait in Fig. 9 achieved for $T_d = 0.5$ shows almost perfect properties both at the plant input and output in whole considered range of constraints.

Transient response in Fig. 10 explain, why it is possible that the y_2-IAE values are over whole analyzed area practically constant – due to the relatively long dead time the controller gains must be relatively low and the control signal does mostly not saturate.

In difference to this the PP in Fig. 11 shows that for $T_d = 0.1$ the controller gives almost perfect properties over whole considered range of control constraints, just for $U_{\max} \to 2$ and $U_{\min} \to -0.05$ there occur some deviations from ideal shapes.

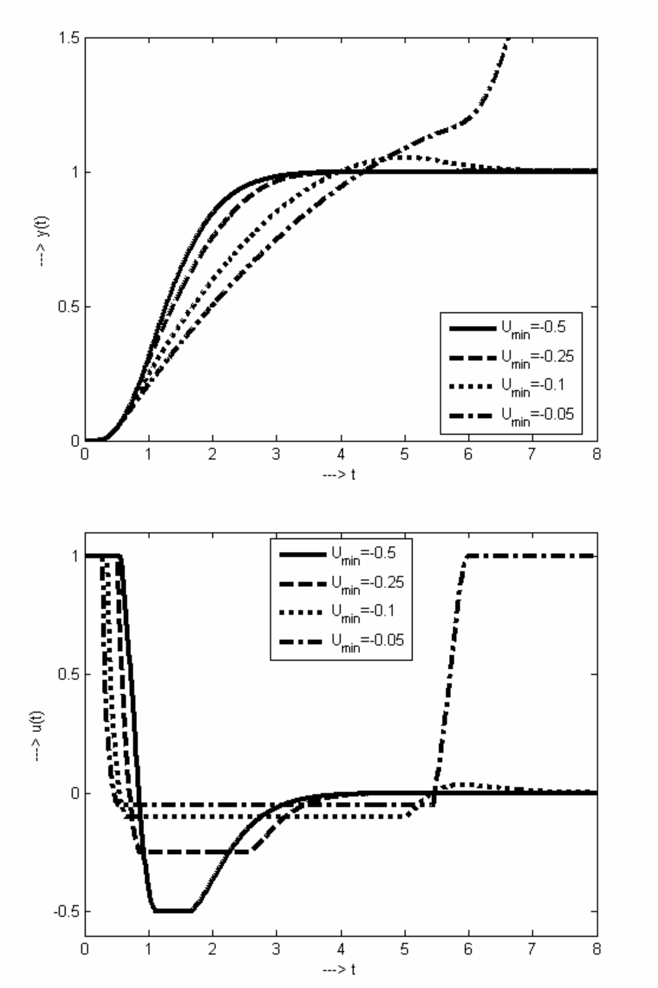


Fig. 8 Transient responses corresponding to $U_{\max} = 1$ and several values of $U_{\min} \in \langle -0.05, -0.5 \rangle$ for complex poles (42-43) and $K_s = 1;\ w = 1;\ T_d = 0.2$

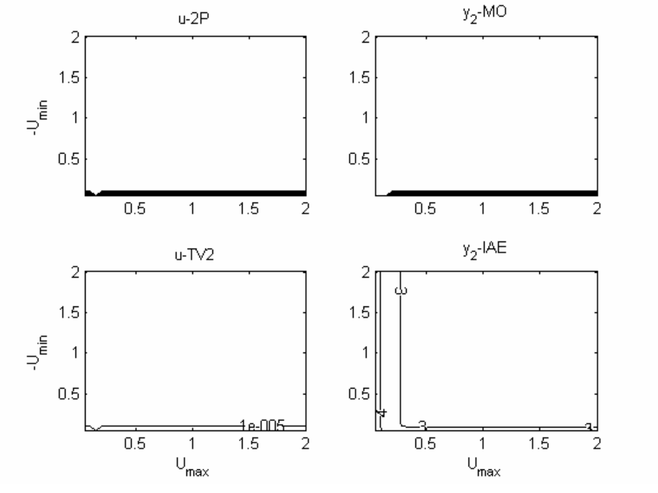


Fig. 9 Performance Portrait of the proposed gain-scheduling constrained pole assignment controller (34) for complex poles (42-43); 41x41 points (from gray to white); $K_s = 1;$ $\varepsilon = \varepsilon_u = \varepsilon_y = \left\{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\right\};$ $w = 1;$ $T_d = 0.5$

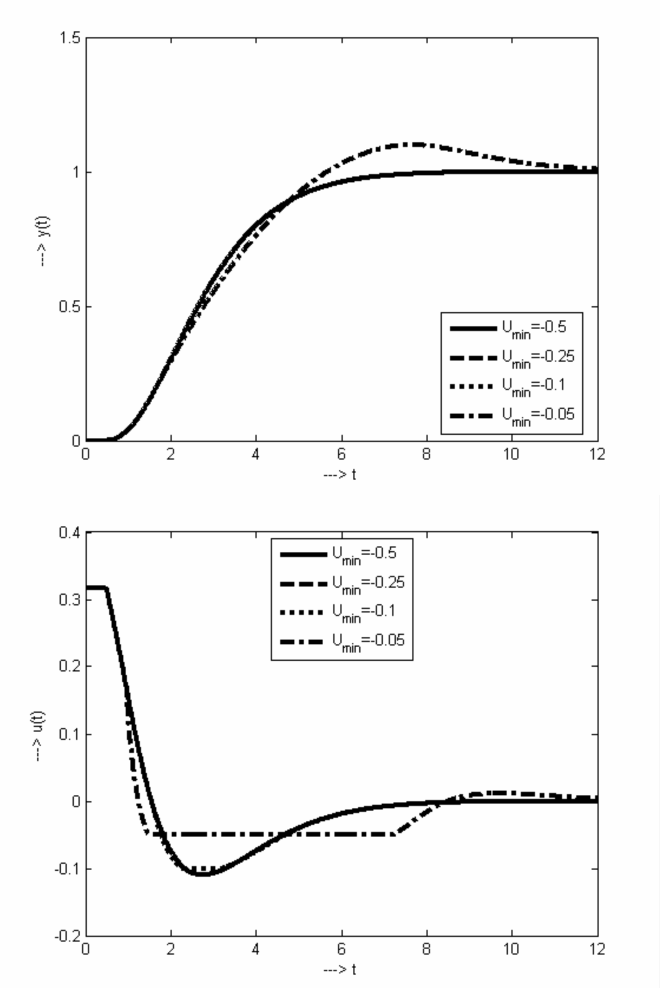


Fig. 10 Transient responses corresponding to $U_{\max}=1$ and several values of $U_{\min}\in\langle -0.05,-0.5\rangle$ for complex poles (42-43) and $K_s=1$; $w=1$; $T_d=0.5$

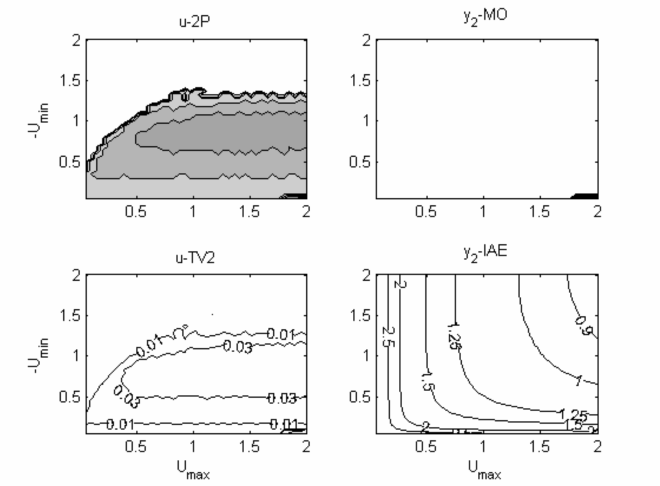


Fig. 11 Performance Portrait of the proposed gain-scheduling constrained pole assignment controller (34) for complex poles (42-43); 41x41 points (from gray to white); $K_s=1$;

$\varepsilon=\varepsilon_u=\varepsilon_y=\{0.1,0.05,0.02,0.01,10^{-3},10^{-4},10^{-5}\}$; $w=1$; $T_d=0.1$

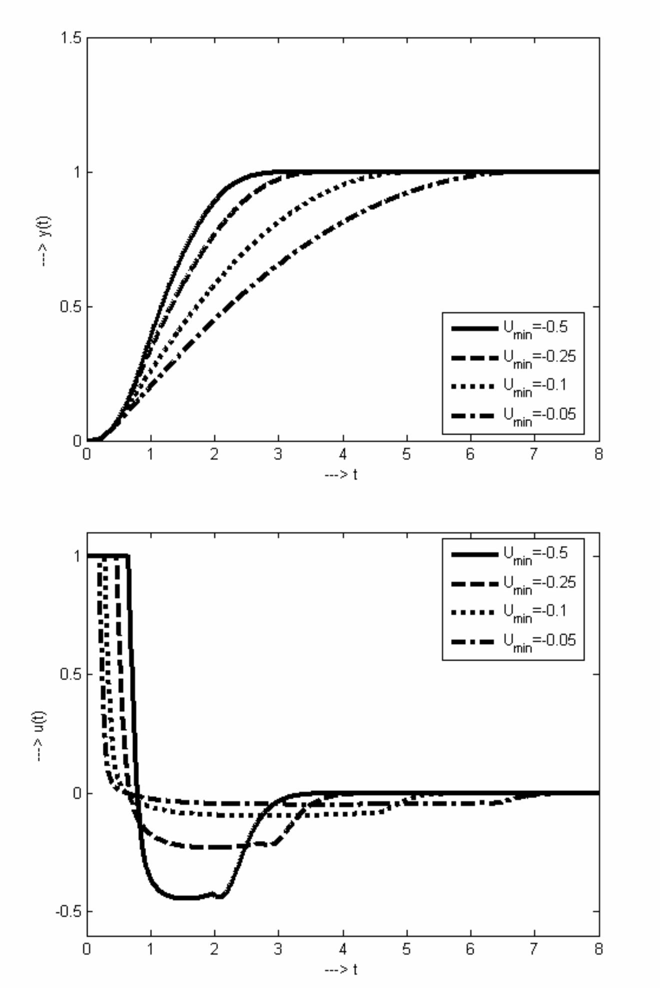


Fig. 12 Transient responses corresponding to $U_{\max}=1$ and several values of $U_{\min}\in\langle -0.05,-0.5\rangle$ for complex poles (42-43) and $K_s=1$; $w=1$; $T_d=0.2$

From the control signal transients it is to see slight imperfection occurring in the transition from nonlinear to linear control algorithm. A question is, if it is due to some imperfection of the algorithm, or it is result of limited precision in carried out calculations that might also influence results achieved for the strongest considered limit values $U_{\min}=-0.05$ and all analyzed dead time values.

Repeating the above analysis for controller with tuning (44-45) corresponding to $\varepsilon=10^{-5}$ it is possible to see from Fig. 13-18 that this relatively more aggressive tuning enlarges areas with stronger deviations from ideal shapes at the plant input and output and makes the imperfection for large ratio $|U_{\max}/U_{\min}|>10$ yet more visible. In the acceptable area of control that is still sufficiently large and interesting for many applications it shows, however, reasonable improvement of dynamics, when the IAE values dropped in the area of linear control for $T_d=0.5$ from above 2.95 to 2.35.

In the step responses influence of the more aggressive tuning is to observe by enlarged overshooting that appears also in situations when the previous tuning gave monotonic output responses. These results make the needs on revision of the controller design yet more acute.

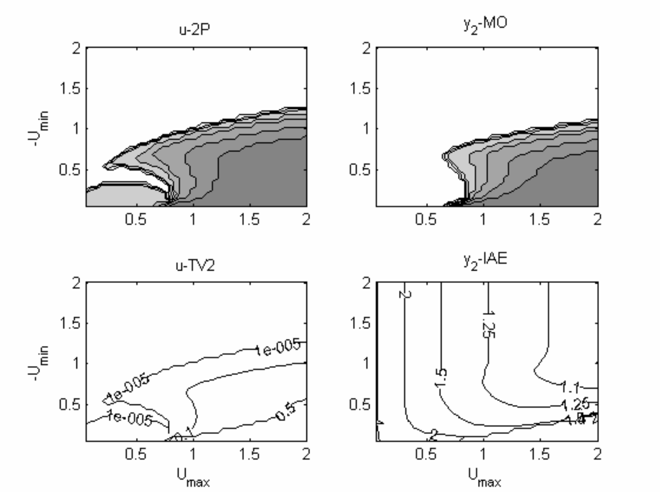


Fig. 13 Performance Portrait of the proposed gain-scheduling constrained pole assignment controller (34) for complex poles (44-45), $\varepsilon = 10^{-5}$; 41x41 points (from gray to white);

$\varepsilon = \varepsilon_u = \varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\}$; $K_s = 1$; $w = 1$; $T_d = 0.2$

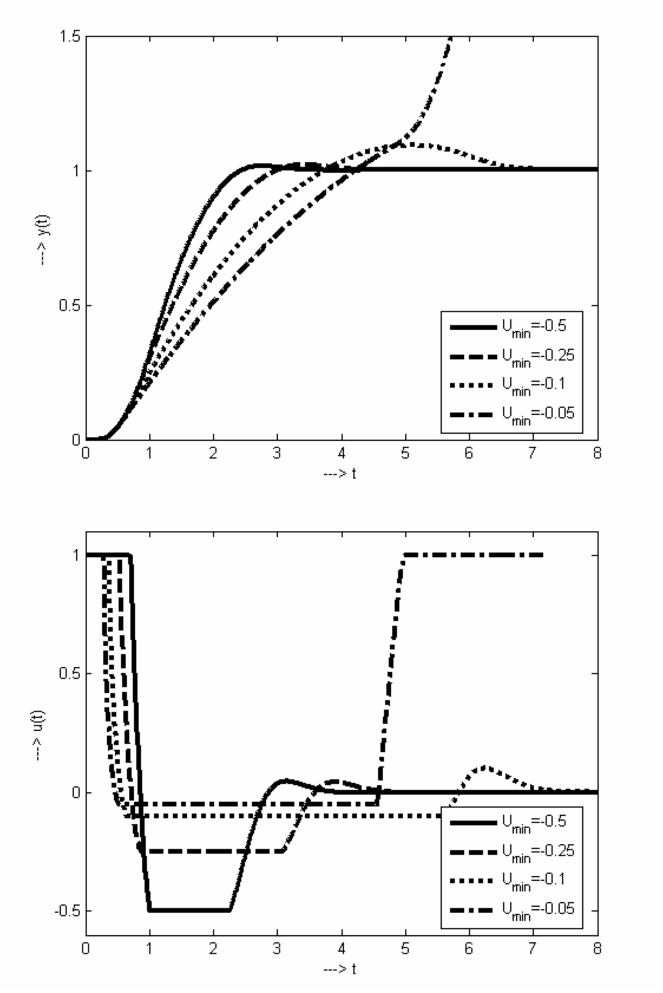


Fig. 14 Transient responses corresponding to $U_{\max} = 1$ and several values of $U_{\min} \in \langle -0.05, -0.5 \rangle$ for complex poles (44-45), $\varepsilon = 10^{-5}$ and $K_s = 1$; $w = 1$; $T_d = 0.2$

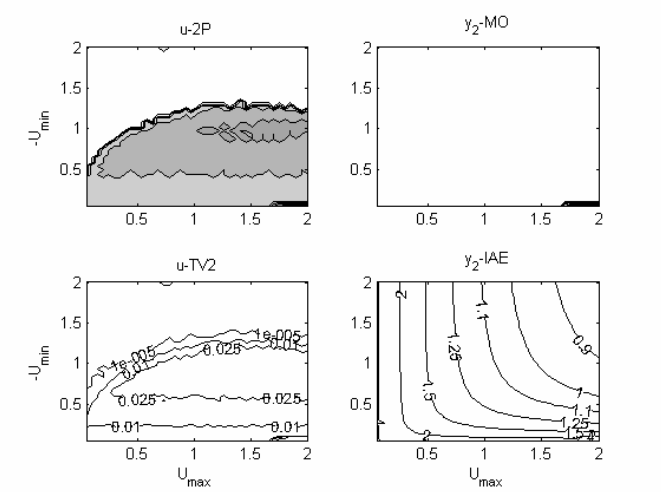


Fig. 15 Performance Portrait of the proposed gain-scheduling constrained pole assignment controller (34) for complex poles (44-45), $\varepsilon = 10^{-5}$; 41x41 points (from gray to white);

$\varepsilon = \varepsilon_u = \varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\}$; $K_s = 1$; $w = 1$; $T_d = 0.1$

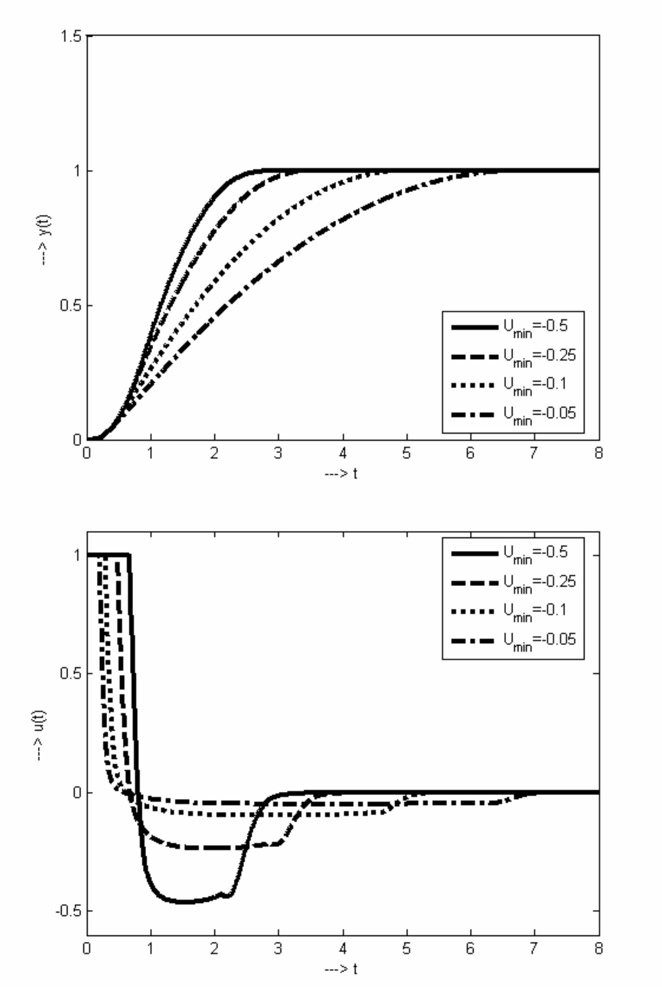


Fig. 16 Transient responses corresponding to $U_{\max} = 1$ and several values of $U_{\min} \in \langle -0.05, -0.5 \rangle$ for complex poles (44-45), $\varepsilon = 10^{-5}$ and $K_s = 1$; $w = 1$; $T_d = 0.1$

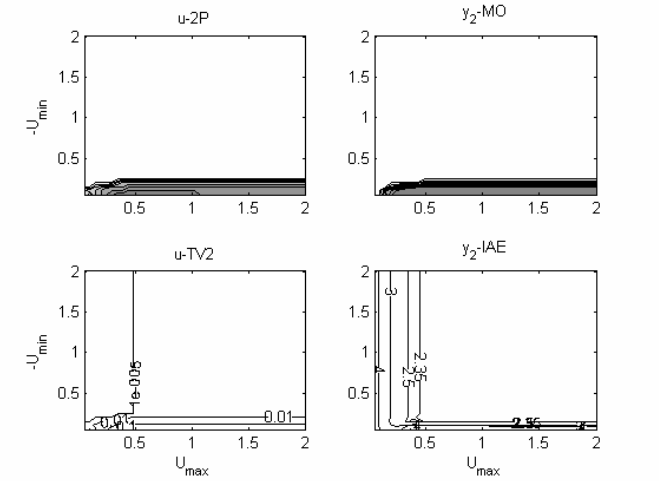


Fig. 17 Performance Portrait of the proposed gain-scheduling constrained pole assignment controller (34) for complex poles (44-45), $\varepsilon = 10^{-5}$; 41x41 points (from gray to white);

$\varepsilon = \varepsilon_u = \varepsilon_y = \{0.1, 0.05, 0.02, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\}$; $K_s = 1$; $w = 1$; $T_d = 0.5$

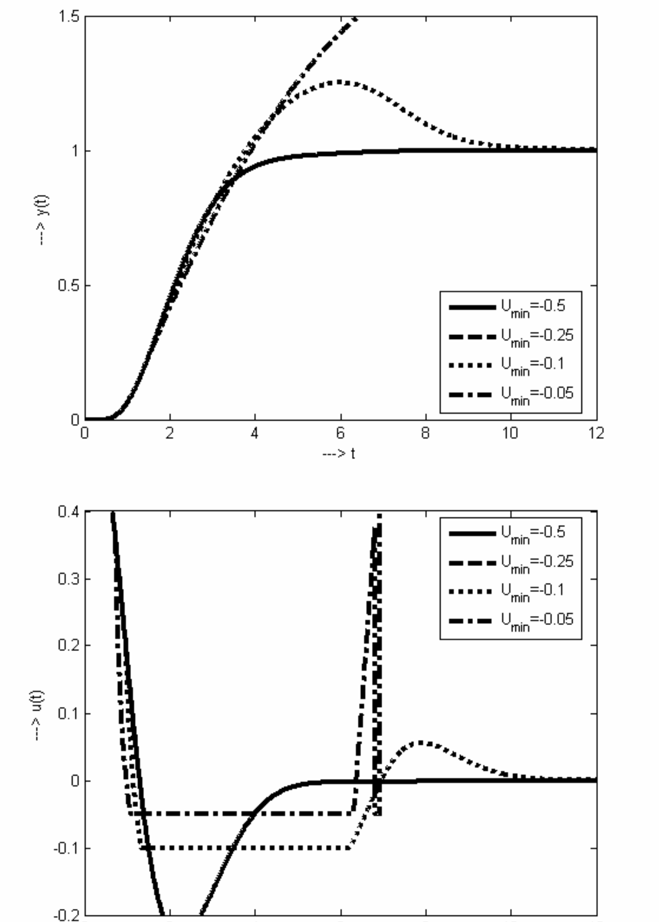


Fig. 18 Transient responses corresponding to $U_{\max} = 1$ and several values of $U_{\min} \in \langle -0.05, -0.5 \rangle$ for complex poles (44-45), $\varepsilon = 10^{-5}$ and $K_s = 1$; $w = 1$; $T_d = 0.5$

8. CONCLUSIONS

Analysis of the previous works dealing with the constrained design of SOIPDT system with closed loop performance specified by complex closed loop poles enabled detailed performance inspection of considered controllers and showed several interesting moments.

The analyzed controller may be interpreted as a variable structure one, or a gain scheduled controller with reference signal shaping. It does not introduce any new design parameter. So, the tuning procedure remains simple. It is even simpler to tune than in the linear case, since the closed loop poles can be specified directly by given dead time - independently from the given constraints!

Since the design uses just the well-known behavior patterns of the linear pole assignment trajectories and of the on-off control, which are well established in the engineering community, it is to assume that the new controllers will be also well accepted in practice.

Firstly it was shown that the previously used controller tuning based on the triple real dominant closed loop pole may be enriched by tuning given by the performance portrait method and corresponding to limit case of output monotonic and input 2P transients with the lowest IAE value of the output variable.

Detailed analysis of influence of constraints shown some not yet sufficiently covered areas occurring for $T_d = 0.2$ and $T_d = 0.5$ and very strong limitation put on the braking value $U_{\min} = -0.05$, while the limit value for acceleration remains relatively large (roughly for $|U_{\max} / U_{\min}| > 10$). Next research should show if the problem is caused by some neglected features of the considered constrained dynamics, or it is caused by insufficient precision of carried out calculations.

ACKNOWLEDGEMENTS

This work was partially supported by the Project VEGA 1/0656/09: Integration and development of nonlinear and robust control methods and their application in controlling flying vehicles, by the project KEGA 3/7245/09 Building virtual and remote experiments for network of online laboratories. It was also supported by the grant NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Åström, K. J., Hägglund, T. (1995a). *PID controllers: Theory, design, and tuning.* 2nd ed., ISA, Research Triangle Park, NC.

Åström, K. J., Hägglund, T. (1995b). New Tuning Methods for PID-controllers. *Proc. 3rd. European Control Conference*, Rome Italy, 1995, pp. 2456-2462.

Åström, K. J., Hägglund, T. (2004). Revisiting the Ziegler–Nichols step response method for PID control. *J. Process Control* 14, 635–650

Åström and T. Hägglund (2005). *Advanced PID Control*, ISA, Research Triangle Park, NC.

Bemporad, A., M. Morari, V. Dua and E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems. *Automatica,* **38**, 2002, pp. 3-20.

Górecki, H. (1971). *Analiza i synteza ukladów regulacji z opóznieniem.* Wydawnictwo Naukowo Techniczne Warszawa.

Guzmán,J.L. et al. (2008). Interactive tool for analysis of time - delay systems with dead - time compensators. *Control Eng. Practice*, 16, 7, 824-835.

Huba, M., P. Bisták, Z. Skachová and K. Žáková, Predictive Antiwindup PI-and PID-Controllers based on I1 and I2 Models With Dead Time. *6th IEEE Mediterranean Conference on Control and Systems*, Alghero 1998.

Huba, M. (1999). Dynamical Classes in the Minimum Time Pole Assignment Control. In: *Computing Anticipatory Systems - CASYS '98*. Woodbury: American Institute of Physics.

Huba, M. and P. Bisták, Dynamic Classes in the PID Control. In: *Proceedings of the 1999 American Control Conference*. San Diego: AACC, 1999.

Huba, M., D. Sovišová and I. Oravec, Invariant Sets Based Concept of the Pole Assignment Control. In: *European Control Conference ECC'99. VDI/VDE* Düsseldorf 1999.

Huba, M. (2003). *Constrained systems design. Vol.1 Basic controllers. Vol.2 Basic structures.* STU Bratislava (in Slovak).

Huba, M. (2004a). Design of Gain Scheduled Pole Assignment for SISO Systems. *EMCSR'04* Vienna.

Huba, M. (2004b) Gain Scheduled Constrained Controller For SISO Plants. In: *Southeastern Europe, USA, Japan and European Community Workshop on Research and Education in Control and Signal Processing REDISCOVER 2004*, Cavtat, Croatia, 1-104-107.

Huba, M. (2004c). Gain Scheduled Constrained Controller for SISO Plants. *Preprints IFAC Conf. NOLCOS* Stuttgart.

Huba, M. (2005). P- und PD-Polvorgaberegler für Regelstrecken mit begrenzter Stellgrösse. In: *Automatisierungstechnik AT* 53, 6, 273-283

Huba, M. (2006). Constrained pole assignment control, *In: Current Trends in Nonlinear Systems and Control*, L. Menini, L. Zaccarian, Ch. T. Abdallah, Edts., Boston: Birkhäuser, 163-183

Huba, M. (2010). Designing Robust Controller Tuning for Dead Time Systems. *IFAC Int. Conf. System Structure and Control*, Ancona, Italy.

Huba, M. (2011a). Basic Notions of Robust Constrained PID Control. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim.

Huba, M. (2011b). Robust Controller Tuning for Constrained Double Integrator. In: *NIL Workshop on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim,

Kothare, M.V., P.J. Campo, M. Morari and C.N.Nett, A unified framework for the study of anti-windup designs. *Automatica*, **30**, 1994, pp. 1869- 1883.

Kvasnica et al. (2011). Complexity Reduction in Explicit Model Predictive Control. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohaľ-Ilkiv Editors, STU Bratislava - NTNU Trondheim.

Shinskey, G. (1990). How good are Our Controllers in Absolute Performance and Robustness. *Measurement and Control*, Vol. 23, 114-121.

Skogestad, S. and I. Postlethwaite (2007). *Multivariable Feedback Control Analysis and Design*, John Wiley, N.York.

Ziegler,J.G. a Nichols,N.B. (1942). Optimum settings for automatic controllers. *Trans. ASME*, 759-768.

Comments – Remarks

Minimum Time PD-Controller Design for 2nd Order Plants

Mikuláš Huba*, and Pavol Bisták***

** Institute of Control and Industrial Informatics, FEI STU, Ilkovičova 3, SK-812 19 Bratislava
Slovakia (Tel: +421-2 -60291 771; e-mail: mikulas.huba@ stuba.sk).
** MI/PRT, FernUniversität in Hagen, Universitätsstr. 27, D-58084 Hagen
Germany (e-mail: mikulas.huba@fernuni-hagen.de)*

Abstract: This paper shows older approach to the design of the minimum time sampled-data control algorithms for linear 2nd order systems. Based on the general theory (Desoer and Wing, 1961) and its modification for the non-symmetrical amplitude constraints and the fully digital implementation (Huba, 1992), it enables to deal also with optimal control processes having more than one sign variation of the optimal control. Non-uniqueness of the optimal control arising at specific sampling periods in systems with complex poles is analyzed. As the main contribution, possibilities for simplification of the minimum time (pole assignment) control are pointed out that result from special shape of the band of proportional control that enables an easy segmentation and parametrization.

Keywords: Control system design; minimum time control; optimal control; saturation.

1. INTRODUCTION

After longer period of dominance of linear approaches, last trends in control design reflect the importance of the control signal constraints in combination with the discrete time essence of control. These can neither be fully replaced by the principles of rely (continuous-time) Minimum Time Control (MTC), or by those of the linear control extended by some anti-windup measures which are still not able to deal effectively with all situations, e.g. when the control signal hits both the upper and the lower saturation limit (Rönnbäck, 1996), or they seem to be too complex (Hippe, 2006). The impact of this situation is evident e.g. by inflation of different “optimal” PID tunings appearing at practically each control conference, or in conclusions that the PD controller cannot be optimally tuned (Aström and Hägglund, 1995).

The Model Predictive Control (MPC) developed due to the demand of practice on robust reliable controllers appropriate for a broad spectrum of industrial applications with different performance specifications and offering a broad spectrum of functionalities. The overall theoretical setting of this approach is much broader than that of the minimum time control. The early works on the minimum time sampled data systems (Kalman, 1957; Desoer and Wing, 1961), mentioned yet in some textbooks in 70s (Kuo, 1970) have been practically forgotten. But, the key problems, principles and solutions investigated in these early papers are still included within the problems of the MPC and so it might be interesting to recall them in looking for new inspiration and ideas.

The aim of this paper is to show that in the area, where the minimum time pole assignment control is not directly applicable, the pure minimum time control is not unique and relatively not sensitive to the actual choice of the control signal. This approves use of simplified control algorithms in this area. Simultaneously, simplified interpretation of optimal control valid rigorously just for a limited range of initial states is formulated, which is easier to implement and also to understand.

In chapters 2-8, basic properties of minimum time control systems are summarized based on early works by Desoer and Wing (1961), Busa and Huba (1986); Huba (1992); Huba and Bistak (2003). Chapters 8 and 9 show properties of minimum time control occurring for special sampling periods, which explain possibilities of the controller simplification. Chapter 10 deals with possible nonuniqueness of the vector representation and control design. Short summary and one possible development of the MTC are discussed in Conclusions in Chapter 11.

2. STATEMENT OF THE PROBLEM

The paper considers sampled data control of a continuous linear system ($\mathbf{A}_c$, $\mathbf{b}_c$) described as

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{b}u_{k+1}$$

$$\mathbf{A} = \mathbf{A}(T) = e^{-\mathbf{A}_c T} \ ; \ \mathbf{b} = \mathbf{b}(T) = \int_0^T e^{\mathbf{A}_c \tau}\mathbf{b}_c d\tau \qquad (1)$$

whereby $\mathbf{x}_k$ is the phase vector at time $t=kT$ and

$$u_{k+1} \in \langle U_1, U_2 \rangle; \ U_1 < 0 \ ; \ U_2 > 0 \qquad (2)$$

is the control over the interval $kT \le t < (k+1)T$. The minimum time regulator problem may be stated as follows: given any admissible initial state $\mathbf{x}_0$, it is required to bring the system to the demanded state **w**=**0** with a control constrained by (2).

Applying the control sequence $\{u_i\}_1^N;\ u_i \in \langle U_1, U_2 \rangle$ for a sampling period *T* one gets series of states

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{A}\mathbf{x}_0 + \mathbf{b}u_0 \\ \mathbf{x}_2 &= \mathbf{A}\mathbf{x}_1 + \mathbf{b}u_1 = \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{b}u_0 + \mathbf{b}u_1 \\ &\vdots \\ \mathbf{x}_N &= \mathbf{A}^N\mathbf{x}_0 + \mathbf{A}^{N-1}\mathbf{b}u_0 + \ldots + \mathbf{A}\mathbf{b}u_{N-2} + \mathbf{b}u_{N-1} \end{aligned} \tag{3}$$

If after that $\mathbf{x}_N$=**0**, Eqs. (3) can be rearranged by introducing vectors

$$\mathbf{v}_k = -\mathbf{A}^{-k}\mathbf{b} = -\mathbf{A}(-kT)\mathbf{b}(T)\ ;\ k = 1,2,.... \tag{4}$$

into the canonical representation of the initial states

$$\mathbf{x}_0 = \mathbf{v}_1 u_1 + \mathbf{v}_2 u_2 + \cdots + \mathbf{v}_N u_N \tag{5}$$

Basic Acronyms	
FAB	Full-Acceleration-Boundary
FBB	Full-Acceleration-Boundary
CPAC	Constrained Pole Assignment Control
MTC	Minimum Time Control
$\mathrm{P_b}$	Zone of Proportional Control
PD	Proportional Derivative
$\mathrm{R_N}$	Reachability set from which the origin is reachable in N steps
RBC	Reference Braking Curve
ZC	Zero (Critical) Curve

3. SETS $\mathbf{R_N}$ AND $\mathbf{I_n}$; BOUNDARY STATES

$\mathbf{R}_N$ is the set of all initial states that can be brought to the origin by the control (2) in *N* sampling period or less:

$$R_N = \left\{ \mathbf{x}_0 \mid \mathbf{x}_0 = \sum_{i=1}^{N} \mathbf{v}_i u_i\ ;\ u_i \in \langle U_1; U_2 \rangle; i = 1,2,\ldots,N \right\} \tag{6}$$

Desoer and Wing have shown that $\mathbf{R}_N$ are convex:
$\mathbf{R}_1$ is the line segment with vertices

$$V_1^1 = \mathbf{v}_1 U_1\ ;\ V_1^2 = \mathbf{v}_1 U_2 \tag{7}$$

$\mathbf{R}_2$ is the parallelogram whose edges are parallel to $\mathbf{v}_1$ and $\mathbf{v}_2$. Its vertices (see also Huba, 1992) are:

$$\begin{aligned} V_2^{11} &= \mathbf{v}_1 U_1 + \mathbf{v}_2 U_1\ ;\ V_2^{12} = \mathbf{v}_1 U_1 + \mathbf{v}_2 U_2 \\ V_2^{21} &= \mathbf{v}_1 U_2 + \mathbf{v}_2 U_1\ ;\ V_2^{22} = \mathbf{v}_1 U_2 + \mathbf{v}_2 U_2 \end{aligned} \tag{8}$$

For *N*>2, $\mathbf{R}_N$ is obtained inductively. Suppose $\mathbf{R}_N$ is known, let us construct $\mathbf{R}_{N+1}$. $\mathbf{R}_N$ being a convex polygon is completely described by its vertices. Let us classify these vertices in two classes: let δ be an arbitrarily small positive number. The class P_N^2 contains all the vertices V_N of $\mathbf{R}_N$, such that $V_N + \delta\mathbf{v}_{N+1} \notin \mathbf{R}_N$. P_N^1 contains all vertices V_N of $\mathbf{R}_N$, such that $V_N - \delta\mathbf{v}_{N+1} \notin \mathbf{R}_N$. These two classes usually have common vertices characterized by the maximum distance from a line going through the origin parallel to $\mathbf{v}_{N+1}$. The vertices of $\mathbf{R}_{N+1}$ are obtained from those of $\mathbf{R}_N$ as follows:

- to each vertex V_N of P_N^1 corresponds the vertex of $\mathbf{R}_{N+1}$: $V_N + \mathbf{v}_{N+1}U_1$.
- to each vertex V_N of P_N^2 corresponds the vertex of $\mathbf{R}_{N+1}$: $V_N + \mathbf{v}_{N+1}U_2$.

The boundary of $\mathbf{R}_{N+1}$ is obtained from that of $\mathbf{R}_N$ by adding vectors $\mathbf{v}_{k+1}U_j$ in an outward direction. In all linear 2nd order systems the origin is surrounded by sets $\mathbf{R}_N$ with 2*N* vertices of the form

$$V_{k,N-k}^j = U_j \sum_{i=1}^{k} \mathbf{v}_i + U_{3-j} \sum_{i=k+1}^{N} \mathbf{v}_i \tag{9}$$

corresponding to sequences where: 1) each u_i is maximum in absolute value (i.e. equal to U_1 or U_2, respectively); 2) the sequence consists of 2 subsequences of u_i's that have the same sign and each subsequence has a sign opposite to that of the another one, *k* being the number of steps of the 1st subsequence with u_i=U_j, *N-k* the number of steps with the opposite limit value U_{3-j}.

In systems with complex poles there are also possible sequences consisting of more than 2 subsequences (with more than 1 sign variation). A vertex with 2 sign variations can then be expressed as

$$V_{k,l,N-k-l}^j = U_j \sum_{i=1}^{k} \mathbf{v}_i + U_{3-j} \sum_{i=k+1}^{k+l} \mathbf{v}_i + U_j \sum_{i=k+l+1}^{N} \mathbf{v}_i \tag{10}$$

with *k* and *1* denoting the number of steps of the 1st or of the 2nd subsequence, respectively. It is yet convenient to partition the set off all initial states into sets $\mathbf{I}_n$ in terms of the number of trains of pulses with a constant sign in the optimal control sequence.

From the construction of $\mathbf{R}_N$ it is obvious that all its edges are parallel to one of the vectors. Points of an edge of $\mathbf{I}_2$ parallel to $\mathbf{v}_\mathrm{m}$ can then be expressed as

$$\begin{aligned} B_N^j &= \mathbf{v}_1 U_j + \cdots + \mathbf{v}_{m-1} U_j + \mathbf{v}_m u_m + \mathbf{v}_{m+1} U_{3-j} + \cdots + \mathbf{v}_N U_{3-j} \\ u_m &\in \langle U_1, U_2 \rangle,\ m \ne N;\ u_m \in \left(0, U_{3-j}\right\rangle,\ m = N \end{aligned} \tag{11}$$

In $\mathbf{I}_3$ the boundary states can be expressed as

$$\begin{aligned} B_N^j &= U_j \sum_{i=1}^{m-1} \mathbf{v}_i + \mathbf{v}_m u_m + U_{3-j} \sum_{i=m+1}^{m+l} \mathbf{v}_i + U_j \sum_{i=m+l+1}^{N} \mathbf{v}_i \\ u_m &\in \langle U_1, U_2 \rangle \end{aligned} \tag{12}$$

$$\begin{aligned} B_N^j &= U_j \sum_{i=1}^{k} \mathbf{v}_i + U_{3-j} \sum_{i=k+1}^{m-1} \mathbf{v}_i + \mathbf{v}_m u_m + U_j \sum_{i=m+1}^{N} \mathbf{v}_i \\ u_m &\in \langle U_1, U_2 \rangle,\ m \ne N;\ u_m \in \left(0, U_j\right\rangle,\ m = N \end{aligned} \tag{13}$$

4. UNIQUENESS PROPERTIES

Desoer and Wing (1961) have shown that if $\mathbf{x}_0 = \mathbf{v}_1 u_1 \in \mathbf{R}_1$, one and only one value of $\mathbf{x}_0$ corresponds to each value of u_1 and the canonical representation is unique. If $\mathbf{x}_0 = \mathbf{v}_1 u_1 + \mathbf{v}_2 u_2 \in \mathbf{R}_2$, two rows of this vector equation uniquely determine the values u_1 and u_2 under the condition that the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are not parallel, which is equivalent to the controllability condition $\det[\mathbf{b} \quad \mathbf{Ab}] \neq 0$.

The expression of the vertices and the points B_N on the boundary of $\mathbf{R}_N$ is also unique. The canonical representation (6) is, however, not unique in interior states of $\mathbf{R}_N$, N>2. For these points they have proposed a recursive construction of canonical representation, which has seemed to them to be the simplest one: An interior point of $\mathbf{R}_N$ can be obtained from a boundary point of $\mathbf{R}_{N-1}$ by adding the vector $\mathbf{v}_N u_N$ in an outward direction

$$\mathbf{x}_0 = B_{N-1}^j + \mathbf{v}_N u_N \tag{14}$$

From (11-14) it follows that an optimal control sequence can involve, besides the trains of saturated u_i's, two non-saturated steps: the first occurring at the sign change and the second in the last control step. However, Desoer and Wing have not observed another significant feature of the construction (14): It gives $|u_N| = \min$! From (3) it immediately follows that the control error at the last-but-one sampling instant takes minimal possible value:

$$\mathbf{x}_{N-1} = -\mathbf{A}(-T)\mathbf{b}(T) = \mathbf{v}_1 u_N \tag{15}$$

The output variable reaches an ε-neighborhood of the origin in the minimum time!

5. PROPORTIONAL BAND

In an usual application it is not necessary to produce the optimal control sequence all at once at time t=0. Instead, it is more advantageous to compute only the control signal u_1 for the next control interval and to repeat the computation at each new sampling instant. This does note change the system behavior and at the same time simplifies the computer operation and enables registration of unexpected disturbances. Let us consider those $\mathbf{x}_0 \in \mathbf{I}_2$ for which $u_1 \in \langle U_1, U_2 \rangle$. From (11) and (14) for m=1 it follows

$$\mathbf{x}_0 = \mathbf{v}_1 u_1 + U_j \sum_{i=2}^{N-1} \mathbf{v}_i + \mathbf{v}_N u_N$$
$$u_1 \in \langle U_1, U_2 \rangle; \quad u_N \in (0, U_j\rangle; \; j = 1,2 \tag{16}$$

Equivalently, for $\mathbf{x}_0 \in \mathbf{I}_3$ from (12) and (14)

$$\mathbf{x}_0 = \mathbf{v}_1 u_1 + U_j \sum_{i=2}^{l+1} \mathbf{v}_i + U_{3-j} \sum_{i=l+2}^{N-1} \mathbf{v}_i + \mathbf{v}_N u_N$$
$$u_1 \in \langle U_1, U_2 \rangle; \; u_N \in (0, U_{3-j}\rangle; \; j = 1,2 \tag{17}$$

The set of all points with $u_1 \in \langle U_1, U_2 \rangle$ is denoted as the **Proportional Band (P$_\mathbf{b}$)**. By the second term in (15) and (16) denoted as $\mathbf{z}_N^j$, whereby in $\mathbf{I}_2$

$$\mathbf{z}_N^j = U_j \sum_{i=2}^{N-1} \mathbf{v}_i + \mathbf{v}_N u_N \; ; \; u_N \in (0, U_j\rangle; \; j = 1,2 \tag{18}$$

the **Critical** or **Zero Curve (ZC)** is defined, which consists of points at which u_1=0 (dividing the phase plane according to the sign of u_1). In $\mathbf{I}_2$ this polygonal curve is obtained by joining vertices

$$Z_N^j = \begin{bmatrix} z_N^j \\ \dot{z}_N^j \end{bmatrix} = U_j \sum_{i=2}^{N} \mathbf{v}_i, \; N > 1 \; ; \; Z_0^j = \mathbf{0} \; ; \; j = 1,2 \tag{19}$$

ZC of P$_\mathrm{b}$ in $\mathbf{I}_3$ has interior points

$$\mathbf{z}_N^j = U_j \sum_{i=2}^{l+1} \mathbf{v}_i + U_{3-j} \sum_{i=l+2}^{N-1} \mathbf{v}_i + \mathbf{v}_N u_N \; ; u_N \in (0, U_{3-j}\rangle; \; j = 1,2 \tag{20}$$

and vertices

$$Z_N^j = U_j \sum_{i=2}^{l+1} \mathbf{v}_i + U_{3-j} \sum_{i=l+2}^{N} \mathbf{v}_i ; \; j = 1,2 \tag{21}$$

For u_1=U_j in (16) and (17) a new polygonal curve is defined, referred originally as the **Switching Curve** (SC) due to its relation to the switching curve of the relay minimum time systems. In $\mathbf{I}_2$ it has vertices

$$X_N^j = V_{N,0}^j = Z_N^j + \mathbf{v}_1 U_j = U_j \sum_{i=1}^{N} \mathbf{v}_i \; ; \; j = 1, \, 2 \tag{22}$$

In $\mathbf{I}_2$ SC represents an invariant set, along which the system is braked to the origin, it could also be denoted as the **Reference Braking Curve (RBC)**. Since the last property does not hold in $\mathbf{I}_n$ n>2**,** in such situations it will be denoted as the **Full-Braking-Boundary (FBB)**. In $\mathbf{I}_3$

$$X_N^j = Z_N^j + \mathbf{v}_1 U_j = U_j \sum_{i=1}^{l+1} \mathbf{v}_i + U_{3-j} \sum_{i=l+2}^{N} \mathbf{v}_i ; \; j = 1,2 \tag{23}$$

Putting u_1=$U_{3\text{-}j}$ in (16) and (17), vertices of the **Full-Acceleration-Boundary (FAB)** are defined

$$Y_N^j = V_{1,N-1}^{3-j} = Z_N^j + \mathbf{v}_1 U_{3-j} = X_N^j + (U_{3-j} - U_j)\mathbf{v}_1 \; ; \; j = 1, \, 2 \tag{24}$$

Both these polygonal curves represent boundaries of P$_\mathrm{b}$, in which the control signal is not saturated and therefore has to be computed. Outside of P$_\mathrm{b}$ u_1=U_1 or U_2, which depends only on the representative point position with respect to the ZC.

The proportional zone is divided into two parts by a line parallel to $\mathbf{v}_1$ and crossing the origin

$$\mathbf{x} = q\mathbf{v}_1, \; q \in (-\infty, \infty) \tag{25}$$

By eliminating q the line equation can be written as

$$p = 0 \; ; \qquad p = y - \dot{y}\frac{v_1}{\dot{v}_1} \tag{26}$$

To the left of p, j=1 will be substituted into (17-24) and to the right j=2. In order to avoid the ambiguous situation p=0, "j" can be determined according to

if p < 0 then j=1 else j=2 (27)

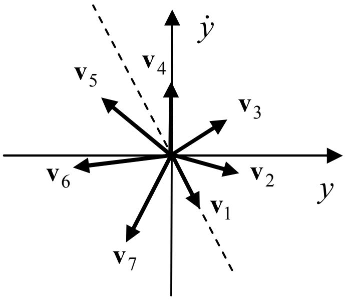


Fig.1. Vektors $\mathbf{v}_k$ of the oscillating plant; a_0= 0.82; a_1=0.2; T=1.

Now the question is, how long P$_b$ can be constructed according to the formulas for $\mathbf{I}_2$ and when one has to use formulas derived for $\mathbf{I}_n$, n>2. Let the characteristic polynomial of the original system with complex poles be $A(s)=s^2+a_1s+a_0$. Observe that increasing k the length of vectors $\mathbf{v}_k$ (4) is increasing in stable systems (Fig.1) and decreasing in unstable systems. Simultaneously, $\mathbf{v}_k$ also rotate in each step by an angle $\Delta\varphi$. If the formula (19) is used for $N=\mathrm{int}(\pi/\Delta\varphi)+1$, *int*=integer part, the new point belongs to $\mathbf{R}_{N\text{-}1}$, $\mathbf{v}_NU_j$ points from Z_{N-1}^j into $\mathbf{R}_{N\text{-}1}$ (the angle between $\mathbf{v}_1$ and $\mathbf{v}_N$ is greater then π). Therefore, the new vertex Z_N^j of ZC can only be constructed from the previous one by adding the vector $\mathbf{v}_NU_{3\text{-}j}$ pointing in the outwards direction. That holds for all $\mathbf{v}_N$ having tips to the left of p: The new vertex should be constructed using recursion

$$Z_N^j=Z_{N-1}^j+\mathbf{v}_NU_i \tag{28}$$

whereby

$$\text{if } p(\mathbf{v}_N)>0 \quad i\text{=}j \tag{29}$$

$$\text{if } p(\mathbf{v}_N)<0 \quad i\text{=}3\text{-}j$$

Equivalently, points $\mathbf{z}_N\in Z_{N-1}^jZ_N^j$ are expressed as

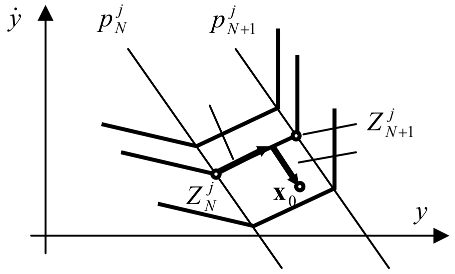


Fig.2. Computation of the control signal u_1.

$$\mathbf{z}_N^j=Z_{N-1}^j+\mathbf{v}_Nu_N\,;\ u_N\in(0,U_i) \tag{30}$$

6. CONTROL ALGORITHM

From modified expressions (16-20)

$$\begin{gathered}x_0=\mathbf{v}_1u_1+\mathbf{z}=\mathbf{v}_1u_1+Z_N^j+\mathbf{v}_{N+1}u_{N+1}\\ N=2,3,...;\ j=1,2\,;\\ u_1\in<U_1,U_2>\,;\ u_{N+1}\in(0,U_i>\end{gathered} \tag{31}$$

whereby the value U_i is given by (29), it is obvious that the value u_1 can be determined as the distance of the initial state $\mathbf{x}_0$ to ZC in the direction of $\mathbf{v}_1$.

The line crossing $\mathbf{x}_0$ in direction $\mathbf{v}_1$ (Fig.2) is given by the equation $\mathbf{x}=\mathbf{z}+\mathbf{v}_1u_1\ ;\ u_1\in(-\infty,\infty)$ whereby $\mathbf{z}=Z_N^j+\mathbf{v}_{N+1}u_{N+1}$. Eliminating $\mathbf{z}$ and u_N yields

$$\begin{aligned}u_1&=(\dot{v}_{N+1}y-v_{N+1}\dot{y}+\dot{z}_Nv_{N+1}-z_N\dot{v}_{N+1})/(\dot{v}_{N+1}v_1-v_{N+1}\dot{v}_1)=\\&=r_0(N)y+r_1(N)\dot{y}+r_c(N)\end{aligned} \tag{32}$$

7. STRIPS S_N^j; DETERMINATION OF N

After limiting the calculated value (32) according to (2), the control algorithms can be used in all points of the strip $S_N{}^j$ determined by parallel lines p_{N+1}^j and p_N^j crossing Z_{N+1}^j and Z_N^j in direction of $\mathbf{v}_1$. p_N^j is described as $\mathbf{x}=Z_N^j+q\mathbf{r}_1\ ;\ q\in(-\infty,\infty)$, or after eliminating the parameter q by the equation

$$p_N^j(\mathbf{x})=0;$$

$$p_N^j(\mathbf{x})=y-\frac{v_1\left(\dot{y}-\dot{z}_N^j\right)}{\dot{v}_1}-z_N^j=p+c_N^j \tag{33}$$

The value of the parameter N in (32) is determined for the point $\mathbf{x}_0$ to be between lines p_N^j and p_{N+1}^j ($\mathbf{x}_0\in S_N^j$), which can be realized by an iterative procedure, until the inequalities

$$\left(p.p_{N+1}^j<0\right)AND\left(p.p_N^j>0\right) \tag{34}$$

are fulfilled (Huba, Sovišová and Spurná, 1987).

8. BEHAVIOUR IN $\mathbf{I}_2$

Considering an initial point $\mathbf{x}_0\in P_b\subset\mathbf{I}_2$ (16) and the applied optimal control u_1, the representative point takes in the next sampling instant value

$$\begin{aligned}\mathbf{x}_1&=\mathbf{A}\left[\mathbf{v}_1u_1+U_j\sum_{i=2}^{N-1}\mathbf{v}_i+\mathbf{v}_Nu_N\right]+\mathbf{b}u_1=\\&=U_j\sum_{i=1}^{N-2}\mathbf{v}_i+\mathbf{v}_{N-1}u_N\ ;\ u_N\in\left(0,U_j\right\rangle;\ j=1,2\end{aligned} \tag{35}$$

This is a point of the line segment $X_{N-1}^jX_{N-2}^j$ of the RBC. So, from points of P$_b$ the optimal solution tends in the next sampling period to RBC!

When expressing the distance of an initial point $\mathbf{x}_0=V_{k,N-k}^{3-j}\in\mathbf{I}_2$ (9) from the RBC by the number k of steps with $u=U_{3-j}$, it is clear from

$$\begin{aligned}\mathbf{x}_1&=\mathbf{A}\left(U_{3-j}\sum_{i=1}^{k}\mathbf{v}_i+U_j\sum_{i=k+1}^{N}\mathbf{v}_i\right)+\mathbf{b}U_{3-j}=\\&=U_{3-j}\sum_{i=1}^{k-1}\mathbf{v}_i+U_j\sum_{i=k}^{N-1}\mathbf{v}_i\end{aligned} \tag{36}$$

that under optimal control with u_1=$U_{3\text{-}j}$ this distance decreases by one sampling period in each control step. So, in general, the aim of the optimal control for $\mathbf{x}_0\in\mathbf{I}_2$ can be interpreted as: **To reach RBC in the minimum time possible**. This task can be easily solved also without the reachibility sets: RBC is traced out by vertices $X_N{}^j$ computed by N backward steps from the origin under the control u=U_j as

$$X_N^j=\mathbf{b}(-NT)U_j\ ;\ j=1,2 \tag{37}$$

The control algorithms can be determined by the requirement to reach RBC in one control step

$$\mathbf{A}\mathbf{x}_0 + \mathbf{b}u_1 = X_{N-1}^j + q\mathbf{v}_N \tag{38}$$

If the computed control signal exceeds the given constraints (2), it has simply to be limited, so that

$$\begin{aligned} u_1 &= sat\left\{[0 \quad 1]\left[\mathbf{v}_N \middle| \, \mathbf{b}\right]^{-1}\left(X_{N-1}^j - \mathbf{A}\mathbf{x}_0\right)\right\} = \\ &= sat\{r_0(N)y + r_1(N)\dot{y} + r_c(N)\} \end{aligned} \tag{39}$$

9. SPECIAL CASE

In the case of an oscillating undamped system with $a_1 = 0, a_0 = \omega^2$ and the sampling period satisfying

$$K\omega T = \pi \tag{40}$$

(K being positive integer and $\Delta\varphi = \omega T$ the angle traced out by $\mathbf{v}_k$ and $\mathbf{v}_{k+1}$), P$_\text{b}$ acquires a regular shape and all its vertices can be expressed in a relatively simple way. As the consequence of (40)

$$\mathbf{v}_{k+iK} = (-1)^i \mathbf{v}_k \tag{41}$$

It is now convenient to introduce strips ${P_m}^j$, ${N_m}^j$ and ${Q_M}^j$ (Fig.3): with m denoting the number of begun half rounds of $\mathbf{v}_N$ around the origin and M denoting the begun full rounds of $\mathbf{v}_N$.

The strips ${P_m}^j$ are defined as a union of all strips ${S_N}^j$ in $\mathbf{I}_n$ n=m+1 with Z_N^j constructed using U_i=U_j in (28) (vector $\mathbf{v}_N$ is to the right of p)

$${P_m}^j = \bigcup_{N=(2m-2)K+1}^{(2m-1)K} S_N^j \tag{42}$$

The strips ${N_m}^j$ are defined as a union of strips ${S_N}^j$ in $\mathbf{I}_n$, n=m+1 with Z_N^j constructed using U_i=U_{3-j} in (28) (vector $\mathbf{v}_N$ is to the left of p)

$${N_m}^j = \bigcup_{N=(m-1)K+1}^{2(m-2)K} S_N^j \tag{43}$$

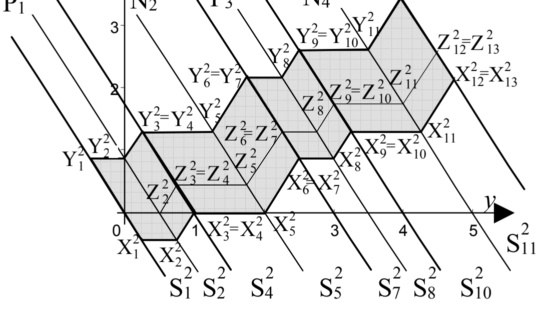


Fig.3. Special case: K=3. P$_\text{b}$ corresponding to (58c).

The union of all strips ${S_N}^j$ corresponding to the complete M-th round of $\mathbf{v}_N$ will be denoted as the strip ${Q_M}^j$

$${Q_M}^j = \bigcup_{N=2(M-1)K+1}^{2MK} S_N^j = P_{2M-1}^j \bigcup N_{2M}^j \tag{44}$$

The strips ${P_1}^j$ and ${N_2}^j$ are separated by a line (33) crossing the point

$$Z_K^j = \mathbf{b}(-KT)U_j - \mathbf{v}_1 U_j = \mathbf{z}_K U_j \tag{45}$$

The lines p_N^j (33) crossing X_N^j in $\mathbf{I}_2$ are described by

$$\begin{aligned} p &= y + \frac{\tan(\omega T/2)}{\omega}\dot{y}; \\ c_N^j &= \frac{\cos(\omega T(N-1/2)) - \cos(\omega T/2)}{\omega^2 \cos(\omega T/2)} Uj \end{aligned} \tag{46}$$

From (41) and from (28-29) it follows that

$$Z_{2K}^j = Z_K^j - \mathbf{z}_K U_{3-j} \tag{47}$$

and the constant c_{2K}^j of the line crossing Z_{2K}^j and separating ${N_2}^j$ from ${P_3}^j$ and, simultaneously, ${Q_1}^j$ from ${Q_2}^j$ is

$$c_{2K}^j = \frac{\cos\left(\dfrac{\omega T(2K-1)}{2}\right) - \cos\left(\dfrac{\omega T}{2}\right)}{\omega^2 \cos\left(\dfrac{\omega T}{2}\right)} \left(U_j - U_{3-j}\right) \tag{48}$$

In general, the strips ${Q_M}^j$ and ${Q_{M+1}}^j$ are separated by the lines (46) with

$$c_{2MK}^j = Mc_{2K}^j \tag{49}$$

Putting such a line through a point of a strip Q_M^j

$$p(\mathbf{x}_0) + c_{2MK}^j = p(\mathbf{x}_0) + Mc_{2K}^j = 0 \tag{50}$$

and solving it in terms of M results in

$$M = -p(\mathbf{x}_0)/c_{2K}^j \tag{51}$$

which takes in ${Q_M}^j$ values $M^* \in (M, M+1)$. Hence, the index M can be determined as

$$M = \operatorname{int}(M^*) \tag{52}$$

The strips ${Q_M}^j$ are partitioned into the strips ${P_m}^j$ and ${N_m}^j$ by the lines (46) with constants

$$c_{K+2K(M-1)} = c_K^j + (M-1)c_{2K}^j \tag{53}$$

By evaluating the sign of expression (33) with constant (53) one can determine, in which of the two strips the initial point is. Then the computation of the parameter N of the actual strip can be finished iteratively. But, it is also possible to derive an explicit solution based on the solution in P_1^j . Here, writing equation of line (46) crossing a point $\mathbf{x}_0 \in {S_N}^j$ and solving it in terms of N gives

$$N = \operatorname{int}\left\{\frac{1}{2} + \frac{1}{\omega T}\arccos\left[\cos\left(\frac{\omega T}{2}\right)\left(1 + \frac{\omega^2 p(\mathbf{x}_0)}{U_j}\right)\right]\right\} \tag{54}$$

The control signal can be computed by means of (54) and (32) directly, without iterations. The same can be generalized for N_2^j with $N = \overline{N} + K$ and by determining $\overline{N}$ according to (54) with $p(\mathbf{x}_0)/U_j$ being replaced by $p(\mathbf{x}_0 - Z_K^j)/\left[|U_{3-j}|sign(U_j)\right]$.

10. NONUNIQUENESS OF CANONICAL REPRESENTATION

Choice of the sampling period (40) also varies the shape of sets $\mathbf{R}_N$. In $\mathbf{I}_3$ the edges of $\mathbf{R}_N$ parallel to $\mathbf{v}_1$ can be expressed using (12) with m=1 as

$$B_N^j = \mathbf{v}_1 u_1 + U_{3-j}\sum_{i=2}^{K}\mathbf{v}_i + \mathbf{v}_{K+1}U_j + U_j\sum_{i=K+2}^{N}\mathbf{v}_i \qquad (55a)$$

$$u_1 \in \langle U_1, U_2 \rangle$$

and the edges parallel to $\mathbf{v}_{K+1}$ with m=K+1 as

$$B_N^j = \mathbf{v}_1 U_j + U_{3-j}\sum_{i=2}^{l+1}\mathbf{v}_i + \mathbf{v}_{K+1}u_{K+1} + U_j\sum_{i=K+2}^{N}\mathbf{v}_i \qquad (55b)$$

$$u_{K+1} \in \langle U_1, U_2 \rangle$$

Both these edges have one common vertex

$$V_{1,l,N-l-1}^j = \mathbf{v}_1 U_j + U_{3-j}\sum_{i=2}^{K}\mathbf{v}_i + U_j\sum_{i=K+1}^{N}\mathbf{v}_i \qquad (56)$$

Changing u_1 from U_j to $U_{3\text{-}j}$ the point (55a) moves from the vertex (56) and traces out line segment $(U_{3\text{-}j}\text{-}U_j)\mathbf{v}_1$. In accordance with (41), a motion of the point (55b) with the length of $(U_j\text{-}U_{3\text{-}j})\mathbf{v}_1$ corresponds to equivalent change of u_{K+1}. Both edges form together a line segment with the length $2(U_{3\text{-}j}\text{-}U_j)\mathbf{v}_1$. The point (56) becomes to be an internal point of this line segment. From the relation

$$\mathbf{v}_1 U_j + \mathbf{v}_{\mathrm{K}+1}U_j = \mathbf{v}_1(U_j - U_j) = \mathbf{v}_1(u_1 + u_{K+1}) \qquad (57)$$

it is obvious that the point (56) can be expressed by any couple u_1, u_{K+1}. Thus, driving this system from the initial point (56) the optimal control is irrelevant to the value $u_1 \in \langle U_1, U_2 \rangle$!

The canonical representation of interior states of edges (55) is also nonunique, but increasing the distance from the point (56) the range of possible combinations (u_1, u_{K+1}) becomes narrower and, finally, the points corresponding to u_1=u_{K+1}=$U_{3\text{-}j}$ have a unique canonical representation.

The prolongation of edges parallel to $\mathbf{v}_1$ (Fig.4) brings some degree of freedom into the construction of the proportional zone. The rule (29) for constructing the new vertices of polygonal curves (28) can be modified by any of the possibilities:

a) putting U_i=U_j;
b) putting U_i=$U_{3\text{-}j}$;
c) putting U_i=0; (58)
d) determining U_i by means of $\mathbf{v}_{N\text{-}1}$;
e) determining U_i by means of $\mathbf{v}_{N+1}$.

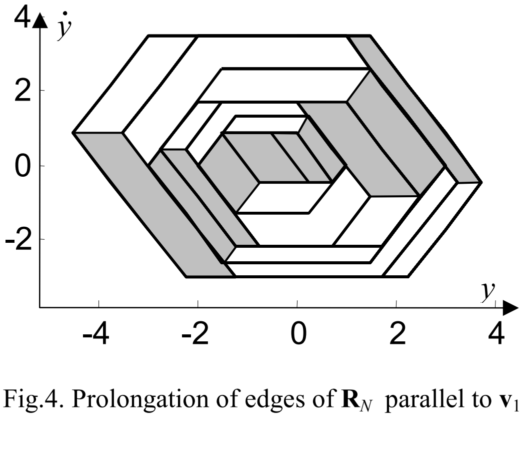


Fig.4. Prolongation of edges of $\mathbf{R}_N$ parallel to $\mathbf{v}_1$.

without changing the shape of the strips S, P, N and Q, since the questionable vectors are parallel to $\mathbf{v}_1$.

In Fig.3, $\mathrm{P_b}$ is shown for the case (58c) guaranteeing continuity of the control over the phase plane.

11. CONCLUSIONS

It was shown that choosing sampling period (40) the control algorithm can be expressed explicitly for arbitrary initial point, reasonably simplified and speeded up. Of course, one could ask, if it has a high practical importance, since, in fact, all used functions (like sin, etc.) are internally calculated iteratively. So, the only difference between an iterative and the explicit controller is that in the 2nd case the designer is using already available procedures. Once such controller is implemented, the differences are no more important.

Two approaches to the minimum time controller design have been shown: The 1st one based on the reachability sets is generally valid; the 2nd one is restricted just to the initial states $\mathbf{x}_0 \in \mathbf{I}_2$ consisting of two control intervals. At $\mathbf{x}_0 \in \mathbf{I}_n, n > 2$, for the sampling period (40) a nonuniqueness of the synthesis occurs. This can be considered as a consequence of the fact that in such points the trajectories corresponding to various control inputs differ only slightly. Such a property holds also in the case of a general sampling period, but it is not so easy to demonstrate. Analogically as in the continuous-data systems (Athans-Falb, 1966), this can be used for the controller simplification based on the 2nd approach; even here the convenience of such solution is yet more stressed by the effect of the time quantization.

The problem of linear PD controller with constrained output and a strip-like proportional band $\mathrm{P_b}$ is that it cannot sufficiently approximate the optimal $\mathrm{P_b}$ traced out by the polygonal curves. If it is optimally tuned for the relatively small initial conditions, overshoot, or even instability occurs for higher initial disturbances. If these are suppressed by modified controller setting, sluggish transient occurs in the vicinity of demanded state.

For implementation of the MTC algorithms it is important that in special cases corresponding to the most important plant poles configurations it is possible to determine actual segment of the zone of proportional control required for computing the control signal analytically. This was firstly published by Huba (1986) for the double integrator plant and later for all such linear plants in Huba (1992). The analytical control algorithms are possible for such 2nd order plants for which there exists inverse expression for the parameter N from coordinates of vertices of the polygonal curves, e.g. from (37).

It is, however, to remind that the minimum time control algorithms are mostly not practically usable in practice. With the aim to achieve softer transients considering the always present nonmodelled dynamics, the first modifications of algorithms inspired by predictive control approaches used the so called “premature sampling”, when the minimum time control algorithm was working with the prediction horizon used as the sampling period, whereas the real sampling period was shortened (Huba and Bistak, 1992; Huba et al., 1992). This, however, enabled to modify just one of the two closed loop poles.

Modification enabling to change both closed loop poles from the minimum time values $\lambda_{1,2} = 0$ to $\lambda_1 > 0, \lambda_2 > 0$ by keeping all basic features of the minimum time control was then presented for the double integrator by Huba (1994) and extended to other 2nd order plants e.g. in Huba (1998). The minimum time strategy – to reach in one control step the RBC from a general initial state and to reach in one step origin from point of the RBC was here modified by requirement to decrease in one step distance from the RBC by λ_1 (from a general initial state) and to decrease the distance from the origin by λ_2 (from points of RBC). Similarly as in the MTC, the resulting control may be interpreted as piecewise linear one with one parameter determining the actual segment for control computation (Fig.5).

Later was in Huba (2000) shown that the highest possible speed of transients still guaranteeing monotonic responses may be achieved by control considering more complicated structure of the phase plane with two piecewise linear partitions requiring determination of two integer parameters (Fig. 6). Later research was then oriented on possibilities of practical applications of derived control (see e.g. Ivanov et al., 2000) what was leading to necessity to deal with robust tuning of achieved controllers. Recent results achieved in this area show that it should be possible by the Performance Portrait method (Huba, 2010; Huba, 2011a). Although its use was up to now demonstrated just for continuous time robust control design (Huba, 2011b), it is supposed to be usable also for discrete time control.

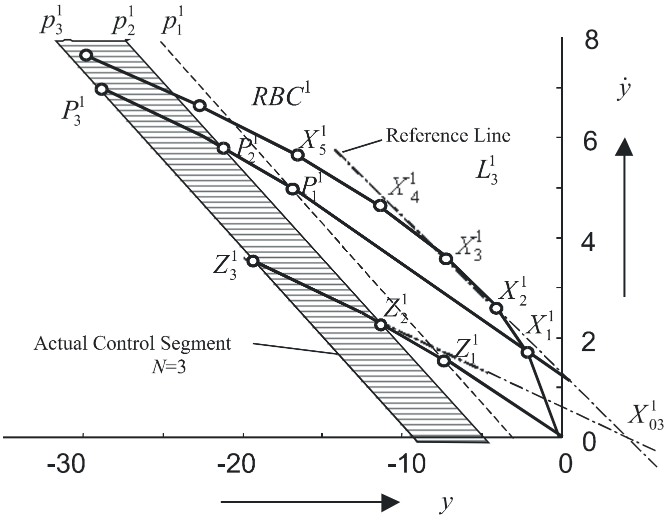


Fig. 5 Piecewise linear algorithm of the constrained pole assignment control decreasing distance from reference line corresponding to one segment of RBC.

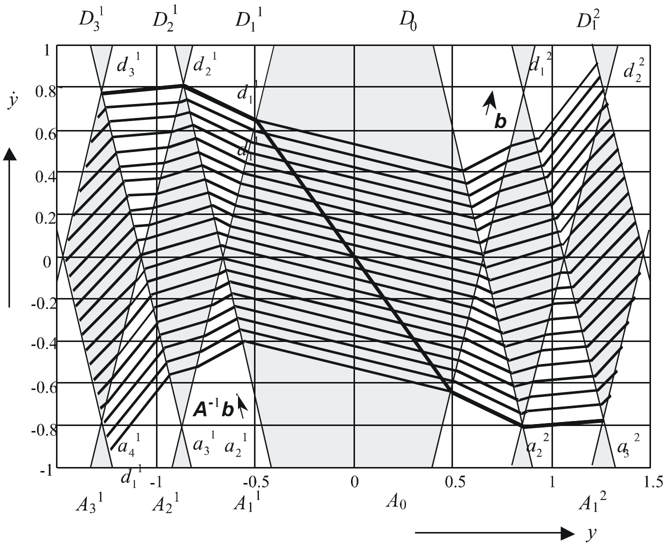


Fig. 6 Phase plane structure of the constrained pole assignment control decreasing distance from with two partitions and a piecewise linear control algorithm depending on two integer parameters.

ACKNOWLEDGEMENTS

This work was partially supported by the Project VEGA 1/0656/09: Integration and development of nonlinear and robust control methods and their application in controlling flying vehicles, by the project KEGA 3/7245/09 Building virtual and remote experiments for network of online laboratories. It was also supported by the grant NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic

REFERENCES

Aström, K. J., T.Hägglund (1995). *PID controllers : Theory, design, and tuning* - 2. ed., ISA, Research Triangle Park, NC

Athans, M. and P.L.Falb (1966). *Optimal control*. McGraw-Hill, N.York, pp.569-589.

Buša,J. und Huba,M. (1986). Entwurf von zeitoptimalen Abtastreglern für die Regelstrecke mit doppeltem integralen Verhalten. *Automatisierungstechnik* 34, H7, 287-288.

Desoer, C.A. and J.Wing (1961). The minimal time regulator problem for linear sampled-data systems. *J. Franklin Inst.*, **272**, 208-228.

Hippe, P. (2006). Windup in control. Springer, Berlin.

Huba, M. et al. (1987). Digital time-optimal control of nonlinear second-order system. In: *Preprints 10th IFAC World Congress,* Munich, **8**, 29-34.

Huba,M. (1992). Control Algorithms for 2nd Order Minimum Time Systems. *Electrical Engineering Journal* **43**, 233-240. (http://www.kar.elf.stuba.sk/~huba/EE92.pdf)

Huba,M., Bisták,P. (1992). Premature sampling - an analogy to the pole assigment for the saturating minimum time systems. 11 European meeting on cybernetics and system research, Vienna.

Huba,M., Bisták, P. and M.Molnár (1992). Control Algorithms for Saturating Systems. Preprints of IFAC Symposium on Low Cost Automation, LCA'92 Vienna, 229-234.

Huba, M. (1994). Saturating Pole Assignment Controller. Construction and Geometrical Interpretation in the Phase Plane. *Preprints 1st IFAC Workshop "New Trends in Design of Control Systems"* Smolenice, 121-126.

Huba,M., (1998). Minimum Time Pole Assignment PD Controller for Linear 2nd Order Systems with Constrained Input. IFAC *SSC'98* Nantes.

Huba, M., Skachová, Z., Bisták, P. (1998). Minimum Time Pole Assignment Controllers for I1 and I2 Plants. *J. Electrical Eng*., 49, No.11-12, 289-298.

Huba, M., et al. (1999). Invariant Sets Based Concept of the Pole Assignment Control. In: *ECC'99*, Düsseldorf: VDI/VDE

Huba, M., Bisták, P. (1999). Dynamic Classes in the PID Control. In: *Proc. 1999 American Control Conference*. San Diego: AACC.

Huba, M. (2000). Proportional Band Structure of 2nd Order Systems under the Minimum Time Pole Assignment Control. *45th International Scientific Colloquium Technical University Ilmenau*, October 04 – 06, 2000, 385-390

Ivanov,I., Masár, I., Huba, M. and H.Hoyer (2000). Minimum Time Pole Assignment Control of a Position System Servo-Drive. IFAC Conference Control Systems Design, Bratislava, June 2000, 593-598

Huba, M., Bisták, P. (2003). Minimum Time PD-Controller Design. In: 2nd IFAC Conference on Control Systems Design 2003, Bratislava, Slovak republic, 7-10 September 2003, 65-70.

Huba, M. (2010). Designing Robust Controller Tuning for Dead Time Systems. *IFAC Int. Conf. System Structure and Control*, Ancona, Italy.

Huba, M. (2011a). Basic Notions of Robust Constrained PID Control. In: *Selected topics on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohal'-Ilkiv Editors, STU Bratislava - NTNU Trondheim.

Huba, M. (2011b). Robust Controller Tuning for Constrained Double Integrator. In: *NIL Workshop on constrained and nonlinear control.* M. Huba, S. Skogestad, M. Fikar, M. Hovd, T.A. Johansen, B. Rohal'-Ilkiv Editors, STU Bratislava - NTNU Trondheim.

Kalman, R. E. (1957). Optimal nonlinear compensation of saturating systems by intermittent action, *IRE WESCON Convention Record*, pt. 4, 130-135

Kuo, B.C. (1970). Discrete-data control systems. *Prentice-Hall*, Englewood Clifs, N.Jersey.

Rönnbäck,S. (1996). Nonlinear Dynamic Windup Detection in Anti-Windup Compensators. Preprints CESA '96, Lille, 1014-101

Comments – Remarks

Dead Time Compensators for FOPDT Plants

M. Huba [*] **P. Ťapák** [**]

[*] *MI/PRT, FernUniversität in Hagen, Universitätsstr. 27,
D-58084 Hagen, Germany (e-mail: mikulas.huba@stuba.sk)*
[**] *Institute of Control and Industrial Informatics
Faculty of Electrical Engineering and IT, Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: peter.tapak@stuba.sk)*

Abstract: This paper introduces modifications of the Filtered Smith Predictor with primary P-controller that reasonably simplify treatment of the control constraints. Testing the traditional and the new solutions by real time experiments on a stable plant fully confirms superiority of the new controllers in constrained control. It also shows that despite to the formal input-to-output equivalence of the new and the traditional solutions, by considering limit situation of the "quasi continuous" implementation, due to different schemes they give responses to the setpoint and disturbance steps that are different in terms of the speed, monotonicity, as well as noise attenuation. As a by-product of this paper it is shown that the solutions with the 2nd order disturbance filter are much more sensitive than those with the 1st order one, what may be crucial for their practical use.

Keywords: proportional control, dead-time compensators, robustness, predictive control, Smith predictor

1. INTRODUCTION

The Smith Predictor (SP) Smith (1957) represents one of the oldest structures of the Dead-Time Compensators (DTCs). Due to the infinity dimension of dead time systems, design of DTCs is still surrounded by not sufficiently explained myths and in focus of current research Guzmán et al. (2008); Normey-Rico and Camacho (2008); Normey-Rico et al. (2009); Normey-Rico and Camacho (2009); Panda (2009); Zhang et al. (2008). Despite that just few of the known DTCs are interpreted as disturbance observer (DO) based structures, in fact, all of them may be shown to include observers for reconstruction of either input or output disturbances that are then compensated by modifying output or input of the primary controller. Due to this the primary controller does not need to include the integral action that is known to produce windup. The Filtered Smith Predictor (FSP) Normey-Rico et al. (1997); Normey-Rico and Camacho (2007, 2008); Normey-Rico et al. (2009); Normey-Rico and Camacho (2009) using the parallel plant model (PPM) is based on reconstruction of the output disturbance. Since the pioneering work by Smith Smith (1957) and continued later within the concept of the Internal Model Control (IMC) Lu et al. (2005) PPP played a key role in several DTC schemes. Here, it will be shown that the FSP may be further simplified by replacing the primary PI controller by a two-degree-of-freedom (2DOF) P-controller, what reasonably simplifies its use in constrained control without decreasing its capability in the disturbance rejection. The paper is structured as follows: Section 2 gives a brief overview of the FSP design. Modification of the FSP design for the First Order Plus Dead Time (FOPDT) plants with simplified primary loop enabling constrained control is described in Section 3. Comparison of presented solutions by real time experiments for the relatively short and long dead times is reported in Sections 5 and 6. Contributions of the paper and potential for further development are summarized in Conclusions.

2. THE FSP FOR FOPDT PLANTS

The FSP was originally proposed in Normey-Rico et al. (1997) for stable FOPDT processes to improve robustness of the traditional SP. Later, the disturbance filter $F_r(s)$ has been also proposed to decouple the reference setpoint and the disturbance response and to stabilize the controller loop in case of unstable and integral plants Normey-Rico and Camacho (2009). It may be interpreted as a structure with the dynamical feedforward control and the reference plant model Aström and Hägglund (2005); Visioli (2006), or the 2DOF IMC structure.

The unified approach to designing FSPs for the FOPDT plants introduced in Normey-Rico et al. (2009); Normey-Rico and Camacho (2009) considers compensation of an output disturbance by correction of the reference value, whereby the disturbance is reconstructed by using the PPM. However, despite to the proclaimed unification, it separately presents solutions corresponding to stable, integral and unstable plants. Thereby, for integral plant $P(s)$ with the fast and nominal dynamics $P_0(s)$ and $P_n(s)$

$$\begin{aligned} P(s) &= \frac{K_v e^{-\theta s}}{s} \\ P_0(s) &= \frac{K_{v0}}{s} \\ P_n(s) &= \frac{K_{v0} e^{-\theta_0 s}}{s} \end{aligned} \tag{1}$$

the primary controller is proposed as the P-controller

$$C_0(s) = K_c \tag{2}$$

whereby

$$K_c = 1/(K_{v0}T_r) \tag{3}$$

whereby T_r is the time constant of the fast closed loop described for $K_{v0} = K_v$ by the transfer function

$$C(s) = \frac{U(s)}{E(s)} = \frac{C_0}{1 + C_0 P_0} = \frac{K_c s}{s + K_c K_v} = \frac{1}{K_v}\frac{s}{1 + T_r s} \tag{4}$$

$$e(t) = r(t) - y(t) \tag{5}$$

$$E(s) = R(s) - Y(s) \tag{6}$$

are the control error and its Laplace transform, respectively. The nominal (reference) setpoint-to-output transfer function is

$$H_r(s) = \frac{Y(s)}{R(s)} = C(s)P(s) = \frac{e^{-\theta s}}{1 + T_r s} \tag{7}$$

When extending the disturbance compensation loop by a disturbance filter, for the primary controller loop the equivalent controller may be introduced as

$$C_e(s) = \frac{C}{1 - CPF_r} \tag{8}$$

The transfer functions corresponding to the output d_o and the input d_i disturbances and to the measurement noise n become

$$\begin{aligned} H_o(s) &= \frac{Y(s)}{D_o(s)} \\ &= \frac{1}{1 + C_e P F_r} = 1 - CPF_r \\ &= 1 - \frac{F_r(s) e^{-\theta s}}{1 + T_r s} \end{aligned} \tag{9}$$

Table 1. Acronyms

2DOF	Two Degree of Freedom
DO	Disturbance Observer
DTC	Dead-Time Compensator
FOPDT	First Order Plus Dead Time
PI-F1SP, PI-F2SP	Filtered Smith Predictors with PI-controller and the 1st, or the 2nd order disturbance filters
P-F1SP, P-F2SP	Filtered Smith Predictors with P-controller and the 1st, or the 2nd order disturbance filters
IAE	Integrated Absolute Error
IMC	Internal Model Control
P-controller	Proportional controller
PPM	Parallel Plant Model
SP	Smith Predictor
TV	Total Variance

$$\begin{aligned} H_i(s) &= \frac{Y(s)}{D_i(s)} \\ &= \frac{P}{1 + C_e P F_r} \\ &= P(s)\left(1 - F_r(s)\frac{e^{-\theta s}}{1 + T_r s}\right) \end{aligned} \tag{10}$$

$$H_n(s) = \frac{U(s)}{N(s)} = \frac{C_e F_r}{1 + C_e P F_r} = C(s)F_r(s) \tag{11}$$

For the 2nd order filter with T_f representing time constant of the disturbance responses

$$F_{r2} = \frac{(1 + T_r s)(1 + \beta_{12} s)}{(1 + T_f s)^2} \tag{12}$$

after substituting (12) into (9,10,11) for F_r one gets

$$H_o(s) = \left(1 - \frac{1 + \beta_{12} s}{(1 + T_f s)^2} e^{-\theta s}\right) \tag{13}$$

$$H_i(s) = P(s)\left(1 - \frac{1 + \beta_{12} s}{(1 + T_f s)^2} e^{-\theta s}\right) \tag{14}$$

$$H_i(s) = C(s)\frac{1 + \beta_{12} s}{(1 + T_f s)^2} \tag{15}$$

After determining β_{12} to fulfill

$$\begin{aligned} \left[(1 + T_f s)^2 - (1 + \beta_{12} s)e^{-\theta s}\right]_{s=0} &= 0 \\ \frac{d}{ds}\left[(1 + T_f s)^2 - (1 + \beta_{12} s)e^{-\theta s}\right]_{s=0} &= 0 \end{aligned} \tag{16}$$

the stepwise constant disturbances and the plant pole will be eliminated by a double zero of H_i at $s = 0$, when

$$\beta_{12} = 2T_f + \Theta \tag{17}$$

Similarly, for the stable FOPDT plant represented in the form

$$\begin{aligned} P(s) &= \frac{K e^{-\theta s}}{Ts + 1} \\ P_0(s) &= \frac{K_0}{T_0 s + 1} \\ P(s) &= \frac{K_0 e^{-\theta_0 s}}{T_0 s + 1} \end{aligned} \tag{18}$$

papers Normey-Rico et al. (2009); Normey-Rico and Camacho (2009) propose the PI primary controller

$$\begin{aligned} C_0(s) &= K_c \frac{1 + T_i s}{T_i s} \\ T_i &= T_0 \\ K_c &= \frac{T_0}{T_r K_0} \end{aligned} \tag{19}$$

Then, the primary (fast) closed loop transfer function becomes

$$C(s) = \frac{C_0}{1 + C_0 P_0} = \frac{K_c(1 + T_0 s)}{T_0 s + K_c K_0} = \frac{1}{K_0}\frac{1 + T_0 s}{1 + T_r s} \tag{20}$$

Similarly as (4), (20) gives inversion of the fast plant dynamics $P_0(s)$ filtered with the time constant T_r. From

$$\begin{aligned} H_i(0) &= 0 \\ H_i(-1/T) &= 0 \end{aligned} \tag{21}$$

for the 2nd order filter (12) one gets according to Normey-Rico et al. (2009); Normey-Rico and Camacho (2009)

$$\beta_{12} = T\left(1 - (1 - T_f/T)^2 e^{-\Theta/T}\right) \tag{22}$$

$H_o(s)$,$H_i(s)$ and $H_n(s)$ are then given by (13),(14) and (15). Besides of this controller denoted in the following as the PI-F2SP, it is also possible to work with simpler disturbance filters fulfilling $F_r(0) = 1$ and (21), as e.g.

$$F_{r1}(s) = \frac{1 + \beta_{11}s}{1 + T_f s} \tag{23}$$

This defines the PI-F1SP controller characterized with

$$\begin{aligned} \beta_{11} &= T\left(1 - (1 - T_f/T)(1 - T_r/T)e^{-\theta/T}\right) \\ H_i(s) &= P(s)\left(1 - \frac{1 + \beta_{11}s}{(1 + T_f s)(1 + T_r s)}e^{-\theta s}\right) \\ H_n(s) &= C(s)\frac{1 + \beta_{11}s}{1 + T_f s} \end{aligned} \tag{24}$$

For unstable FOPDT plants

$$\begin{aligned} P(s) &= \frac{Ke^{-\theta s}}{Ts - 1} \\ P_0(s) &= \frac{K_0}{T_0 s - 1} \\ P(s) &= \frac{K_0 e^{-\theta_0 s}}{T_0 s - 1} \end{aligned} \tag{25}$$

papers Normey-Rico et al. (2009); Normey-Rico and Camacho (2009) propose again the primary PI controller (19), however, its tuning is now

$$\begin{aligned} T_i &= T_r(2 + T_r/T_0) \\ K_c &= (T_r + 2T_0)/(T_r K_0) \end{aligned} \tag{26}$$

This gives

$$C(s) = \frac{C_0}{1 + C_0 P_0} = \frac{1}{K_0}\frac{(1 + T_i s)(T_0 s - 1)}{(1 + T_r s)^2} \tag{27}$$

To decouple the responses and to eliminate the pole $s = 1/T$ the 3rd order filter is proposed

$$F_{r3}(s) = \frac{(1 + T_r s)^2(1 + \beta_{13}s)}{(1 + T_f s)^2(1 + T_i s)} \tag{28}$$

with time constant

$$\beta_{13} = T\left((1 + T_f/T)^2 e^{\theta/T} - 1\right) \tag{29}$$

3. MODIFIED FSP FOR FOPDT PLANTS

As it was shown e.g. by Zhang and Jiang (2008), SP with constrained output of the primary PI controller tend to produce windup what further complicates the controller design and represents one of the reasons for not using this structure much more frequently than today. In Visioli (2006) the actuator saturation problem in controlling FOPDT plant was solved by a variable structure controller. Simpler P-controllers instead of PI ones were recommended in a slightly modified setting already in Liu et al. (2005b); Lu et al. (2005). Here, by simplifying the primary controller it will be possible to decrease the controller complexity and in combination with different disturbance filters to modify the loop properties.

Next, we will firstly show that all three above situations may be formally unified by considering the 2DOF P-controller as the primary controller C_0. Besides of unification and simplification, this will enable to work with control schemes not generating windup that may reasonably simplify implementation and tuning of achieved controllers. Furthermore, it will also enable to understand phenomena related to the nonlinear process character.

3.1 Primary 2DOF P-controller

To describe the FOPDT plants in the form enabling a unified treatment of stable ($a > 0$), unstable ($a < 0$) and integral plants ($a = 0$), the pole-zero notation

$$\begin{aligned} P(s) &= \frac{K_s}{s + a}e^{-\theta s} \\ P_n(s) &= \frac{K_+ s_0}{s + a_0}e^{-\theta_0 s} \\ P_0(s) &= \frac{K_+ s_0}{s + a_0} \end{aligned} \tag{30}$$

will be used instead of (1), (18) and (25). The 2DOF controller will be expressed as the P controller with the gain K_P extended by the static feedforward control u_0

$$\begin{aligned} u &= K_P e + u_0 \\ u_0 &= a_0 r/K_{s0} \\ K_P &= (1/T_r - a_0)/K_{s0} \end{aligned} \tag{31}$$

Thereby, the fast model parameters a_0,K_{s0} correspond to the estimates of the plant parameters a and K_s . T_r represents the reference time constant of the fast primary loop. In the nominal case with $P_n = P$ and neglected control signal constraints, the structure in Fig. 1 yields transfer function between the setpoint and the not delayed output x

$$\begin{aligned} H_{r0}(s) &= \frac{X(s)}{R(s)} = \frac{1}{T_r s + 1} \\ T_r &= \frac{1}{K_P K_s + a} \end{aligned} \tag{32}$$

The dynamics between the setpoint and the control signal

$$\begin{aligned} C(s) &= \frac{U(s)}{R(s)} = \frac{1}{K_s}\frac{s + a}{T_r s + 1} \\ T_r &= \frac{1}{K_P K_s + a} \end{aligned} \tag{33}$$

represents filtered inversion of the fast plant dynamics. To achieve stability of the primary control loop, condition

$$K_P K_s + a > 0 \tag{34}$$

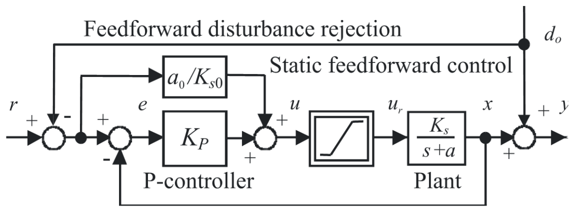


Fig. 1. The primary 2DOF P-controller with a static feedforward control; the measurable output disturbance d_0 may be compensated by a feedforward to the controller input;

must be fulfilled for all possible loop parameters, what guarantees monotonic decrease of the control error (5) to zero.

3.2 Respecting the control signal constraints

Control signal of real plants is always subject to control constraints expressed e.g. in form of the saturation function

$$u_r = sat(u) = \begin{cases} U_{max} & \text{if } u > U_{max} \\ u & \text{if } U_{min} \le u \le U_{max} \\ U_{min} & u < U_{min} \end{cases} \tag{35}$$

In controlling stable "fast" plant P_0 (30) with built in constraints (35) by the P-controller (31) satisfying to (34) it may be shown by choosing appropriate Ljapunov function, by the circle criterion, by the Popov criterion, or by the passivity approach Föllinger (1993); Glattfelder and Schaufelberger (2003); Hsu and Meyer (1968) that the loop remains stable without taking any additional measures for any transients with the final value $y_\infty = r$ satisfying

$$\begin{aligned} r &\in (Y_{min}, Y_{max}) \\ Y_{min} &= \frac{K_s(U_{min}+d_i)}{a} + d_o \\ Y_{max} &= \frac{K_s(U_{max}+d_i)}{a} + d_o \end{aligned} \tag{36}$$

In controlling unstable plants $P_0(s)$, besides of (34), condition (36) must be fulfilled already by the initial plant output $y_0 \in (Y_{min}, Y_{max})$) and by all its subsequent values during the transient responses what increases the motivation to achieve monotonic output changes that are connected with the minimal requirements on the admissible output values. In order to guarantee proper function of the D/A converters used usually in the digital controller implementation, already in such simple above mentioned situation the controller output has to be limited to the extent not exceeding the admissible range of the converter. In the much more complicated DTCs for the plant $P(s)$, one output of the primary loop (33) yields inversion of the fast dynamics. In order to respect constraints imposed on the real plant input, this output can not be simply generated by the transfer function, but the loop must be implemented by including at least so strong constraints as those at the plant input. Consequences of not respecting constraints in the feedforward control design are to find e.g. in Visioli (2006). Furthermore, in order to guarantee relevance of information used in the disturbance reconstruction, the DO used in the disturbance reconstruction must be supplied with the constrained control signal and the corresponding predicted output $\hat{x}_p$ ($\hat{x}_p$ plays in the structure in Fig. 2 role of the output predicted with respect to the delayed output x_d), what again requires work with constrained primary loop instead of simply using the corresponding transfer functions. However, for the sake of simplicity, saturation indicated in Fig. 1 was omitted from the scheme in Fig. 2. In order not to loose information about the signals corresponding to the constrained primary loop, it is also not possible to implement directly the equivalent controller (8).

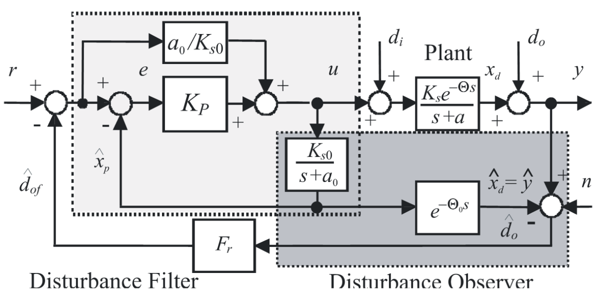


Fig. 2. Modified P-FSP with the primary loop using 2DOF P-controller with the disturbance filters (12), or (23)

3.3 Modified P-F1SP and P-F2SP Controllers

Since the primary controllers (4) and (20) are under notation

$$\begin{aligned} K_v = K_s \quad & \text{if } a = 0 \\ K = K_s/a; T = 1/a \quad & \text{if } a > 0 \end{aligned} \tag{37}$$

equivalent to (33), equivalent will also be all transfer functions (9,10,11), (13,14,15) and (24). This means that for the nominal tuning ($P = P_n$) with a chosen setpoint time constant T_r and without constraining the control signal, all loops with the controllers (4), (20) and (33) are fully equivalent and yield for the setpoint step r the IAE value

$$IAE/r = \theta + T_r \tag{38}$$

By using requirements (21) and denotation (37), for the P-F2SP controller with the disturbance filter (12) one gets

$$\beta_{12} = \frac{1}{a}\left(1 - (1 - aT_f)^2 e^{-\theta a}\right) \tag{39}$$

and for the integral plant with $a = 0$

$$\beta_{12} = \lim_{a \to 0} \frac{1}{a}\left(1 - (1 - aT_f)^2 e^{-\theta a}\right) = 2T_f + \theta \tag{40}$$

For the 1st order filter (23), one gets the P-F1SP with

$$\beta_{11} = \frac{1}{a}\left(1 - (1 - aT_f)(1 - aT_r)e^{-\theta a}\right); a \neq 0 \tag{41}$$

$$\begin{aligned} \beta_{11} &= \lim_{a \to 0} \frac{1}{a}\left(1 - (1 - aT_f)(1 - aT_r)e^{-\theta a}\right) \\ &= T_f + T_r + \theta \end{aligned} \tag{42}$$

For (37), the achieved formulas (41,42) are equal to (24). In both F1SP alternatives, the disturbance response is

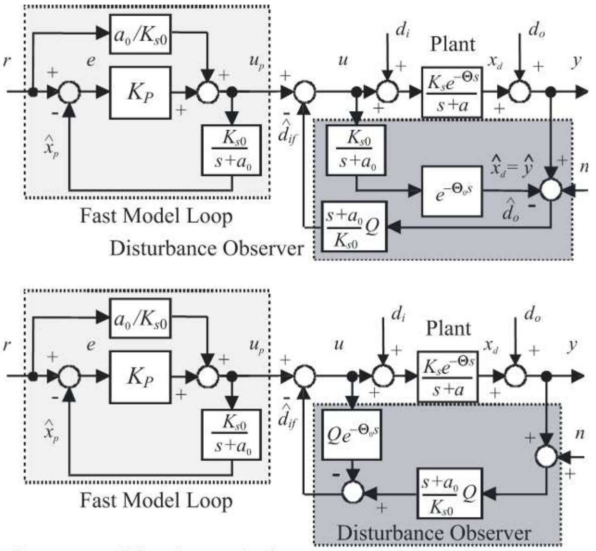


Fig. 3. Modification of the DO based on the PPM for compensation of the input disturbance (above) and the loop with DO-DTC based on the IPM

formally not fully decoupled from the setpoint response, i.e. T_r influences both the setpoint as well as the disturbance response, but still it is possible to tune both these responses separately.

4. MODIFICATIONS OF THE SMITH PREDICTOR

In order to compensate by analogue means shortage of the above solutions in controlling unstable and integral plants, it was e.g. proposed
1, to use a modified plant model that gives an input controllable state error instead of the not controllable one (Normey-Rico et al. (1997); Normey-Rico and Camacho (1999, 2007); Watanabe and Ito (1981))or to use a stable plant model instead of the nominal unstable plant (DePaor (1985)).
2, to introduce local stabilization of the unstable plant model with a corrective signal influencing also the real plant (Liu et al. (2005b,a); Lu et al. (2005); Majhi and Atherton (1998); Tan et al. (2003)).
3, to make the input of the unstable plant model independent from the input of the unstable plant and to eliminate work with equivalent unconstrained output disturbance (Normey-Rico and Camacho (2007, 2008); Normey-Rico et al. (2009); Normey-Rico and Camacho (2009); Zhong and Normey-Rico (2002); Zhong (2003)),
4, to introduce loop stabilization (Lu et al. (2005); Majhi and Atherton (1998); Matausek and Micic (1996, 1999)).

5. COMPARING PI-FSPS WITH P-FSPS: THE LAG DOMINANT OPTICAL PLANT

The thermo-optical laboratory plant (Fig. 4) offers control of 8 measured process variables by 3 manipulated (voltage) variables (0 - 5 V) influencing the bulb (main heat & light source), the light-diode (disturbance light source) and the fan (system cooling). Within Matlab/Simulink or Scilab/Scicos schemes the plant is represented as a single block and so limiting needs on costly and complicated software packages for real time control. The (supported) external converter cards are necessary just for sampling periods below 60ms. The optical channel used in following experiments consists of the light intensity measured by a photodiode and filtered by an analogue low pass filter with the time constant about 20s.

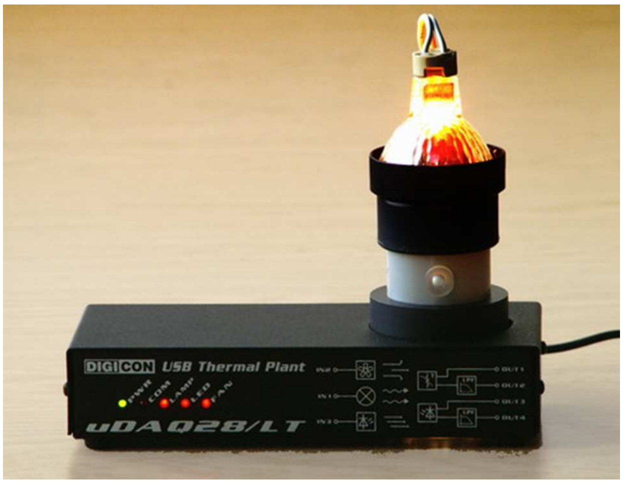


Fig. 4. Thermo-optical plant

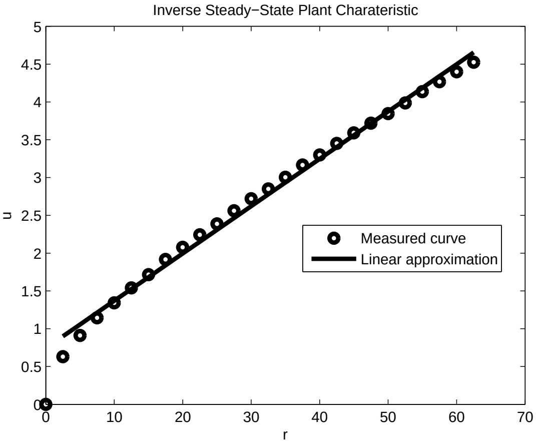


Fig. 5. Optical channel: the steady state controller output versus the reference signal and the linear feedforward control

The input-output characteristic of the plant is reasonably nonlinear, the steady state gains change significantly over the range of control. The dependence of the output y on the control signal u and on the disturbance signal d_i was approximated by measuring response to input steps of different amplitudes and around different working points as

$$\begin{aligned} Y(s) &= \frac{Ke^{-\theta s}}{Ts+1}\left(U(s)+K_i D_i(s)\right) \\ &= \frac{K_s e^{-\theta s}}{s+a}\left(U(s)+K_i D_i(s)\right) \qquad (43) \end{aligned}$$

As a result, the plant was represented by uncertain model with

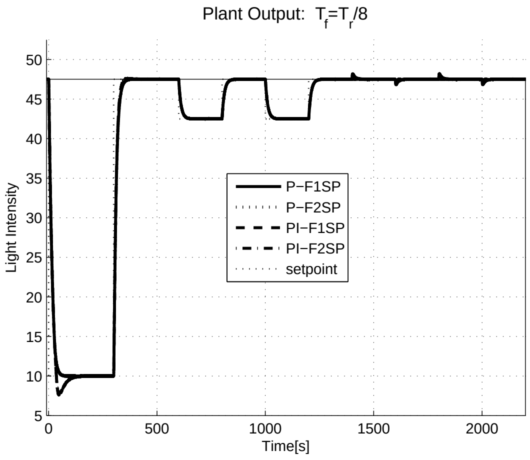


Fig. 6. Testing sequence: large (0-600s) and small (600-1400s) setpoint steps and disturbance steps $\Delta d = \pm 3V$ (1400-2200s) produced by LED equivalent to $\Delta d_i = \pm 0.78V$. Plant output.

$$\begin{aligned} K_s &\in [0.2282, 0.7060] \\ K_i &\in [4.31, 4.58] \\ a &\in [0.0387, 0.0588] \\ \theta &\in [0, 0.5] \\ T &\in [17.0096, 25.8687] \end{aligned} \tag{44}$$

Obviously, it is lag dominant. The achieved transient responses correspond to the operating point chosen as

$$\begin{aligned} K_{s0} &= 0.7060 \\ \theta_0 &= 0.34 \\ a_0 &= 0.0477 \\ T_0 &= 1/a_0 = 20.9644 \\ T_r &= T/2 = 10.4822 \end{aligned} \tag{45}$$

and to the quasi-continuous control with the sampling period $T_{samp} = 0.1sec$. The testing sequences (as shown in Fig. 6) were evaluated for the disturbance filter time constants

$$\begin{aligned} T_f &= T_r/c \\ c &= 1, 2, 8, 16 \\ T_f &\in [0.6551, 10.4822] \end{aligned} \tag{46}$$

In evaluating transients achieved with particular controllers, speed and duration of the plant output responses was characterized by the IAE performance index defined as

$$IAE = \int_0^\infty |e(t)|\, \mathrm{d}t = r(t) - y(t) \tag{47}$$

Besides of this, monotonicity of the output responses has been evaluated. The control effort was characterized by the TV criterion (Total Variance Skogestad and I.Postlethwaite (1996)) defined as

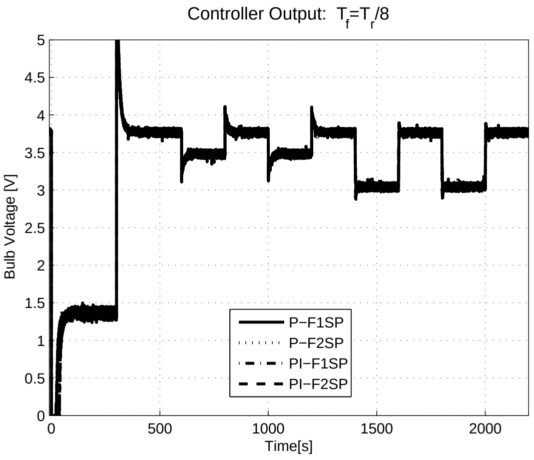


Fig. 7. Testing sequence: large (0-600s) and small (600-1400s) setpoint steps and disturbance steps $\Delta d = \pm 3V$ (1400-2200s) produced by LED equivalent to $\Delta d_i = \pm 0.78V$. Controller output.

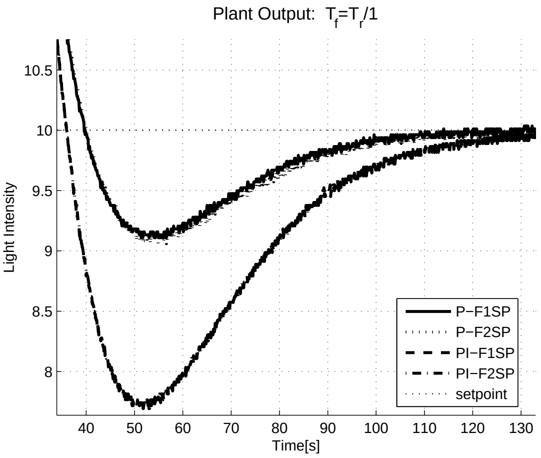


Fig. 8. Details of the large downward with typical windup of both PI-FSPs remaining also for shorter T_f

$$TV = \int_0^\infty |\frac{du}{dt}|\, \mathrm{d}t \approx \sum_i |u_{i+1} - u_i| \tag{48}$$

5.1 Setpoint steps

As it is obvious from Fig. 5, for the chosen operating point the static feedforward control $u_0 = a_0 r / K_{s0}$ used in (31) yields lower control values than those corresponding to the static controller characteristic. Due to this, for a week corrective feedback the setpoint responses will initially tend to lower output values than required. This imperfection of the linear static feedforward control may be interpreted as an internal input disturbance. This may be eliminated by the DO. For long disturbance filter time constant $T_f \approx T_r$ this imperfection will lead to a visible asymmetry of the setpoint steps: to a slightly slower output increase for the upwards reference steps and to undershooting of the downwards steps (Fig. 8).

When speeding up corrective disturbance rejection by $T_f \leq T_r/8$, both the P-F1SP and the P-F2SP controllers

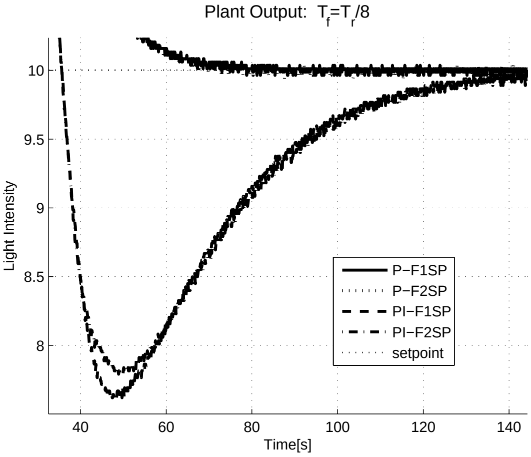


Fig. 9. Details of the large downward with typical windup of both PI-FSPs remaining also for shorter T_f

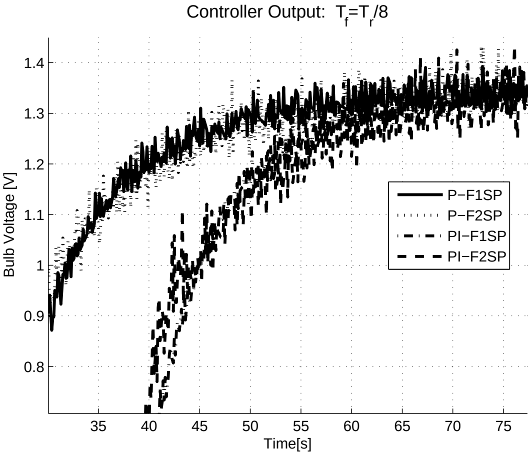


Fig. 10. Details of the large downward with typical windup of both PI-FSPs remaining also for shorter T_f

(immune against windup) yield already monotonic output responses that are clearly superior over the PI-FSP ones (influenced by the windup) - as illustrated by the plant input and output responses in Fig. 6 - 10, or by the IAE values in Fig. 11. Dependence of the achieved dynamics on the ratio $c = T_r/Tf$ shows in all cases that decreasing of the filter time constant T_f (i.e. increasing of c) influences the resulting IAE values just negligibly, i.e. it confirms the above expectation of independence of the setpoint dynamics from the choice of T_f. However, characteristics of the IAE, TV and IAE*TV values for the small setpoint steps (Fig. 12-14) that enable already a finer analysis of the dependence on $c = T_r/Tf$ (not superimposed by the windup and plant nonlinearity) show that in all cases decreasing of the filter time constant T_f leads to increased figures both in IAE as well as TV. Thereby, lower values correspond to the P- and PI-F1SP controllers with the 1st order disturbance filter (23) and both P-FSP controllers are at least equivalent to the PI-FSP ones. The existing differences may be explained as due to the

1, imperfections of the quasi-continuous implementation based on continuous transfer functions but depending (a) on the chosen sampling period taking values close to the shortest closed loop time constants and (b) on the level quantization of the measured signals,
2, imperfections (uncertainty, nonlinearity) of the used plant models, when the DO is active and influencing the loop behaviour also in situations with zero external disturbances, whereby the reconstructed disturbances correspond to the plant-model mismatch,
3, statistical character of the measurement noise.

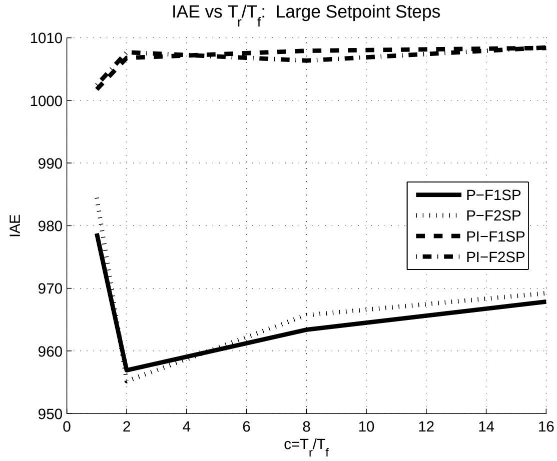


Fig. 11. IAE values corresponding to $T_r = T/2$ and T_f (46) for large setpoint steps

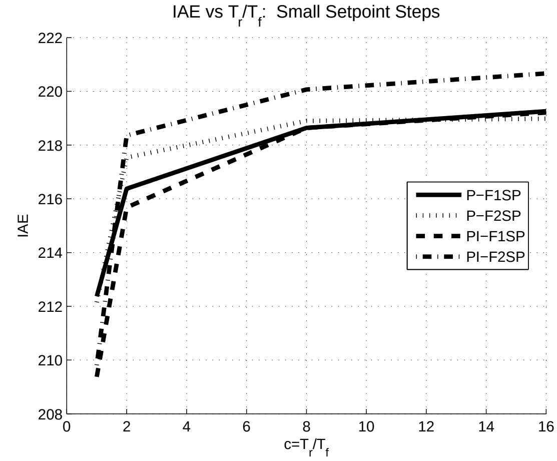


Fig. 12. IAE, TV and IAE*TV values corresponding to $T_r = T/2$ and T_f (46) for small setpoint steps

This set of experiments may be concluded by statement that both P-FSP controllers yield performance similar, or even superior to the PI-FSP ones whereby they do not exhibit the windup effect and are easier to interpret.

5.2 Disturbance steps

With respect to the formally equal disturbance responses of the PI- and P-F2SP corresponding to the filter (12), or PI- and P-F1SP corresponding to (23) one could expect confirmation of this fact by the real time experiments. Example of the measured disturbance responses is in Fig. 15,16. As it is also evident from the characteristics in Fig.

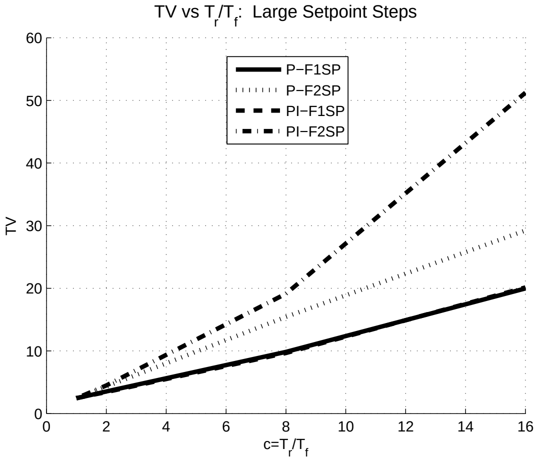


Fig. 13. IAE, TV and IAE*TV values corresponding to $T_r = T/2$ and T_f (46) for small setpoint steps

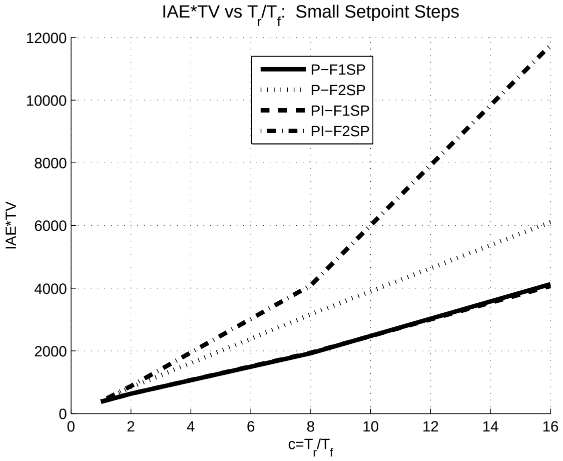


Fig. 14. IAE, TV and IAE*TV values corresponding to $T_r = T/2$ and T_f (46) for small setpoint steps

17-19 by decreasing the filter time constant T_f the IAE values decrease, but the required control effort expressed by the TV values increases. Optimum in the IAE*TV values indicates that it is possible to look for a compromise between the achievable speed of responses and the required control effort. The fastest responses correspond to the PI-F2SP, but on the cost of the largest TV value. This controller gives also the lowest IAE*TV value, however, just for a limited range of the filter time constants T_f. The P- and PI-F1SP controllers yield practically the same dynamics that is less depending on the filter time constant T_f than in the case of F_{2r}.

6. COMPARING PI- AND P-FSP: LONGER DEAD TIME VALUES

In order to be able to evaluate the closed loop performance corresponding to longer dead time values and to compare them with those achieved for the lag-dominated loop, the natural plant delay θ_0 (45) was increased in Simulink by an additional artificial delay

$$\theta_a = 20sec \tag{49}$$

Due to this the total dead time

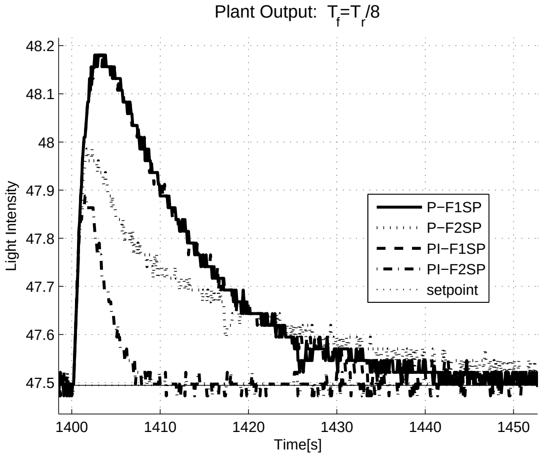


Fig. 15. Responses corresponding to the disturbance step

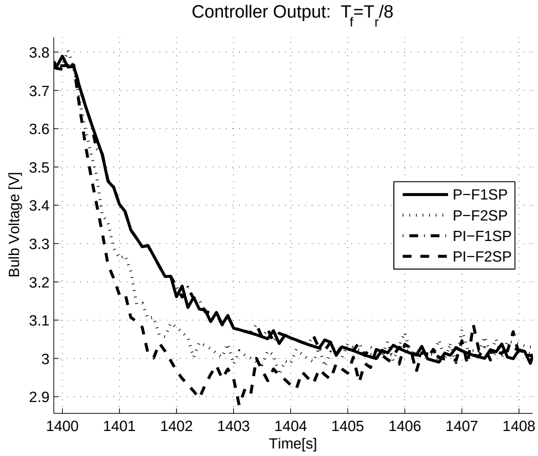


Fig. 16. Responses corresponding to the disturbance step

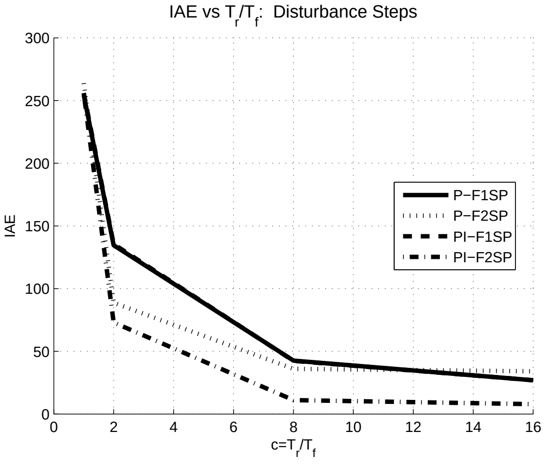


Fig. 17. IAE, TV and IAE*TV values corresponding to $T_r = T/2$ and T_f (46) for disturbance steps

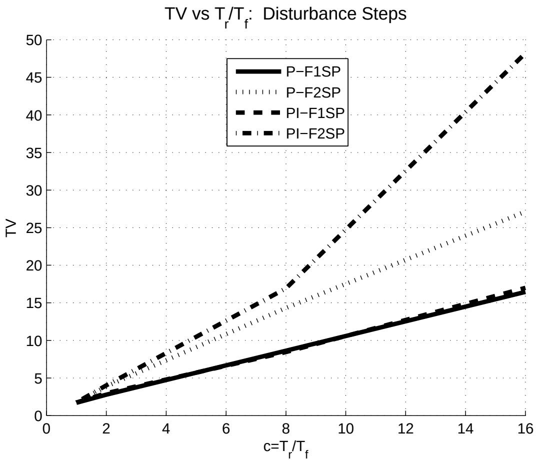


Fig. 18. IAE, TV and IAE*TV values corresponding to $T_r = T/2$ and T_f (46) for disturbance steps

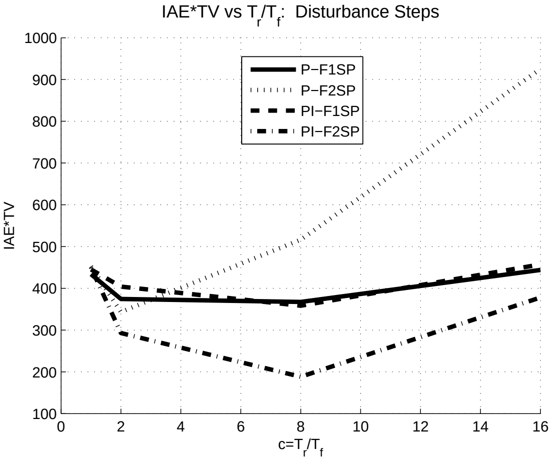


Fig. 19. IAE, TV and IAE*TV values corresponding to $T_r = T/2$ and T_f (46) for disturbance steps

$$\theta = \theta_0 + \theta_a \tag{50}$$

was increased from the relatively short plant value θ_0 up to the value close to the time constant T. Increased dead time values result into the reasonably increased closed loop sensitivity to the model uncertainty (leading to increased amount of higher harmonics) and decreased controller ability to compensate for the model uncertainty, what is visible especially in the large downward setpoint steps, when it results in undershooting of all tested controllers (Fig. 20,21). The increased sensitivity prevents the P- and PI-F2SP controllers with the 2nd order filters to work with the time constants $T_f < T_r/2$, since already for $c = 8$ the resulting transients are practically not acceptable (Fig. 24) due to permanent oscillation at the controller output and the not converging plant output. This is accompanied by the reasonably increased IAE values (see characteristics in Fig. 25,26) in comparing with those in Fig. 12, 13,14 and Fig. 17, 18,19. This set of experiments could again be concluded by the statement that the P-FSP controllers yield performance that is at least comparable, or even superior to that of the PI-FSP controllers. For longer dead time values, due to the increased sensitivity is the use of 2nd order filter (12) not recommended.

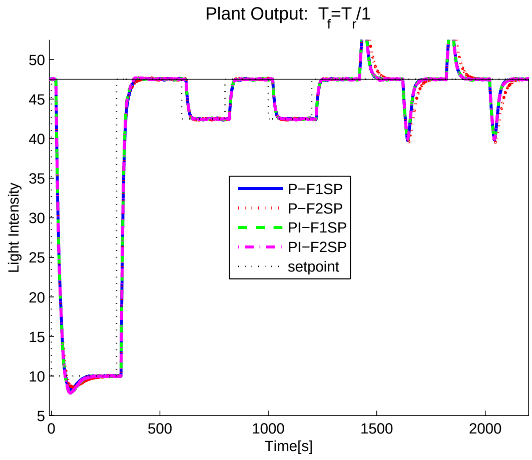


Fig. 20. Testing sequence corresponding to long dead time (50), $T_f = T_r$,$T_r = T/2$

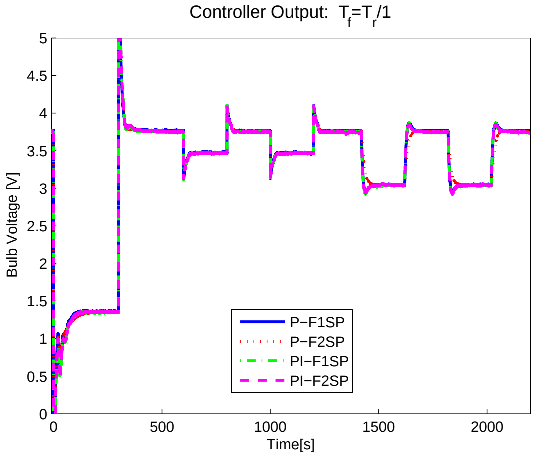


Fig. 21. Testing sequence corresponding to long dead time (50), $T_f = T_r$,$T_r = T/2$

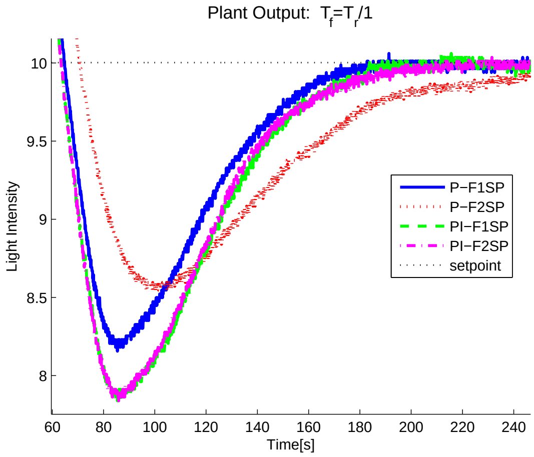


Fig. 22. Detail of downward step

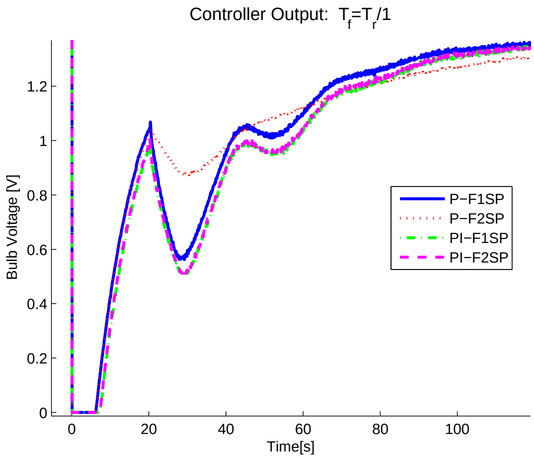


Fig. 23. Detail of downward step

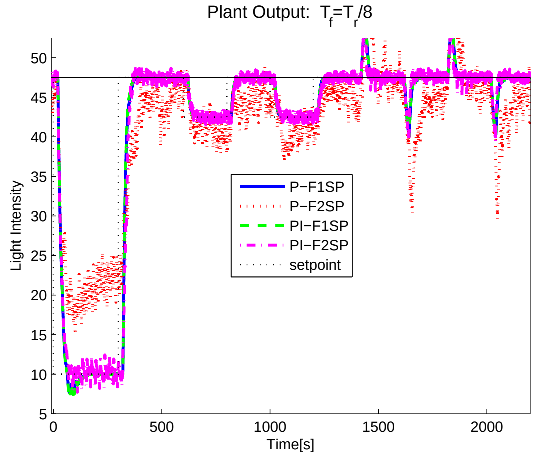


Fig. 24. Testing sequence corresponding to long dead time (50), $T_f = T_r/8$,$T_r = T/2$

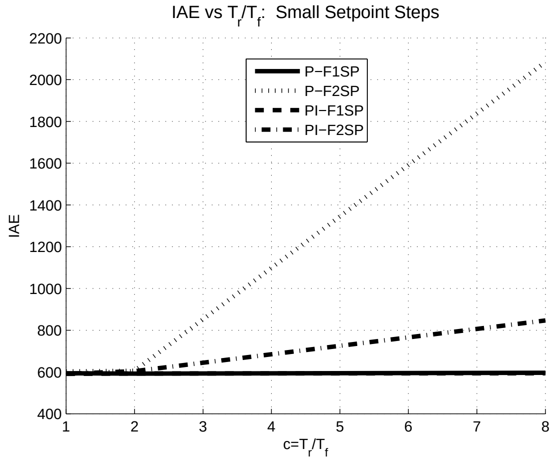


Fig. 25. IAE values corresponding to $Tr = T/2$ for small setpoitn steps

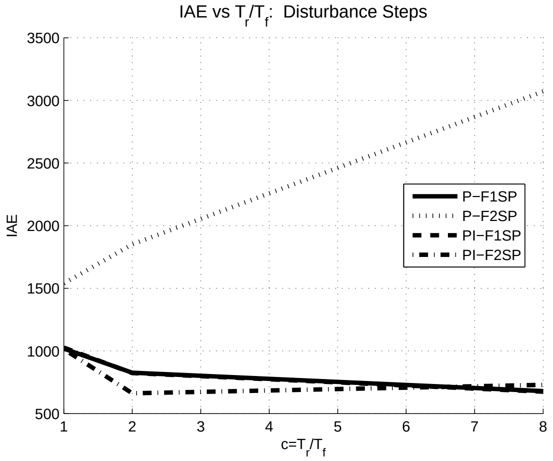


Fig. 26. IAE values corresponding to $Tr = T/2$ for disturbance steps

7. CONCLUSION

New formulations of the FSPs were proposed based on simplified primary loop with the 2DOF P-controller. Two solutions denoted as P-F1SP and P-F2SP corresponding to the 1st and the 2nd order filters were considered and compared with the PI based FSP (Normey-Rico and Camacho (2008); Normey-Rico et al. (2009); Normey-Rico and Camacho (2009)) with equivalent disturbance filters. The essential advantage of the new solutions is that, due to the memoryless controller in the primary loop and due to the IMC structure (Morari and Zafiriou (1989)) they do not generate the windup effect, in contrast to the PI-FSP ones that under constrained control tend to the windup and so they require appropriate anti-windup measures (Zhang and Jiang (2008)). Therefore, to be fair in comparing the traditional and the new solutions by real time experiments, with respect to the PI-FSP it was not possible to choose arbitrarily fast tuning parameters T_r (setpoint response time constant) and T_f (disturbance response time constant) and arbitrarily large steps, even when the dead time was relatively short and when the relatively low uncertainty enabled to use faster tuning leading to saturation during more significant setpoint steps. Due to different implementation schemes and the plant-model mismatch, the real time experiments corresponding to particular solutions are different. Thereby, the traditional solutions based on the primary PI controller give better performance just for a relatively narrow range of parameters for the disturbance response. So, the experiments confirm attractiveness of the new solutions that is not restricted just to the case of constrained control. Besides of this primary advantage they also show interesting properties in the setpoint response. Experience achieved in the DTCs tuning shows that there is no principal difference in dealing with the lag dominant and the dead time dominant systems. It is just to remember that the closed loop dead time is limiting speed of the correcting processes within the loop and so increasing demands on precision of the model of the plant. In the considered case of the optical plant, higher requirements on control quality lead in the case of longer dead time to necessity of considering a more precise nonlinear plant model. Use of the 2nd order disturbance filter (12) that enables formally fully independent tuning of the setpoint and disturbance responses shows to be much more sensitive to different loop imperfections and therefore it has be used very carefully.

ACKNOWLEDGEMENTS

The work was supported by a grant (No. NIL-I-007-dright) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Aström, K. and Hägglund, T. (2005). Advanced pid control. In *ISA-The Instrumentation, Systems, and Automation Society*. Research Triangle Park, NC.

DePaor, A. (1985). A modified smith predictor and controller for unstable processes with time delay. *Int. J. Control*, 41, 1025–1036.

Föllinger, O. (1993). *Nichtlineare Regelungen*. Oldenbourg Verlag, München.

Glattfelder, A. and Schaufelberger, W. (2003). *Control Systems with Input and Output Constraints*. Springer, London.

Guzmán, J., Garca, P., Hägglund, T., Dormido, S., Albertos, P., and Berenguel, M. (2008). Interactive tool for analysis of time - delay systems with dead - time compensators. *Control Engineering Practice*, 16, 824–835.

Hsu, J. and Meyer, A. (1968). *Modern Control Principles and Applications*. McGraw-Hill.

Liu, T., Cai, Y., Gu, D., and Zhang, W. (2005a). New modified smith predictor scheme for integrating and unstable processes with time delay. *IEEE Proceedings*, 238–246.

Liu, T., Zhang, W., and Gu, D. (2005b). Analytical design of two-degree-of-freedom control scheme for open-loop unstable processes with delay. *J. Process Control*, 15, 559–572.

Lu, X., Yang, Y., Wang, Q., and Zheng, W. (2005). A double two-degree-of-freedom control scheme for improved control of unstable delay processes. *J. Process Control*, 15, 605–614.

Majhi, S. and Atherton, D. (1998). A new smith predictor and controller for unstable and integrating processes with time delay. *Proceedings IEEE conference on control and decision, Tampa, USA*, 1341–1345.

Matausek, M. and Micic, A. (1996). A modified smith predictor for controlling a process with an integrator and long dead-time. *IEEE Trans. Autom. Control*, 41, 1199–1203.

Matausek, M. and Micic, A. (1999). : On the modified smith predictor for controlling a process with an integrator and long dead-time. *IEEE Trans. Autom. Control*, 44, 1603–1606.

Morari, M. and Zafiriou, E. (1989). *Robust Process Control*. Prentice Hall, Englewood Cliffs, N.Jersey.

Normey-Rico, J.E., Bordons, C., and Camacho, E.F. (1997). Improving the robustness of dead-time compensating pi controllers. *Control Engineering Practice*, 5, 801–810.

Normey-Rico, J.E. and Camacho, E.F. (1999). Robust tuning of deadtime compensators for process with an integrator and long dead-time. *IEEE Trans. Autom. Control*, 44,(8), 1597–1603.

Normey-Rico, J.E. and Camacho, E.F. (2007). *Control of dead-time processes*. Springer.

Normey-Rico, J.E. and Camacho, E.F. (2008). Dead-time compensators: A survey. *Control Engineering Practice*, 16, 407–428.

Normey-Rico, J.E. and Camacho, E.F. (2009). Unified approach for robust dead-time compensator design. *J. Process Control*, 19, 38–47.

Normey-Rico, J.E., Guzman, J., Dormido, S., Berenguel, M., and Camacho, E.F. (2009). An unified approach for dtc design using interactive tools. *Control Engineering Practice*, 17, 1234–1244.

Panda, R. (2009). Synthesis of pid controller for unstable and integrating processes. *Chemical Engineering Science*, 64, 2807–2816.

Skogestad, S. and I.Postlethwaite (1996). *Multivariable Feedback Control Analysis and Design*. John Wiley, New York.

Smith, O. (1957). Closer control of loops with dead time. *Chem.Eng.Prog*, 53, 217–219.

Tan, W., Marquez, H., and Chen, T. (2003). Imc-based design for unstable processes with time delays. *J. Process Control*, 13, 203–213.

Visioli, A. (2006). *Practical PID Control*. Springer, London.

Watanabe, K. and Ito, M. (1981). A process-model control for linear systems with delay. *IEEE Trans. Automat. Contr.*, AC-26,6, 1261–1266.

Zhang, M. and Jiang, C. (2008). Problem and its solution for actuator saturation of integrating process with dead time. *ISA Transactions*, 47,5, 80–84.

Zhang, W., Rieber, J., and Gu, D. (2008). Optimal dead-time compensator design for stable and integrating processes with time delay. *J. Process Control*, 18,5, 449–457.

Zhong, Q. (2003). Control of integral processes with dead-time. part 3: Dead-beat disturbance response. *IEEE Trans. Autom Control*, 48, 153–159.

Zhong, Q. and Normey-Rico, J. (2002). Control of integral processes with dead-time. part 1: Disturbance observer-based 2dof control scheme. *IEEE Proc.-Control Theory Appl.*, 149, 285–290.

Comments – Remarks

PWA System Identification with Strong Nonlinearities [★]

J. Stevek [*] **S. Kozak** [**]

[*] *Slovak University of Technology, Faculty of Informatics and Information Technologies, Bratislava, Slovakia (e-mail: stevek@fiit.stuba.sk)*
[**] *Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Bratislava, Slovakia (e-mail: stefan.kozak@stuba.sk).*

Abstract: This paper is dedicated to issue of approximation of nonlinear functions and nonlinear dynamical systems by Piecewise Affine (PWA) linear model. The article presents new identification Matlab toolbox for modelling and simulation of nonlinear systems. Functions of the toolbox together with GUI application simplified and accelerates identification of so called PWA OAF model. Identification of nonlinear systems is based on novel method of PWA modelling by generalized Fourier series. The approach provides identification of nonlinear functions of an arbitrary number of variables and identification of nonlinear dynamical systems in ARX model structure fashion from input-output data.

Keywords: PWA systems, Generalized Fourier series, Matlab toolbox, Chebyshev polynomial, PWA identification.

1. INTRODUCTION

In the recent research many methods were developed for modelling of hybrid systems and general nonlinear functions at all (Roll et al., 2004; Ferrari-Trecate, 2005; Julian et al., 1999). Many model structures were developed for hybrid systems and nonlinear systems. Much attention is dedicated to system modeling in MLD (Mixed Integer Dynamical) form (Bemporad and Morari, 1999) and PWA (Piecewise Affine). In (Bemporad et al., 2000), the formal equivalence between MLD systems and PWA systems is established and also effective algorithms were developed for transformation from one model structure to another (Villa et al., 2004; Bemporad, 2002). In (Heemels et al., 2001b,a), the equivalence between the following five classes of hybrid systems is, under certain conditions, established: MLD systems, Linear Complementarity (LC) systems, Extended Linear Complementarity (ELC) systems, PWA systems and Max-Min-Plus-Scaling (MMPS) systems. The important result of these equivalences is that derived theoretical properties and tools can easily be transferred from one class to another.

In this paper we present an effective tool for modeling of nonlinear systems by PWA using novel approach based on generalized Fourier series (Kozak and Stevek, 2010). This approach belongs to black-box identification methods of general nonlinear models (Sjöberg et al., 1995).

We use methodology of generalized Fourier series with orthogonal polynomials. In Leondes (1997), orthogonal polynomials were used as activation functions for special case of neural network with one hidden layer - Orthogonal Activation Function based Neural Network (OAF NN). For this type of neural network online and offline training algorithm has been defined with fast convergence properties. After simple modification of OAF NN it is possible to use this technique for PWA approximation of a common nonlinear system.

The paper is divided in six sections. First, we formulate the identification and linearization problem of nonlinear function. Next, we present modeling of nonlinear process by OAF NN, topology of the fourier series (PWA OAF NN) and network transformation to state space PWA form. In Section 3 PWA OAF identification toolbox is presented on three case studies. In Section 3.3 is identified nonlinear dynamical system from input-output data and designed explicit mpc control law.

2. PROBLEM FORMULATION

PWA linear approximation of hybrid systems depends on defining guardlines of the PWA mappping. If guardlines are known, the problem of identifying PWA systems can easily be solved using standard techniques for linear systems (Roll et al., 2004). The method based on finding mapping guardlines is suitable for linear system with nonlinear discrete parts like switches which changes system behavior in step. Other methods a priori assume that the system dynamics is continuous (Ferrari-Trecate, 2005). Both mentioned approaches use for identification clustering-based algorithms.

As will be pointed out, nonlinear identification techniques can be used under specific conditions in order to obtain linear PWA model. Many neural network based identification techniques use nonlinear neuron functions of one variable which are easier linearizable than whole model of many variables. The key idea is based on linearization of nonlinear neural network functions of single variable. Similarly as

[★] This paper was supported by Vega project No. 1/1105/11.

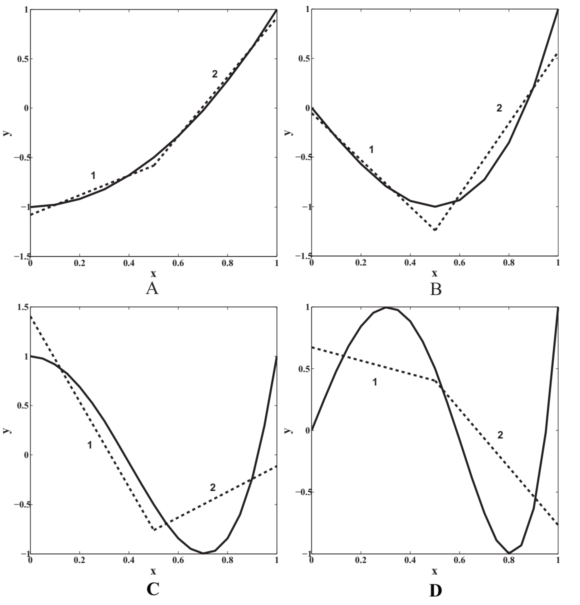


Fig. 1. PWA approximation of T2, T3, T4, T5 Chebyshev polynomials

Taylor series, it is possible to define any nonlinear function as a series of nonlinear functions. This approach leads to generalized Fourier polynomial series. Generalized Fourier series is based on a set of one-dimensional orthonormal functions $\phi_i^{(N)}$ defined as

$$\int_{x_1}^{x_2} \phi_i^{(N)}(x)\phi_j^{(N)}(x) = \delta_{ij} \tag{1}$$

where δ_{ij} is the Kronecker delta function and $[x_1, x_2]$ is the domain of interest. Several examples of orthonormal functions are the normalized Fourier (harmonic) functions, Legendre polynomials, Chebyshev polynomials and Laguerre polynomials (Leondes, 1997). In this paper only Chebyshev polynomials will be discussed.

Orthogonal Activation Function based Neural Network (OAF NN) is employed in the task of nonlinear approximation. PWA approximation of every used orthonormal polynomial creates Piecewise Affine Orthogonal Activation Function based Neural Network (PWA OAF NN).

2.1 Chebyshev polynomial

The Chebyshev polynomials of the first kind can be defined by the trigonometric identity

$$T_n(x) = \cos(n \arccos(x)) \tag{2}$$

with norm defined as follows

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}}(T_n(x))^2 dx = \begin{cases} \pi & n = 0 \\ \pi/2 & n \neq 1 \end{cases} \tag{3}$$

Recursive generating formula for Chebyshev polynomials:

$$T_0(x) = 1, \tag{4}$$
$$T_1(x) = x, \tag{5}$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \tag{6}$$
$$T_n(x) = U_{n+1}(x) - U_{n-1}(x). \tag{7}$$

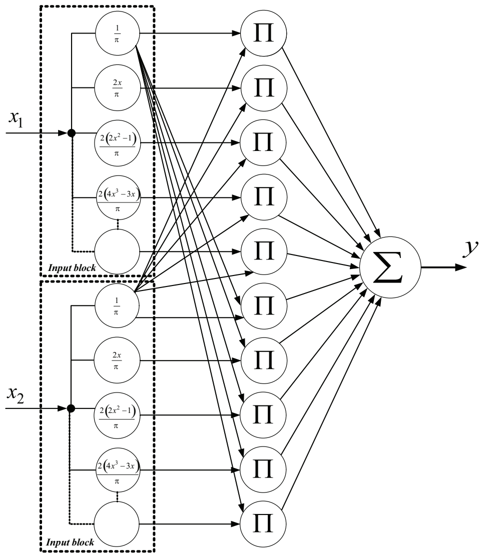


Fig. 2. Adjusted OAF NN structure

where U_n is the Chebyshev polynomial of the second kind generated by the recursive formula:

$$U_0(x) = 1, \tag{8}$$
$$U_1(x) = 2x, \tag{9}$$
$$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x), \tag{10}$$

The first few Chebyshev polynomials of the first kind are

$$T_0(x) = 1, \tag{11}$$
$$T_1(x) = x, \tag{12}$$
$$T_2(x) = 2x^2 - 1 \tag{13}$$
$$T_3(x) = 4x^3 - 3x \tag{14}$$
$$T_4(x) = 8x^4 - 8x^2 + 1. \tag{15}$$

The first few Chebyshev polynomials of the second kind are

$$U_0(x) = 1, \tag{16}$$
$$U_1(x) = 2x, \tag{17}$$
$$U_2(x) = 4x^2 - 1 \tag{18}$$
$$U_3(x) = 8x^3 - 4x \tag{19}$$
$$U_4(x) = 16x^4 - 12x^2 + 1. \tag{20}$$

It is possible to define Generalized Fourier series with orthogonal polynomilas by neural network with one hidden layer. In this work we use a Matlab function framework for orthogonal activation function based neural networks which is part of the toolbox. Aftr slight revision it is possible to use this methodology for modeling the fourier series. Example of the network for modeling function of two variables is depicted in Fig. 2.

If we consider general structure of the network in ARX fashion with na, nb, and nk parameters we get network output equation:

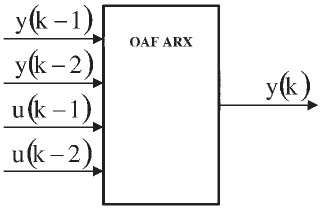


Fig. 3. OAF ARX model for na=2, nb=2, nk=0 or nk=1

$$
\begin{aligned}
y &= w_1 \frac{1}{\pi}^{p} + \dots \\
&\frac{2}{\pi^p}(w_2 T_1(y(k|1)) + \dots + w_n T_{n-1}(y(k|1)))+ \\
&\vdots \\
&\frac{2}{\pi^p}(w_{i_1} T_1(y(k|na)) + \dots + w_{i_2} T_{n-1}(y(k|na)))+ \\
&\frac{2}{\pi^p}(w_{i_3} T_1(u(k|nk)) + \dots + w_{i_4} T_{n-1}(u(k|nk)))+ \\
&\vdots \\
&\frac{2}{\pi^p}(w_{i_5} T_1(u(k|i_7)) + \dots + w_{i_6} T_{n-1}(u(k|i_7))).
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
i_1 &= (na-1)(n-1)+2 \\
i_2 &= na(n-1)+1 \\
i_3 &= nb(n-1)+2 \\
i_4 &= (nb+1)(n-1)+1 \\
i_5 &= (nb+na-1)(n-1)+2; \\
i_6 &= (nb+na)(n-1)+1; \\
i_7 &= nk+nb-1 \\
p &= na+nb
\end{aligned}
$$

where $y(k|na)$ denotes $y(k-na)$ and similarly $u(k|nk) \equiv u(k-nk)$. Every Chebyshev polynomial is aproximated by set of lines (Fig. 1)

$$T(x) \approx a_i x + b_i \quad \text{for} \quad i = \{1, 2, \dots, n_{div}\} \tag{22}$$

Then output equation becomes difference equation.
A convenient feature of all Chebyshev polynomial is their symmetry. All polynomials of even order are symmetrical by vertical axis and all polynomial of odd order are symmetrical by origin. These properties allow decreasing number of linearization points to half while keeping precision. To get the lowest number of shift cases of generated PWA model we linearized the polynomials in the same points.The term 'linearization point' denotes the interval division point where the PWA function breaks.

2.2 Transformation to state space PWA form

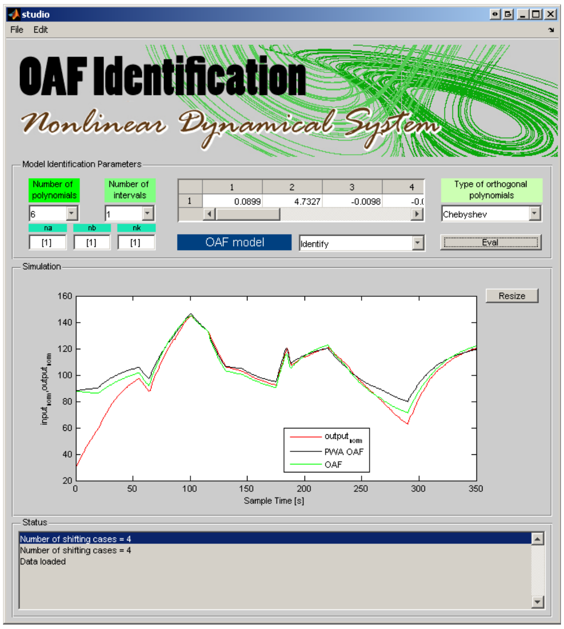
Accuracy of the approximation of nonlinear system is significantly increased when the function is linearized around multiple distinct linearization points. State space PWA structure describes behavior of nonlinear dynamical systems in multiple linearization points.

$$
\begin{aligned}
x(k+1) &= A_i x(k) + B_i u(k) + f_i \\
y(k) &= C_i x(k) + D_i u(k) + g_i
\end{aligned} \tag{23a}
$$

$$IF \begin{bmatrix} x \\ u \end{bmatrix} \in \mathrm{D}_i, \quad i = 1, \dots, n_L \tag{23b}$$

Every dynamic i is active in polyhedral partition (23b) which can be expressed by inequality

$$guardX_i x(k) + guardU_i u(k) \le guardC_i \tag{24}$$


studio

File Edit

OAF Identification

Nonlinear Dynamical System

Model Identification Parameters

Number of polynomials: 6

Number of intervals: 1

na: [1] nb: [1] nk: [1]

	1	2	3	4
1	0.0899	4.7327	-0.0098	-0.(

OAF model: Identify

Type of orthogonal polynomials: Chebyshev

Eval

Simulation

Resize

Status

Number of shifting cases = 4
Number of shifting cases = 4
Data loaded

Fig. 4. PWA OAF ID studio

Difference equation (21) can be easily transformed to state space form. In Matlab difference equation can be expressed by discrete transfer function. It is possible to use transformation function `tf2ss`. But this policy doesn't lead to desired state space PWA form. Desired state space form has to keep all outputs of difference equation (21) in state vector. So we can correctly define guard-line inequality (24).
Here we present transformation example for system with parameters **na=2, nb=2, nk=0 or nk=1**, Fig.3 . Difference equation:

$$
\begin{aligned}
y(k) \;=\; & c^{(i)} + c^{(i)}_{y1} y(k-1) + c^{(i)}_{y2} y(k-2) + \\
& c^{(i)}_{u1} u(k-1) + c^{(i)}_{u-2} u(k-2)
\end{aligned} \tag{25}
$$

In PWA form guard-lines are defined for $x1 = u(k-2), x2 = y(k-2), x3 = y(k-1)$ and $u = u(k-1)$
PWA state space model:

$$x(k+1) = A_i x(k) + B_i u(k) + f_i \tag{26a}$$

$$y(k) = C_i x(k) + D_i(k) + g_i \tag{26b}$$

$$A_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ c^{(i)}_{u2} & c^{(i)}_{y2} & c^{(i)}_{y1} \end{bmatrix} \tag{26c}$$

$$B_i = \begin{bmatrix} 1 \\ 0 \\ c^{(i)}_{u1} \end{bmatrix} \tag{26d}$$

$$C_i = [\,0 \;\; 0 \;\; 1\,] \tag{26e}$$

$$D_i = 0 \tag{26f}$$

$$f_i = \begin{bmatrix} 0 \\ 0 \\ c^{(i)} \end{bmatrix} \tag{26g}$$

$$g_i = 0; \tag{26h}$$

$$x \in < 3 \times 1 > \tag{26i}$$

3. PWA OAF IDENTIFICATION TOOLBOX

PWA identification problem has garnered great interest in the research community. In Matlab enviroment several toolboxes were developed for identification hybrid and nonlinear systems (Roll et al., 2004; Ferrari-Trecate, 2005; Julian et al., 1999). The main aim of the PWA OAF Identification Toolbox (PWA OAF IT) is to provide efficient tools for analysis, identification and simulation of PWA OAF model. In following section we present toolbox functionality on several identification examples.
In PWA OAF IT the model is represented by the following fields of the model structure:

`model.na` - Number of past output terms
`model.nb` - Number of past input terms
`model.nk` - Delay from input to the output
`model.npoly` - Number of Chebys. polynomials
`model.ndiv` - Division of {0,1} interval
`model.Fi` - Connection matrix of network
`model.w` - Network parameters
`model.type` - Type of polynomials 'Cebys'
`model.const` - Constant in difference equation
`model.yconst` - Y-cons in difference equation
`model.uconst` - U-const in difference equation
`model.sysStruct` - PWA state space struct
`model.ynorm` - Normalized output data
`model.unorm` - Normalized input data
`model.u` - Input data
`model.y` - Output data
`model.ypar` - Normalization param. of output
`model.upar` - Normalization param. of input

So far PWA OAF ID supports only MISO systems. In order to obtain identified model, call

```
>>model = pwaoafid(y,u,modelstruct,param)
```

Input arguments are in standard notation well known from PWAID toolbox. For more information type

```
>>help oafpwaid
```

For using gui application Fig. 4, call

```
>> oafpwaid_studio
```

3.1 Identification of 2-D function

2-D function is defined by formula:

$$\begin{aligned} y \;=\; & a_1 e^{-((x-b_1)/c_1)^2} + a_2 e^{-((x-b_2)/c_2)^2} + \\ & a_3 e^{-((x-b_3)/c_3)^2} + a_4 e^{-((x-b_4)/c_4)^2} \\ & \\ & a_1 = 53.4,\ b_1 = 5.165,\ c_1 = 8.815, \\ & a_2 = 31.25,\ b_2 = 18.69,\ c_2 = 5.109, \\ & a_3 = 20.2,\ b_3 = 13.89,\ c_3 = 2.381, \\ & a_4 = 4.316,\ b_4 = 9.864,\ c_4 = 0.992, \end{aligned} \tag{27}$$

We have made sample data in interval $\{7, 22\}$ (Fig. 6). In our example we did approximation in one point, by two lines. Before parameter estimation it was necessary to normalize data into the interval $\{-1, 1\}$ where Chebyshev polynomials are orthogonal. We used the first four Chebyshev polynomials $T_0 \div T_3$. Mean square error for this approximation is mse = 5.1947. To choose a best position of linearization points is a state of art of many algorithms. Through fast network parameters computation it is possible to use even genetic approach to get better position of linearization point and number of chebyshev polynomials.

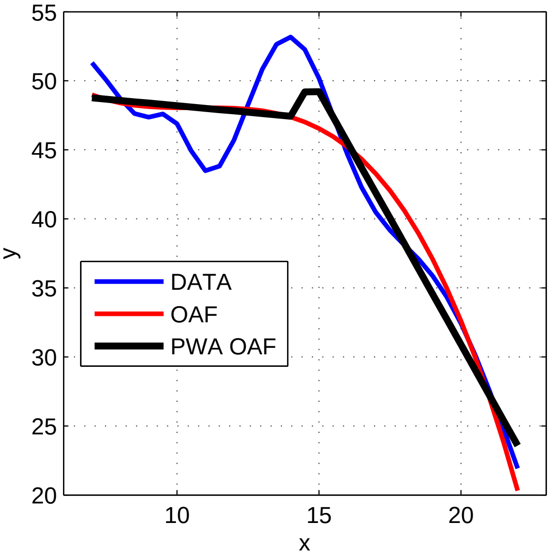


Fig. 5. 2-D function example

3.2 Identification of 3-D function

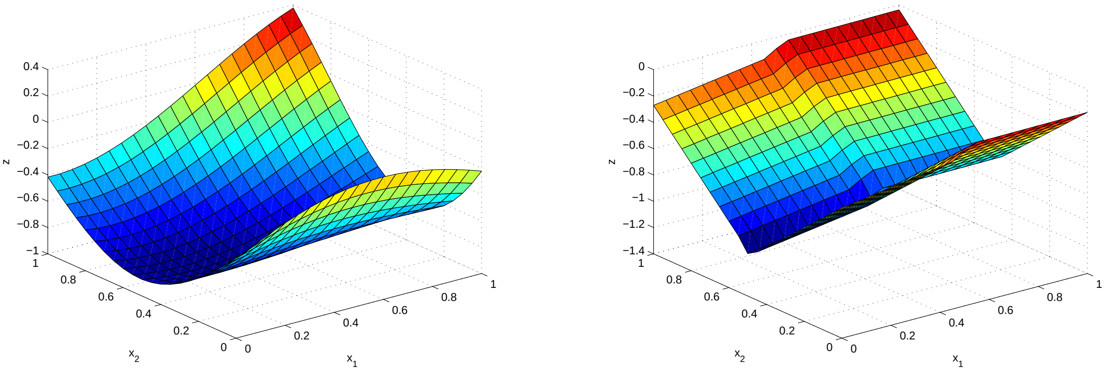
Consider a 3-D nonlinear function defined as

$$\begin{aligned} f(\bar{x}) \;=\; & -.2(\sin(x_1 + 4x_2)) - 2\cos(2x_1 + 3x_2) \\ & -3\sin(2x_1 - x_2) + 4\cos(x_1 - 2x_2) \\ & x_1 \in \{0, 1\}, \\ & x_2 \in \{0, 1\}, \end{aligned} \tag{28}$$

We used the first six Chebyshev polynomials, up to the fifth order $T_0 \div T_5$, linearized in 1 point, each polynomial by two lines. The total number of shifting cases for the resulting PWA function is n_u^{lp+1} where n_u is the number of neural network inputs and lp is the number of linearization points. For the 3-D function example (28) we get $2^2 = 4$ shifting cases. The result is plotted in Fig. 6b. For this approximation mse=0.0144.

3.3 Modeling and control of nonlinear dynamic system

In next example we will try to capture vehicle nonlinear dynamic from input output data for purpose of predictive control design of automatic cruise control. We used Simulink vehicle model with automatic transmission controller (Veh, 2006). Input for model is throttle and break torque signal. Output is vehicle velocity. From the character of input signals we can merge throttle and break torque signal to one input signal (Fig. 7a). Positive part of the input signal is proportional to accelerator pedal pressing and negative part of the input signal is proportional to breaking pedal pressing. Input-output data and identified system output are captured in Fig. 7. We used following identification parameters:



(a) 3-D function data

(b) PWA approximation of 3-D function

Fig. 6. 3D function

$$
\begin{aligned}
na &= 1 \\
nb &= 1 \\
nk &= 1 \\
npoly &= 4 \quad \text{polynomials:} T_0, T_1, T_2, T_3 \\
ndiv &= 1 \quad \text{approximation by two lines}
\end{aligned}
\tag{29}
$$

These parameters leads to state space model with one state variable and one input. Acquired PWA state space model has four dynamics (four shifting cases) and it is possible to design an automatic cruise control for such system.
For control design we used MPT toolbox (Kvasnica et al., 2004). We designed explicit mpc controller with time varying reference tracking property. We choosed quadratic cost control problem:

$$
\begin{aligned}
\min_{u(0),\dots,u(N-1)} &= x(N)^T P_N x(N) + \\
&\sum_{k=1}^{N-1} u(k)^T R u(k) + x(k)^T Q x(k)
\end{aligned}
\tag{30a}
$$

$$
s.t.: \begin{cases} x(k+1|t) = f_{dyn}(x(k), u(k)) \\ u_{min} \leq u(k) \leq u_{max} \\ \Delta u_{min} \leq u(k) - u(k-1) \leq \Delta u_{max} \\ y_{min} \leq g_{dyn}(x(k), u(k)) \leq y_{max} \\ x(N) \in T_{set} \end{cases}
\tag{30b}
$$

Parameters of control design:

```
      norm: 2
subopt_lev: 0
         N: 3
  tracking: 1
         Q: 100
         R: 1
        Qy: 700
```

Thanks to few PWA dynamics it is possible choose higher prediction horizon to refine control performance. Resulting control law is defined over 430 regions. It is possible to get satisfactory performance with control law defined over fewer number of regions. Designed control law was used in feedback control with nonlinear vehicle model, Fig. 8b.

4. CONCLUSION

PWA OAF toolbox significantly improves identification and modeling of nonlinear systems. Transformation to PWA state space model allows to use existing control design tools. So far PWA OAF ID supports only MISO systems. Three studied cases were presented. It was shown that the proposed approach was effective in model precision and universal in various input configuration. Computation of network parameters is fast and it allows to execute identification for various parameters (order of used Chebyshev polynomials, number of linearization points) to get better performance or even to use genetic approach. Accuracy of the PWA OAF NN approximation depends on the number of linearization points, the highest order of used Chebyshev polynomials and absolute value of computed parameters of the neural network. More linearization points give better precision of the approximation but complexity of the PWA model increases. It is necessary to find suitable proportion between the number of linearization points and required precision.

ACKNOWLEDGEMENTS

This paper was supported by Vega project No. 1/1105/11.

REFERENCES

(2006). Modeling an automatic transmission controller. URL `http://www.mathworks.com/`.

Bemporad, A. (2002). An efficient technique for translating mixed logical dynamical systems into piecewise affine systems. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 2, 1970 – 1975 vol.2.

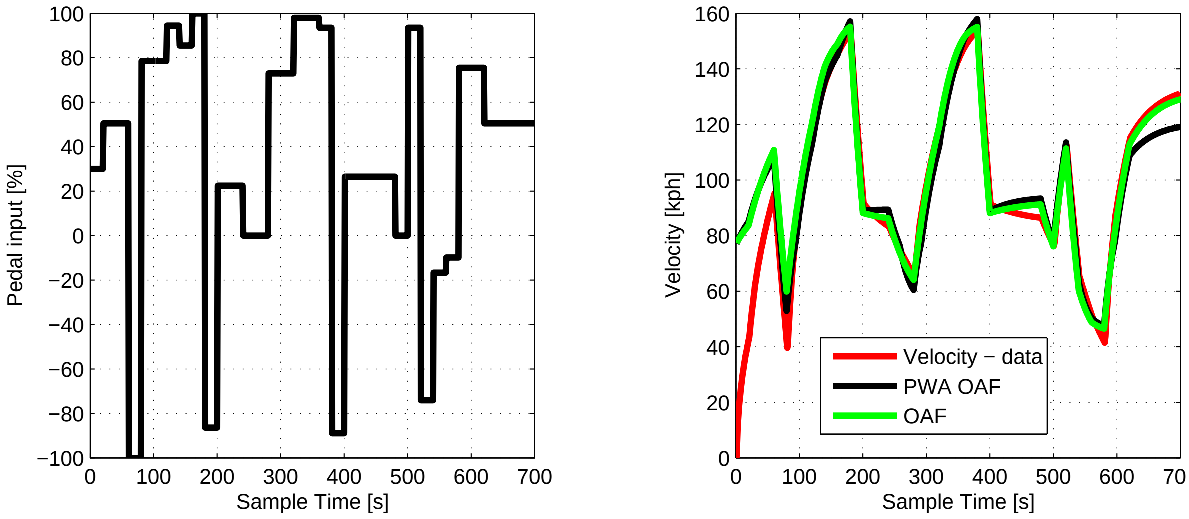
Bemporad, A., Ferrari-Trecate, G., and Morari, M. (2000). Observability and controllability of piecewise affine and hybrid systems. *Automatic Control, IEEE Transactions on*, 45(10), 1864 – 1876.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35, 407–427.

Ferrari-Trecate, G. (2005). Hybrid Identification Toolbox (HIT).

Heemels, W., De Schutter, B., and Bemporad, A. (2001a). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.

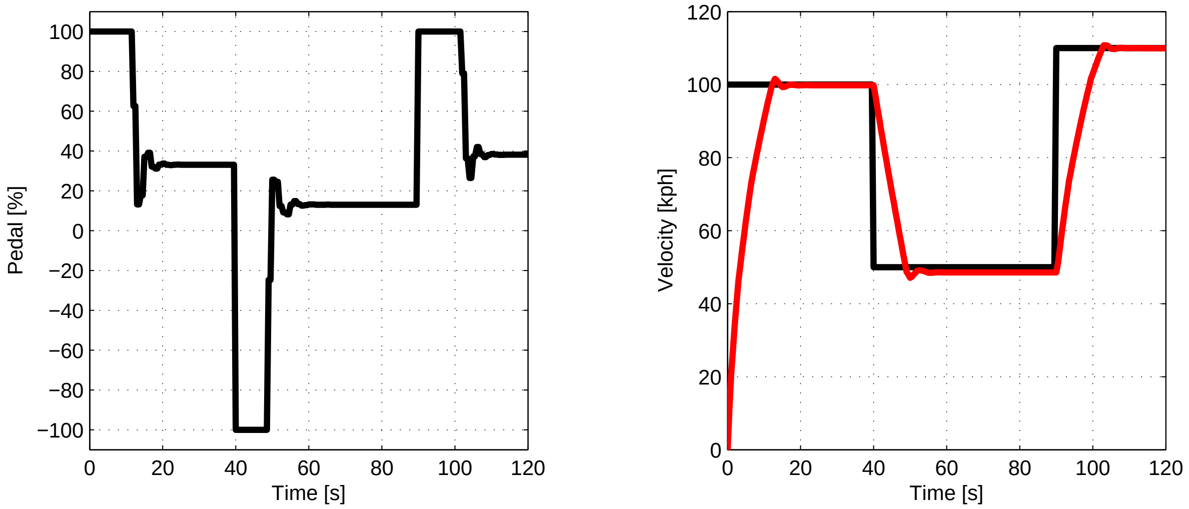
Heemels, W., De Schutter, B., and Bemporad, A. (2001b). On the equivalence of classes of hybrid dynamical models. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on.*



(a) Pedal time response

(b) Vehicle speed time response

Fig. 7. Vehicle identification data



(a) Control and reference input time response

(b) Controlled output time response

Fig. 8. Automatic cruise control

Julian, P., Desages, A., and Agamennoni, O. (1999). High-level canonical piecewise linear representation using a simplicial partition. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 46(4), 463 –480.

Kozak, S. and Stevek, J. (2010). Improved piecewise linear approximation of nonlinear functions in hybrid control.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). URL `http://control.ee.ethz.ch/ mpt/`.

Leondes, C.T. (ed.) (1997). *Control and Dynamic Systems, Neural Network Systems Techniques and Applications, Volume 7*. Academic Press.

Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1), 37 – 50.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.Y., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31, 1691–1724.

Villa, J., Duque, M., Gauthier, A., and Rakoto-Ravalontsalama, N. (2004). Translating pwa systems into mld systems. In *Intelligent Control, 2004. Proceedings of the 2004 IEEE International Symposium on*, 37 – 42.

Comments – Remarks

Robust Time Delay Controller Design Using Discretized Lyapunov Krasovskii Functionals

Nguyen Quang Thuan*

**Slovak University of Technology,*
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics, Ilkovičova 3, 812 19 Bratislava, Slovak Republic
(tel.: +421 2 602 91 544, e-mail: thuan.quang@stuba.sk)

Abstract: The main aim of this paper is to design a robust output feedback PI controller for Networked Control Systems (NCSs) with time-varying delay using discretized Lyapunov-Krasovskii functional method and polytopic linear model. In this framework, time delay of NCSs is partitioned into two parts. The obtained PI controller will guarantee performance and stability of NCSs for all admissible uncertainties and time-varying delays. Finally, one numerical example is given to illustrate the effectiveness of the proposed method.

Key words: Lyapunov-Krasovskii Functional (LKF), Networked Control Systems (NCSs), polytopic linear model, time-varying delay, Integral Quadratic Constraint (IQC).

1. INTRODUCTION

Networked Control Systems (NCSs) is defined as “Feedback control systems wherein the loops are closed through real-time networks” (Ray and Halevi, 1988; Nilson, 1998; Walsh, Ye, Bushnell, 1999; Zhang, Branicky and Philips, 2001). Advantages of using NCSs in the control area include simplicity, cost-effectiveness, ease of system diagnosis and maintenance, increased system agility and testability. However, the integration of communication real-time networks into feedback control loops inevitable leads to some problems. As a result, it leads to a network-induced delay in networked control closed-loop system. The existence of such kind of delay in a network-based control loop can induce instability or poor performance of control systems (Jiang and Han, 2008).

There are two approaches for controller designing and studying of stability of closed-loop systems in the time domain such that Razumikhin theorem and Lyapunov-Krasovskii functional (LKF) approach. It is well know that the LKF approach can provide less conservative results than Razumikhin theorem (Friedman and Niculescu, 2008; Richard, 2003; Kharitonov and Melchor-AGuilar, 2000). The challenge of all approaches using simple LKF is the conservatism of algorithms. Indeed, the delay-independent stability condition is very conservative if the delay is known. Although the simple delay-dependent condition using model transformation is intended to improve the situation, it is not necessary less conservative in all the situations. And the method with implicit model transformation is indeed less conservative than two previous methods; however, it seems to involve substantial conservatism too and requires the system to be stable if the delay is set to zero (Kequin and Niculescu, 2000). To reduce the conservatism efficiently, there are two techniques used. The first one is discretizing scheme of the Lyapunov-Krasovskii matrices (Gu, Kharitonov and Chen, 2003). At a price of an increasing number of variables to be optimized, the result tends to become a necessary and sufficient condition. Another one, developed in a Lyapunov and robust frameworks use an augmented state vector formulation to construct some new LKF for the original system. Hence, a partitioning delay scheme is developed in order to construct a LKF which depends on a discretizing version of the whole state $x_t(\theta)$ (Gouaisbaut and Peaucelle, 2006).

The guaranteed cost control approach has been extended to the uncertain time-delay systems, for the state feedback case, see (Yu and Chu, 1999; Lee and Gyulee, 1999; Zhang, Boukas and Haidar, 2008) and for output feedback (Chen, Guan, and Lu, 2004; Vesely and Nguyen, 2010). In the paper Vesely and Nguyen, 2010 the authors considered the design of robust guaranteed cost PID controller for NCSs. However, it seems that there is not close to a necessary and sufficient condition of stability any more, and the algorithm also involves conservatism. And thus, a partitioning scheme of time-varying delay and IQC are used from (Ariba and Gouaisbaut, 2008) to overcome these disadvantages.

Motivated by the above observation, in this article, a new discretized Lyapunov-Krasovskii functional method will be studied to design a robust output feedback PI controller achieving a guaranteed cost such that the NCSs can be stabilized for all admissible polytopic-type uncertainties and time-varying delays with less conservatism. Sufficient condition for existence of a guaranteed cost output feedback controller is established in term of matrix inequalities.

This paper is organized as follows. Section 2 gives the problem formulation. Section 3 explains main results of the paper. And in section 4 one numerical example is presented to show the effectiveness of the proposed method.

Notation: Throughout this paper, for real matrix M , the notation $M \geq 0$ (respectively $M > 0$) means that matrix M is symmetric and positive semi-definite (respectively positive definite); "*" denotes a block that is readily inferred by symmetry; Matrices, if not explicitly stated, are assumed to have compatible dimensions.

2. PRILIMINARIES AND PROBLEM FORMULATION

Consider the following linear time-delay system described

$$\begin{aligned} &\dot{x}(t) = A(\xi)x(t) + A_d(\xi)x(t-\tau(t)) + B(\xi)u(t) \\ &y(t) = Cx(t) \\ &x(t) = \varphi(t), t \in [-\tau_M, 0] \end{aligned} \quad (1)$$

where $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ is the control input, $y(t) \in R^l$ is the controlled output (measured output). The matrices $A(\xi), A_d(\xi), B(\xi) \in S$ belong to convex hull, and S is a polytope with N vertices $S_1, S_2, .., S_N$ which can formally defined as:

$$S := \left\{ \begin{array}{l} A(\xi), A_d(\xi) \in R^{nxn}, B(\xi) \in R^{nxm} : \\ A(\xi) = \sum_{i=1}^{N} \xi_i A_i \ , \ A_d(\xi) = \sum_{i=1}^{N} \xi_i A_{di} \ , \\ B(\xi) = \sum_{i=1}^{N} \xi_i B_i \ , \ \sum_{i=1}^{N} \xi_i = 1 \ , \ \xi_i \geq 0 \end{array} \right\} \quad (2)$$

where A_i, A_{di}, B_i constant matrices with are appropriate dimensions and ξ_i is time-invariant uncertainty; τ_M is the upper bound of time delay and $\varphi(t)$ is a continuously differentiable initial function. Note S is a convex and bounded domain.

We assume that a real-time communication network is integrated into feedback control loops of system (1), and the network induced delay in NCS $\tau(t)$ is given by $0 < \tau(t) \leq \tau_M$ and the derivative of $\tau(t)$ is bounded as $|\dot{\tau}(t)| \leq \mu \leq 1$.

And now suppose that the time interval $[t - \tau(t) \ , \ t]$ is portioned into N_d=2 parts. The discretization-like method is employed considering the state vector shifted by a fraction $\frac{\tau(t)}{2}$ of the delay. The discretized extended states are constructed with signals (Ariba and Gouaisbaut, 2008):

$$x_i(t)=x\big(t_i(t)\big) \ ; i=0,1 \quad (3)$$

where

$$t_0(t) = t - \frac{\tau(t)}{2} \quad , \quad t_1(t) = t - \tau(t) + \frac{\delta_0(t)}{2} \quad (4)$$

$$\delta_0(t) = \tau(t) - \tau\left(t - \frac{\tau(t)}{2}\right) = \int_{t-\frac{\tau(t)}{2}}^{t} \dot{\tau}(s)ds \leq \frac{\tau_M \mu}{2} \quad (5)$$

These latter variables can be rewritten as:

$$\begin{aligned} &x_0(t)=x(t)- \int_{t_0(t)}^{t} \dot{x}(s)ds \\ &x_1(t)=x(t)- \int_{t_1(t)}^{t_0(t)} \dot{x}(s)ds- \int_{t_0(t)}^{t} \dot{x}(s)ds \end{aligned} \quad (6)$$

The last component $x_1(t)$ is hardly suitable to describe the delayed instantaneous state $x(t-\tau(t))$. In order to clarify the relations between these two signals, we introduce an additional operator $\nabla[.]$ (Ariba and Gouaisbaut, 2008) from L_2 to L_2 as

$$\nabla : x(t) \rightarrow \int_{t-\tau(t)}^{t_1(t)} x(s)ds = \int_{t-\tau(t)}^{t-\tau(t)+\frac{\delta_0(t)}{2}} x(s)ds \quad (7)$$

The L_2 - norm of the operator ∇ is defined by:

$$\|\nabla(x)\|_{L_2}^2 = \int_0^\infty \left(\int_{t-\tau(t)}^{t-\tau(t)+\frac{\delta_0(t)}{2}} x(s)ds \right)^2 dt \quad (8)$$

Then, following the Cauchy-Schwarz inequality, the operator ∇ is bounded as:

$$\begin{aligned} \|\nabla(x)\|_{L_2}^2 &\leq \int_0^\infty \frac{\tau_M \ \mu}{4} \left(\int_{t-\tau(t)}^{t-\tau(t)+\frac{\delta_0(t)}{2}} \|x(s)\|^2 ds \right) dt \\ &\leq \int_0^\infty \frac{\tau_M \ \mu}{4} \left(\int_0^{\frac{\tau_M \ \mu}{4}} \|x(\vartheta + t - \tau(t))\|^2 d\vartheta \right) dt \end{aligned}$$

Consider the substitution $u = \vartheta + t - \tau(t)$, we obtain

$$\begin{aligned} \|\nabla(x)\|_{L_2}^2 &\le \int_0^{\infty} \frac{\tau_M\,\mu}{4} \frac{1}{1-\mu} \left(\int_0^{\frac{\tau_M\,\mu}{4}} \|x(u)\|^2 du \right) d\vartheta \\ &\le \left(\frac{\tau_M\,\mu}{4}\right)^2 \frac{1}{1-\mu} \|x\|_{L_2}^2 \end{aligned} \tag{9}$$

The bounded L_2- norm of operator ∇ (11) is used to construct an IQC to reduce the conservatism of the proposed methodology.

For system (1), we consider the following PI control algorithm

$$u(t) = K_P y(t-\tau(t)) + K_I \int_0^t y(t-\tau(t))dt \tag{10}$$

Consider $z(t) = \int y(t-\tau(t))dt$, by using Newton-Leibniz formulas $x(t-\tau(t)) = x(t) - \int_{t-\tau(t)}^{t} \dot{x}(s)ds$, the PI control algorithm (12) can be written as

$$\begin{aligned} u(t) &= FC_n X(t) - F_P C_P \int_{t-\tau(t)}^{t} \dot{X}(s)ds \\ &= FC_n X(t) + (-F_P C_P) \int_{t-\tau(t)}^{t-t_1(t)} \dot{X}(s)ds + \\ &(-F_P C_P) \left[\int_{t-t_1(t)}^{t-t_0(t)} \dot{X}(s)ds + \int_{t-t_0(t)}^{t} \dot{X}(s)ds \right] \end{aligned} \tag{11}$$

Where $C_n = diag(C, I)$, $C_P = diag(C, 0)$,

$X(t) = \begin{bmatrix} x^T(t) & z^T(t) \end{bmatrix}^T, F = \begin{bmatrix} K_P & K_I \end{bmatrix}, F_P = \begin{bmatrix} K_P & 0 \end{bmatrix}$

Consider $\dot{z}(t) = C_i x(t-\tau(t)) = C_i x(t) - C_i \int_{t-\tau(t)}^{t} \dot{x}(s)ds$

where C_i is output matrix for integral output feedback, the system (1) can be expanded in the following form

$$\dot{X}(t) = A_n(\xi)X(t) + B_n(\xi)u(t) - A_{dn}(\xi) \int_{t-\tau(t)}^{t} \dot{X}(s)ds \tag{12}$$

where

$$\begin{aligned} A_n(\xi) &= \sum_{i=1}^{N} \xi_i A_{ni} \quad , \quad A_{ni} = \begin{bmatrix} A_i + A_{di} & 0 \\ C_i & 0 \end{bmatrix} \\ B_n(\xi) &= \sum_{i=1}^{N} \xi_i B_{ni} \quad , \quad B_{ni} = \begin{bmatrix} B_i \\ 0 \end{bmatrix} \\ A_{dn}(\xi) &= \sum_{i=1}^{N} \xi_i A_{dni} \quad , \quad A_{dni} = \begin{bmatrix} A_{di} & 0 \\ C_i & 0 \end{bmatrix} \end{aligned} \tag{13}$$

Note that dimension of state vector of system integrated PI controller is extended to $n := n + l$.

Applying the PI control algorithm (4) to system (5) will result in the closed-loop feedback system

$$\dot{X}(t) + A_c(\xi)X(t) + A_{dc}(\xi) \int_{t-\tau(t)}^{t} \dot{X}(s)ds = 0 \tag{14}$$

where

$$\begin{aligned} A_c(\xi) &= \sum_{i=1}^{N} \xi_i A_{ci} \quad , \quad A_{ci} = -\left(A_{ni} + B_{ni} F C_n\right) \\ A_{dc}(\xi) &= \sum_{i=1}^{N} \xi_i A_{dci} \quad , \quad A_{dci} = A_{dni} + B_{ni} F_P C_P \end{aligned} \tag{15}$$

Given positive definite symmetric matrices Q, R we will consider the cost function

$$J = \int_0^{\infty} J(t)dt \,,\; J(t) = X^T(t)QX(t) + u^T(t)Ru(t) \tag{16}$$

Consider

$$\eta^T(t) = \begin{bmatrix} \dot{X}^T(t) & X^T(t) & \int_{t-t_0(t)}^{t} \dot{X}^T(s)ds \;;\ldots \\ & \int_{t-t_1(t)}^{t-t_0(t)} \dot{X}^T(s)ds & \int_{t-\tau(t)}^{t-t_1(t)} \dot{X}^T(s)ds \end{bmatrix}$$

and by substituting $u(t)$ from (13) to $u^T(t)Ru(t)$ we obtain

$$J(t) = \eta^T(t) \begin{bmatrix} 0 & 0 & 0 \\ * & Q + C_n^T F^T RFC_n & -C_n^T F^T RF_P C_P \mathrm{I}_P \\ * & * & \mathrm{I}_P{}^T C_P^T F_P^T RF_P C_P \mathrm{I}_P \end{bmatrix} \eta(t) \tag{17}$$

where $I_P = \begin{bmatrix} I & I & I \end{bmatrix}_{n x 3n}$

Associated with the cost, the guaranteed cost controller is defined as follows:

Definition 1.

Consider the uncertain system (1). If there exist a controller of form (12) and a positive scalar J_0 such that for all uncertainties (2), the closed-loop system (16) is asymptotically stable and closed-loop value of the cost function (18) satisfies $J \le J_0$ then J_0 is said to be a guaranteed cost and the controller (12) is said to be guaranteed cost controller.

Finally we introduce the well known results from LQ theory.

Lemma 1.

Consider the continuous-time delay system (14) with control algorithm (12). The control algorithm (12) is the guaranteed cost control for system (14) if and only if there exists LKF $V(\xi,t)$ and IQC $\Pi(\xi,t)$ such that the following condition holds:

$$\frac{d}{dt}\left(V(\xi,t)+\Pi(\xi,t)\right)+J(t)\le 0 \qquad (18)$$

The objective of this paper is to develop a procedure to design a robust PI controller of form (12) which ensure the closed-loop system stability and guaranteed cost.

3. MAIN RESULT

Theorem 1

Consider the uncertain linear time-delay system (1) with network-induced delay $\tau(t)$ satisfying $0<\tau(t)\le\tau_M$, $\|\dot{\tau}(t)\|\le\mu\le 1$ and the cost function (18). If there exist a PI controller of form (12), scalar J_0, and matrices $P_i>0$, $Q_{0i}>0$, $Q_{1i}>0$, $Q_{2i}>0$, $R_{0i}>0$, $R_{1i}>0$ $(i=1,...,N)$, N_1, N_2, N_3, that satisfy the following matrix inequality $W_i \le 0$

$$\begin{aligned}W_i=&\begin{bmatrix}W_{11}^i & W_{12}^i & W_{13}^i\\ * & W_{22}^i & W_{23}^i\\ * & * & W_{33}^i\end{bmatrix}+M_{Q_0}^T\begin{bmatrix}\mu Q_{0i} & (1-\mu)Q_{0i}\\ * & \text{-}(1\text{-}\mu)Q_{0i}\end{bmatrix}M_{Q_0}\\ &+M_{R_0}^T\begin{bmatrix}\tau_M R_{0i} & 0\\ 0 & -\frac{1}{\tau_M}R_{0i}\end{bmatrix}M_{R_0}\\ &+\left\{M_{Q_{1a}}^T\,Q_{1i}\,M_{Q_{1a}}-\left(1-\frac{\mu}{2}\right)M_{Q_{1b}}^T\,Q_{1i}\,M_{Q_{1b}}\right\}\\ &+M_{R_1}^T\begin{bmatrix}\frac{\tau_M}{2}R_{1i} & 0\\ 0 & -\frac{2}{\tau_M}R_{1i}\end{bmatrix}M_{R_1}\end{aligned} \qquad (19)$$

$$\begin{aligned}W_{11}^i&=N_1+N_1^T+\left[\frac{\tau_M\mu}{4}\right]^2\frac{1}{1-\mu}Q_{2i}\\ W_{12}^i&=N_1A_{ci}+N_2^T+P_i\\ W_{13}^i&=N_1A_{dci}I_P+N_3^T\\ W_{22}^i&=N_2A_{ci}+A_{ci}^TN_2^T+C_n^TF^TRFC_n+Q\\ W_{23}^i&=N_2A_{dci}I_P+A_{ci}^TN_3^T-C_n^TF^TRF_PC_PI_P\\ W_{33}^i&=N_3A_{dci}I_P+I_P^TA_{dci}^TN_3^T-diag\left(0_{2n},Q_{2i}\right)\\ &\quad+I_P^TC_P^TF_P^TRF_PC_PI_P\end{aligned} \qquad (20)$$

where

$$M_{Q_0}=\begin{bmatrix}0 & I & 0 & 0 & 0\\ 0 & 0 & I & I & I\end{bmatrix}\in R^{2nx5n},$$

$$M_{R_0}=\begin{bmatrix}I & 0 & 0 & 0 & 0\\ 0 & 0 & I & I & I\end{bmatrix}\in R^{2nx5n},$$

$$M_{R_1}=\begin{bmatrix}I & 0 & 0 & 0 & 0\\ 0 & 0 & I & 0 & 0\end{bmatrix}\in R^{2nx5n},$$

$$M_{Q_{1a}}=\begin{bmatrix}0 & I & 0 & 0 & 0\\ 0 & I & -I & 0 & 0\end{bmatrix}\in R^{2nx5n},$$

$$M_{Q_{1b}}=\begin{bmatrix}0 & I & -I & 0 & 0\\ 0 & I & -I & -I & 0\end{bmatrix}\in R^{2nx5n}$$

Then the uncertain system (1) with controller (12) is parameter-dependent quadratically asymptotically stable and the cost function (18) satisfies the following bound

$$J\le J_0=\sqrt{\lambda_{MP}^2+\lambda_{MQ_0}^2+\lambda_{MR_0}^2+\lambda_{MR_1}^2+\lambda_{MQ_1}^2}*J_M \qquad (21)$$

where

$$\lambda_{MP}=\underset{i=1..N}{Max}(Max(Eigenvalue(P_i))),$$
$$\lambda_{MQ_0}=\underset{i=1..N}{Max}(Max(Eigenvalue(Q_{0i}))),$$

$$\lambda_{MR_0}=\underset{i=1..N}{Max}(Max(Eigenvalue(R_{0i}))),$$
$$\lambda_{MQ_1}=\underset{i=1..N}{Max}(Max(Eigenvalue(Q_{1i}))),$$

$$\lambda_{MR_1}=\underset{i=1..N}{Max}(Max(Eigenvalue(R_{1i})))$$

$$J_M=\sqrt{\begin{aligned}&\|x_0\|^4+\left(\int_{-\tau_M}^{0}\|\varphi(s)\|^2ds\right)^2+\left(\int_{-\tau_M}^{0}d\theta\int_{\theta}^{0}\|\dot{\varphi}(s)\|^2ds\right)^2\\ &+\left(2\int_{-\frac{\tau_M}{2}}^{0}\|\varphi(s)\|^2ds\right)^2+\left(\int_{-\frac{\tau_M}{2}}^{0}d\theta\int_{\theta}^{0}\|\dot{\varphi}(s)\|^2ds\right)^2\end{aligned}}$$

Proof 1

The proof is based on the Lyapunov-Krasovskii approach and Integral Quadratic Constraint (IQC). Let us define the following Lyapunov-Krasovskii functional candidate:

$$V(\xi,t)=V_1(\xi,t)+V_2(\xi,t)+V_3(\xi,t)+V_4(\xi,t)+V_5(\xi,t) \tag{22}$$

where

$$V_1(\xi,t)=X^T(t)P(\xi)X(t) \tag{23}$$

$$V_2(\xi,t)=\int_{t-\tau(t)}^{t} X^T(s)Q_0(\xi)X(s)ds \tag{24}$$

$$V_3(\xi,t)=\int_{-\tau_M}^{0} d\theta \int_{t+\theta}^{t} \dot{X}^T(s)\,R_0(\xi)\,\dot{X}(s)ds \tag{25}$$

$$V_4(\xi,t)=\int_{t-\frac{\tau(t)}{2}}^{t} \begin{bmatrix} X(s)\\ X_0(s)\\ X_1(s)\end{bmatrix}^T Q_1(\xi)\begin{bmatrix} X(s)\\ X_0(s)\\ X_1(s)\end{bmatrix} ds \tag{26}$$

$$V_5(\xi,t)=\int_{-\frac{\tau_M}{2}}^{0} d\theta \int_{t+\theta}^{t} \dot{X}^T(s)\,R_1(\xi)\,\dot{X}(s)ds \tag{27}$$

and the IQC

$$\Pi(\xi,t)=\int_0^t \begin{bmatrix} \left(\dfrac{\tau_M\,\mu}{4}\right)^2 \dfrac{1}{1-\mu}\dot{X}^T(s)Q_2(\xi)\dot{X}(s) \\ -\nabla\left[\dot{X}(s)\right]^T Q_2(\xi)\nabla\left[\dot{X}(s)\right]\end{bmatrix} ds \tag{28}$$

$\nabla[.]$ is an operator defined as (8).

Consider

$P(\xi)=\sum_{i=1}^{N}\xi_i P_i\,,P_i>0\,;Q_k(\xi)=\sum_{i=1}^{N}\xi_i Q_{ki}\,,Q_{ki}>0,\;\;k=0..2$

and $R_j(\xi)=\sum_{i=1}^{N}\xi_i R_{ji}\,,R_{ji}>0,\,j=0{,}1;$ then $V(\xi,t)>0$. The IQC $\Pi(\xi,t)$ is positive definite (see proof in Ariba and Gouaisbaut, 2008).

The derivative of the each element of (24) along the trajectories of (1) leads to

$$\dot{V}_1(\xi,t)=2X^T(t)P(\xi)\dot{X}(t) \tag{29}$$

$$\begin{aligned}\dot{V}_2(\xi,t)&=X^T(t)Q_0(\xi)X(t)\\ &\quad-(1-\mu)X^T(t-\tau)Q_0(\xi)X(t-\tau)\\ &=\eta^T(t)M_{Q_0}^T\begin{bmatrix}\mu Q_0(\xi) & (1-\mu)Q_0(\xi)\\ * & -(1-\mu)Q_0(\xi)\end{bmatrix}M_{Q_0}\eta(t)\end{aligned} \tag{30}$$

$$\begin{aligned}\dot{V}_3&=\int_{-\tau_M}^{0}\dot{X}^T(t)R_0(\xi)\dot{X}(t)ds-\int_{t-\tau_M}^{t}\dot{X}^T(t)R_0(\xi)\dot{X}(t)ds\\ &=\tau_M\dot{X}^T(t)R_0(\xi)\dot{X}(t)-\int_{t-\tau_M}^{t}\dot{X}^T(t)R_0(\xi)\dot{X}(t)ds\end{aligned}$$

Invoking the Jensen's inequality $\dot{V}_3(\xi,t)$ can be bounded by

$$\begin{aligned}\dot{V}_3(\xi,t)&\le\tau_M\dot{X}^T(t)R_0(\xi)\dot{X}(t)\\ &\quad-\frac{1}{\tau_M}\left(\int_{t-\tau(t)}^{t}\dot{X}^T(s)ds\right)R_0(\xi)\left(\int_{t-\tau(t)}^{t}\dot{X}(s)ds\right)\\ &\le\eta^T(t)M_{R_0}^T\begin{bmatrix}\tau_M R_0(\xi) & 0\\ 0 & -\dfrac{1}{\tau_M}R_0(\xi)\end{bmatrix}M_{R_0}\eta(t)\end{aligned} \tag{31}$$

$$\begin{aligned}\dot{V}_4(\xi,t)&=\begin{bmatrix}X(t)\\ X_0(t)\end{bmatrix}^T Q_1(\xi)\begin{bmatrix}X(t)\\ X_0(t)\end{bmatrix}\\ &\quad-\left(1-\frac{\mu}{2}\right)\begin{bmatrix}X_0(t)\\ X_1(t)\end{bmatrix}^T Q_1(\xi)\begin{bmatrix}X_0(t)\\ X_1(t)\end{bmatrix}\\ &=\eta^T(t)\begin{bmatrix}M_{Q_{1a}}^T Q_1(\xi)M_{Q_{1a}}\\ -\left(1-\dfrac{\mu}{2}\right)M_{Q_{1b}}^T Q_1(\xi)M_{Q_{1b}}\end{bmatrix}\eta(t)\end{aligned} \tag{32}$$

$$\begin{aligned}&\dot{V}_5(\xi,t)\le\eta^T(t).K.M_{R_1}\eta(t)\\ &K=M_{R_1}^T diag\left(\frac{\tau_M}{2}R_1(\xi),-\frac{2}{\tau_M}R_1(\xi)\right)\end{aligned} \tag{33}$$

The derivative of the IQC (30) along the trajectories of (1), we obtain

$$\begin{aligned}\dot{\Pi}(\xi,t)&=\left[\frac{\tau_M\mu}{4}\right]^2\frac{1}{1-\mu}\dot{X}^T(t)Q_2(\xi)\dot{X}(t)\\ &\quad-\left(\int_{t-\tau(t)}^{t-t_{N-1}}\dot{X}(s)ds\right)^T Q_2(\xi)\left(\int_{t-\tau(t)}^{t-t_{N-1}}\dot{X}(s)ds\right)\end{aligned} \tag{34}$$

where the matrices $M_{Q_0},M_{R_0},M_{R_1},M_{Q_{1a}},M_{Q_{1b}}$ were defined as (22).

Applying the free-weighting matrices technique, the equation (16) is represented in the following equivalent form

$$\alpha(t) = 2\eta^T(t)\begin{bmatrix} N_1^T & N_2^T & N_3^T \end{bmatrix}^T .$$
$$\begin{bmatrix} I & A_c(\xi) & A_{dc}(\xi)\mathrm{I}_P \end{bmatrix}\eta(t) = 0 \tag{35}$$
$$= \eta^T(t)\begin{bmatrix} \boxed{\begin{matrix} N_1 \\ +N_1^T \end{matrix}} & \boxed{\begin{matrix} N_1A_c(\xi) \\ +N_2^T \end{matrix}} & \boxed{\begin{matrix} N_1A_{dc}(\xi)\mathrm{I}_P \\ +N_3^T \end{matrix}} \\ * & \boxed{\begin{matrix} N_2A_c(\xi) \\ +A_c^T(\xi)N_2^T \end{matrix}} & \boxed{\begin{matrix} N_2A_{dc}(\xi)\mathrm{I}_P \\ +N_3^T A_c^T(\xi) \end{matrix}} \\ * & * & \boxed{\begin{matrix} N_3A_{dc}(\xi)\mathrm{I}_P \\ +\mathrm{I}_P^T A_{dc}^T(\xi)N_3^T \end{matrix}} \end{bmatrix}\eta(t)$$

Due to lema1, the closed-loop system (16) is robustly asymptotically stable and give an upper bound (a guaranteed cost) for the cost function (18) if

$$\dot{V}(\xi,t) + \dot{\Pi}(\xi,t) + J(t) \le \eta^T(t)W(\xi)\eta^T(t) \le 0$$
$$\Leftrightarrow \; W(\xi) \le 0 \tag{36}$$

where $W(\xi) = \sum_{i=1}^{N} \xi_i W_i$.

If for each $W_i \le 0$, $i = 1...N$, then $W(\xi) = \sum_{i=1}^{N} \xi_i W_i \le 0$. Therefore, $\dot{V}(\xi,t) + \dot{\Pi}(\xi,t) \le -J(t) \le 0 \;\; (J(t) \ge 0)$, resp. $J(t) \le -\dot{V}(\xi,t)$. With $X(t) = [\varphi^T(t) \quad 0]$, $\forall t \in [-\tau_M, 0]$, after integrating $\int_0^{\infty} J(t)dt \le -\int_0^{\infty}\left(\dot{V}(\xi,t) + \dot{\Pi}(\xi,t)\right)dt$ we obtain

$$J \le V_0 \le \lambda_{MP}\|x_0\|^2 + \lambda_{MQ_0}\int_{-\tau_M}^{0}\|\varphi(s)\|^2 ds$$
$$+ \lambda_{MR_0}\int_{-\tau_M}^{0} d\theta \int_{\theta}^{0}\|\dot{\varphi}(s)\|^2 ds$$
$$+ \lambda_{MQ_1}\left(2\int_{-\frac{\tau_M}{2}}^{0}\|\varphi(s)\|^2 ds\right)$$
$$+ \lambda_{MR_1}\int_{-\frac{\tau_M}{2}}^{0} d\theta \int_{\theta}^{0}\|\dot{\varphi}(s)\|^2 ds$$

It is known, that for two arbitrary vectors $\bar{X}, \bar{Y}$, the following inequality hold:

$$\left|\bar{X}^T\,\bar{Y}\right| \le \left\|\bar{X}\right\|\left\|\bar{Y}\right\| \tag{37}$$

Consider $\bar{X} = \begin{bmatrix} \lambda_{MP} & \lambda_{MQ_0} & \lambda_{MR_0} & \lambda_{MQ_1} & \lambda_{MR_1} \end{bmatrix}^T$

$$\bar{Y}^T = \begin{bmatrix} \|x_0\|^2 & \int_{-\tau_M}^{0}\|\varphi(s)\|^2 ds & \int_{-\tau_M}^{0} d\theta\int_{\theta}^{0}\|\dot{\varphi}(s)\|^2 ds & \dots \\ \dots & 2\int_{-\frac{\tau_M}{2}}^{0}\|\varphi(s)\|^2 ds & \int_{-\frac{\tau_M}{2}}^{0} d\theta\int_{\theta}^{0}\|\dot{\varphi}(s)\|^2 ds & \end{bmatrix}$$

Applying the inequality (39), to above equation the upper bound of cost function (18) J_0 is obtained as (23).

The theorem 1 is proved.

4. EXAMPLE

In this section we present the results of numerical calculations of one example to demonstrate the effectiveness of proposed method. In this example, we compare results of proposed method with PI controller corresponding to partitioning of time-delay $N_d = 1,2$ and method in (Thuan and Vojtech, 2010) through of 1000 generated examples. Numerical calculation is realized in PEN-BMI.

In this case 1000 examples are generated with following parameters: the system $2nd$ order may be unstable with $0 < \max(real(eig)) \le 0.1$, each system has two inputs two outputs with one uncertainty (two vertices are calculated), generated time-delay middle value $\tau_{middle} = 200[ms]$, time-delay rate $\dot{\tau}(t) \le 0.5$. Results are in Table 1.

N_d \Controller	*PI*
1, (Vesely and Nguyen, 2010)	888
2	897

The parameters of cost function are $R = rI, r = 1;$ $Q = qI, q = 0.1$.

5. CONCLUSIONS

The paper addresses the problem to output feedback guaranteed cost controller design for Networked Uncertain Control Systems with time-varying delay and polytopic uncertainties. Base on partitioning scheme of time-varying delay (time delay is partitioned into two parts) and using integral quadratic constraint, a new discretized Lyapunov-Krasovskii functional method is obtained to synthesize a robust PI controller achieving a guaranteed cost and parameter-dependent quadratic stability such that the NCSs is stable for all admissible uncertainties and bounded time-varying delays. The solution has been obtained using PEN-BMI script.

The obtained numerical results exhibit that when partitioning scheme of time-varying delay and IQC are used the robust controller design procedure conservatism is decreased.

ACKNOWLEDGMENT

The work has been supported by Grant N1/0544/09 and 1/0592/10 of the Slovak Scientific Grant Agency.

REFERENCES

Ariba, Y.; Gouaisbaut, F. 2007. Delay-dependent stability analysis of linear systems with time-varying delay. In: *IEEE Conference on Decision and Control*, December 2007, 2053–2058.

Kharitonov, V. L.; Melchior Aquila, D. 2000. On delay-dependent stability conditions. *Systems and Control Letters*, 40, 71-76.

Chen, W.H.; Guan, Z.H.; Lu, X. 2004. Delay-dependent output feedback guaranteed cost for uncertain time-delay systems. *Automatica*, 40, 1263-1268.

Gouaisbaut, F.; Peaucelle, D. 2006. Delay-dependent stability analysis of linear time delay systems. In: *IFAC Workshop on Time Delay System (TDS'06)*, Aquila, Italy, July 2006.

Fridman, E.; Niculescu, S. J. 2008. On complete Lyapunov-Krasovskii functional techniques for uncertain systems with fast varying delays. *Int. J. of Robust and nonlinear Control*, 18, 364-374.

Gu, K.; Niculescu, S. I. 2000. Additional dynamics in transformed time-delay systems. *IEEE Trans. Auto. Control*, 2000, 45(3): 572–575.

Gu, K.; Kharitonov, V. L.; CHEN, J. 2003. *Stability of Time-Delay Systems*. Birkhäuser Boston, 2003. Control engineering.

Lee, B.; Guylee, J. 1999. Robust stability and stabilization of linear delayed systems with structured uncertainty. *Automatica*, 35, 1149-1154.

Jiang, X.; Han, Q. L. 2008. New stability criteria for linear systems with interval time-varying delay. *Automatica* , 44, 2680-2685.

Nilson, J. 1998. Real-time Control System with Delays. *PhD dissertation, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden*.

Ray, A.; Halevi, Y. 1988. Integrated communication and control systems: Part II Design considerations. *ASME Journal of Dynamic Systems, Measurement and control,* 110, 374-381.

RICHARD, J. P. 2003. Time-delay systems: an overview of some recent advances and open problem. *Automatica*, 39, 1667-1694.

Vesely, V.; Nguyen Quang, T. 2010. Robust Output Networked Control System Design, ICIC submitted.

Walsh, G. C.; Ye, H.; Bushnell, L. 1999. Stability analysis of networked control systems. *Proceedings of the American Control Conference*, 2876-2880.

Yu, L.; Chu, J. 1999. An LMI approach to guaranteed cost control of linear uncertainty time-delay systems. *Automatica*, 35, 1155-1159.

Zhang, W.; Branicky; Philips, S. M. 2001. Stability of networked control systems. *IEEE Control System Magazine*, 21(1), 84-99.

Zhang, L.; Boukas El-Kebir; Haidar, A. 2008. Delay-range dependent control synthesis for time-delay systems with actuator saturation. *Automatica*, 44, 2691-2695.

Comments – Remarks

Stabilizing Model Predictive Controller Design with Reduced On-line Computation

Vojtech Veselý, Danica Rosinová

Institute of Control and Industrial Informatics, Slovak University of Technology
Faculty of Electrical Engineering and Information Technology:
Bratislava, Slovak Republic
(e-mail: danica.rosinova@stuba.s)

Abstract: The paper studies the problem of output feedback stabilizing model predictive control design with guaranteed cost. Considering a prediction horizon *N*, the proposed control design method is based on the idea of step-by-step sequential design for the whole prediction horizon, so that in each step, an one-step ahead stable model predictive control is designed. This approach enables to reduce the on-line computational load significantly. Numerical examples are given to illustrate the effectiveness of the proposed method.

Keywords: Model predictive control, Guaranteed cost, Lyapunov function, Output feedback, Diophantine equation

1. INTRODUCTION

Model predictive control (MPC) has gained considerable attention in control of dynamic systems. The idea of MPC can be summarized in following basic points, (Camacho and Bordons, 2004; Maciejovski, 2002, Rossiter, 2003):

- Predict the future behaviour of the process state/output over the finite time horizon (prediction horizon).
- Compute the future input signals on line at each step by minimizing a cost function under inequality constraints on the manipulated (control) and/or controlled variables.
- Apply on the controlled plant only the first of vector control variable and repeat the previous step with new measured input/state/output variables.

As indicated above, the presence of the plant model is a necessary condition for the development of the predictive control. The main criticism related to MPC is that because of the finite prediction horizon the algorithm in its original formulation does not guarantee stability of closed-loop system. Several approaches have been developed in this field, robust control issues has been studied in (Kothare et al., 1996). The excellent survey on stability, robustness properties and optimality of MPC are given in (Mayne et al, 2000).

In this paper, the two MPC design methods are proposed based on the idea of sequential design for prediction horizon *N*, using one step ahead model predictive stabilizing control design approach. First sequential design method is based on classical LQ state feedback and solution of diophantine equation. The second method is based on the Lyapunov function approach with guaranteed cost which is adapted to design model predictive control with output feedback control for prediction horizon *N* and constraints on input variables. The proposed sequential design strategy significantly reduces the problem size which enables to solve the resulting bilinear matrix inequality (BMI) even for longer prediction horizon. For both design approaches, the respective BMI is solved, its feasible solution provides corresponding output feedback gain matrices.

The paper is organized as follows. Section 2 gives a problem formulation and some preliminaries about a predictive output/state model. In Section 3, two sequential design methods are proposed to design output feedback predictive control for prediction horizon *N*. Section 4 provides examples to illustrate the design procedure results for several exaples.

2. PROBLEM FORMULATION AND PRELIMINARIES

Consider a time invariant linear discrete-time system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

where $x(t) \in R^n, u(t) \in R^m, y(t) \in R^l$ are state, control and output variables of the system, respectively; *A*, *B*, *C* are known matrices of corresponding dimensions.

The following quadratic cost function is considered for prediction horizon *N*

$$J(t) = F(x(N)) + \sum_{t=t_0}^{N-1} \left[x^T(t)Qx(t) + u^T(t)Ru(t) \right] \tag{2}$$

where $F(x(N))$ is a given terminal constraint at time N and $Q = Q^T \geq 0 \in R^{n\times n}$, $R = R^T > 0 \in R^{m\times m}$ are corresponding weighting matrices.

The problem studied in this paper is to design a model predictive control with the following control algorithm

$$u(t) = F_{11}y(t) + F_{12}y(t+1) \tag{3}$$

and, in the second design step, a model predictive control for a given prediction horizon *N* is designed in the form

$$u(t+k-1) = \sum_{i=1}^{k+1} F_{ki}y(t+i-1), \qquad k = 2,3,...,N \tag{4}$$

where $F_{ki} \in R^{m \times l}, k = 1,2,...,N$. $i = 1,2,...,k+1$ is output (state) feedback gain matrix to be determined so that the given cost function (2) is minimized with respect to given constraint.

3. MODEL PREDICTIVE CONTROL DESIGN

The MPC sequential design procedure is described in this section, basically it includes two steps: design of one step ahead model predictive control, considering the measured output variable; design of MPC for prediction horizon *N* to obtain predicted model outputs for the whole prediction horizon.

Substituting from system model (1) to (3), the control law (3) can be rewritten as

$$u(t) = F_{11}Cx(t) + F_{12}Cx(t) \;\; \Rightarrow$$

$$u(t) = (I - F_{12}CB)^{-1}(F_{11}C + F_{12}CA)x(t) = K_1x(t) \tag{5}$$

where $K_1 = (I - F_{12}CB)^{-1}(F_{11}C + F_{12}CA)$, *I* is identity matrix of corresponding dimension.

The respective closed-loop system is then described by

$$x(t+1) = (A + BK_1)x(t) = D_1x(t) \tag{6}$$

where $D_1 = A + BK_1$.

The control objective is usually to steer the state to the origin or to an equilibrium state x_r for which the output $y_r = Cx_r = w$, where *w* is the constant reference. A suitable change of coordinates reduces the latter problem to the former one which is therefore considered in the sequel.

To design the state feedback matrix K_1 for system (6), the cost function (2) for this step is in the form

$$J_1 = F_1(x(N_1)) + \bar{J}_1(t) \tag{7}$$

$$\bar{J}_1(t) = \sum_{t=t_0}^{N_1-1} x^T(t)Q_1x(t) + u^T(t)R_1u(t)$$

where $F_1(x(N_1))$ is a given terminal constraint at time N_1 and $Q_1 = Q_1^T \geq 0 \in R^{n \times n}$, $R_1 = R_1^T > 0 \in R^{m \times m}$ are corresponding weighting matrices.

Definition 1
Consider the system (1). If there exists a control law u^* given by (5) and a positive scalar J_1^* such that the closed-loop system (6) is stable and the value of the closed-loop cost function (7) satisfies $J_1 \leq J_1^*$, then J_1^* is said to be *guaranteed cost* and u^* is said to be *guaranteed cost control* law for system (1).

Consider the constraints on the maximal value of state and control input respectively: $\max_t(x^T(t)x(t))$, $\max_t(u^T(t)u(t))$ are given in the form

$$\frac{\max_t(u^T(t)u(t))}{\max_t(x^T(t)x(t))} = \rho \, . \tag{8}$$

Then, for the worst case, the following linear matrix inequality (LMI) constraint for matrix K_1 can be formulated

$$\begin{bmatrix} \rho I & K_1^T \\ K_1 & I \end{bmatrix} \geq 0 \tag{9}$$

Inequality (9), cost function (7) and system (1) are basical for a calculation of gain matrix K_1. This can be done through several approaches. If the constraint (8) is omitted, the classical LQ design approach may be used; if $N_1 \to \infty$ and $F(x(N_1)) = 0$, bilinear matrix inequality could be formed which can be transformed using linearization approach, (de Oliveira et al, 2000), into the linear (LMI) one with respect to unknown matrix K_1. For the choice of different terminal constraint $F(x(N_1))$, the different approaches for the calculation of K_1 can be used, see (Mayne et al, 2000). As soon as the state feedback gain matrix K_1 is known, using (5), the following diophantine equation with respect to matrices F_{11}, F_{12} is obtained

$$K_1 = F_{11}C + F_{12}CD_1 \tag{10}$$

Note, that if (*A*,*B*) is controllable for given Q_1, R_1, there exists a matrix K_1 which guarantees minimum of cost (7) and stability of the closed-loop system (6). Moreover, if there exists a solution to diophantine equation (10) with respect to matrices F_{11}, F_{12}, the proposed one-step ahead predictive control algorithm (5) for closed-loop with output feedback guarantees the same properties as the closed-loop system obtained by state feedback and gain matrix K_1. Note, that according to receding horizon strategy, only *u*(*t*) is applied to plant. This complete the one-step ahead design procedure is completed.

Consider now the case *N* = 2. Model predictive control algorithm and system are

$$x(t+2) = Ax(t+1) + Bu(t+1)$$

$$u(t+1) = F_{21}y(t) + F_{22}y(t+1) + F_{23}y(t+2)$$

or after some manipulation

$$u(t+1) = K_2x(t)$$

where

$$K_2 = (I - F_{23}CB)^{-1}(F_{21}C + F_{22}CD_1 + F_{23}CAD_1)$$

and the respective closed-loop system

$$x(t+2)=(AD_1+BK_2)x(t)=D_2(t)x(t) \tag{11}$$

The closed-loop system (11) is stable with respect to state $x(t+2)$ if and only if the matrix D_2 is stable, i.e. $real(\lambda_i(D_2))<0$, $i=1,2,...,n$. The matrix K_2 can be calculated using the same approaches as used for matrix K_1. Control objective for calculation of K_2 may be the same as for calculation of K_1 or the new control objective can be defined

$$J_2=F_2(x(N_1))+\bar{J}_2(t)$$

where

$$\bar{J}_2(t)=\sum_{t=t_0}^{N_1}x^T(t)Q_2x(t)+u^T(t+1)R_2u(t+1)$$

Note, that control input $u(t+1)$ is applied for a calculation of future output $y(t+2)$ by predictive model (11). The constraints (9) for K_2 calculation need not be used, since the respective control input is used only in model prediction and is not directly applied to the real process.

Sequentially, for the case of "*k*" step prediction, the following closed-loop system is obtained

$$x(t+k)=(AD_{k-1}+BK_k)x(t)=D_k(t)x(t) \tag{12}$$

where

$$D_0=I,\ D_k=AD_{k-1}+BK_k$$

$k=1,2,...,N$. The respective diophantine equation for a calculation of output feedback matrices (4) is

$$K_k=\sum_{j=1}^{k}F_{kj}CD_{j-1}+F_{kk+1}CD_k \tag{13}$$

The obtained results are summarized in the following theorem.

Theorem 1
Consider the closed-loop system (12), control algorithm (3) and (4) and receding horizon *N*. The respective closed-loop system is stable with guaranteed cost if the following conditions hold: pair (AD_{k-1},B) is controllable for $k=1,2,...,N$, that is, there exists a matrix K_k which guarantees the closed-loop system matrix stability with guaranteed cost and for a given K_k there exists solution of diophantine equation (13) with respect to output feedback matrices F_{kj}, $k=1,2,...,N$, $j=1,2,...,k+1$. Note that some matrices F_{kj}; $k=1,2,...,N$, $j\le k-1$ may be equal to zero.

To find a solution to diophantine equation (13), the following way is proposed. Introduce the following control algorithm

$$u(t+k-1)=F_{kk}y(t+k-1)+F_{kk+1}y(t+k) \tag{14}$$

$k=1,2,...,N$. For model prediction *k*, the closed-loop system (12) is

$$x(t+k)=(A+BF_{kk}C)D_{k-1}x(t)+BF_{kk+1}Cx(t+k) \tag{15}$$

The aim of the proposed second-step predictive control design procedure is to design gain matrices F_{kk}, F_{kk+1}, $k=1,2,...,N$ such that closed-loop system (15) is stable with guaranteed cost when $F(x(N_1))=0$, $N_1\to\infty$. The following theorem gives sufficient conditions to design the above output feedback matrices.

Theorem 2
Closed-loop system (15) is stable for $k=1,2,...,N$ with guaranteed cost if there exist matrices $N_{k1}\in R^{n\times n}$, $N_{k2}\in R^{n\times n}$, F_{kk}, F_{kk+1} and a positive definite matrix $P\in R^{n\times n}$ such that the following bilinear matrix inequality holds

$$\begin{bmatrix} G_{k11} & G_{k12} \\ G_{k12}^T & G_{k22} \end{bmatrix}\ge 0 \tag{16}$$

where

$$\begin{aligned} G_{k11}&=N_{k1}^TM_{ck}+M_{ck}^TN_{k1}+C^TF_{kk+1}^TR_kF_{kk+1}C+P \\ G_{k12}^T&=D_{k-1}^TC^TF_{kk}^TR_kF_{kk+1}C+D_{k-1}^TA_{ck}^TN_{k1}+N_{k2}^TM_{ck} \\ G_{k22}&=Q_k-P+D_{k-1}^TC^TF_{kk}^TR_kF_{kk}CD_{k-1}+N_{k2}^TA_{ck}D_{k-1}+ \\ &\quad +D_{k-1}^TA_{ck}^TN_{k2} \end{aligned}$$

for $k=1,2,...,N$

where

$$\begin{aligned} M_{ck}&=BF_{kk+1}C-I \\ A_{ck}&=A+BF_{kk}C \\ D_k&=AD_{k-1}+BK_k \\ K_k&=(I-F_{kk+1}CB)^{-1}(F_{kk}C+F_{kk+1}CA)D_{k-1} \end{aligned}$$

If there exists a feasible solution of (16) with respect to matrices F_{kk}, F_{kk+1}, N_{k1}, N_{k2}, $k=1,2,...,N$ and positive definite matrix *P*, then the proposed model predictive algorithm (14) guarantees the stability of the closed-loop system with guaranteed cost.

Note, that for the case of *N* + 1 one-step ahead predictive control and polytopic systems $A=\sum_{i=1}^{S}A_i\alpha_i$, parameter dependent Lyapunov function $P=\sum_{i=1}^{S}P_i\alpha_i$, for more details see (deOliveira et al, 2000), (Kothare et al, 1996), a feasible solution of (16) guarantees the robustness properties of the respective closed-loop model predictive control.

4. EXAMPLES

Example 1
This example serves as a benchmark. Continuous-time model of double integrator has been converted to a discrete-time one with sampling time 0.1s, the model turns to (1) where

$$A = \begin{bmatrix} 2 & -0.5 \\ 2 & 0.0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.125 \\ 0.0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.08 & 0.4 \end{bmatrix}$$

Eigenvalues of matrix A are $\{1,1\}$. For prediction horizon $N = 3$ and cost function matrices $Q_1 = 5I, R_1 = I$, $Q_2 = 10I, R_2 = I$ and $Q_3 = I, R_3 = I$, the following results are obtained.

Gain matrices for state feedback

$K_1 = [8.2747 \quad -2.727]$

$K_2 = [2.2454 \quad -1.0169]$

$K_3 = [0.1934 \quad -0.3645]$

Solution of diophantine equations (10) for output feedback gain matrices are:

for $u(t)$:

$F_{11} = -6.4985 \qquad F_{12} = 10.0251$

for $u(t+1)$:

$F_{21} = -1.8969 \qquad F_{22} = 1.1255 \qquad F_{23} = 1.7097$

for $u(t+2)$:

$F_{31} = -0.8204 \qquad F_{32} = -0.0608$

$F_{33} = 0.2495 \qquad F_{34} = 1.2098.10^{-7}$

Eigenvalues of the closed-loop system for N=3 are $\{-0.035 \pm 0.3329i\}$.

Example 2

Consider the following 5th order model with four outputs and two control inputs

$$A = \begin{bmatrix} 0.6 & 0.0097 & 0.0143 & 0 & 0 \\ 0.012 & 0.9754 & 0.0049 & 0 & 0 \\ -0.0047 & 0.01 & 0.46 & 0 & 0 \\ 0.0488 & 0.0002 & 0.0004 & 1 & 0 \\ -0.0001 & 0.0003 & 0.0488 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0425 & 0.0053 \\ 0.0052 & 0.01 \\ 0.0024 & 0.0474 \\ 0.0011 & 0.0001 \\ 0 & 0.0012 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For the case of $N = 1$, $Q_1 = 5I, R_1 = I$, the following results are obtained.

State feedback matrix K_1 obtained by LQ method

$$K_1 = \begin{bmatrix} 0.4642 & 0.4052 & 0.0091 & 2.1971 & -0.154 \\ 0.0536 & 0.6707 & 0.3362 & 0.1361 & 2.1947 \end{bmatrix}$$

Solution of diophantine equation for output feedback

$$F_{11} = \begin{bmatrix} 4.4269 & 6.514 & 1.5663 & -1.3642 \\ 6.6108 & 11.0884 & -0.9085 & 0.1941 \end{bmatrix}$$

$$F_{12} = \begin{bmatrix} -7.0219 & -14.42 & -0.1983 & -0.3201 \\ -11.6202 & -23.8367 & -0.3284 & -0.5298 \end{bmatrix}$$

Eigenvalues of the closed-loop system are $\{0.5909, 0.4541, 0.9627, 0.9874, 0.9923\}$. It is interesting to note that for prediction horizon N = 3, eigenvalues of the respective closed-loop system are changed into $\{0.208, 0.0941, 0.9100, 0.9768, 0.9814\}$.

Example 3

The same system as in Example 2 is considered. The design problem is reformulated into decentralized one.

Design two PS (PI) decentralized controllers for control of input variable $u(t)$; N = 5, when cost functions are defined by $Q_1 = Q_2 = Q_3 = I$; $R_1 = R_2 = R_3 = I$; $Q_4 = Q_5 = 0.5I$ $R_4 = R_5 = I$.

The obtained output feedback gain matrices (solution of (16)) are :

$u(t) = F_{11}y(t) + F_{12}y(t+1)$

$$F_{11} = \begin{bmatrix} -0.2512 & 0 & -0.4353 & 0 \\ 0 & -0.269 & 0 & -0.3435 \end{bmatrix}$$

where the decentralized controller consists of two subsystem controllers: the first one has proportional gain $K_{1p} = 0.2512$ and integral gain $K_{1i} = 0.4353\,0$ and parameters of the second controller are $K_{2p} = 0.269$; $K_{2i} = 0.3435$. Because output $y(t+1)$ is obtained from a predictive model, there is no need to use decentralized structure for F_{12}

$$F_{12} = \begin{bmatrix} -0.2253 & -0.1198 & -0.5957 & 0.1095 \\ -0.2024 & -0.2268 & -0.0547 & -0.6476 \end{bmatrix}$$

and finally for

$u(t+4) = F_{55}y(t+4) + F_{56}y(t+5)$

where output feedback gain matrices are

$$F_{55} = \begin{bmatrix} -0.0046 & -0.0046 & -0.0874 & 0.0099 \\ -0.0102 & -0.0054 & -0.0044 & -0.0841 \end{bmatrix}$$

$$F_{56} = \begin{bmatrix} -0.00437 & -0.006 & -0.1067 & 0.0024 \\ -0.0134 & -0.0479 & -0.0141 & -0.1051 \end{bmatrix}$$

Eigenvalues of the respective closed-loop system for $N = 5$ are
$\{0.074, 0.0196, 0.8864, 0.9838, 0.9883\}$.

The above examples illustrates that the proposed sequential design procedure for model predictive control guarantees the closed-loop system stability and guaranteed cost.

5. CONCLUSION

The paper addresses the problem of the sequential design of model predictive control. Sequential design consists of *N* steps, respective to prediction horizon *N*. Firstly, one-step ahead model predictive control is designed by either of two design approaches proposed in the paper. Repeating the one-step ahead design procedure to *N* steps, one obtains the results – model predictive control for a prediction horizon *N*. The proposed design procedures guarantee the stability of the closed-loop system and guaranteed cost corresponding to a minimization of cost function given for each step. The results of design procedure are the output feedback gain matrices for model predictive control and real plant control input u(t) for the given prediction and control horizon.

Acknowledgment

The work has been partially supported by the Slovak Scientific Grant Agency, Grant N.1/0544/09.

REFERENCES

Camacho E.F , and Bordons C. (2004). *Model predictive control*. Springer-Verlag, London.

Clarke D.W., Mohtadi C. (1989). Properties of generalized predictive control. *Automatica*, 25(6), 859-875.

Clarke D.W. and Scattolini (1991). Constrained Receding-horizon Predictive Control. *IEE Proceedings*, 138(4), 347-354.

Dermicioglu H., Clarke D.W. (1993). Generalized predictive control with end-point weighting. *IEE Proceedings Part D*, 140(4). 275- 282.

de Oliveira M.C., Camino J.F. , and Skelton R.E. (2000). A convexifying algorithm for the design of structured linear controllers. *In: Proc.39th IEEE Conference on Decision and Control*, Sydney, 2781-2786.

Haber R., Schmitz U., Bars R. (2002). Stabilizing predictive control with infinite control error weighting. *In: Conf. Process Control*, Kouty nad Desnou, Czech Republik, 1-11.

Kothare M.V., Balakrishnan V. and Morari M. (1996). Robust Constrained Model Predictive Control Using Linear Matrix Inequalities. *Automatica*, 32(10), 1361-1379.

Maciejowski J.M. (2002). *Predictive Control with Constraints*. Prentice Hall.

Mayne D.Q., Rawlings J.B., Rao C.V., Scokaert P.O.M. (2000). Contrained model predictive control: stability and optimality, *Automatica* 36, 789- 814.

Rawlings J. and Muske K. (1993). The stability of constrained Receding Horizon Control. *IEEE Trans. on Automatic Control*, 38, 1512- 1516.

Rossiter J.A (2003). *Model Based Predictive Control: A Practical Approach*, Control Series, CRC Press.

Rossiter J.A. and Kouvaritakis B. (1994). *Advances in model based Predictive Control, Chapter Advances in Generalized and Constrained Predictive Control,* Oxford University Press.

Comments – Remarks

Institute of Information Engineering, Automation, and Mathematics, FCFT STU in Bratislava

Parallel distributed Fuzzy Control of a Heat Exchanger

Vasičkaninová A., Bakošová M.

Department of Information Engineering and Process Control
Faculty of Chemical and Food Technology
Slovak University of Technology
Radlinského 9, 81237 Bratislava, Slovakia
{e-mail: anna.vasickaninova, monika.bakosova}@stuba.sk}

Abstract: Fuzzy controllers have found popularity in many practical situations. Many complex plants have been controlled very well using fuzzy controllers without any difficult analysis common in classical control design. Fuzzy controllers are general nonlinear ones and their benefits are well-known. The paper presents a fuzzy control based on parallel distributed fuzzy controllers for a heat exchanger. First, a Takagi-Sugeno fuzzy model is employed to represent a system. Each subcontroller is LQR designed and provides local optimal solutions. The simulation results are compared with classical PID control and illustrate the validity and applicability of the presented approach.

1. INTRODUCTION

Fuzzy controllers have found popularity in many practical situations. Many complex plants have been controlled very well using fuzzy controllers without any difficult analysis common in classical control design. Fuzzy controllers are general nonlinear ones and their benefits are well-known (Slotine and Li, 1991). In spite of these advantageous properties of fuzzy controllers, the main crisis of them was the absence of a formal method for proving the system's stability. However, after introducing the fuzzy plant modelling (Takagi and Sugeno, 1985), some methods for stable controller design have arisen.

The described approach is based on fuzzy modelling of a nonlinear plant as a sum of nonlinear-weighted linear subsystems (Wang and Tanaka, 1996). Following this approach, one can design a linear controller for each subsystem and satisfying some constraints expressible as linear matrix inequalities (LMIs), stability of the whole system can be proved (Lam *et al.*, 2001; Tanaka *et al.*, 1997; Volosencu 2008). The idea is similar to traditional gain scheduling method in which controlling gains change according to the state of the controlled system (Alata, 2001; Packard, 1994). The ability of converting linguistic descriptions into automatic control strategy makes it a practical and promising alternative to the classical control scheme for achieving control of complex nonlinear systems.

Many real systems can be represented by TS fuzzy models (Lagrat *et al.*, 2008; Takagi and Sugeno, 1985; Tanaka *et al.*, 1997). A TS fuzzy model approximates the system using simple models in each subspace obtained from the decomposition of the input space. The dynamic TS models are easily obtained by linearization of the nonlinear plant around different operating points. After the TS fuzzy models are obtained, linear control methodology can be used to design local state feedback controllers for each linear model. Aggregation of the fuzzy rules results in a generally nonlinear model (Wang and Tanaka, 1995).

Stability and optimality are the most important requirements for any control system (Lam *et al.*, 2001). Most of the existed works are based on Takagi–Sugeno type fuzzy model combined with parallel distribution compensation concept and applying Lyapunov's method to do stability analysis (Tanaka and Wang, 2001). Tanaka and co-workers reduced the stability analysis and control design problems to the linear matrix inequality (LMI) problems.

The state feedback gain design method was developed in (Chang *et al.*, 2001; Tanaka *et al.* 1997; Tanaka and Hori, 1999; Wang and Tanaka, 1996) and it is based on assigning a common positive definite matrix P in accordance with the Lyapunov stability concept. It is important to find a suitable P such that the stable feedback gains exist. Some theorems will be presented to find the suitable P. After assigning a suitable common positive definite matrix P, one can obtain the feedback gains for each rule of the TS type fuzzy system.

A heat exchanger is a device in which energy is transferred from one fluid to another across a solid surface. Heat exchanger analysis and design therefore involve both, convection and conduction. The heat exchangers are widely used in many industrial power generation units, chemical, petrochemical, and petroleum industries and they are robust units that work for wide ranges of pressures, flows and temperatures (Taborek, 1983).

In this paper, a stable nonlinear fuzzy controller based on parallel distributed fuzzy controllers is proposed. Each subcontroller is LQR designed and provides local optimal solution. The Takagi-Sugeno fuzzy model is employed to approximate the nonlinear model of the controlled plant. Based on the fuzzy model, a fuzzy controller is developed to guarantee not only the stability of fuzzy model and fuzzy control system for the heat exchanger but also control the transient behaviour of the system. The design procedure is conceptually simple and natural. Moreover, the stability analysis and control design problems are reduced to LMI problems. Therefore, they can be solved very efficiently in practice by convex programming techniques for LMIs. Simulation results shows that the proposed control approach is robust and exhibits a superior performance to that of established traditional control methods.

2 PROBLEM FORMULATION

Fuzzy modelling is a framework in which different modelling and identification methods are combined, providing a transparent interface with the designer or operator *and.* It is a flexible tool for nonlinear system modelling and control too. The rule-based character of fuzzy models allows for a model interpretation in a way that is similar to the one humans use to describe reality.

Using fuzzy systems it is possible to define very general nonlinearities. In order to be able to derive any analytical useful results it is necessary to constrain the classes of nonlinearities that one consider. The class of systems that has achieved most attention is linear and affine Takagi-Sugeno systems on state-space form. For these systems both stability and synthesis results are available based on Lyapunov theory.

Quadratic Lyapunov functions are wery powerful if they can be found. In many cases it is very difficult to find a common global Lyapunov function. The feasible solution is to use a piecewise quadratic Lyapunov function that is tailored to fit the cell partition of the system (Johansson and Rantzer, 1997). The search for piecewise quadratic Lyapunov function can also be formulated as an LMI-problem.

2.1 Fuzzy Plant Model

Consider a nonlinear controller. We can assume that the plant can be represented by a fuzzy plant model. Our goal is designing a nonlinear state feedback controller.

The continuous fuzzy dynamic model (Takagi and Sugeno, 1985) is described by fuzzy *if-then* rules. It can be seen as a combination of linguistic modelling and mathematical regression, in the sense that the antecedents describe fuzzy regions in the input space in which consequent functions are valid. The i^{th} rule is of the following form (Tanaka, K. and M. Sugeno, 1992).

Plant Rule *i:*

if $z_1(t)$ is M_1^i and ... and $z_s(t)$ is M_s^i then

$$\begin{aligned}\dot{x}(t) &= A_i x(t) + B_i u(t)\\ y(t) &= C_i x(t)\end{aligned} \quad i=1,.., N \qquad (1)$$

where $x(t)=[x_1(t),\ x_2(t),\ ...,\ x_n(t)]^T \in R^n$ is the state vector, $u(t)=[u_1(t),\ u_2(t),\ ...,\ u_m(t)]^T \in R^m$ is the control input, $y(t)=[y_1(t),\ y_2(t),\ ...,\ y_p(t)]^T \in R^p$ is the controlled output, M_j^i are fuzzy sets, $z(t)=[z_1(t),\ z_2(t),\ ...,\ z_s(t)]$ are the premise parameters, $A_i \in R^{nxn}$ is the state transition matrix, $B_i \in R^{nxm}$ is input matrix, $C_i \in R^{pxn}$ is output matrix. Let us use product as *t*- norm operator of the antecedent part of rules and the center of mass method for defuzzification. The final output of the fuzzy system is inferred as follows (Tanaka and Sugeno, 1992):

$$\dot{x}(t) = \frac{\sum_{i=1}^{N} \mu_i(z(t))(A_i x(t) + B_i u(t))}{\sum_{i=1}^{N} \mu_i(z(t))} = \qquad (2)$$

$$= \sum_{i=1}^{N} h_i(z(t))(A_i x(t) + B_i u(t))$$

$$y(t) = \frac{\sum_{i=1}^{N} \mu_i(z(t))(C_i x(t))}{\sum_{i=1}^{N} \mu_i(z(t))} = \sum_{i=1}^{N} h_i(z(t))(C_i x(t)) \qquad (3)$$

where

$$h_i(z(t)) = \frac{\mu_i z(t)}{\sum_{i=1}^{N} \mu_i z(t)} \qquad (4)$$

$$\mu_i(z(t)) = \prod_{j=1}^{s} M_i^j z_j(t) \qquad (5)$$

$$\sum_{i=1}^{N} h_i(z(t)) = 1 \qquad (6)$$

2.2 Quadratic Stability

After defining the model, the conditions are found under which the system is stable.

Theorem 1. The continuous uncontrolled ($u=0$) fuzzy system of (1) - (3) is globally quadratically stable if there exists a common positive definite matrix $P=P^T$ such that

$$A_i^T P + PA_i < 0, \quad i = 1,..,N \qquad (7)$$

This is equivalent to saying that one must find a single function $V(x) = x^T Px$ as a candidate for Lyapunov function.

Finding a common P can be considered as linear matrix inequality (LMI) problem. Matlab LMI toolbox presents simple appliance for solving this problem (Gahinet *et al.*, 1995).

2.3 Parallel distributed compensation

Having TS plant model, it can be used parallel distributed compensation control defined as follows:

Control Rule *j:*

if *z1(t)* is M_1^i and ... and *zs(t)* is M_s^i then

$$u(t) = -K_j x(t)\ j=1\ ,.., N \tag{8}$$

Hence, the fuzzy controller is given

$$u(t) = -\sum_{j=1}^{N} h_j(z(t))K_j x(t) \tag{9}$$

in which *Kj* are state feedback gains. We can see it as local gains of gain scheduling design which overall control signal is made from combining each local control signal with different weights according to the closeness to the each rule's region.

The closed loop system can be expressed by combining (2) and (7) as following system

$$\begin{aligned}\dot{x}(t) &= \sum_{i=1}^{N}\sum_{j=1}^{N} h_i(z(t))h_j(z(t))\left[A_j - B_i K_j\right]x(t) = \\ &= \sum_{j=1}^{N} h_i^2(z(t))G_{ij}x(t) + \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{N} h_i(z(t))h_j(z(t))\left[\frac{G_{ij}+G_{ji}}{2}\right]x(t)\end{aligned} \tag{10}$$

where $G_{ij} = A_i - B_i K_j$.

It is easy to obtain the following result using Theorem 1: The fuzzy system (2), (3) with fuzzy control of (9) is globally stable if there exists $P = P^T$ such that

$$\begin{gathered}\left(\frac{G_{ij}+G_{ji}}{2}\right)^T P + P\left(\frac{G_{ij}+G_{ji}}{2}\right) < 0, \\ i = 1,.., N \\ j = 1,.., N\end{gathered} \tag{11}$$

2.4 Locally optimal design

Since the local fuzzy system (i.e., fuzzy subsystem) is linear, its quadratic optimization problem is the same as the general linear quadratic (LQ) issue (Alata and Demirli, 2001; Burl 1999; Kim and Rhee, 2001; Slotine and Li, 1991). Therefore, solving the optimal control problem for fuzzy subsystem can be achieved by simply generalizing the classical theorem from the deterministic case to fuzzy case.

$$J(x(t), u(t)) = \int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right)dt \tag{12}$$

where *Q* is a real symmetric positive semidefinite weighting matrix and *R* is a real symmetric positive definite weighting matrix. Solution of the optimization problem, i. e. minimization of *J* for any $\boldsymbol{x_0}$ satisfies the feedback control law

$$u(t) = -Kx(t) \tag{13}$$

where $K = R^{-1}B^T P$.

The optimal gain is *K* in which *P* is a symmetric positive semidefinite solution of the matrix Ricatti equation (Botan and Ostafia, 2008)

$$PA + A^T P + Q - PBR^{-1}B^T P = 0\,(13) \tag{14}$$

If the matrix (A - $BR^{-1}B^T$ P) is stable, i.e. (A - BK) is stable, the closed-loop system is stable.

3 SIMULATIONS AND RESULTS

3.1 Shell heat exchangers

Consider two heat exchangers shown in Fig. 1.

The measured and controlled output is temperature from second exchanger. The control objective is to keep the temperature of the output stream close to a desired value 353 K. The control signal is input volumetric flow rate of the heated liquid. Assume ideal liquid mixing and zero heat losses. We neglect accumulation ability of exchangers walls. Hold-ups of exchanger as well as flow rates and liquid specific heat capacity are constant.

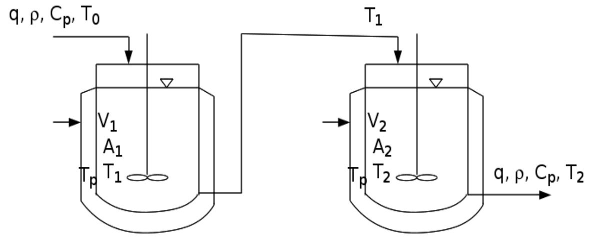


Fig.1. Two shell heat exchangers in series

Under these assumptions the mathematical model of the exchangers is given as

$$\frac{dT_1}{dt} = \frac{q}{V_1}\left(T_0 - T_1\right) + \frac{A_1 k}{V_1 \rho C_p}\left(T_p - T_1\right) \tag{15}$$

$$\frac{dT_2}{dt} = \frac{q}{V_2}\left(T_1 - T_2\right) + \frac{A_2 k}{V_2 \rho C_p}\left(T_p - T_2\right) \tag{16}$$

where T_1 is temperature in the first exchanger, T_2 is temperature in the second exchanger, T_0 is liquid temperature in the inlet stream of the first tank, q is volumetric flow rate of liquid, ρ is liquid density, V_1, V_2 are liquid volumes, A_1, A_2 are heat transfer areas, k is heat transfer coefficient, C_p is specific heat capacity. The superscript *s* denotes the steady-state values in the main operating point.

Parameters and inputs of the exchangers are enumerated in Table 1.

Table 1. Parameters and inputs of heat exchangers

Variable	Unit	Value
q	m^3min^{-1}	1
V_1	m^3	5
V_2	m^3	5
C_p	$kJ\ kg^{-1}K^{-1}$	3.84
A_1	m^2	16

Variable	Unit	Value
A_2	m^2	16
k	$kJ\ m^{-2}min^{-1}K^{-1}$	72
ρ	$kg\ m^{-3}$	900
T_0^s	K	293
T_p^s	K	373
T_1^s	K	313
T_2^s	K	328

3.2 Takagi-Sugeno fuzzy model and LQ control design

The system was approximated by nine fuzzy models

$$\text{if } x \text{ is } M_1^i \text{ and } u \text{ is } M_2^i \text{ then}$$
$$\dot{x}(t) = A_i x(t) + B_i u(t) \quad i=1,\dots,9 \qquad (17)$$

The bell curve membership functions for the premise variables *x* and *u* in each rule are adopted:

$$f(x;[a,b,c]) = \left(\left(1+\left|\frac{(x-c)}{a}\right|\right)^{2b}\right)^{-1} \qquad (18)$$

The parameters *a*, *b* and *c* for bell shaped membership functions are listed in the Table 2 and membership functions are shown in Figures 2, 3. The consequent parameters are given in Table 3 and the resulting plot of the output surface of a described fuzzy inference system is presented in Figure 4.

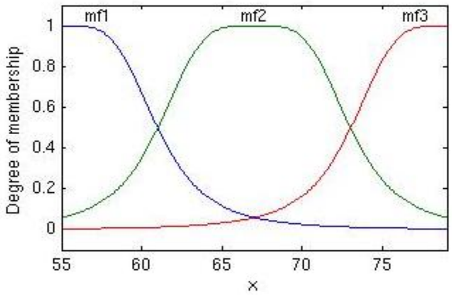


Fig. 2. Bell curve membership functions for *x*

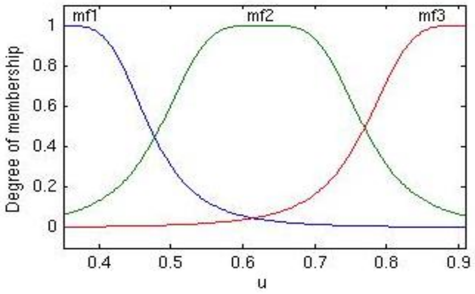


Fig. 3. Bell curve membership functions for *u*

Table 2. Bell curve membership functions parameters

	x			u	
a_i	b_i	c_i	a_i	b_i	c_i
6	2	55	0.13	2	0.34
6	2	67	0.14	2	0.62
6	2	79	0.14	2	0.91

Table 3. Consequent parameters

A_i	B_i
0.73	-13.41
0.27	-4.94
-0.33	9.44
0.01	0.22
0.13	-3.04
-0.37	11.11
-0.25	0.72
2.33	-74.17
2.05	-59.04

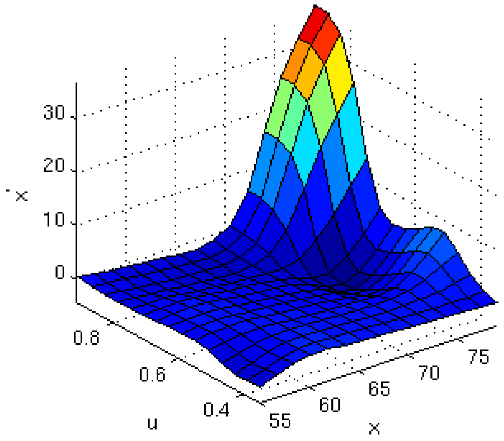


Fig. 4. Output surface of a fuzzy inference system $x' = f(x,u)$

After obtaining A_i, B_i, gains K_j were calculated of each subsystem using LQR design and then tested for stability of the system.

The problem was solved using LMI optimization toolbox in Matlab software package.

The results for different performance measures are compared in Table 4.

Table 4. Fuzzy LQR controllers performance comparison

performance measure	set-point tracking	disturbance rejection
K=-0.1803		
Q=1*I(2,2)	iae = 0.41 e3	iae = 0.46 e3
R=1	ise = 2.67 e3	ise = 2.54 e3
K=-1.1858		
Q=100*I(2,2)	iae = 0.19 e3	iae = 0.18 e3
R=1	ise = 1.91 e3	ise = 1.56 e3
K=-0.1803		
Q=40*I(2,2)	iae = 1.41 e3	iae = 0.46 e3
R=40	ise = 2.67 e3	ise = 2.54 e3
K=-3.5888		
Q=100*I(2,2)	iae = 0.16 e3	iae = 0.14 e3
R=0.1	ise = 1.88 e3	ise = 1.54 e3

3.3 PID control

For feedback controller tuning, the approximate model of a system with complex dynamics can have the form of a first-order-plus-time-delay transfer function (19). The process is characterised by a steady-state gain *K*, an effective time constant *T* and an effective time delay *D*.

$$G_P(s) = \frac{K}{Ts+1} e^{-Ds} \tag{19}$$

The transfer function describing the controlled heat exchangers was identified from step response data in the form (19) with parameters: *K* = -38.57, *T* = 11.3 min, *D* = 2 min. These parameters were used for feedback controller tuning. The feedback PID controllers were tuned by various methods (Ogunnaike and Ray,1994). Two controllers were used for comparison: PID controller (20) tuned using Rivera-Morari method with parameters $K_C = -0.1063$, $T_I = 12.3$, $T_D = 0.91$ and PID controller tuned using Ziegler-Nichols method with parameters $K_C = -0.17$, $T_I = 4$, $T_D = 1$. The transfer function of the used PID controller is following

$$G_C(s) = K_C\left(1 + \frac{1}{T_I s} + T_D s\right) \tag{20}$$

The step changes of the reference y_r were generated and the fuzzy LQR and PID controllers were compared. Figure 5 presents the comparison of the simulation results obtained by fuzzy LQR controller and PID controllers tuned using Rivera-Morari and Ziegler-Nichols methods. Figure 6 presents the comparison control inputs generated by above mentioned controllers. Figure 7 presents the simulation results of the fuzzy LQR and PID control of the heat exchanger in the case when disturbances affect the controlled process. Disturbances were represented by temperature changes from 373 K to 353 K at t=25 min, from 353 K to 383 K at t=75 min and from 383 K to 368 K at t=125 min. The comparison of the controllers output is shown in Figure 8.

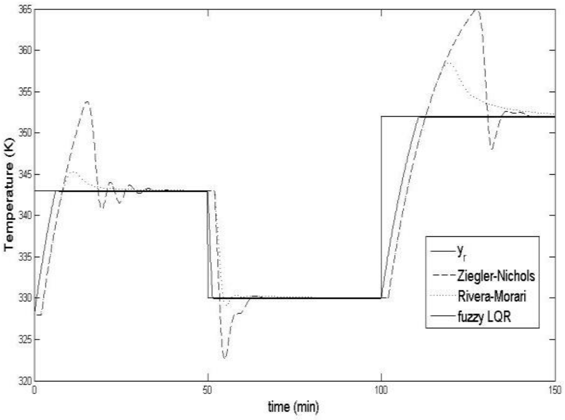


Fig. 5. Comparison of the temperature of the output stream from second heat exchanger: PID controllers: Rivera-Morari (......), Ziegler-Nichols (- - -), fuzzy LQR (__), reference trajectory (__)

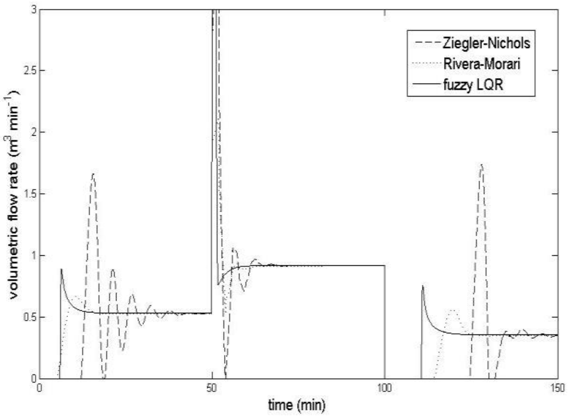


Fig. 6. Comparison of the control inputs

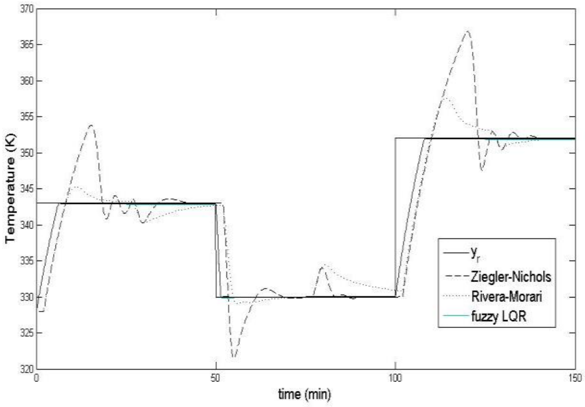


Fig. 7. Control responses in the presence of disturbances: PID controllers: Rivera-Morari (......), Ziegler-Nichols (- - -), fuzzy LQR (__), reference trajectory (__)

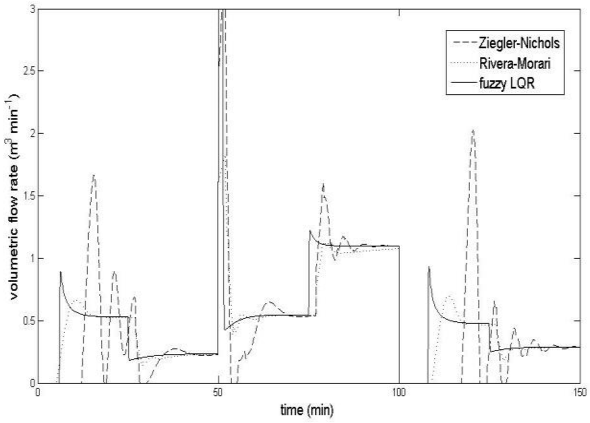


Fig. 8. Comparison of control inputs in the presence of disturbances

The comparison of the fuzzy LQR controller with PID controllers was made using *iae* and *ise* criteria described as follows:

$$iae = \int_0^T \left| e \right| dt \qquad (21)$$

$$ise = \int_0^T e^2 dt \qquad (22)$$

The *iae* and *ise* values are given in Table 5.

Table 5. Comparison of the simulation results by integrated absolute error *iae* and integrated square error *ise*

control method	set-point tracking	disturbance rejection
fuzzy LQR: *K =-3.5888*	*iae = 0.16 e3* *ise = 1.88 e3*	*iae = 0.14 e3* *ise = 1.54 e3*
PID (Rivera-Morari)	*iae = 0.36 e3* *ise = 4.09 e3*	*iae = 0.40 e3* *ise = 3.64 e3*
PID (Ziegler-Nichols)	*iae = 0.52 e3* *ise = 5.79 e3*	*iae = 0.50 e3* *ise = 5.48 e3*

Used fuzzy is simple, and it offers the smallest values *iae* and *ise*. The disadvantage of the fuzzy LQR controllers is, that using these controllers can lead to nonzero steady-state errors, but without overshoots practically. In the case of the heat exchanger control in the presence of disturbances, the control responses with fuzzy LQR controllers do not show any overshoots and undershoots.

Comparison of the LQR simulation results with classical PID control demonstrates the effectiveness and superiority of the proposed approach.

ACKNOWLEDGMENTS

This project is supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism and also co-financed from the state budget of the Slovak Republic.

4 REFERENCES

Alata, M. and K. Demirli (2001). Fuzzy control and gain scheduling-case study: robust stabilization of an inverted pendulum, *Joint 9th IFSA World Cong. and 20th NAFIPS Int. Conf.*

Burl, J.B. (1999). Linear Optimal Control: *H_2 and H_∞ methods*, Addison Wesley Longman, Inc.

Gahinet, P., A. Nemirovski, A. Laub, and M. Chilali (1994). *LMI Control Toolbox*, The Math. Works Inc.

Chang, W.J., C.C. Sun and C.C. Fuh (2001). Continuous output feedback fuzzy controller design with a specified common controllability Gramian. *Int. J. Fuzzy Syst*. **3**, (1), pp. 356–363.

Johansson, M. and A. Rantzer (1997). Computation of piecewise quadratic Lyapunov functions for hybrid systems. *European Control Conference, ECC97*.

Kim, D. and S. Rhee (2001). Design of an optimal fuzzy logic controller using response surface methodology, *IEEE Trans. Fuzzy Syst.*, vol. 9. no. 3.

Lagrat, I., A. El Ougli and I. Boumhidi (2008). Optimal Adaptive Fuzzy Control for a Class of Unknown Nonlinear Systems. *Wseas, Transactions on Systems and Control*, vol. 3, pp. 89 - 98

Lam, H.K., F.H.F. Leung and P.K.S. Tam (2001). Nonlinear state feedback controller for nonlinear systems: stability analysis and design based on fuzzy plant model, *IEEE Trans. Fuzzy Syst., vol. 9.*, no. 4, pp. 657-661.

Ogunnaike, B. A. and W.H. Ray (1994). *Process Dynamics, Modelling, and Control*. 536-541. Oxford University Press. New York. 1259. ISBN 0-19-509119-1.

Packard, A. (1994). Gain scheduling via linear fractional transformations, *Syst. Contr. Lett.*, vol. 22, pp. 79-92.

Slotine, J.-J. E. and W. Li (1991). *Applied Nonlinear Control*. Englewood Cliffs, New Jersey: Prentice-Hall.

Taborek, J. (1983). Shell-and-tube exchangers: Single-phase flow, in: E. U. Schlunder (Ed.), *Heat exchangers design handbook*, Vol. 3, Section 3.3, Hemisphere Publishing Corp.

Takagi, K. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Syst. Man. Cybern*. 15, 116-132.

Tanaka, K. and M. Sugeno (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets Syst.*, vol. 45, pp. 135-156.

Tanaka, K., T. Ikeda and H.O. Wang (1996). Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stability, H_∞ control theory, and linear matrix inequalities. *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 1-13.

Tanaka, K. and H.0. Wang (2001). Fuzzy Control Systems Design and Analysis; A linear Matrix Inequality Approach. New York, (John Wiley & Sons, Inc).

Volosencu, C. (2008). Stabilization of Fuzzy Control Systems. *Wseas, Transactions on Systems and Control*, vol. 3, pp. 879 – 896.

Wang, H.O., K. Tanaka and M. Griffin (1995). Parallel distributed compensation of nonlinear systems by Takagi and Sugeno's fuzzy model. *Proc. FUZZ - IEEE' 95*. pp. 531–538.

Wang, H.O., K. Tanaka and M. Griffin (1996). An approach to fuzzy control of nonlinear system: Stability and design issues. *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 14 – 23.

Wang, H.O. and K. Tanaka (1996). An LMI-based stable fuzzy control of nonlinear systems and its application to control of chaos. *Proc. FUZZ - IEEE' 96*, vol. 2, pp. 1433–1438.

Comments – Remarks

Control Design of Chemical Reactor with Disturbances by Neuro-fuzzy Methods

L. Blahová, J. Dvoran

**Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Radlinského 9, 812 37 Bratislava (e-mail: lenka.blahova@stuba.sk, jan.dvoran@stuba.sk)*

Abstract: This paper present the control design via the combination of the neural predictive controller and neuro-fuzzy controller type of ANFIS. The ANFIS works in parallel with the predictive controller. The performance of our proposal is demonstrate on the Continuous Stirred-Tank Reactor control problem with disturbances. Simulation result demonstrate the effectiveness and robustness of the proposed approach.

1. INTRODUCTION

The aim of process control is to achieve the target value of given variable. This is mainly the task of the properly designed controller. The controller should also provide some flexibility in the case of unexpected failure or change of condition, etc.

This paper describes the above mentioned combination of two methods of intelligent system controlling. By the parallel connection of predictive and neural-fuzzy controller, we aimed to obtain better results of the reference variable in terms of lowering its overshooting and reducing the control time. The designed system with two connected controllers was tested using the chemical reactor in the nominal state and in the perturbed state (disturbances were in the input concentrate of substance A, input temperature of reaction mixture, input temperature of coolant and flow rate of reaction mixture). The chemical reactor introduces one of the complex types of the chemical-technological process where full sailed control without expressive overshooting is needed.

2. PREDICTIVE CONTROL

MBPC (Model-Based Predictive Control) is a name of a several different control techniques (Vasičkaninová et al., 2008). All are associated with the same idea. The prediction is based on the model of the process (Fig.1).

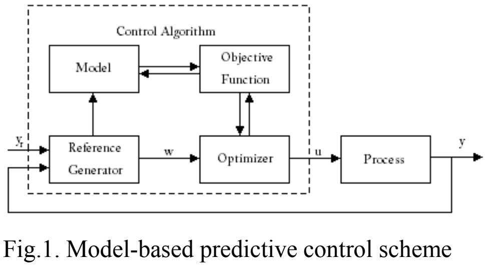


Fig.1. Model-based predictive control scheme

The controller uses a neural network model to predict future plant responses to potential control signals. An optimization algorithm then computes the control signals that optimize future plant performance. The neural network plant model is trained offline, in bath form, using some of the training algorithms.

The controller, however, requires a significant amount of online computation, because an optimization algorithm is performed at each sample time to compute the optimal control input. The model predictive control method is based on the receding horizon technique. The neural network model predicts the plant response over a specified time horizon. The predictions are used by a numerical optimization program to determine the control signal that minimizes the following performance criteria over the specified horizon.

$$J(t,u(k)) = \sum_{i=N_1}^{N_2} (y_m(t+i) - y_r(t+i))^2 + \lambda \sum_{i=1}^{N_u} (\Delta u(t+i-1))^2 \qquad (1)$$

where N_1, N_2 and N_u define the horizons over the tracking error and the control increments are evaluated. The $_u$' variable is the tentative control signal, y_r is the desired response and y_m is the network model response. The λ value determines the contribution of the sum of squares of the control increments and the performance index.

The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of u that minimize J, and then the optimal u is input to the plant.

Equation (1) is used in combination with input and output constraints:

$$\begin{aligned} u_{\min} &\le u \le u_{\max} \\ \Delta u_{min} &\le \Delta u \le \Delta u_{max} \\ y_{\min} &\le y \le y_{\max} \\ \Delta y_{min} &\le \Delta y \le \Delta y_{max} \end{aligned} \qquad (2)$$

3. NEURO-FUZZY CONTROLLER

The neural predictive controller can be extended with neuro-fuzzy controller, connected in parallel (Fig.2).

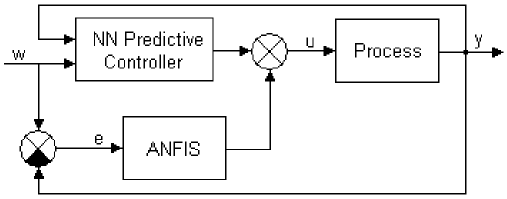


Fig.2. Neuro-fuzzy control scheme

Neuro-fuzzy systems, which combine neural networks and fuzzy logic, have recently gained a lot of interest in research and application. A specific approach in neuro-fuzzy development is the ANFIS (Adaptive Network-based Fuzzy Inference System) (Agil et al., 2007). ANFIS uses a feed forward network to search for fuzzy decision rules that perform well on a given task. Using a given input-output data set, ANFIS creates a Fuzzy Inference System for which membership function parameters are adjusted using a combination of a back propagation and least square method. The ANFIS architecture of the first-order Takagi-Sugeno inference system is shown in Fig.3.

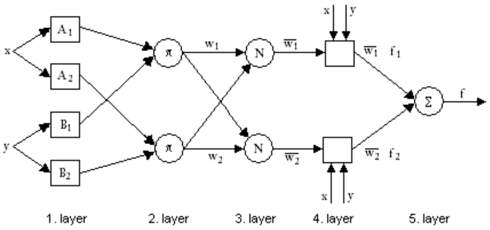


Fig.3. System architecture ANFIS

4. EXPERIMENTAL

4.1 Continuous Stirred-Tank Reactor (CSTR)

Consider CSTR (Mikleš et al., 2007) with first-order irreversible parallel reaction according to the scheme (3).

$$A \xrightarrow{k_1} B \qquad A \xrightarrow{k_2} C \tag{3}$$

The simplified dynamical mathematical model of CSTR is:

$$\frac{dc_A}{dt} = \frac{q}{V} c_{Av} - \frac{q}{V} c_A - k_1 c_A - k_2 c_A \tag{4}$$

$$\frac{dc_B}{dt} = \frac{q}{V} c_{Bv} - \frac{q}{V} c_B + k_1 c_A \tag{5}$$

$$\frac{dc_C}{dt} = \frac{q}{V} c_{Cv} - \frac{q}{V} c_C + k_2 c_A \tag{6}$$

$$\frac{d\vartheta}{dt} = \frac{q}{V}\vartheta_v - \frac{q}{V}\vartheta - \frac{Ak}{Vc_p\rho}\left[\vartheta - \vartheta_c\right] + \frac{Q_r}{Vc_p\rho} \tag{7}$$

$$\frac{d\vartheta_c}{dt} = \frac{q}{V_c}\vartheta_{vc} - \frac{q}{V_c}\vartheta_c + \frac{Ak}{V_c\rho_c c_{pc}}\left[\vartheta - \vartheta_c\right] \tag{8}$$

The rate of reaction is a strong function of temperature:

$$k_i = k_{i\infty} e^{\frac{-E_i}{R\vartheta}} \tag{9}$$

For reaction heat gives:

$$Q_r = k_1 c_A V\left(-\Delta_r H_1\right) + k_2 c_A V\left(-\Delta_r H_2\right) \tag{10}$$

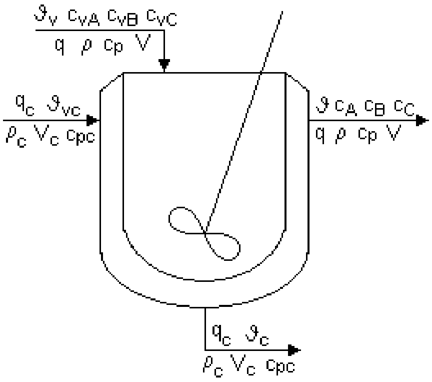


Fig.4. Signification scheme of chemical reactor

Temperature of reaction mixture ϑ is controlled variable and volume flow rate of coolant q_c is input variable. The process state variables are molar concentration of A, B and C (c_A, c_B and c_C) and temperatures of reaction mixture ϑ and coolant ϑ_c. The model parameters are summarized in Table 1.

Table 1. Parameters of the chemical reactor

Variable	Unit	Value
c_{vA}	kmol m^{-3}	4,22
c_{vB}	kmol m^{-3}	0
c_{vC}	kmol m^{-3}	0
Q	$m^3 min^{-1}$	0,015
ϑ_v	K	328
P	kg m^{-3}	1020
c_p	kJ $kg^{-1}K^{-1}$	4,02
V	m^3	0,23
q_{vc}	$m^3 min^{-1}$	0,004
ϑ_{vc}	K	298
ρ_c	kg m^{-3}	998
c_{pc}	kJ $kg^{-1}K^{-1}$	4,182
V_c	m^3	0,21

A	m^2	1,51
K	kJ $min^{-1}m^{-2}K^{-1}$	42,8
E_1/R	K	9850
ΔrH_1	kJ $kmol^{-1}$	$-8{,}6.10^4$
$k_{1\infty}$	min^{-1}	$1{,}55.10^{11}$
E_2/R	K	22019
ΔrH_2	kJ $kmol^{-1}$	$-1{,}82.10^4$
$k_{2\infty}$	min^{-1}	$4{,}55.10^{25}$

4.2 Process Control in the Nominal State

Firstly, CSTR was simulated with neural predictive controller (NNPC). To set this controller neural network process model was needed. Neural network model of CSTR was trained offline based on nonlinear process input and output data by Levenberg-Marquardt back propagation method. When optimization parameters were adjusted, CSTR was further controlled by NNPC controller.

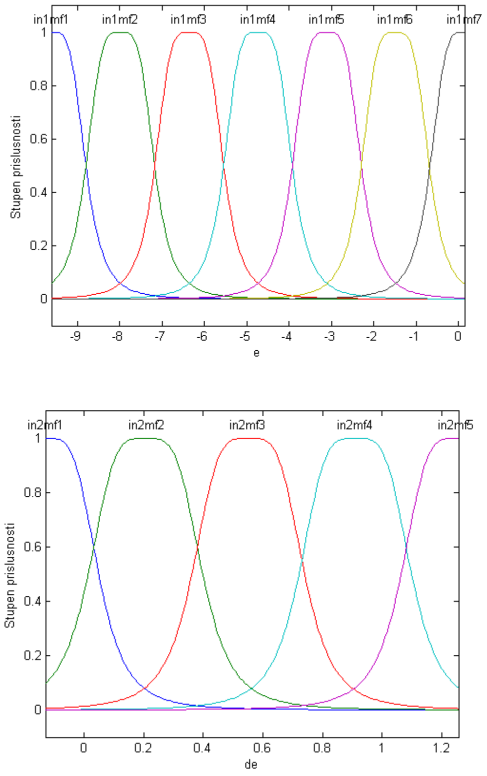


Fig.5. Membership functions for input variables *e* and *de*

Secondly, CSTR was controlled with neuro-fuzzy controller (NFC) formed from neural predictive controller and ANFIS controller. ANFIS was trained by PID controller. PID parameters were designed by Smith-Murrill method in five training periods. ANFIS have two inputs: set-point error *e* and derivation of set-point error *de*. Twelve membership function bell shape were chosen for ANFIS input: seven for variable *e* and five for variable *de* (Fig.5).

The neural predictive and the neuro-fuzzy controller were tested in MATLAB/SIMULINK ® environment using neural network toolbox and fuzzy logic toolbox. This experiment was designed to compare a neural predictive controller with neuro-fuzzy controller performance while controlling a nominal process.

In Fig.6, set-point changes of the desired temperature profile were tracked with satisfactory results in both considered cases. However, it can be seen, that the controlled variable (ϑ) profiles exhibit differences for both controllers compared. The neuro-fuzzy controller had more fainting performance that the neural predictive controller.

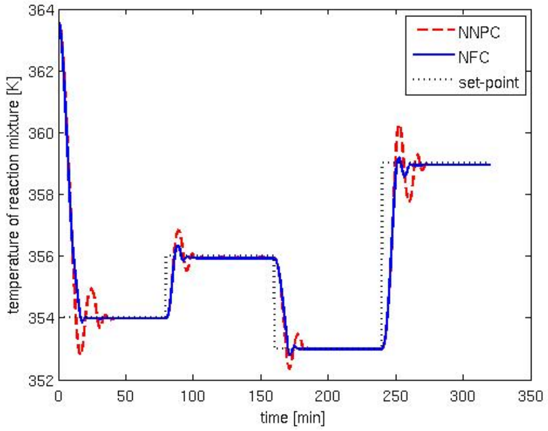


Fig.6. Comparison of NNPC and NFC performance for nominal plant

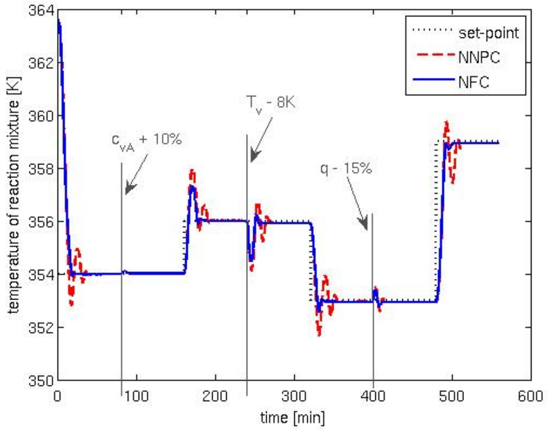


Fig.7. Comparison of NNPC and NFC performance for perturbed state – step change of c_{vA} + 10 % from the nominal value, ϑ_v – 8K from the nominal value and q – 15% from the nominal value.

4.3 Process Control in the Perturbed State

Besides the good regulatory performance tested above, tracking abilities of controllers proposed in the presence of disturbances is of utmost importance. Disturbances were applied during the control curse and they were set as step change of input concentration of substance A (c_{vA}), input temperature of reaction mixture (ϑ_v), input temperature of coolant (ϑ_{cv}) and flow rate of reaction mixture (q). Input concentration of substance A (c_{vA}) was change in range ± 10% from the nominal value. Input temperature of reaction mixture (ϑ_v) was change in range ± 8K from the nominal value. Input temperature of coolant (ϑ_{cv}) was change in range ± 8K from the nominal value. Flow rate of reaction mixture was change in range ± 15% from the nominal value.

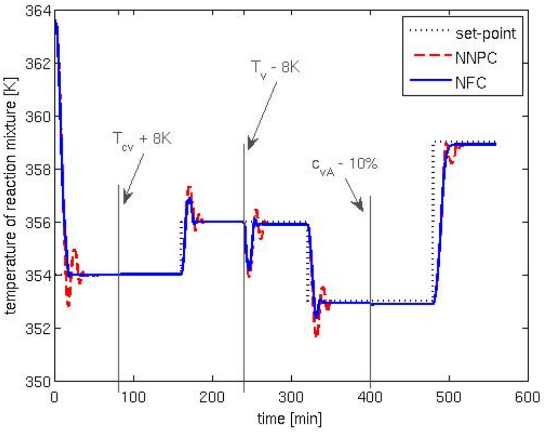


Fig.8. Comparison of NNPC and NFC performance for perturbed state – step change of ϑ_{cv} + 8K from the nominal value, ϑ_v – 8K from the nominal value and c_{vA} - 10 % from the nominal value.

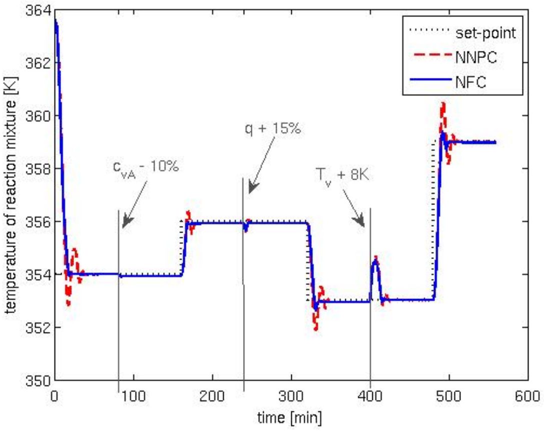


Fig.9. Comparison of NNPC and NFC performance for perturbed state – step change of c_{vA} - 10 % from the nominal value, q + 15% from the nominal value and ϑ_v +8K from the nominal value.

A comparison of the neural predictive controller and the neuro-fuzzy controller performance tested in the presence of process parameter perturbation is demonstrate in Fig.7, 8 and 9 (the arrows are to show the time instants when disturbances were applied).

5. CONCLUSION

In this paper, we present intelligent control system of a continuous stirred-tank reactor. This intelligent control system is composed from two individual controllers: neural predictive controller and ANFIS controller.

The main goal of the resulting control system was to enhance a profile of temperature of reaction mixture in the CSTR by manipulating the volume flow rate of coolant. Experimental results obtained demonstrated the usefulness and robustness of the proposed control system, and general advantages of the innovative technique in control application.

Acknowledgments

Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/0537/10, 1/0095/11, and the Slovak Research and Development Agency under the project APVV-0029-07.

REFERENCES

Agil, M., I. Kita, A. Yano and S. Nishiyama (2007). Analysis and prediction of flow from local source in a river basin using a Neuro-fuzzy modeling tool. In: *Jurnal of Enviromental Management*, 85, 215 – 223.

Mikleš, J. and M. Fikar (2007). *Process Modelling, Identification and Control.* New York: Springer, Berlin Heidelberg.

Vasičkaninová, A., M. Bakošová, A. Mészáros and J. Závacká (2008). Model-based predictive control of a chemical reactor. In: *18th International Congress of Chemical and Process Engineering*, Orgit s.r.o., Praha, Czech Republic, 0623-1 – 0623-6.

Comments – Remarks

Explicit Minimum-Time Controllers for Fuzzy Systems

M. Kvasnica [*], **M. Herceg** [**], **Ľ. Čirka** [*], and **M. Fikar** [*]

[*] *Institute of Information Engineering, Automation, and Mathematics,
Slovak University of Technology in Bratislava,
81237 Bratislava, Slovakia
e-mail : {michal.kvasnica,lubos.cirka,miroslav.fikar}@stuba.sk*
[**] *Automatic Control Laboratory, ETH Zurich, 8006 Zurich,
Switzerland, e-mail : herceg@control.ee.ethz.ch*

Abstract: : The paper deals with synthesis of MPC feedback laws for Takagi-Sugeno fuzzy systems. The procedure is based on constructing time-optimal controllers for the class of piecewise affine (PWA) systems with parametric uncertainties. It is shown that the fuzzy system can be embed into the PWA framework where the membership activation coefficients are represented as parametric uncertainties. A robust MPC controller is then derived which is able to utilize on-line measurements of the unknown parameters to further optimize for performance.

Keywords: Model Predictive Control, real-time control, parametric optimization

1. INTRODUCTION

Takagi-Sugeno fuzzy systems represent a popular modeling framework due to their ability to act as universal approximators of general nonlinear systems (Tanaka and Wang, 2002). Following the original contribution by (Takagi and Sugeno, 1985), fuzzy systems have garnered increased attention by the control community (Feng, 2006). This was mainly to their ability to model complex nonlinear behavior in a human-friendly way. MPC is one of the frequently used methodologies for control of fuzzy systems (Mollov et al., 2004; Vasičkaninová and Bakošová, 2007). The approach is based on optimizing the predicted process behavior while taking constraints into account. However, most available MPC techniques for fuzzy systems are based on solving nonlinear optimization problems on-line, which has implications on the minimal admissible sampling rate of the closed-loop system. Issues with solving the MPC problem to global optimality could also arise.

Therefore in this work we propose to over-approximate TS models by PWA models, for which rigorous globally optimal control schemes can be synthesized using the MPC framework, see e.g. (Bemporad et al., 2002; Borrelli, 2003; Grieder et al., 2005). In addition, as shown in the referenced works, the MPC problem for PWA models can be solved off-line parametrically, which leads the control law in a form of a look-up table, which allows for very fast on-line implementation of MPC-based controllers. Similarly to TS systems, PWA systems (Sontag, 1981) can also be efficiently employed to describe the dynamical behavior of nonlinear systems by utilizing the concept of multiple linearization. Although PWA systems are still nonlinear due to the presence of "IF-THEN" switching rules, the underlying piecewise linearity allows for simpler control synthesis compared to full nonlinear setups. Motivated by these upsides and following the ideas of (Johansson et al., 1999), this paper is aimed at establishing a bridge between fuzzy TS models and the PWA modeling framework.

In this paper, we show how TS models with trapezoidal membership functions can be embedded into the PWA framework. The influence of the membership functions on state evolution is captured by means of an unknown (at the time of the synthesis of the control law), but bounded variation of the system matrices. The control synthesis is based on solving an MPC problem with a minimum-time objective. We show that if the value of the variation is measured on-line, the evolution of the PWA model mirrors the one of the TS model. We also illustrate how the minimum-time problem can be solved parametrically such that the control law is obtained in the form of a look-up table, allowing fast implementation and easy analysis of the closed-loop system. The main advantage of the proposed method is that satisfaction of input and state constraints, as well as time-optimal performance is achieved by construction.

2. TAKAGI-SUGENO FUZZY SYSTEMS

We consider the class of discrete-time Takagi-Sugeno models described by fuzzy "IF ... THEN" rules where the dynamical behavior is driven by an affine state-space dynamics. Generally, the ith TS rule can be expressed as

$$\begin{aligned}&\text{IF } x_{1,k} \text{ is } \mu_{i1} \text{ and } \dots\ x_{n,k} \text{ is } \mu_{in} \\ &\text{THEN } x_{k+1} = A_i x_k + B_i u_k + f_i\end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the input vector, and μ_{ij} are input fuzzy sets for $i = 1, \dots, r$ rules. In addition, $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times m}$, $f_i \in \mathbb{R}^n$ are the matrices which are used to obtain the successor state x_{k+1} with $k \geq 0$ denoting the sampling instance. The dynamics of the aggregated system can be written in a more compact way as

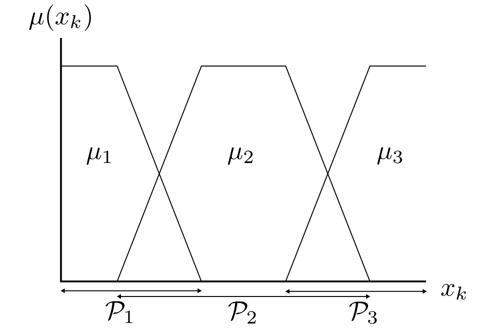


Fig. 1. Fuzzy membership functions.

$$x_{k+1} = \frac{\sum_{i=1}^{r} \left(w_i(x_k)(A_i x_k + B_i u_k + f_i)\right)}{\sum_{i=1}^{r} w_i(x_k)} \tag{2}$$

with

$$w_i(x_k) = \prod_{j=1}^{n} \mu_{ij}(x_{j,k}), \tag{3}$$

where the membership function $\mu_{ij}(x_{j,k})$ measures the activation of the fuzzy set j in the rule i. Using the definition

$$\alpha_i(x_k) = \frac{w_i(x_k)}{\sum_{i=1}^{r} w_i(x_k)}, \tag{4}$$

the overal system model can be described by

$$x_{k+1} = \sum_{i=1}^{r} \alpha_i(x_k)(A_i x_k + B_i u_k + f_i). \tag{5}$$

Each membership function μ_{ij} is defined over a region of the state space which, if the function is defined by linear hyperplanes, can be described as a convex polytope

$$\mathcal{P}_i = \{x_k \in \mathbb{R}^{n_x} \mid w_i(x_k) > 0\}. \tag{6}$$

Depending on the shape of the membership functions, all such polytopes $\mathcal{P}_i$ can either be disjoint or they can overlap. From the aggregation rule (2) it is clear that the interpolation of different dynamics occurs only in the overlapping parts, as illustrated in Fig 1. In the next section we show how to transform a given Takagi-Sugeno model into a PWA form with parametric uncertainties, which allows for synthesis of MPC feedback laws in explicit form. The control synthesis is based on the following two assumptions.

Assumption 2.1. The membership functions in (3) are trapezoidal functions, i.e.

$$\sum_i w_i(x_k) = 1, \quad \forall i \in [1, \dots, r] \tag{7}$$

Assumption 2.2. In the TS model (1), the matrices B_i and f_i are constant $\forall i \in [1, \dots, r]$, i.e. (5) can be written as

$$x_{k+1} = \Big(\sum_i^{r} \alpha_i(x_k) A_i\Big) x_k + B u_k + f \tag{8}$$

Although these assumptions may be restrictive in practice, they are vital for performing a control synthesis which features three strong points: constraint satisfaction, time-optimal performance and cheap implementation of the control law.

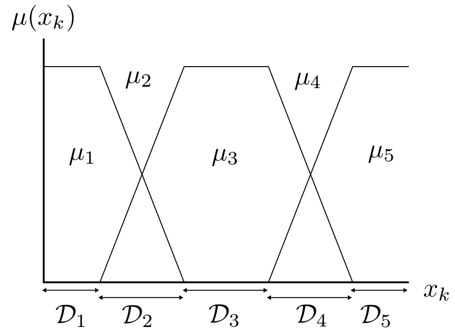


Fig. 2. Intersections of several membership functions defines the PWA partitioning.

3. TRANSFORMATION OF TAKAGI-SUGENO MODELS INTO PWA FORM

The aim of this section is to transform the TS model (1) into a corresponding Piecewise Affine form such that convenient MPC approaches could be used for control synthesis. Discrete-time PWA systems with an uncertain system matrix can be described by

$$\begin{aligned} x_{k+1} &:= f_{\text{PWA}}(x_k, \lambda_k, u_k) \\ &:= A_d(\lambda_k) x_k + B_d u_k + f_d, \text{ if } x_k \in \mathcal{D}_d, \end{aligned} \tag{9}$$

where λ_k represents a vector of parameters whose entries are unknown, but are assumed to be bounded. The domain $\mathcal{D} := \cup_{d=1}^{N_\mathcal{D}} \mathcal{D}_d$ of $f_{\text{PWA}}(\cdot,\cdot)$ is a non-empty compact set in $\mathbb{R}^{n_x}$ with $N_\mathcal{D} < \infty$ elements represented as a union of polytopes in hyperplane representation:

$$\mathcal{D}_d := \left\{x_k \mid D_d^x x_k \le D_d^0\right\} \tag{10}$$

The system matrices $A_d(\lambda_k)$ in (9) can be expressed as a convex combination of a total of m_d vertices $A_{d,1}, \dots, A_{d,m_d}$, i.e.

$$A_d(\lambda_k) = \lambda_{k,1} A_{d,1} + \ldots + \lambda_{k,m_d} A_{d,m_d} \tag{11}$$

with $\sum_j \lambda_{k,j} = 1$ and $0 \le \lambda_{k,j} \le 1$. For the PWA system to be well-posed (Bemporad and Morari, 1999), it is required that the regions $\mathcal{D}_d$ do not overlap, i.e.

$$\mathcal{D}_p \cap \mathcal{D}_q = \emptyset \quad \forall p \neq q, \quad p = 1, \dots, N_\mathcal{D}. \tag{12}$$

This non-overlapping property represents the main difference between Takagi-Sugeno models (8) and the PWA description (9). Therefore in order to transform the TS model into a PWA form, we first need to identify such overlapping regions and then, in the second step, assign a unique dynamics to each such intersection. As sketched in Fig. 2, if the membership functions are given by linear hyperplanes (which is true for trapezoidal functions considered in Assumption 2.1), the intersecting parts can be obtained by defining new regions for each intersection of the neighboring fuzzy sets. Formally, we are looking for a non-overlapping polyhedral partitioning $\mathcal{D}$ of the set $\mathcal{P}$ such that (12) holds and $\bigcup_d \mathcal{D}_d = \bigcup_i \mathcal{P}_i$. The partitioning can be obtained in two steps. In the first stage, the regions corresponding to subsets of $\mathcal{P}$ over which only one membership function μ_i takes a non-zero value are found by a set-difference operation:

$$\mathcal{D}_a = \mathcal{P}_i \setminus \bigcup_{j \neq i} \mathcal{P}_j, \quad \forall i = 1, \dots, r \tag{13}$$

Such sets are represented by $\mathcal{D}_1$, $\mathcal{D}_3$, and $\mathcal{D}_5$ in Figure 2. Then, in the second step, the regions in which multiple membership functions overlap can be found by another set-difference operation:

$$\mathcal{D}_b = \mathcal{P} \diagdown \bigcup \mathcal{D}_a. \tag{14}$$

For the case in Figure 2, regions $\mathcal{D}_2$ and $\mathcal{D}_4$ are an example of $\mathcal{D}_b$. The resulting partitioning $\mathcal{D}_d$ is thus given as a union of $\mathcal{D}_a$ and $\mathcal{D}_b$:

$$\mathcal{D}_d = \{x_k \mid x_k \in \mathcal{D}_a \vee x_k \in \mathcal{D}_b\}. \tag{15}$$

Once the strictly separated regions $\mathcal{D}_d$ are obtained, the next task is to associate one local model to each such region. This can be done, for each d, by transforming the dynamics (8) into the form of (9) in such a way that the cross-term $\alpha_i(x_k)A_i$ is replaced by the linear combination of the form of (11). The vertices $A_{d,1}, \ldots, A_{d,m_d}$ of the uncertainty set (11) can be easily obtained from (8) by evaluating $\alpha_i(x_k)$ for all vertices of the region $\mathcal{D}_d$ for all i such that $\mathcal{D}_d \cap \mathcal{P}_i \neq \emptyset$.

Once the non-overlapping regions $\mathcal{D}_d$ and the corresponding vertices $A_{d,1}, \ldots, A_{d,m_d}$ of (11) are computed, the PWA model (9) is an over-approximation of the TS model (8). However, if the value of λ_k in (11) is expressed as $\lambda_{k,i} = \alpha_i(x_k)$, value of which can be measured at each time since x_k is assumed to be measurable and α_i are known, the one-step equivalence between the two models can be established:

Lemma 3.1. The PWA model (9) is equivalent to the TS model (8), for one time step, if

$$\lambda_{k,i} = \alpha_i(x_k) \tag{16}$$

can be measured at each time k.

Proof. Follows directly from Assumptions 2.1 and 2.2 and from the definition of $A_d(\lambda_k)$ as in (11) since $\mathcal{D}_d$ given by (15) satisfies $\bigcup_d \mathcal{D}_d = \bigcup_i \mathcal{P}_i$. This one-step equivalence allows to synthesize an MPC feedback law by applying the concept of dynamic programming, i.e. by solving one-step problems backwards in time. In the next section we show how an MPC controller for PWA systems of the form of (9) can be designed by considering λ_k as a parametric uncertainty. The value of λ_k is not known at the time of the synthesis, but its assumed to be bounded. The control policy will be parameterized in the influence of λ_k, such that the optimal control action could be recovered once the particular value of $\lambda_k = \alpha(x_k)$ becomes known when the controller is implemented on-line. Therefore the feedback policy could be viewed at as an optimization-based adaptive controller which takes into account the knowledge of the activation of individual fuzzy rules.

4. CONTROL SYNTHESIS

The problem which we aim at solving can be stated as follows:

Problem 4.1. For the PWA system (9), find a feedback policy of the form

$$u_k = g(x_k, \lambda_k), \tag{17}$$

which takes into account measurements of the current state x_k and the measurements of the vector λ_k, and drives all system states towards a given terminal set in the minimal admissible number of time steps, i.e. finds the smallest integer $N_{\max}$ satisfying $0 \leq N_{\max} < \infty$ such that $x_{N_{\max}} \in T_{\text{set}}$ for all admissible values of the vector λ_k with T_{set} representing a user-defined terminal set.

Problem 4.1 is commonly referred to as a *minimum-time problem*, a problem frequently tackled in the literature, see e.g. (Keerthi and Gilbert, 1987; Grieder et al., 2005; Raković et al., 2004). However, the cited approaches cannot be applied to solve Problem 4.1 directly, as (9) is nonlinear in the bi-product between x_k and λ_k (cf. (11)) even if a particular mode d is fixed.

However, it is easy to observe that the state-update equation of the PWA system (9) for a fixed mode d is linear in the joint product $A_i(\lambda_k)x_k$. Following the ideas of (Baric et al., 2008) and (Besselmann et al., 2008) we propose to introduce an auxiliary information variable z_k to convert the bi-linear PWA form (9) into a linear one:

$$\begin{aligned} x_{k+1} &= f_{\text{PWA}}(z_k(x_k, \lambda_k, d), u_k) \\ &= z_k(x_k, \lambda_k, d) + Bu_k + f, \end{aligned} \tag{18}$$

where the information variable $z_k(\cdot)$ given by

$$z_k(x_k, \lambda_k, d) = A_d(\lambda_k)x_k \tag{19}$$

captures both the knowledge of the mode d active at the time instance k as well as the state contribution $A_d(\lambda_k)x_k$ for the actually measured value of the parameter vector λ_k.

Remark 4.1. Important to notice is that the augmented PWA system (18) is equivalent, for one time step, to the original form of (9) if the current state x_k, the current value of the parameter vector λ_k, and the active mode d are known such that $z_k(\cdot)$ can be evaluated per (19). This is not a restrictive requirement, but a direct consequence of the adaptive control strategy. As for any other state-feedback policy, the current state x_k has to be measured (or estimated), and the values of the parameters λ_k can be directly calculated from $\mu_i(x_k)$ per the equivalence (16). The active mode d is uniquely determined by (10) for each x_k. At the next time step, a new value of $z_{k+1}(\cdot)$ will be calculated from x_{k+1} and the whole procedure can be repeated.

This equivalence allows to re-formulate Problem 4.1 as follows:

Problem 4.2. For the augmented PWA system (18) with $z_k(\cdot)$ defined as in (19), find a feedback policy of the form

$$u_k = g(z_k(x_k, \lambda_k, d)), \tag{20}$$

which takes into account the measurements of the information variable $z_k(\cdot)$ and minimizes the number of time steps needed to push all system states towards the terminal set, i.e. find the smallest integer $N_{\max}$ satisfying $0 \leq N_{\max} < \infty$ such that $x_{N_{\max}} \in T_{\text{set}}$ for all admissible values of λ_k.

Even though the augmented PWA system (18) is now linear in all variables, Problem 4.2 is still not trivial to solve as the solution has to robust against all possible variations of $\lambda_k \in \Lambda$ entering the system matrices $A_d(\lambda)$. We remind the reader that only the bounds Λ are known at the time of control synthesis, λ_k will only become available once the controller is calculated and implemented on-line.

The idea of the proposed control synthesis is based on solving Problem 4.2 parametrically, such that the optimal solution is "pre-calculated" for all admissible values of the information variable $z_k(x_k, \lambda_k, d)$. Once the feedback strategy is obtained in a form of a look-up table, the on-line implementation procedure then consists of identifying the values of the parameter vector λ_k, evaluating the information variable $z_k(x_k, \lambda_k, d)$, and evaluating the feedback policy $u_k^* = g(z_k(\cdot))$ for the respective value of $z_k(\cdot)$ in order to obtain the optimal control action.

As was shown in (Grieder et al., 2005) for PWA systems with no parametric uncertainties, a minimum-time controller could be found by solving a series of horizon-1 optimization problems for each feasible switching sequence (i.e. for each allowed change of d). In this paper we extend the synthesis procedure to cover PWA systems with parametric uncertainties. To do so, we propose to solve a series of $d = 1, \dots, N_{\mathcal{D}}$ problems of the form

$$\begin{aligned}
\min_{u_k} \ & |Qx_{k+1}|_1 + |Ru_k|_1 && \text{(21a)}\\
\text{s.t. } & u_k \in \mathcal{U}, && \text{(21b)}\\
& x_k \in \mathcal{X}, && \text{(21c)}\\
& D_d^x x_k \le D_d^0, && \text{(21d)}\\
& x_{k+1} = z_k(x_k, \lambda_k, d) + Bu_k + f, && \text{(21e)}\\
& x_{k+1} \in T_{\text{set}}, && \text{(21f)}\\
& A_d^1 x_k + Bu_k + f \in T_{\text{set}} && \text{(21g)}\\
& \vdots \\
& A_d^{m_d} x_k + Bu_k + f \in T_{\text{set}}, && \text{(21h)}
\end{aligned}$$

with $\mathcal{X}$ and $\mathcal{U}$ representing, respectively, the polyhedral state and input constraints sets. Constraints (21g)–(21h) enforce that the resulting control law will push all system states in one step to a given terminal set for all admissible values of the parametric uncertainty λ_k. We remark that the value of λ_k is unknown at the time of control synthesis, therefore one has to consider all vertices of the parametric uncertainty in (21g)–(21h). The objective function (21a) also takes λ_k into account through the use of the information variable $z_k(x_k, \lambda_k, d)$ via the definition of x_{k+1} as in (18). Therefore the resulting control law will be parameterized in the influence of λ_k and its on-line measurements can be used to adjust the control action accordingly.

The optimal solution to the linear programming problem (21) can be obtained for all feasible initial conditions x_k (or $z_k(\cdot)$) using techniques of parametric programming as summarized by the following theorem.

Theorem 4.1. ((Borrelli, 2003)). The solution to the problem (21) for all admissible initial conditions $y_k \equiv x_k$ or $y_k \equiv z_k(\cdot)$ is a piecewise affine state feedback optimal control law of the form

$$u_k^*(y_k) = F_r^k y_k + G_r^k \quad \text{if} \quad y_k \in \mathcal{R}_r^k, \qquad (22)$$

where $\mathcal{R}_r^k = \left\{y_k \in \mathbb{R}^n | H_r^k y_k \le K_r^k\right\}, r = 1, \dots, R^k$ is a set of polyhedral (or polytopic) regions. Moreover, the set $\mathcal{P}^k = \bigcup_r \mathcal{R}_r^k$ of initial parameters y_k for which problem (21) is feasible at time k is a convex set.

The important implication of Theorem 4.1 is that one can find, *simultaneously*, (i) the closed-form expression for the optimizer u_k^* to the the problem (21) and (ii) the set of feasible initial values of the parameters. We can freely choose whether we want to solve the problem by considering x_k or $z_k(\cdot)$ as the free parameter. The parametric solution can be easily obtained e.g. by using the freely available Multi-Parametric Toolbox (Kvasnica et al., 2004).

We can now state the main result of the paper, which is an algorithm for synthesis of an adaptive MPC feedback strategy which solves Problem 4.2 parametrically in a dynamic programming fashion, i.e. by solving a series of one-step problems backwards in time.

(1) Choose the initial terminal set $T_{\text{set}} \subset \mathcal{X}$.
(2) Set the iteration counter $k = 0$ and set $\mathcal{S}_k = T_{\text{set}}$.
(3) For each mode $d \in [1, \dots, N_{\mathcal{D}}]$ of the PWA system (9):
    (a) Solve the optimization problem (21) parametrically by considering $z_k(\cdot)$ as the parameter. Store the optimizer $u_k^*(z_k(\cdot))$ and the corresponding regions $\mathcal{R}_r^{d,k}$.
    (b) Solve the optimization problem (21) parametrically by considering x_k as the parameter. Store the set $\mathcal{P}^{d,k}$ of states x_k, for which (21) was feasible for the mode d.
(4) Denote the feasible set of $N_{\mathcal{D}}$ problems (21) for $d = 1, \dots, N_{\mathcal{D}}$ by $\mathcal{S}_{k+1} = \bigcup_d \mathcal{P}^{d,k}$.
(5) If $\mathcal{S}_{k+1} = \mathcal{S}_k$, stop, the algorithm has converged.
(6) Increase k by 1 and jump back to Step 3.

At every run of Step 3a of the algorithm above, a control law of the form (22) is obtained as a look-up table parameterized in the information variable $z_k(x_k, \lambda_k, d)$. Moreover, since $z_k(\cdot)$ enters the objective function of the optimization problem (21a), the obtained feedback policy will be optimal for any measured value of λ_k via $z_k(x_k, \lambda_k, d)$. By construction, the feedback law pushes all system states one step closer to the initially chosen terminal set for all choices of λ. The union of the sets calculated in Step 3b is then used at the next iteration as a new terminal set constraint.

Remark 4.2. To attain stability, T_{set} along with a feedback law u_{set}^* active for all $x \in T_{\text{set}}$ must be chosen such that the terminal set is invariant. Finding such terminal set along with the terminal controller is, however, outside of the scope of this work.

Note that, in general for PWA systems, the sets $\mathcal{S}_k$ for $k > 0$ will be non-convex unions of a finite number of convex polytopes, even when a convex terminal set T_{set} is used initially. In such a case one would need to solve a series of problems (21) by employing individual components of $\mathcal{S}_k$ as a terminal set in (21g)–(21h).

The iterative nature of the algorithm guarantees that once the procedure converges, for all feasible initial conditions there will exist a feedback law such that the terminal set will be reached in the minimal possible number of time steps (Grieder et al., 2005) for all admissible realizations of the uncertain parameter vector λ in (9). Moreover, the k^* for which Algorithm 4 converges, defines the lowest possible value of $N_{\max}$ in Problem 4.2 (Grieder et al., 2005). The on-line implementation of the resulting feedback law can then be performed as follows. First, the quantity $z_k = \left(\sum_i \alpha_i(x_k)A_i\right) x_k$ is calculated based on the measurements of x_k and the knowledge of the corresponding values of $\mu_i(x)$ entering α_i as in (4). Then, in the second step, the look-up table (22) is evaluated for the respective value of

z to obtain the optimal control action $u_k^*(z_k)$. This is done by selecting from the sets $\mathcal{S}_i$ the lowest value of the index i such that $x_k \in \mathcal{S}_i$, i.e. finding the set with the lowest "step distance" wrt. the initial terminal set. Once the index i is known, $u_k^*(z_k)$ can be extracted from the i-th look-up table $\mathcal{R}_r^i$, cf. (22). Therefore the control law calculated by Algorithm 4 acts as a minimum-time controller for the TS fuzzy system (1) when implemented on-line.

5. EXAMPLE

We have applied the proposed synthesis scheme to design a minimum-time controller for a continuously stirred tank reactor (CSTR), where the reaction $A \to B$ takes place. By considering the normalized conversion rate as $x^{(1)}$ and the normalized mixture temperature as $x^{(2)}$, and the coolant temperature as the system input u, the model of the reactor is given by (Cao and Frank, 2000)

$$\dot{x}^{(1)} = f_1(x) + \left(\frac{1}{\rho} - 1\right) x^{(1)} \tag{23a}$$

$$\dot{x}^{(2)} = f_2(x) + \left(\frac{1}{\rho} - 1\right) x^{(2)} + \beta u \tag{23b}$$

where

$$\begin{aligned} f_1(x) =& -\frac{1}{\rho} x^{(1)} + D_a(1 - x^{(1)}) e^{\left(\frac{x^{(2)}}{1 + x^{(2)}/\gamma_0}\right)} \\ f_2(x) =& -\left({}^1\!/\!_\rho + \beta\right) x^{(2)} + \\ &+ H D_a (1 - x^{(1)}) e^{\left(\frac{x^{(2)}}{1 + x^{(2)}/\gamma_0}\right)} \\ \gamma_0 =& 20,\ H = 8,\ \beta = 0.3,\ D_a = 0.072,\ \rho = 0.8. \end{aligned}$$

The states are subject to constraints $0 \le x^{(1)} \le 1$ and $0 \le x^{(2)} \le 10$, respectively. The control signal is bounded by $-20 \le u \le 20$.

The nonlinear model (23) can be approximated (Cao and Frank, 2000) by a Takagi-Sugeno fuzzy system with three membership functions. Assuming the sampling time $T_s = 0.1$ minutes, the TS model (8) is given by

$$x_{k+1} = \left(\sum_{i=1}^{3} \alpha_i(x_k) A_i\right) x_k + B u_k \tag{24}$$

where $\alpha_i(x_k)$ can be calculated from $\mu_i(x_k)$ by (4). The numerical values of A_i and B are given by

$$A_1 = \begin{pmatrix} 0.866 & 0.006 \\ -0.126 & 0.909 \end{pmatrix},\ A_2 = \begin{pmatrix} 0.802 & 0.038 \\ -0.627 & 1.162 \end{pmatrix},$$
$$A_3 = \begin{pmatrix} 0.604 & 0.026 \\ -2.193 & 1.065 \end{pmatrix},\ B = \begin{pmatrix} 0.0001 \\ 0.0286 \end{pmatrix}.$$

Dynamics A_1 and A_3 capture the stable operating points of the CSTR, while dynamics A_2 corresponds to the unstable mode of the reactor. The membership functions $\mu_1(x)$, $\mu_2(x)$, and $\mu_3(x)$ are given as trapezoidal functions with centers around respective linearization points:

$$\mu_1(x) = \begin{cases} 1 & \text{if } 0 \le x^{(2)} \le 2.5 \\ 1 - \dfrac{x^{(2)} - 2.5}{2.652 - 2.5} & \text{if } 2.5 \le x^{(2)} \le 2.652 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_2(x) = \begin{cases} \dfrac{x^{(2)} - 2.5}{2.652 - 2.5} & \text{if } 2.5 \le x^{(2)} \le 2.652 \\ 1 & \text{if } 2.65 \le x^{(2)} \le 2.85 \\ 1 - \dfrac{x^{(2)} - 2.852}{3 - 2.852} & \text{if } 2.852 \le x^{(2)} \le 3 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_3(x) = \begin{cases} 1 & \text{if } x^{(2)} \ge 3 \\ \dfrac{x^{(2)} - 2.852}{3 - 2.852} & \text{if } 2.852 \le x^{(2)} \le 3 \\ 0 & \text{otherwise.} \end{cases}$$

In order to synthesize the minimum-time controller for such a TS system, we first need to convert the TS model (24) into a corresponding PWA form (9) by using the procedure described in Section 3. The partitioning $\mathcal{D}_d$ results directly from the respective domains of individual μ_i's, i.e.

$$\begin{aligned} \mathcal{D}_1 :=& \{x \in R^2 \mid 0 \le x^{(2)} \le 2.5\} \\ \mathcal{D}_2 :=& \{x \in R^2 \mid 2.5 \le x^{(2)} \le 2.652\} \\ \mathcal{D}_3 :=& \{x \in R^2 \mid 2.652 \le x^{(2)} \le 2.852\} \\ \mathcal{D}_4 :=& \{x \in R^2 \mid 2.852 \le x^{(2)} \le 3\} \\ \mathcal{D}_5 :=& \{x \in R^2 \mid 3 \le x^{(2)} \le 10\}. \end{aligned} \tag{26}$$

Dynamics assigned to each element of $\mathcal{D}_d$ is represented by

$$x_{k+1} = \begin{cases} A_1 x_k + B u_k & \text{if } x^{(2)} \in \mathcal{D}_1(x) \\ \{A_1, A_2\} x_k + B u & \text{if } x^{(2)} \in \mathcal{D}_2(x) \\ A_2 x_k + B u_k & \text{if } x^{(2)} \in \mathcal{D}_3(x) \\ \{A_2, A_3\} x_k + B u_k & \text{if } x^{(2)} \in \mathcal{D}_4(x) \\ A_3 x_k + B u_k & \text{if } x^{(2)} \in \mathcal{D}_5(x) \end{cases} \tag{27}$$

where $\{A_1, A_2\}$ represents a system matrix A as a convex combination of the vertices A_1 and A_2.

To perform the controller synthesis, we have implemented Algorithm 4 using the Multi-Parametric Toolbox (Kvasnica et al., 2004) and YALMIP (Löfberg, 2004). The control objective was to drive the system states towards the terminal set $T_{set} = \{x \mid 2.652 \le x^{(2)} \le 2.852\}$ in the minimal possible number of time steps, while minimizing the objective function $J = |u_k|_1 + |10(x_{k+1} - 2.751)|_1$ at each step k. This operating range corresponds to the unstable mode of the reactor. Our implementation of Algorithm 4 resulted in a PWA feedback law of the form (22) defined over 34 regions in the two-dimensional $z(\cdot)$ space. The proposed Algorithm 4 was compared with a standard parallel distributed compensation (PDC) approach of (Wang et al., 1996). Input constraints have been incorporated into PDC design using LMI techniques and PDC controller was computed via YALMIP interface. For the initial condition $x_0 = [0.6,\ 9]^T$, Figure 3 shows the closed-loop evolution of system state $x^{(2)}$ for both controllers. While PDC controller reacts on the initial condition conservatively, minimum-time (MT) controller is significantly faster it drives system states towards the chosen terminal set in a minimum-time fashion. The corresponding values of the control actions are shown in Figure 4.

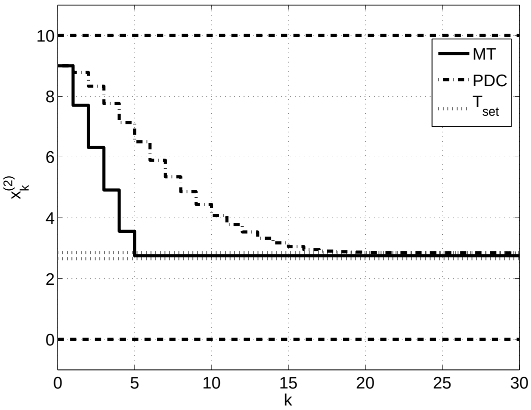


Fig. 3. Closed-loop evolutions of the dimensionless temperature.

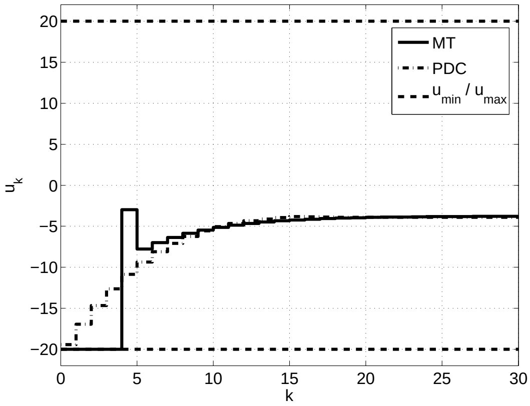


Fig. 4. Values of the control actions.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/4055/07, and 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

M. Baric, Sasa V. Rakovic, Th. Besselmann, and M. Morari. Max-Min Optimal Control of Constrained Discrete-Time Systems. In *IFAC World Congress*, July 2008.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3): 407–427, March 1999.

A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica*, 38(1):3–20, January 2002. URL `http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=12`.

Th. Besselmann, J. Löfberg, and M. Morari. Explicit MPC for systems with linear parameter-varying state transition matrix. In *International Federation of Automatic Control World Congress*, July 2008.

F. Borrelli. Constrained Optimal Control of Linear and Hybrid Systems. In *Lecture Notes in Control and Information Sciences*, volume 290. Springer, 2003.

Yong-Yan Cao and P.M. Frank. Analysis and synthesis of nonlinear time-delay systems via fuzzy control approach. *IEEE Transactions on Fuzzy Systems*, 8(2):200–211, April 2000.

G. Feng. A Survey on Analysis and Design of Model-Based Fuzzy Control Systems. *IEEE Transactions on Fuzzy Systems*, 14(5):676–697, Oct. 2006.

P. Grieder, M. Kvasnica, M. Baotic, and M. Morari. Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica*, 41, issue 10:1683–1694, Oct. 2005.

M. Johansson, A. Rantzer, and K.-E. Årzén. Piecewise Quadratic Stability of Fuzzy Systems. *IEEE Transactions on Fuzzy Systems*, 7(6):713–722, Dec. 1999.

S. S. Keerthi and E. G. Gilbert. Computation of Minimum-Time Feedback Control Laws for Discrete-Time Systems with State-Control Constraints. *IEEE Trans. on Automatic Control*, AC-32:432–435, May 1987.

M. Kvasnica, P. Grieder, M. Baotic, and M. Morari. Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004. Available from `http://control.ee.ethz.ch/~mpt`.

J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004. Available from `http://control.ee.ethz.ch/~joloef/yalmip.php`.

S. Mollov, R. Babuška, J. Abonyi, and H.B. Verbruggen. Effective Optimization for Fuzzy Model Predictive Control. *IEEE Transactions on Fuzzy Systems*, 12(5):661–675, Oct. 2004.

S.V. Raković, P. Grieder, M. Kvasnica, D.Q. Mayne, and M. Morari. Computation of Invariant Sets for Piecewise Affine Discrete Time Systems subject to Bounded Disturbances. In *Proceeding of the 43rd IEEE Conference on Decision and Control*, pages 1418–1423, Atlantis, Paradise Island, Bahamas, December 2004.

E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2): 346–358, April 1981.

T. Takagi and M. Sugeno. Fuzzy identications of fuzzy systems and its applications to modelling and control. *IEEE Trans. Systems Man and Cybernetics*, 15:116–132, 1985.

K. Tanaka and H.O. Wang. T-S Fuzzy Model as Universal Approximator. *Fuzzy Control Systems Design and Analysis*, pages 277–289, 2002.

A. Vasičkaninová and M. Bakošová. Fuzzy model-based predictive control of a chemical reactor. In *Proceedings of European Congress of Chemical Engineering (ECCE-6)*, Copenhagen, September 2007. EFCE.

H.O. Wang, K. Tanaka, and M.F. Griffin. An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues. *IEEE Transactions on Fuzzy Systems*, 4 (1), February 1996.

Comments – Remarks

Robust Explicit Time-Optimal Control of PWA Systems with Parametric Uncertainties

M. Kvasnica *, **M. Herceg** **, **Ľ. Čirka** *, and **M. Fikar** *

* *Institute of Information Engineering, Automation, and Mathematics,
Slovak University of Technology in Bratislava,
81237 Bratislava, Slovakia
e-mail : {michal.kvasnica,lubos.cirka,miroslav.fikar}@stuba.sk*
** *Automatic Control Laboratory, ETH Zurich, 8006 Zurich,
Switzerland, e-mail : herceg@control.ee.ethz.ch*

Abstract: : The paper shows how to design model predictive controllers which are robust against parametric uncertainties of the prediction models. It is illustrated that the feedback controller can be obtained in a closed-loop form, represented by a piecewise affine (PWA) function. Such controllers can then be easily implemented in real-time by converting the PWA function into a look-up table. Given a prediction model which includes a-priori unknown parameters, the procedure described in the paper shows how to construct robust time-optimal MPC controllers which are able to utilize on-line measurements of the parameters to optimize for performance. Closed-loop stability and constraint satisfaction is maintained for all values of the parameters from a given range.

Keywords: Model Predictive Control, real-time control, parametric optimization

1. INTRODUCTION

PWA systems represent a powerful tool to describe the evolution of hybrid systems (Sontag, 1981) and can be shown to be equivalent to many other hybrid system classes (Heemels et al., 2001) such as mixed logical dynamical systems, linear complementary systems, and max-min-plus-scaling systems and thus form a very general class of linear hybrid systems. Moreover, PWA systems can be used to identify or approximate generic nonlinear systems via multiple linearizations at different operating points (Sontag, 1981). Although hybrid systems (and in particular PWA systems) are a special class of nonlinear systems, most of the nonlinear system and control theory does not apply because it usually requires certain smoothness assumptions. For the same reason we also cannot simply use linear control theory in some approximate manner to design controllers for PWA systems.

Model predictive control of PWA systems has garnered increasing interest in the research community because it allows optimal control inputs for discrete-time PWA systems to be obtained by solving mixed-integer optimization problems on-line (Bemporad and Morari, 1999; Mayne and Raković, 2003), or as was shown in (Baotić et al., 2003; Borrelli et al., 2003; Kerrigan and Mayne, 2002; Borrelli, 2003), by solving off-line a number of multi-parametric programs. By multi-parametric programming, a linear (mpLP) or quadratic (mpQP) optimization problem is solved off-line for a range of parameters. The associated solution (the explicit representation of the optimal control law) takes the form of a PWA state feedback law. In particular, the state-space is partitioned into polyhedral regions in which the optimal control law is given as an affine function of the state. In the on-line implementation of such controllers, input computation reduces to a simple set-membership test. Even though the benefits of this procedure in terms of cheap implementation are self evident, one major drawback of the parametric approach to MPC is the solution itself. Once calculated off-line, the solution is, so to say, “set in stone” and it can only be changed by repeating the off-line calculation. This might be necessary e.g. when the knowledge of plant model used to formulate the underlying optimization problem is updated. This is a frequent requirement in control of real plants, because the precise values of some (or all) model parameters are not known exactly and they often fluctuate in time.

This issue is usually tackled by adopting the framework of adaptive control (Bitmead et al., 1990; Mosca, 1995; Dostál et al., 2004). In this policy, the values of unknown parameters are measured, estimated or identified on-line and the process model is updated accordingly. For the newly obtained model, a new control problem is formulated and solved to take the updated knowledge into account. This repetitive parameter estimation and control optimization is particularly suitable in classical on-line MPC. However, it goes against the spirit of parametric solutions to MPC problems, where the solution is calculated just once for a fixed process model. To circumvent this problem and to keep the advantages of the off-line MPC approach, (Baric et al., 2008) and (Besselmann et al., 2008) proposed, for linear and LPV systems, respectively, a method of solving a max-min control problem parametrically while providing (i) robust feasibility and (ii) optimal performance in terms of minimizing a given objective function over a fixed prediction horizon.

In this paper we first extend these ideas to PWA systems, whose dynamics is affected by a some time-varying parameters. Parameter values are not known at the time of the synthesis of the control law, but will become available when the controller is evaluated on-line. The goal is to synthesize a control policy which drives all system states towards a given terminal set in the least possible number of time steps, i.e. in a *minimum time* fashion with respect to the system dynamics and constraints on states and control inputs. We illustrate how this problem can be solved off-line to obtain the feedback law in a form of a look-up table, hence mitigating the on-line implementation effort. As will be shown later, this boils down to a non-convex problem. Therefore, in this article we also provide a new methodology of synthesizing feedback laws under the circumstances that the terminal sets are non-convex unions of convex polytopes. Once the closed-form representation of the control law is obtained off-line, robust feasibility and guaranteed convergence towards a chosen terminal set for any variation of the unknown system parameters within a given range is provided. Moreover, since the influence of the parameters is consider in the optimization objective, on-line measurements of these parameters can be used to update the control policy. Hence the proposed strategy acts as a robust minimum-time adaptive controller, where the control inputs are time-optimal for all values of the parametric uncertainty. Therefore, the updated knowledge of the process model can be taken into account by the controller at each time step.

2. PROBLEM STATEMENT

In this work we consider discrete-time PWA systems of the following form

$$\begin{aligned} x_{k+1} &= f_{\text{PWA}}(x_k, \lambda_k, u_k) \\ &= A_d(\lambda_k)x_k + B_d u_k + f_d \quad \text{if} \quad x_k \in \mathcal{D}_d, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ is the system state, $u_k \in \mathbb{R}^{n_u}$ is the manipulated input, $k \geq 0$ denotes the sampling instant, and $A_d(\lambda_k)$, B_d, f_d are matrices of appropriate dimensions. Variable $d \in [1, \dots, n_D]$ denotes the mode of the PWA system, with n_D being the total number of modes. The system states and inputs are assumed to be bounded by, respectively, $x_k \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $u_k \in \mathcal{U} \subset \mathbb{R}^{n_u}$ where $\mathcal{X}$ and $\mathcal{U}$ are nonempty convex and compact sets. We assume that the system matrices $A_d(\lambda_k)$ depend on an unknown, but bounded parameter vector $\lambda_k \in \Lambda$. Furthermore we assume that Λ is a convex and compact set and that the unknown parameters λ_k enter $A_d(\lambda_k)$ in a linear fashion

$$A_d(\lambda_k) = \sum_{j=1}^{n_\lambda} \lambda_{k,j} A_i^j, \tag{2}$$

with $\sum_j \lambda_{k,j} = 1$ and $0 \leq \lambda_{k,j} \leq 1$ for $j \in [1, \dots, n_\lambda]$ and the total of n_λ vertices $A_d^1, \dots, A_d^{n_\lambda}$ being given.

If the whole state space domain is denoted by $\mathcal{D}$, then the overall PWA model is built by $\bigcup_{d=1}^{n_D} \mathcal{D}_d$ regions, whereas one local model is valid in each region. Formally $\mathcal{D}_d$ is a nonempty compact set, defined in the state space, and it is given by a set of linear inequalities of the form

$$\mathcal{D}_d = \left\{ x_k \mid D_d^x x_k \leq D_d^0 \right\} \tag{3}$$

where D_d^x and D_d^0 are matrices of suitable dimensions specifying the borders of the d-th region $\mathcal{D}_d$.

For the PWA system (1) this paper shows how to solve the following problem:

Problem 2.1. For the PWA system (1), find a feedback policy of the form

$$u = g(x, \lambda), \tag{4}$$

which takes into account measurements (or estimates) of the current state x and measurements (or estimates) of the parameter vector λ, and drives all system states into $\mathcal{T}_{\text{set}}$ in the least possible number of steps for all possible values of the parameter vector $\lambda \in \Lambda$ while respecting input and state constraints.

Problem 2.1 is commonly referred to as a *minimum-time* problem (Keerthi and Gilbert, 1987; Blanchini, 1992; Mayne and Schroeder, 1997; Grieder et al., 2005; Raković et al., 2004). As shown e.g. in (Grieder et al., 2005), if the PWA system (1) is *not* subject to the parametric uncertainty (2) (i.e. for $n_\lambda = 1$), the minimum-time problem can be solved using dynamic programming (DP), i.e. by solving 1-step problems backwards in time. At each iteration of the DP procedure the feedback law $u_k^*(x_k)$ minimizing a given performance measure $J(x_k, u_k)$ is obtained, explicitly, by solving a multi-parametric program. The feedback is such that for all $x_k \in \mathcal{S}_k$ the one-step predicate x_{k+1} is pushed "one step closer" to the given initial terminal set, i.e. $f_{\text{PWA}}(x_k, u_k^*(x_k)) \in \mathcal{S}_{k-1}$.

There are two reasons why these standard approaches cannot be directly applied to solve Problem 2.1: (i) λ_k and x_k are optimized parameters, thus the PWA dynamics is nonlinear due to their bi-product in (1), and (ii) the resulting variations of the parameter vector λ_k in the set Λ. This is equivalent to solution of a non-convex minimum-time problem for a PWA system with state-dependent disturbances. Both issues make Problem 2.1 far from trivial.

3. SYNTHESIS OF AN ADAPTIVE MINIMUM-TIME CONTROLLER

In this section we show how to solve Problem 2.1 parametrically, i.e. we obtain an explicit representation of the function $g(x_k, \lambda_k)$ for all admissible values of x_k and λ_k. Solving the problem faces following challenges:

C1: Deal with the fact that the PWA dynamics (1) is bilinear in x_k and λ_k.

C2: Give a procedure for computing *robust* one-step reachable sets for PWA systems with parametric uncertainties, i.e. find

$$\begin{aligned} Pre(\mathcal{S}_k) = \{ x_k \mid \exists u_k \in \mathcal{U}, \text{ s.t.} \\ f_{\text{PWA}}(x_k, \lambda_k, u_k) \in \mathcal{S}_k, \ \forall \lambda_k \in \Lambda \}. \end{aligned} \tag{5}$$

C3: Find an explicit representation of the feedback law (4) in such a way that measurements of λ_k are taken into account when minimizing

$$J(x_k, \lambda_k, u_k) = \|Q_x x_{k+1}\|_1 + \|Q_u u_k\|_1, \tag{6}$$

allowing (4) to adapt the control action to the currently available value of λ_k. Here, $x_{k+1} = f_{\text{PWA}}(x_k, \lambda_k, u_k)$, cf. (1), $\|\cdot\|_1$ denotes a standard 1-norm of a vector and Q_x, Q_u are weighting matrices of suitable dimensions.

Challenge C1 can be attacked by observing that (1) is linear in the joint product $A_d(\lambda_k)x_k$ for a fixed mode d. Following the ideas of (Baric et al., 2008) and (Besselmann et al., 2008) we propose to introduce an auxiliary information variable z_k to convert the bi-linear PWA form into a linear one:

$$\begin{aligned} x_{k+1} &= f_{\text{PWA}}(z_k, u_k) \\ &= z_k + B_d u_k + f_d. \end{aligned} \tag{7}$$

The information variable z_k given by

$$z_k(x_k, \lambda_k, d) = A_d(\lambda_k)x_k \tag{8}$$

captures both the knowledge of the mode d active at the time instance k as well as the state contribution $A_d(\lambda_k)x_k$ for the actually measured value of the parameter vector λ_k.

Remark 3.1. Important to notice is that the augmented PWA system (7) is equivalent to the original form of (1), for one time step, if the state x_k, the value of the parameter vector λ_k, and the active mode d are known at time k such that $z_k(\cdot)$ can be evaluated per (8). This is not a restrictive requirement, but a direct consequence of the adaptive control strategy. As for any other state-feedback policy, the current state x_k has to be measured (or estimated), and the values of the parameters λ_k will either be directly measured, or obtained e.g. using recursive identification techniques at each discrete time instance k. For a given x_k, the active mode d is uniquely determined by (3).

To illustrate solution to C2, we denote by $\mathcal{S}_k = \bigcup_{i=1}^{n_\mathcal{S}} \mathcal{S}_{k,i}$ the (possibly non-convex) union of $n_\mathcal{S}$ convex polytopes $\mathcal{S}_{k,i}$. Then we get the following result.

Lemma 3.2. For the PWA system (1) the set of states x_k which can be steered into $\mathcal{S}_k$ by some $u_k \in \mathcal{U}$ in one time step *for all possible values of* $\lambda_k \in \Lambda$ is given as a (possibly non-convex) union of convex polytopes

$$Pre(\mathcal{S}_k) = \bigcup_{i=1}^{n_\mathcal{S}} \bigcup_{d=1}^{n_D} proj_x(\mathcal{Z}_{k,i,d}) \tag{9}$$

with

$$\begin{aligned} \mathcal{Z}_{k,i,d} = \Big\{ \left[\begin{smallmatrix} x_k \\ u_k \end{smallmatrix}\right] \;\Big|\; & u_k \in \mathcal{U}, x_k \in \mathcal{D}_d, \\ & A_d^j x_k + B_d u_k + f_d \in \mathcal{S}_{k,i}, \\ & \forall j \in [1, \ldots, n_\lambda] \Big\} \end{aligned} \tag{10}$$

where $proj_x(\mathcal{Z}_{k,i,d})$ denotes the orthogonal projection of the set $\mathcal{Z}_{k,i,d}$ onto x.

Algorithm 3.3. Notice that $A_d^j x_k + B_d u_k + f_d \in \mathcal{S}_{k,i}$ $\forall j \in [1, \ldots, n_\lambda]$ can be written in an expanded form as

$$\begin{gathered} A_d^1 x + B_d u + f_d \in \mathcal{S}_{k,i}, \\ \vdots \\ A_d^{n_\lambda} x + B_d u + f_d \in \mathcal{S}_{k,i} \end{gathered}$$

which enforces that $f_{\text{PWA}}(x_k, \lambda_k, u_k) \in \mathcal{S}_{k,i}$ holds for all $\lambda_k \in \Lambda$. As $\mathcal{S}_{k,i}$, $\mathcal{U}$, and $\mathcal{D}_d$ are assumed to be convex, the set $\mathcal{Z}_{k,i,d}$ will be a convex polytope and its projection therefore also will be a polytope (Ziegler, 1995). Hence $Pre(\mathcal{S}_k)$ will be a collection of convex polytopes.

Optimal control action u_k minimizing the cost (6) in C3 can be found by solving the following horizon-1 non-convex optimization problem:

$$\min_{u_k} \; J(x_k, \lambda_k, u_k) \tag{11a}$$
$$\text{s.t. } u_k \in \mathcal{U} \tag{11b}$$
$$x_k \in \mathcal{D} \tag{11c}$$
$$x_{k+1} \in \mathcal{S}_k \tag{11d}$$
$$x_{k+1} = f_{\text{PWA}}(z_k(x_k, \lambda_x, d), u_k) \tag{11e}$$

Remark 3.4. By Lemma 3.2 and Remark 3.1 the set of x_k for which (11b)–(11e) is feasible is given by $Pre(\mathcal{S}_k)$. By considering the augmented PWA model (7) in (11e), the influence of the measured parameters λ_k on $J(\cdot)$ is taken into account when optimizing for the values of u_k.

Non-convexity of (11) stems from two reasons. First, the PWA state-update equation in (11e) is nonlinear due to the presence of "IF-THEN" rules in (1). Secondly, $\mathcal{S}_k$ in (11d) is, in general, given as a non-convex union of convex polytopes, i.e. $\mathcal{S}_k = \bigcup_i \mathcal{S}_{k,i}$. However, if the performance measure $J(\cdot)$ in (11) is as in (6), and for a fixed d and fixed i, problem (11) boils down to a convex linear programming (LP) problem in variables u_k and z_k. If z_k is considered as a parameter, feedback law (4) can be obtained, for all admissible values of z_k, by solving (11) as an mpLP:

Theorem 3.5. ((Borrelli, 2003)). The optimal solution to (11) for all admissible values of $z_k(\cdot)$ is, for a fixed i and d, a piecewise affine state-feedback control law and a PWA representation of the optimal cost in the form

$$u^*_{k,i,d}(z_k(\cdot)) = F^r_{k,i,d} z_k(\cdot) + G^r_{k,i,d} \text{ if } z_k(\cdot) \in \mathcal{R}^r_{k,i,d}, \tag{12}$$
$$J^*_{k,i,d}(z_k(\cdot)) = L^r_{k,i,d} z_k(\cdot) + M^r_{k,i,d} \text{ if } z_k(\cdot) \in \mathcal{R}^r_{k,i,d}, \tag{13}$$

where $\mathcal{R}^r_{k,i,d} = \left\{ z_k(\cdot) \mid H^r_{k,i,d} z_k(\cdot) \leq K^r_{k,i,d} \right\}, r = 1, \ldots, R_{k,i,d}$ is a set of polyhedral (or polytopic) regions. Moreover, the set $\mathcal{P}_{k,i,d} = \bigcup_r \mathcal{R}^r_{k,i,d}$ of all z_k for which (11) is feasible at time k is a convex set.

Theorem 3.6. ((Borrelli, 2003)). If (11) is solved consecutively $\forall i \in [1, \ldots, n_\mathcal{S}]$, $\forall d \in [1, \ldots, n_D]$, an explicit representation of the optimal feedback law $u^*_k(z_k(\cdot))$ given by

$$u^*_k(z_k(\cdot)) = \arg\min_{i,d} J^*_{k,i,d}(z_k(\cdot)) \tag{14}$$

is also a PWA function of $z_k(\cdot)$, i.e.

$$u^*_k(z_k(\cdot)) = F^r_k z_k(\cdot) + G^r_k \quad \text{if} \quad z_k(\cdot) \in \mathcal{R}^r_k \tag{15}$$

with $\mathcal{R}^r_k = \{z_k(\cdot) \mid H^r_k z_k(\cdot) \leq K^r_k\}$.

Theorem 3.6 suggests that an explicit representation of $u^*_k(z_k(\cdot))$ solving C3 can be found by solving $n_\mathcal{S} \cdot n_D$ mpLP's and subsequently by taking the minimum among the same number of PWA optimal costs $J^*_{k,i,d}$.

Remark 3.7. Multi-parametric linear programs can be solved e.g. using the freely available Multi-Parametric Toolbox (MPT) (Kvasnica et al., 2004), which also provides calculation of the minimum among several PWA cost functions in (14).

We can now state the main result of the paper, which is a procedure for designing an adaptive controller which will utilize the measurements of the parameter vector λ to update the control policy. The problem is solved parametrically, which means that the whole control synthesis can be performed off-line. On-line implementation of such a controller will reduce to a simple set-membership test.

The computation of the controller is carried out using Algorithm 1.

Algorithm 1 The minimum-time adaptive algorithm

```
INPUT: PWA system (1), weighting matrices $Q_x$, $Q_u$ of (6), initial terminal set $\mathcal{T}_{\text{set}}$.
OUTPUT: Integer $k^*$, sets $\mathcal{S}_k$, PWA feedback laws $u_k^*(z_k(\cdot))$.
1: $k \leftarrow 0$, $\mathcal{S}_k \leftarrow \mathcal{T}_{\text{set}}$.
2: repeat
3:     Obtain $u_k^*(z_k(\cdot))$ in the form of (15) by solving (11) as $n_{\mathcal{S}} \cdot n_D$ mpLPs.
4:     Compute $Pre(\mathcal{S}_k)$ by Lemma 3.2.
5:     $\mathcal{S}_{k+1} \leftarrow Pre(\mathcal{S}_k)$.
6:     $k \leftarrow k+1$.
7: until $\mathcal{S}_{k+1} \neq \mathcal{S}_k$
8: $k^* \leftarrow k$.
```

Theorem 3.8. The feedback laws $u_k^*(z_k(\cdot))$, $k = [0, \ldots, k^*]$ calculated by Algorithm 1 are such that the PWA system (1) can be robustly pushed to $\mathcal{T}_{\text{set}}$ $\forall \lambda \in \Lambda$ in, at most, k^* steps for any $x \in Pre(\mathcal{S}_{k^*})$. Moreover, measurements of λ are taken into account by $u_k^*(z_k(\cdot))$ to further optimize for performance.

Algorithm 3.9. At iteration k for any $x_0 \in \mathcal{S}_k$ the feedback law obtained in Step 3 is such that the one-step predicate $x_1 = f_{\text{PWA}}(x_0, \lambda_0, u_k^*(x_0))$ is pushed into $\mathcal{S}_{k-1}$ in one time step $\forall \lambda_0 \in \Lambda$ by construction (cf. (11d)). Robustness is ensured by taking $\mathcal{S}_k = Pre(\mathcal{S}_{k-1})$ computed by Lemma 3.2. The iterative nature of the algorithm guarantees that for any $x_1 \in \mathcal{S}_{k-1}$ we have $x_2 = f_{\text{PWA}}(x_1, \lambda_1, u_{k-1}^*(x_1)) \in \mathcal{S}_{k-2}$, $\forall \lambda_1 \in \Lambda$ again by Step 3. By consecutively applying the feedback laws $u_{k-2}^*(x_2), u_{k-3}^*(x_3), \ldots, u_0^*(x_k)$ we therefore get $x_{k+1} \in \mathcal{S}_0$ (note that $\mathcal{S}_0 = \mathcal{T}_{\text{set}}$ by Step 1). Hence all states of the PWA system (1) are pushed towards $\mathcal{T}_{\text{set}}$ in, at most, k^* steps due to the stopping criterion in Step 7. By employing (8) in the performance objective (11a) we have that the knowledge of λ_k is taken into account when optimizing for u_k^* at each iteration.

Remark 3.10. To attain stability, $\mathcal{T}_{\text{set}}$ along with a feedback law u_{set}^* active for all $x \in \mathcal{T}_{\text{set}}$ must be chosen such that the terminal set is invariant, i.e. $x_k \in \mathcal{T}_{\text{set}} \Rightarrow x_{k+j} \in \mathcal{T}_{\text{set}}$, $\forall j > 0$. Finding such terminal set along with the terminal controller is however, outside of the scope of this work.

Remark 3.11. The minimal number of steps k^* in which all system states can be steered into $\mathcal{T}_{\text{set}}$ is automatically identified in Step 8 upon convergence of Algorithm 1. This quantity is governed by feasibility of problem (11).

Remark 3.12. It should be noted that, at each iteration k, multiple control actions u_k might exist such that (11b)–(11e) hold. In such a case the performance index (11a) is used to select a unique solution in (14).

Remark 3.13. Complexity of the look-up table can be further reduced at each iteration k of Algorithm 1 in Step 3 by merging together regions whose union is convex and they share the same control law, see e.g. (Geyer et al., 2004). Efficient algorithms to perform such a reduction are included in (Kvasnica et al., 2004).

In general, for PWA systems the sets $\mathcal{S}_k$ would overlap, i.e. there are multiple k's for which $x \in \mathcal{S}_k$. Therefore, the consecutive application of different feedback laws at different steps must be performed such that a proper value of the index k is chosen for each x. Such a procedure is captured by Algorithm 2, which shows how the minimum-time adaptive controller is implemented on-line.

Algorithm 2 On-line implementation

```
INPUT: Measurements of the current state $x$ and the parameter vector $\lambda$, sets $Pre(\mathcal{S}_k)$, feedback laws $u_k^*(z_k(\cdot))$, $\forall k = [0, \ldots, k^*]$.
OUTPUT: Optimal value of the control action $u^*$.
1: Find the minimal value of the index $j$ for which $x \in Pre(\mathcal{S}_j)$.
2: Calculate $z(\cdot)$ from (8) by utilizing the knowledge of $\lambda$ and $x$.
3: From the $j$-th feedback law of the form of (15) find the region index $r$ for which $z(\cdot) \in \mathcal{R}_j^r$.
4: Calculate optimal control action by $u^* = F_j^r z(\cdot) + G_j^r$.
```

Theorem 3.14. The minimum-time adaptive controller calculated by Algorithm 1 and applied to a PWA system (1) in a receding horizon control fashion according to Algorithm 2 guarantees that all states are pushed towards $\mathcal{T}_{\text{set}}$ in the minimal possible number steps.

Algorithm 3.15. Assume the initial state x is contained in the set $Pre(\mathcal{S}_j)$ from which it takes j steps to reach $\mathcal{T}_{\text{set}}$ according to Theorem 3.8. The control law identified by Algorithm 2 will drive the states into the set $Pre(\mathcal{S}_{j-1})$ in one time step. Therefore, the states will enter $\mathcal{T}_{\text{set}}$ in j steps when Algorithm 2 called repeatedly at each sampling instance.

Therefore, the control law calculated by Algorithm 1 acts as an adaptive controller when implemented on-line and the measurements of λ_k can be used to update the process model. The controller is calculated off-line in a form of a look-up table, reducing the on-line implementation effort to a sequence of simple set-membership tests.

4. EXAMPLE

In this section we illustrate the application of Algorithm 1 to a modified version of the periodic PWA system of (Bemporad and Morari, 1999). The dynamics of such a system is given by

$$x_{k+1} = \begin{cases} A_1 x_k + B u_k & \text{IF } x_k^{(1)} < 0 \\ A_2 x_k + B u_k & \text{IF } x_k^{(1)} \geq 0, \end{cases} \tag{16}$$

where $x_k^{(1)}$ denotes the first coordinate of the state vector x_k. State-update matrices for each of the two modes of the PWA system are given by

$$A_i = w \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) \\ \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{17}$$

with $\alpha_1 = -\pi/3$ and $\alpha_2 = \pi/3$, respectively. We assume that the value of the parameter w is unknown at the time of the synthesis of the control law, but it is bounded by $0.7 \leq w \leq 1$. The vertices A_i^1, A_i^2 in (2) can be obtained by evaluating A_i from (17) for boundary values of this interval.

We have then implemented Algorithm 1 by employing the Multi-Parametric Toolbox (Kvasnica et al., 2004) to calculate projections in Lemma 3.2 and to solve (11)

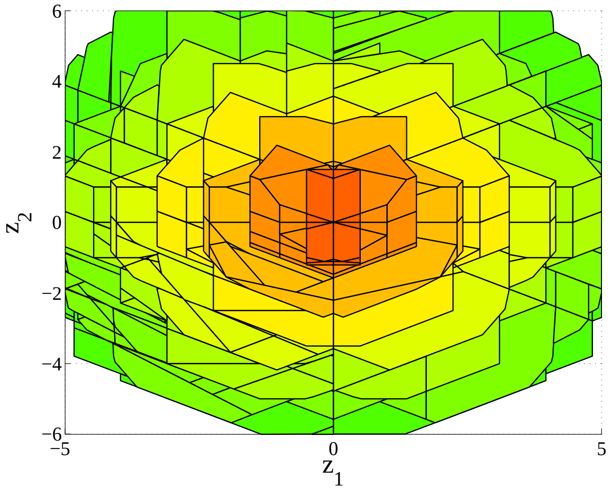


Fig. 1. Regions $\mathcal{R}_k$ of the PWA feedback laws $u_k^*(z(\cdot))$. By the same colors are depicted regions from the same iteration k. More reddish regions correspond to lower values of k.

parametrically. YALMIP (Löfberg, 2004) was used to formulate (11) in a user-friendly fashion. Problem 2.1 was then solved with $\mathcal{X} = \{x \mid -5 \leq x \leq 5\}$, $\mathcal{U} = \{u \mid -1 \leq u \leq 1\}$, and $\mathcal{T}_{\text{set}} = \{x \mid -0.5 \leq x \leq 0.5\}$. Algorithm 1 has converged at iteration 8 after 235 seconds, generating 5 PWA feedback laws of the form (14), which are parameterized in the information variable $z(x, \lambda, d)$. Regions over which these control laws are defined are depicted in Figure 1.

In the spirit of adaptive control we have investigated the behavior of the proposed minimum-time scheme when the value of the uncertainty w fluctuates over time. To do that we have generated a random sequence of w_k satisfying $0.7 \leq w_k \leq 1$ and subsequently performed closed-loop simulations starting from the initial state $x_0 = [0, \ -5]^T$. Profile of the uncertainty, together with the optimal control moves and the closed-loop evolution of system states are depicted, respectively, in Figures 2, 4, and 3. As can be seen from the plots, the minimum-time controllers adapts itself to the current measurements of the parameter w and drives all system states towards the terminal set despite quite substantial variations of the value of the uncertainty.

5. CONCLUSIONS

The paper showed how to synthesize robust adaptive minimum-time controllers for the class of PWA systems affected by parametric uncertainties. The control policy is synthesized in such a way that all system states are pushed towards a prescribed terminal set in the least possible number of time steps for all admissible values of the uncertainty. The controller is calculated using parametric optimization which results in a feedback law in a form of a look-up table, parameterized in the influence of the uncertain parameters. On-line measurements of the uncertainty can thus be used to further optimize for performance when the controller is implemented in the receding horizon fashion.

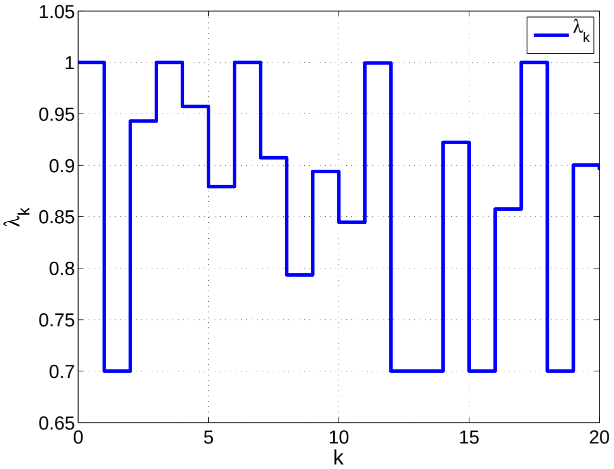


Fig. 2. Time-varying fluctuations of the value of the uncertainty w.

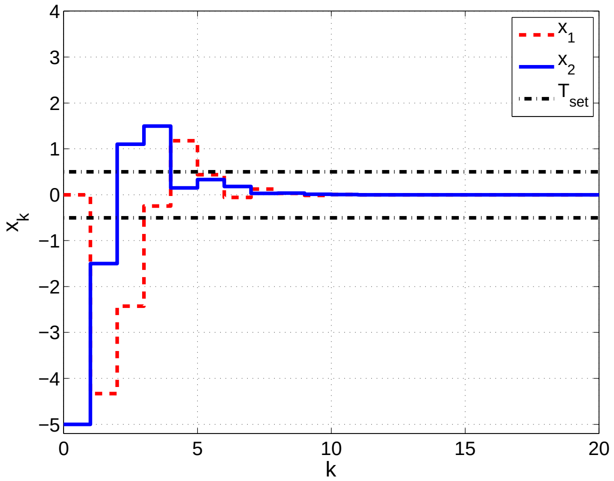


Fig. 3. Closed-loop profiles of x.

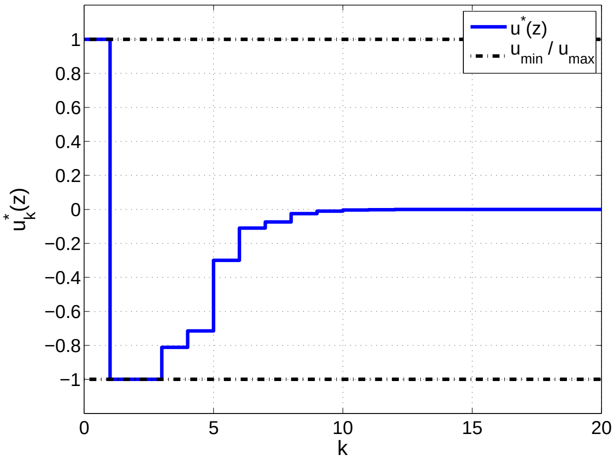


Fig. 4. Closed-loop profiles of u^*.

6. ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/4055/07, and 1/0095/11.

This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

M. Baotić, F. J. Christophersen, and M. Morari. A new Algorithm for Constrained Finite Time Optimal Control of Hybrid Systems with a Linear Performance Index. In *European Control Conference*, Cambridge, UK, September 2003.

M. Baric, Sasa V. Rakovic, Th. Besselmann, and M. Morari. Max-Min Optimal Control of Constrained Discrete-Time Systems. In *IFAC World Congress*, July 2008.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3): 407–427, March 1999.

Th. Besselmann, J. Löfberg, and M. Morari. Explicit MPC for systems with linear parameter-varying state transition matrix. In *International Federation of Automatic Control World Congress*, July 2008.

R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control: The Thinking Man's GPC.* International Series in Systems and Control Engineering. Prentice Hall, 1990.

F. Blanchini. Minimum-Time Control for Uncertain Discrete-Time Linear Systems. In *Proc. of the Conf. on Decision & Control*, pages 2629–2634, Tucson, AZ, USA, December 1992.

F. Borrelli. Constrained Optimal Control of Linear and Hybrid Systems. In *Lecture Notes in Control and Information Sciences*, volume 290. Springer, 2003.

F. Borrelli, M. Baotić, A. Bemporad, and M. Morari. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *Proc. of the American Control Conference*, Denver, Colorado, USA, June 2003.

P. Dostál, M. Bakošová, and V. Bobál. An approach to adaptive control of a cstr. *Chemical Papers*, 58(3):184–190, 2004.

T. Geyer, F. D. Torrisi, and M. Morari. Optimal Complexity Reduction of Piecewise Affine Models Based on Hyperplane Arrangements. In *Proc. on the American Control Conference*, pages 1190–1195, Boston, Massachusetts, USA, June 2004.

P. Grieder, M. Kvasnica, M. Baotic, and M. Morari. Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica*, 41, issue 10:1683–1694, Oct. 2005.

W. P. M. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

S. S. Keerthi and E. G. Gilbert. Computation of Minimum-Time Feedback Control Laws for Discrete-Time Systems with State-Control Constraints. *IEEE Trans. on Automatic Control*, AC-32:432–435, May 1987.

E. C. Kerrigan and D. Q. Mayne. Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proc. 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada, USA, December 2002.

M. Kvasnica, P. Grieder, M. Baotic, and M. Morari. Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004. Available from http://control.ee.ethz.ch/~mpt.

J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004. Available from http://control.ee.ethz.ch/~joloef/yalmip.php.

D. Q. Mayne and S. Raković. Model predictive control of constrained piecewise affine discrete–time systems. *Int. J. of Robust and Nonlinear Control*, 13(3):261–279, April 2003.

D. Q. Mayne and W. R. Schroeder. Robust time-optimal control of constrained linear Systems. *Automatica*, 33 (12):2103–2118, December 1997.

E. Mosca. *Optimal, Predictive, and Adaptive Control.* Prentice Hall, Englewood Cliffs, New York, 1995.

S.V. Raković, P. Grieder, M. Kvasnica, D.Q. Mayne, and M. Morari. Computation of Invariant Sets for Piecewise Affine Discrete Time Systems subject to Bounded Disturbances. In *Proceeding of the 43rd IEEE Conference on Decision and Control*, pages 1418–1423, Atlantis, Paradise Island, Bahamas, December 2004.

E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2): 346–358, April 1981.

G. M. Ziegler. *Lectures on Polytopes.* Graduate Texts in Mathematics. Springer-Verlag, 1995.

Comments – Remarks

Separation Functions used in Simplification of Explicit MPC Feedback Laws

Michal Kvasnica [*],[1], **Ivana Rauová** [*], and **Miroslav Fikar** [*]

[*] *Institute of Automation, Information Engineering and Mathematics,
Slovak University of Technology in Bratislava, 812 37 Bratislava,
Slovakia*

Abstract:
In this paper we focus on the problem of memory storage reduction. We consider explicit MPC feedback laws for linear systems. A controller defined by a continuous Picewise Affine (PWA) function is simplified using separation function. In case that state lies within a saturated region, optimal control value is given by the sign of the separator. Therefore, it is enough to store and evaluate only the unconstrained regions and separator. Construction efficiency of such separators is provided on a large number of problems, even on very complex explicit MPC solutions.

Keywords: model predictive control, constrained control, parametric optimization

1. INTRODUCTION

Real-time implementation of MPC in the Receding Horizon fashion (RHMPC) boils down to repetitively solving a given optimization problem for a given value of the initial condition x. Alternatively, as shown in Bemporad et al. (2002), one can precompute the explicit RHMPC optimizer $u^* = \kappa(x)$ as a PWA function which is defined over a set of polytopic regions. Computing u^* on-line then reduces to a mere function evaluation. However, the number of regions of $\kappa(x)$, which is problem-dependent, tends to be large, easily exceeding the storage capacity of a selected implementation platform. Therefore it is important to keep the number of regions as low as possible.

One approach is to construct a sub-optimal replacement function $\widetilde{\kappa}(x) \approx \kappa(x)$ of substantially lower complexity, see e.g. Bemporad and Filippi (2003); Johansen and Grancharova (2003); Cychowski and O'Mahony (2005); Scibilia et al. (2009). Another line of research is concerned with finding the replacement $\widetilde{\kappa}(x)$ which is simpler than the original function, but maintains the equivalence $\kappa(x) = \widetilde{\kappa}(x)$ for all points x of interest. In Geyer et al. (2008) regions are merged if they share the same expression for the control law. If the PWA function $\kappa(x)$ is convex (or if there exists a convex function $\eta(x)$, defined over the same regions), then the method of Baotic et al. (2008) can be used to reduce the required memory storage. If $\kappa(x)$ is nonconvex, but continuous, its lattice representation (Wen et al., 2009) can be built, again decreasing the memory consumption.

In our previous work (Kvasnica and Fikar, 2010), the performance-lossless replacement $\widetilde{\kappa}(x)$ was constructed by only considering the regions of $\kappa(x)$ where the control action is not saturated. This can considerably reduce the complexity as the number of unsaturated regions is usually significantly smaller compared to the total number of underlying polytopes over which $\kappa(x)$ is defined.

In this paper we improve our previous method and propose to use separation functions. At first, we divide the regions of $\kappa(x)$ into three categories: unsaturated regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ where $u_{\min} < \kappa(x) < u_{\max}$, regions $\mathcal{R}_{\mathcal{I}_{\max}}$ where $\kappa(x) = u_{\max}$, and regions $\mathcal{R}_{\mathcal{I}_{\min}}$ where $\kappa(x) = u_{\min}$. We then search for a function $p(x)$ which separates the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$. When found, the on-line implementation of $u^* = \kappa(x)$ can be substantially simplified by only requiring the storage of unsaturated regions. If $x \notin \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ for a given x, the function $p(x)$ is evaluated, and its sign then governs whether $u^* = u_{\max}$ or $u^* = u_{\min}$. The problem then becomes to find a *simple* separator $p(x)$, such that it is easy to evaluate on-line and requires small amount of memory for its storage. Two choices are proposed: $p(x)$ is either a continuous multivariate polynomial, or a (possibly discontinuous) PWA function encoded as a binary tree. The challenge of finding $p(x)$ stems from the fact that the sets to be separated are in general non-convex. We show how to solve such a separation problem by either solving linear or mixed-integer linear problems. In the case of polynomial separation, additional certification is needed, which can be implemented by finding the roots of a given polynomial. Existence of the separator $p(x)$ then guarantees that the replacement feedback $\widetilde{\kappa}(x)$ will always consists of the unsaturated regions of $\kappa(x)$ only. Such a direct guarantee cannot be given for the clipping-based method of Kvasnica and Fikar (2010). A large case study is provided to confirm viability of the approach.

2. DEFINITIONS

A finite set of n elements $\mathcal{I} := \{\mathcal{I}_1, \dots, \mathcal{I}_n\}$ will be denoted as $\{\mathcal{I}_i\}_{i=1}^n$ and its cardinality by $|\mathcal{I}|$. A polytope is the bounded convex intersection of c closed affine half-spaces, i.e. $\mathcal{R} := \{x \in \mathbb{R}^{n_x} \mid Fx \leq g\}$. We call the collection of polytopes $\{\mathcal{R}_i\}_{i=1}^R$ the *partition* of a polytope $\mathcal{R}$ if $\mathcal{R} = \bigcup_{i=1}^R \mathcal{R}_i$, and $\operatorname{int}(\mathcal{R}_i) \cap \operatorname{int}(\mathcal{R}_j) = \emptyset$ for all $i \neq j$. Each polytope $\mathcal{R}_i$ will be referred to as the *region* of the partition. Function $\kappa(x) : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_z}$ with $x \in \mathcal{R} \subset$

[1] Corresponding author, e-mail: michal.kvasnica@stuba.sk

$\mathbb{R}^{n_x}$, $\mathcal{R}$ being a polytope, is called piecewise affine over polytopes if $\{\mathcal{R}_i\}_{i=1}^R$ is the partition of $\mathcal{R}$ and

$$\kappa(x) := K_i x + L_i \quad \forall x \in \mathcal{R}_i, \tag{1}$$

with $K_i \in \mathbb{R}^{n_z \times n_x}$, $L_i \in \mathbb{R}^{n_z}$, and $i = 1, \dots, R$. PWA function $\kappa(x)$ is continuous if $K_i x + L_i = K_j x + L_j$ holds $\forall x \in \mathcal{R}_i \cap \mathcal{R}_j$, $i \neq j$.

3. EXPLICIT MODEL PREDICTIVE CONTROL

We consider the class of discrete-time, stabilizable linear time-invariant systems

$$x_{k+1} = Ax_k + Bu_k, \tag{2}$$

which are subject to polytopic constraints $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$. Assume the following constrained finite-time optimal control problem:

$$\min_{U_N} \sum_{k=0}^{N-1} x_{k+1}^T Q_x x_{k+1} + u_k^T Q_u u_k \tag{3a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k,\ x_k \in \mathcal{X},\ u_k \in \mathcal{U}, \tag{3b}$$

where x_k and u_k denote, respectively, state and input predictions over a finite horizon N, given the initial condition x_0. It is assumed that $Q_x = Q_x^T \succeq 0$, $Q_u = Q_u^T \succ 0$ in (3a), i.e. that (3) is a strictly convex QP. The receding horizon MPC feedback then becomes $u^*(x_0) = [\mathbb{1}\ \mathbb{0}\ \cdots\ \mathbb{0}]U_N^*$, where the optimal vector $U_N^* := [u_0^T, \dots, u_{N-1}^T]^T$ can be found by solving (3) as a QP for a given value of the initial condition x_0. For problems of modest size (typically for $n_x < 5$), it is also possible to characterize the optimal feedback $u^*(x_0)$ explicitly as a PWA function of x_0 (Bemporad et al., 2002) by solving (3) as a *parametric quadratic program* (pQP).

Theorem 3.1. (Bemporad et al. (2002)). The RHMPC feedback $u^*(x_0)$ for problem (3) is given by $u^*(x_0) = \kappa(x_0)$ where: (i) the set of feasible initial conditions $\Omega := \{x_0 \mid \exists u_0, \dots, u_{N-1} \text{ s.t. (3b) hold}\}$ is a polytope; (ii) $\kappa(x_0) : \Omega \mapsto \mathcal{U}$ is a continuous PWA function defined over R regions $\mathcal{R}_i$, $i = 1, \dots, R$; (iii) $\mathcal{R}_i$ are full-dimensional polytopes $\mathcal{R}_i = \{x \mid F_i x \leq g_i\}$; and (iv) $\{\mathcal{R}_i\}_{i=1}^R$ is a partition of Ω.

In the next section we show how to replace the feedback law $u^*(x_0) = \kappa(x_0)$ by a different function $\widetilde{\kappa}(x_0)$ which requires significantly less memory for its implementation in real-time arrangement and maintains the equivalence $\widetilde{\kappa}(x_0) \equiv \kappa(x_0)\ \forall x \in \Omega$. The procedure is applicable to generic PWA functions $\kappa(x)$ as long as they are continuous and all their regions $\mathcal{R}_i$ are full-dimensional polytopes. The scope of this work therefore extends to cases where 1- or ∞-norms are used in (3a), or when tracking of a non-zero reference is achieved by a suitable augmentation of the state vector.

4. COMPLEXITY REDUCTION VIA SEPARATION

By Theorem 3.1 we have that $\kappa(x)$ is a continuous PWA function defined over convex regions $\mathcal{R}_i$, union of which is the convex polytope Ω. Denote by $\overline{\kappa}$ and $\underline{\kappa}$ the maximal and minimal values which $\kappa(x)$ attains over its domain Ω

$$\overline{\kappa}_i = \max\{K_i x + L_i \mid x \in \mathcal{R}_i\},\ i = 1, \dots, R, \tag{4a}$$

$$\underline{\kappa}_i = \min\{K_i x + L_i \mid x \in \mathcal{R}_i\},\ i = 1, \dots, R, \tag{4b}$$

with $\overline{\kappa} = \max\{\overline{\kappa}_1, \dots, \overline{\kappa}_R\}$, $\underline{\kappa} = \min\{\underline{\kappa}_1, \dots, \underline{\kappa}_R\}$. Then the regions of $\kappa(x)$ can be classified as follows.

(1) If $K_i = 0$ and $L_i = \overline{\kappa}$, then region $\mathcal{R}_i$ is *saturated at the maximum*,
(2) if $K_i = 0$ and $L_i = \underline{\kappa}$, then region $\mathcal{R}_i$ is *saturated at the minimum*,
(3) otherwise the i-th region is *unsaturated.*

Denote by $\mathcal{I}_{\max}$ and $\mathcal{I}_{\min}$ the index lists of regions saturated at the maximum and minimum, respectively, and by $\mathcal{I}_{\text{unsat}}$ the index list of unsaturated regions. With this classification, the RHMPC feedback $\kappa(x)$ can be written as

$$\kappa(x) = \begin{cases} K_i x + L_i & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}, \\ \overline{\kappa} & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\max}}, \\ \underline{\kappa} & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\min}}. \end{cases} \tag{5}$$

Evaluation of $\kappa(x)$ for any $x \in \Omega$ is therefore a two-stage process. First, the index r of region $\mathcal{R}_r$ which contains x needs to be identified. Then, the function value of $\kappa(x)$ is either computed by $K_r x + L_r$ if $r \in \mathcal{I}_{\text{unsat}}$, or $\kappa(x) = \overline{\kappa}$ ($\kappa(x) = \underline{\kappa}$) if $r \in \mathcal{I}_{\max}$ ($r \in \mathcal{I}_{\min}$). Identification of the index r can either be done by searching through all regions $\mathcal{R}_i$, $i = 1, \dots, R$ sequentially, or by traversing a corresponding binary search tree (Tøndel et al., 2003). In either case, the required memory storage is proportional to the total number of regions R.

If the number of saturated regions is non-zero, a simpler representation of $\kappa(x)$ can in fact be obtained. Notice that, since the regions $\mathcal{R}_i$ are non-overlapping due to Theorem 3.1, for any $x \in \Omega$, $x \notin \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$, $\kappa(x)$ can only take two possible values: either $\kappa(x) = \overline{\kappa}$, or $\kappa(x) = \underline{\kappa}$. This fact can be exploited to derive a new PWA function $\widetilde{\kappa}(x)$ which maintains the equivalence $\widetilde{\kappa}(x) = \kappa(x)$ for all $x \in \Omega$, and requires less memory for its description compared to the memory footprint of $\kappa(x)$.

Proposition 4.1. Let a function $p(x) : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ which satisfies $p(x) > 0$ for all $x \in \mathcal{R}_{\mathcal{I}_{\max}}$ and $p(x) < 0$ for all $x \in \mathcal{R}_{\mathcal{I}_{\min}}$ be given. Define

$$\widetilde{\kappa}(x) = \begin{cases} K_i x + L_i & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}, \\ \overline{\kappa} & \text{if } p(x) > 0, \\ \underline{\kappa} & \text{if } p(x) < 0. \end{cases} \tag{6}$$

Then, for all $x \in \Omega$, $\widetilde{\kappa}(x) = \kappa(x)$.

Proof. Follows directly from (5) and from the definition of $p(x)$.

Given $p(x)$, $u^* = \kappa(x)$ can be evaluated by only looking at the unsaturated regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$. If $x \in \mathcal{R}_r$, $r \in \mathcal{I}_{\text{unsat}}$, then $u^* = K_r x + L_r$. Otherwise, based on the sign of $p(x)$, one either takes $u^* = \overline{\kappa}$ or $u^* = \underline{\kappa}$.

If $\kappa(x)$ is a continuous PWA function, then a possibly discontinuous separating function $p(x)$ always exists. Under continuity, the convex regions $\mathcal{R}_j$ and $\mathcal{R}_k$ cannot be adjacent for any $j \in \mathcal{I}_{\max}$, $k \in \mathcal{I}_{\min}$, and therefore they can always be separated. As will be evidenced later, a typical explicit RHMPC feedback laws $\kappa(x)$ contains a significantly smaller number of unsaturated regions as compared to the number of saturated ones, i.e. $|\mathcal{I}_{\text{unsat}}| \ll |\mathcal{I}_{\max}| + |\mathcal{I}_{\min}|$. Therefore $\widetilde{\kappa}(x)$ will require significantly less memory than $\kappa(x)$, and will be faster to evaluate too, if $p(x)$ is a "simple" separator of the two sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$.

Various types of $p(x)$ can be considered, either continuous (e.g. linear or polynomial), or discontinuous (e.g. piecewise linear or piecewise polynomial). In this work we have opted for the polynomial type of $p(x)$ and the problem which we aim at solving is formally stated as follows.

Problem 4.2. Given a RHMPC feedback law $u^* = \kappa(x)$ with $\kappa(x)$ as in (5), construct the replacement function (6) by finding the multivariate polynomial

$$p(x) := \sum_{i_1+\cdots+i_n \leq \delta} \alpha_{i_1,\ldots,i_\delta} x_1^{i_1} \cdots x_n^{i_n}, \tag{7}$$

of *minimum degree* $\delta_{\min}$ such that $p(x)$ strictly separates the sets of regions $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, i.e. $p(x) > 0\ \forall x \in \mathcal{R}_{\mathcal{I}_{\max}}$ and $p(x) < 0\ \forall x \in \mathcal{R}_{\mathcal{I}_{\min}}$.

Solving Problem 4.2 is, however, nontrivial, since the unions of polytopes, i.e. $\mathcal{R}_{\mathcal{I}_{\max}} = \{x \mid x \in \cup_i \mathcal{R}_i,\ i \in \mathcal{I}_{\max}\}$ and $\mathcal{R}_{\mathcal{I}_{\min}} = \{x \mid x \in \cup_j \mathcal{R}_j,\ j \in \mathcal{I}_{\min}\}$, can be non-convex, in general. Even deciding whether they are convex or not is hard (Bemporad et al., 2001).

4.1 Polynomial separation

Given are the (non-convex) sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, each of which consists of a finite number of polytopes $\mathcal{R}_k$. Denote by $\mathcal{V}_k$ the vertices of $\mathcal{R}_k$ and fix some integer $\delta \geq 1$ in (7). Then the *necessary* condition for the existence of a polynomial $p(x)$ which strictly separates $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ is that the following optimization problem is feasible:

$$\epsilon^* = \max_{\epsilon, \alpha_i} \epsilon \tag{8a}$$

$$\text{s.t. } p(v_i) \geq \epsilon, \quad \forall v_i \in \mathcal{V}_{\mathcal{I}_{\max}}, \tag{8b}$$

$$p(v_j) \leq -\epsilon,\ \forall v_j \in \mathcal{V}_{\mathcal{I}_{\min}}. \tag{8c}$$

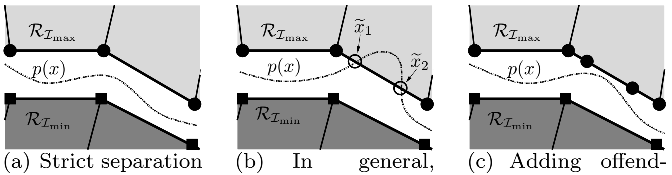
$$\epsilon \geq 0. \tag{8d}$$

The optimal value ϵ^* then denotes the maximal separation gap between the two sets of points $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$. Important to notice is that (8) is a linear program (LP) since, for some fixed argument $x = v_k$, $v_k \in \mathcal{V}_k$, $p(x)$ in (8b)–(8c) are linear functions of the coefficients α_i. If the LP (8) is infeasible, then no polynomial separator $p(x)$ of the form of (7) exists for a given degree δ.

If $\delta = 1$ in (8) then having $\epsilon^* > 0$ is also sufficient for the linear function $p(x) := \alpha_0 + \alpha_1 x$ to strictly separate the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ (Boyd and Vandenberghe, 2004). Consider therefore $\delta > 1$. If (8) is feasible with $\epsilon^* > 0$, then one of the two possible scenarios can occur. In an ideal case, solving for $p(x)$ from (8) by only considering separation of $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$ will also provide a separator for the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, as shown in Fig. 1(a). In a more general case, though, strict separation of vertices is not sufficient for $p(x)$ to separate *all points* from the associated sets, cf. Fig 1(b).

An additional certification step therefore has to be performed. At this point we remind that all regions of $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ are polytopes described by $\mathcal{R}_i = \{x \mid F_i x \leq g_i\}$. Consider the k-th facet of $\mathcal{R}_i$, i.e. $\{x \mid f_{i,k}x - g_{i,k} = 0\}$ where $f_{i,k}$ and $g_{i,k}$ are the k-th rows of the respective matrices F_i and g_i. Denote by $\widetilde{x}_{i,k}$ all (or some) solutions to the polynomial equation $p(x) = f_{i,k}x - g_{i,k}$ on domain $x \in \mathcal{R}_i$:

$$\widetilde{x}_{i,k} = \{x \mid p(x) - f_{i,k}x + g_{i,k} = 0,\ x \in \mathcal{R}_i\}. \tag{9}$$


$\mathcal{R}_{\mathcal{I}_{\max}}$
$p(x)$
$\mathcal{R}_{\mathcal{I}_{\min}}$
$\mathcal{R}_{\mathcal{I}_{\max}}$
$\widetilde{x}_1$
$p(x)$
$\widetilde{x}_2$
$\mathcal{R}_{\mathcal{I}_{\min}}$
$\mathcal{R}_{\mathcal{I}_{\max}}$
$p(x)$
$\mathcal{R}_{\mathcal{I}_{\min}}$

(a) Strict separation of vertices can sometimes imply strict separation of the associated sets.

(b) In general, separation of $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$ does not imply separation of $\mathcal{R}_{\mathcal{I}_{\max}}$ from $\mathcal{R}_{\mathcal{I}_{\min}}$.

(c) Adding offenders $\widetilde{x}_i$ to $\mathcal{V}_{\mathcal{I}_{\max}}$ and resolving (8) leads to a new separating polynomial $p(x)$.

Fig. 1. Sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, vertices $\mathcal{V}_{\mathcal{I}_{\max}}$ (circles) and $\mathcal{V}_{\mathcal{I}_{\min}}$ (squares), polynomial separator $p(x)$.

Clearly, if $\widetilde{x}_{i,k} = \emptyset\ \forall i \in \mathcal{I}_{\max} \cup \mathcal{I}_{\min}$ and $\forall k$, then $p(x)$ as a solution to (8) strictly separates $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ (cf. Figure 1(a)). On the other hand, the situation in Figure 1(b) corresponds to the case where there exist some points $\widetilde{x}_{i,k}$ for which the polynomial $p(x)$ intersects the k-th facet of the i-th region, i.e. when $\widetilde{x}_{i,k} \neq \emptyset$ for some i and k. In such a case, the existence of *any* such point $\widetilde{x}_{i,k}$ provides a certificate that $p(x)$ does not separate $\mathcal{R}_{\mathcal{I}_{\max}}$ from $\mathcal{R}_{\mathcal{I}_{\min}}$.

When at least one offending point $\widetilde{x}_{i,k}$ exists, it can be added to the corresponding set of vertices in (8b)–(8c). I.e., if $\widetilde{x}_{i,k} \neq \emptyset$ for some $i \in \mathcal{I}_{\max}$, then $\mathcal{V}_{\mathcal{I}_{\max}} = \mathcal{V}_{\mathcal{I}_{\max}} \cup \widetilde{x}_{i,k}$. Otherwise, if $i \in \mathcal{I}_{\min}$, then $\mathcal{V}_{\mathcal{I}_{\min}} = \mathcal{V}_{\mathcal{I}_{\min}} \cup \widetilde{x}_{i,k}$. Resolving the LP (8) with the updated list of vertices will then give a new polynomial $p(x)$ for which the certification is repeated, cf. Figure 1(c). If more offenders are found, they are added to the list of vertices and the procedure is repeated. Otherwise, an empty solution to (9) provides a certificate that $p(x)$ strictly separates $\mathcal{R}_{\mathcal{I}_{\max}}$ from $\mathcal{R}_{\mathcal{I}_{\min}}$, whereupon the procedure terminates. The discussed mechanism can be formally stated as Algorithm 1, reported next.

Algorithm 1 Construction of a polynomial separator $p(x)$

```
INPUT: Sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, polynomial degree $\delta$.
OUTPUT: Separating polynomial $p(x)$ as in (7).
 1: Get the lists of vertices $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$.
 2: repeat
 3:     Solve the LP (8) and obtain coefficients $\alpha_i$.
 4:     if $\epsilon^* > 0$ then
 5:         Compute the list of offending points $\widetilde{x}$ from (9).
 6:         Insert $\widetilde{x}$ to $\mathcal{V}_{\mathcal{I}_{\max}}$ or $\mathcal{V}_{\mathcal{I}_{\min}}$.
 7:     else
 8:         No strict separator $p(x)$ of degree $\delta$ exists, abort.
 9:     end if
10: until $\widetilde{x} \neq \emptyset$.
```

Remark 4.3. Vertex enumeration in Step 1 of Algorithm 1 is considered a hard problem in general. However, for the type of small-dimensional problems considered here, enumerating $\mathcal{V}$ does not pose any significant technical difficulty and the vertices can be easily computed e.g. by CDD (Fukuda, 1997) in a matter of seconds.

Remark 4.4. There is no theoretical guarantee that the iterations between Steps 2–10 will terminate in finite time. However, for more than 400 random problems reported in Section 5.2, the number of iterations newer exceeded 4.

Remark 4.5. The list of offending points $\widetilde{x}$ in Step 5 can be obtained by solving (9) in several ways. One option is to compute the real roots of the polynomial $p(x)$ −

$f_{i,k}x + g_{i,k} = 0$ numerically e.g. by using the package of Zeng (2004). Since such a method does not allow to restrict the offenders to a particular domain, the roots which violate $\widetilde{x}_{i,k} \in \mathcal{R}_i$ need to be excluded. Another option is to consider (9) as a feasibility problem with a nonlinear constraint. Nonlinear optimization routines, such as `fmincon` of MATLAB, can then be used to find at least one such offender for each region $\mathcal{R}_i$, provided it exists.

Solving Problem 4.2 involves finding a strict separator $p(x)$ of the minimum degree $\delta_{\min}$. This can be achieved e.g. by using bisection, i.e. by running Algorithm 1 multiple times for various values of δ until a feasible solution is obtained and δ is minimized.

4.2 Separation via binary trees

An another alternative is to separate the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ by a (possibly discontinuous) PWA function $p(x)$, as shown in different context by Fuchs et al. (2010). There the authors search for a separator $p(x)$ represented as a binary search tree. Each node k of the tree represents one linear separator of the form $p_k(x) := \alpha_{k,1}x + \alpha_{k,0}$. The task then becomes to find the coefficients such that $p_k(x)$ correctly separates as many elements of $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ as possible. The misclassified elements are then treated in a recursive fashion while building the tree. The search for $p_k(x)$ is then formulated and solved as a mixed-integer linear program

$$\min \left|\sum R_i - \sum L_j\right| + \sum |R_i + L_j - 1| \tag{10a}$$
$$\text{s.t. } R_i = 1 \Leftrightarrow \{p_k(x) \geq \underline{\epsilon}\ \forall x \in \mathcal{R}_i,\ i \in \mathcal{I}_{\max}\}, \tag{10b}$$
$$L_j = 1 \Leftrightarrow \{p_k(x) \leq -\underline{\epsilon}\ \forall x \in \mathcal{R}_j,\ j \in \mathcal{I}_{\min}\}, \tag{10c}$$

where $\underline{\epsilon} > 0$ is a given minimal separation gap introduced to avoid the trivial solution $\alpha_{k,1} = \alpha_{k,0} = 0$. Binary variables R_i (L_j) denote whether or not the corresponding region of $\mathcal{R}_{\mathcal{I}_{\max}}$ ($\mathcal{R}_{\mathcal{I}_{\min}}$) is correctly classified by $p_k(x)$, while minimizing the number of incorrectly separated regions by (10a). The logic equivalence rules in (10b)–(10c) can be cast as mixed-integer inequalities. The key advantage over the polynomial separation method discussed previously stems from the fact that, by using duality theory, the classification does not require enumerating the vertices of the corresponding regions. The technical details can be found in Fuchs et al. (2010).

Moreover, since a linear separator is sought in each node, no a-posteriori certification step is necessary and hence there is a theoretical guarantee of a finite-time convergence of the tree-building procedure. The crucial downside, however, is that a total of $|\mathcal{I}_{\max}| + |\mathcal{I}_{\min}|$ binaries needs to be introduced. If the number exceeds ~ 700 (which is considered a small case by our standards), the size of the MILP (10) becomes prohibitive to be solved even using state-of-the-art solvers, such as CPLEX.

4.3 Multi-input case

So far we have considered replacing the RHMPC feedback law $\kappa(x)$ by a different function $\widetilde{\kappa}(x)$ of the form of (6), which consists of the unsaturated regions of $\kappa(x)$ and the separator $p(x)$. If $\kappa(x) : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_u}$ is such that $n_u > 1$ in (2), then one can proceed by decomposing $\kappa(x)$ into individual PWA functions $\kappa_j(x) := k_{i,j}x + l_{i,j}$ if $x \in \mathcal{R}_i$, where $k_{i,j}$, $l_{i,j}$ are the j-th rows of K_i and L_i, respectively. Then a set of $j = 1, \ldots, n_u$ polynomial separators $p_j(x)$ can then be obtained by running Algorithm 1 n_u times for different polytopic sets $\mathcal{R}_{\mathcal{I}_{\max,j}}$ and $\mathcal{R}_{\mathcal{I}_{\min,j}}$. Here, the index sets $\mathcal{I}_{\max,j}$ and $\mathcal{I}_{\min,j}$ are obtained based on the scalarized version of (4), i.e.

$$\overline{\kappa}_{i,j} = \max\{k_{i,j}x + l_{i,j} \mid x \in \mathcal{R}_i\},\ i = 1, \ldots, R, \tag{11a}$$
$$\underline{\kappa}_{i,j} = \min\{k_{i,j}x + l_{i,j} \mid x \in \mathcal{R}_i\},\ i = 1, \ldots, R, \tag{11b}$$

with $\overline{\kappa}_j = \max\{\overline{\kappa}_{i,j}\}_{i=1}^R$, $\underline{\kappa}_j = \min\{\underline{\kappa}_{i,j}\}_{i=1}^R$. Naturally, different index sets of unsaturated regions, i.e. $\mathcal{I}_{\text{unsat},j}$, will be obtained for different values of j. Even though the total number of regions of $\widetilde{\kappa}(x)$ is then $\sum_{j=1}^{n_u} |\mathcal{I}_{\text{unsat},j}|$, significant reduction of complexity can still be achieved if $|\mathcal{I}_{\text{unsat},j}| \ll R$ for all $j \in [1, \ldots, n_u]$.

4.4 Complexity analysis

Evaluation of $\widetilde{\kappa}(x)$ as in (6) for a given value of the vector x first requires to asses whether $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$. Searching through the regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ sequentially can answer this query in $\mathcal{O}(|\mathcal{I}_{\text{unsat}}|)$ time, while the binary search tree approach of Tøndel et al. (2003) can provide the answer in $\mathcal{O}(\log_2 |\mathcal{I}_{\text{unsat}}|)$ time. Both approaches require the storage of the unsaturated regions, hence their memory footprint is $\mathcal{O}(|\mathcal{I}_{\text{unsat}}|)$. If $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$, the index r of region $\mathcal{R}_r$ is returned, whereupon the value of $\widetilde{\kappa}(x)$ is given by $K_r x + L_r$. If $x \notin \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$, then the value of the separator $p(x)$ is obtained and its sign is used in (6). The memory and computation requirements associated with storing and evaluating $p(x)$ online is insignificant compared to the description of regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$.

Other approaches can be used to derive the replacement function $\widetilde{\kappa}(x)$. The lattice representation (LR) of Wen et al. (2009) converts the original function $\kappa(x)$ into a series of *min*/*max* operations over the functions $K_i x + L_i$, eliminating the need to store the underlying regions $\mathcal{R}_i$. Evaluation of such a lattice description requires $\mathcal{O}(R_{\text{unique}}^2)$ operations, where R_{unique} is the number of regions where the feedback law is unique. The memory storage is also proportional to $\mathcal{O}(R_{\text{unique}}^2)$, however the constant term in the big-O formulation is small due to the fact that only the matrices K_i and L_i need to be stored. The clipping-based procedure (Kvasnica and Fikar, 2010) removes all saturated regions and replaces them by “extensions” of the unsaturated ones. In the best case, $\widetilde{\kappa}(x)$ is then defined over $|\mathcal{I}_{\text{unsat}}|$ regions, while in the worst case the number of regions remains unchanged. On average, $\widetilde{\kappa}(x)$ consists of $1.3|\mathcal{I}_{\text{unsat}}|$ regions. The memory and runtime requirements of such a scheme are proportional to this figure.

5. EXAMPLES

5.1 Illustrative example

Consider a 2-state 1-input system given by

$$x^+ = \begin{bmatrix} 0.755 & 0.680 \\ 0.651 & -0.902 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.825 \\ -0.139 \end{bmatrix} u, \tag{12}$$

which is subject to constraints $\mathcal{X} = \{[\begin{smallmatrix}x_1\\x_2\end{smallmatrix}] \mid -10 \le [\begin{smallmatrix}x_1\\x_2\end{smallmatrix}] \le 10\}$ and $\mathcal{U} = \{u \in \mathbb{R} \mid -1 \le u \le 1\}$. The MPC problem (3) was formulated with prediction horizon $N = 10$, $Q_x = \mathbb{1}$ and $Q_u = 1$ and solved as a parametric QP according to Theorem 3.1. Using the MPT Toolbox (Kvasnica et al., 2004), the explicit RHMPC feedback $\kappa(x)$ was obtained in 4 seconds[2] as a PWA function defined over 225 regions shown in Fig. 2. The partition of $\kappa(x)$ consists of 29 unsaturated regions, 98 regions where $\kappa(x) = 1$, and 98 regions where $\kappa(x)$ is saturated at -1.

As can be clearly see from the shape of the sets in Figure 2(a), no linear separation between $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ can be found. A polynomial separator $p(x) = -x_1 - x_2 - 0.0011x_1^3 - 0.254x_2^3$ of the minimal degree $\delta_{\min} = 3$ was then found by applying bisection in conjunction with Algorithm 1. The algorithm converged within of two iterations. The vertices in Step 1 were computed by CDD in 0.01 seconds. Coefficients of the polynomial were obtained by solving the LP (8), which only took 0.1 seconds using CPLEX. The subsequent certification check in Step 5 was implemented by solving (9) using `fmincon`, which took 1.1 seconds.

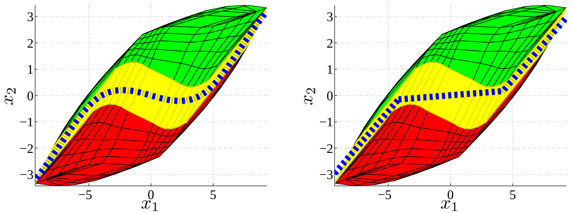
A binary separation tree can also be constructed by recursively solving MILP problems (10). For the sets depicted in Figure 2, the procedure has generated a tree consisting of three nodes: $p_1(x) = 0.084x_1 - x_2 - 0.049$, $p_2(x) = 0.428x_1 - x_2 - 1.393$, $p_3(x) = 0.428x_1 - x_2 + 1.393$. The tree is rooted at $p_1(x)$, with $p_2(x)$ visited if $p_1(x) < 0$. Otherwise, $p_3(x)$ is evaluated and its sign is used in (6). The total runtime of MILPs (10) was 9.1 seconds using CPLEX.

The total memory footprint of $\kappa(x)$ (which consists of the regions $\mathcal{R}_i$ and the feedback laws $K_i x + L_i$) with 225 regions is 27 kilobytes. On the other hand, by devising the polynomial or a tree separator $p(x)$, the storage requirements of $\widetilde{\kappa}(x)$ is a mere 3.5 kiloytes. Here, the unsaturated regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ contribute by 2.8 kB, the associated feedback laws by 0.7 kB, and the memory footprint is just 16 bytes for the polynomial separator, and 36 bytes for the binary tree. It follows that complexity of the on-line implementation of the RHMPC feedback law can be reduced by a factor of 7.7 when using the modified feedback $\widetilde{\kappa}(x)$ instead of the original function $\kappa(x)$.

Enlarging the prediction horizon to $N = 15$ leads $\kappa(x)$ with 489 regions, 39 of which are unsaturated. For this larger case, the minimal degree of the separating polynomial $p(x)$ is again $\delta_{\min} = 3$ and its coefficients can be found by Algorithm 1 in 1.5 seconds. The tree-type of separator which guarantees a minimal separation gap $\underline{\epsilon} = 1 \cdot 10^{-3}$ in (10) consists of 11 nodes and the total runtime of the 11 MILPs (10) was 522.7 seconds. This time, the memory footprint of $\kappa(x)$ is 58.2 kB, while $\widetilde{\kappa}(x)$ only occupies 4.8 kB, a reduction by factor of 12.

5.2 Large-scale analysis

Next, we have analyzed a large number of random RHMPC feedback laws $\kappa(x)$ generated by solving problem (3) for randomly selected LTI systems with 2 to 3 states, and 1 to 2 inputs. 100 random cases were considered for each n_x/n_u category. For each PWA function $\kappa(x)$ we have constructed the replacement $\widetilde{\kappa}(x)$ as in (6). Polynomial (7), as well as tree-type separators $p(x)$ were considered.



(a) Polynomial separator of degree 3.

(b) PWA separator encoded as a binary tree with 3 nodes.

Fig. 2. Sets $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ (yellow), $\mathcal{R}_{\mathcal{I}_{\max}}$ (red), $\mathcal{R}_{\mathcal{I}_{\min}}$ (green), and two possible separators $p(x)$.

Purpose of such a large-scale analysis is to confirm the main two assertions behind this work. First, it verifies that the number of unsaturated regions is indeed considerably smaller compared to the number of saturated ones, i.e. that $|\mathcal{I}_{\text{unsat}}| \ll |\mathcal{I}_{\max}| + |\mathcal{I}_{\min}|$ often holds in practice. Second, it shows that low degrees of the polynomial separator $p(x)$ are typically sufficient to obtain a strict separation. Moreover, the analysis also shows how Algorithm 1 and the MILP procedure of Section 4.2 scale with increasing dimension of the problem size.

Based on the 400 random scenarios, Table 1 shows for how many cases a polynomial separator $p(x)$ of a given minimal degree $\delta_{\min}$ could be found by Algorithm 1. Although only degrees $\delta \le 5$ were considered due to practical reasons, the overall success was 99.5%. Important to notice is that a linear separator exists in a majority of cases. Such a $p(x)$ can be found by solving the LP (8) *without* the need to further certify the separation in Step 5. We remark that for all cases where $\delta = 1$ was sufficient, it never took more than 10 seconds to compute $p(x)$ by Algorithm 1, regardless of problem size, as reported in Table 2. CPLEX was used to solve the LPs (8).

Instances where higher-order polynomials $p(x)$ were necessary are further elaborated in Table 3, which also shows how the computation scales with increasing number of regions. In addition, the reported results confirm the conclusions of Remark 4.4, i.e. that the number of iterations in Algorithm 1 is minor in practice. Majority of the runtime of Alg. 1 consists of the time spent in Step 5, which was implemented using `fmincon`. The results show that the polynomial separation approach scales significantly better than the tree separation. Moreover, size of the MILP (10) quickly becomes of prohibitive for larger number of regions.

Finally, Table 4 reports the minimal, maximal, and average values of the achievable complexity reduction ratio, which is defined as ratio between the total number of regions of $\kappa(x)$ to the number of unsaturated regions, i.e. $\Delta = \frac{R}{|\mathcal{I}_{\text{unsat}}|}$. The results show that the number of unsaturated regions is indeed significantly smaller in practice. Therefore, the replacement function $\widetilde{\kappa}(x)$ (6), which only requires the storage of unsaturated regions, will typically be considerably simpler compared to the original RHMPC

[2] On a 2.4 GHz CPU with 2GB of RAM using MATLAB 7.8 and MPT 2.6.3

Table 1. Likelihood of existence of a polynomial separator $p(x)$ of degree $\delta_{\min}$ (100 cases in each category).

n_x/n_u	$\delta_{\min}=1$	$\delta_{\min}=3$	$\delta_{\min}=5$	Σ
2/1	94	6	–	100
2/2	90	9	1	100
3/1	83	15	–	98
3/2	63	37	–	100

Table 2. Complexity of construction of a linear separator $p(x) = \alpha_0 + \alpha_1 x$ for largest scenarios in respective categories.

n_x/n_u	No. of	Runtime of Alg. 1 [sec]	
	regions	Step 1	Step 3
2/1	505	0.3	0.1
2/2	865	0.4	0.1
3/1	5643	3.7	0.2
3/2	12651	8.2	0.2

Table 3. Complexity of construction of the separator $p(x)$ with $\delta_{\min} > 1$. The symbol † denotes that the MILP was too complex to be solved in 2 hours.

n_x/n_u	No. of	$\delta_{\min}$	No. of	Runtime [sec]	
	regions		iterations	Alg. 1	MILP (10)
2/1	225	3	1	1.0	2.9
	283	3	1	1.2	4.6
	493	3	1	1.7	35.1
	495	3	1	1.3	991.3
2/2	297	3	1	3.2	1.1
	541	3	2	17.3	7.7
	787	5	1	9.2	30.2
	949	3	1	20.1	†
3/1	384	3	2	22.1	14.2
	527	3	4	20.6	†
	1275	3	1	14.4	†
	2513	3	1	35.8	†
3/2	191	3	3	7.5	5.9
	449	3	2	13.5	63.6
	1396	3	4	73.3	†
	3933	3	1	40.8	†

Table 4. Minimal, average, and maximal values of the complexity reduction ratio.

n_x/n_u	$\Delta_{\min}$	Δ_{avg}	$\Delta_{\max}$
2/1	2.3	13.0	31.0
3/1	2.1	7.1	21.0
2/2	1.8	5.9	14.5
3/2	1.9	3.6	10.2

feedback law $\kappa(x)$. We remark that the additional memory due to the storage of $p(x)$ usually amounts to less that 100 bytes.

6. CONCLUSIONS

Given an explicit RHMPC feedback function $\kappa(x)$, we have shown how to construct its simpler replacement $\widetilde{\kappa}(x)$ which maintains the equivalence $\kappa(x) = \widetilde{\kappa}(x)$ for all $x \in \operatorname{dom} \kappa(x)$. The mechanism was based on devising a function $p(x)$, which separates the regions over which $\kappa(x)$ attains a saturated value. The replacement $\widetilde{\kappa}(x)$ then requires only the storage of the unsaturated regions of $\kappa(x)$, along with the separator $p(x)$. We have shown how to build a polynomial separator by solving linear optimization problems, followed by a certification step which requires solution to a polynomial equation. By means of a large case study we have demonstrated that the procedure scales well with increasing problem size, and that significant reduction of complexity can be achieved in general.

ACKNOWLEDGMENT

Authors gratefully acknowledge the contribution of the Scientific The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/4055/07, and 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Baotic, M., Borrelli, F., Bemporad, A., and Morari, M. (2008). Efficient On-Line Computation of Constrained Optimal Control. *SIAM Journal on Control and Optimization*, 47(5), 2470–2489.

Bemporad, A. and Filippi, C. (2003). Suboptimal explicit RHC via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications*, 117(1), 9–38.

Bemporad, A., Fukuda, K., and Torrisi, F.D. (2001). Convexity Recognition of the Union of Polyhedra. *Computational Geometry*, 18, 141–154.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Cychowski, M. and O'Mahony, T. (2005). Efficient off-line solutions to robust model predictive control using orthogonal partitioning. In *Proceedings of the 16th IFAC world congress*.

Fuchs, A., Jones, C., and Morari, M. (2010). Optimized Decision Trees for Point Location in Polytopic Data Sets - Application to Explicit MPC. In *American Control Conference*. Baltimore, USA.

Fukuda, K. (1997). *Cdd/cdd+ Reference Manual*. Available from www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html.

Geyer, T., Torrisi, F., and Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7), 1728–1740.

Johansen, T. and Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Trans. on Automatic Control*, 48, 810–815.

Kvasnica, M. and Fikar, M. (2010). Performance-lossless complexity reduction in explicit MPC. In *Conference on Decision and Control*, (accepted). Atlanta, USA.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/~mpt/.

Scibilia, F., Olaru, S., and Hovd, M. (2009). Approximate explicit linear MPC via delaunay tessellation. In *Proceedings of the 10th European Control Conference*. Budapest, Hungary.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5), 945–950.

Wen, C., Ma, X., and Ydstie, B.E. (2009). Analytical expression of explicit MPC solution via lattice piecewise-affine function. *Automatica*, 45(4), 910 – 917.

Zeng, Z. (2004). MultRoot—a MATLAB package for computing polynomial roots and multiplicities. *ACM Trans. Math. Softw.*, 30(2), 218–236.

Comments – Remarks

Real-Time Implementation of Model Predictive Control Using Automatic Code Generation

Michal Kvasnica [*,1], **Ivana Rauová** [*], and **Miroslav Fikar** [*]

[*] *Institute of Automation, Information Engineering and Mathematics, Slovak University of Technology in Bratislava, 812 37 Bratislava, Slovakia*

Abstract:
Model Predictive Control (MPC) represents one of the control concepts used in the process industry. The main advantage of the framework is ability to optimize behavior of the process while respecting physical and economical constraints. Implementing MPC in real time on low-cost hardware is challenging due to the inherent computational complexity. One way of solving a given MPC problem is to use parametric programming, which encodes the optimal control moves as a lookup table. Such tables can be processed with low computation resources and therefore allow MPC to be employed on low cost devices. Aim of this paper is to show a unique software tool which requires low human effort to design MPC problems and is capable to automatically generate real-time executable code for various target platforms.

Keywords: model predictive control, controller design, real-time implementation

1. INTRODUCTION

Model predictive control (MPC) is an attractive approach widely used in industry and academia to control a broad range of systems due to its ability to provide optimal performance while taking process constraint into account Maciejowski (2002). In their industrial survey, Qin and Badgewell (1997) report successful applications of MPC which are commercially available. In order to introduce feedback, MPC is traditionally implemented in the Receding Horizon fashion (RHMPC). Here, the MPC setup is first converted into a suitable optimization problem, which has the plant state x as a free parameter. Then, once new measurements of the state arrive, the optimization problem is solved and the optimized control input u^* is applied to the plant. The procedure then repeats at the next time steps with fresh state measurements. Therefore it is of crucial importance to be able to solve the optimization problem within of time frame bounded by the sampling time of the plant. Not being able to perform the optimization on time can result, in a better case, to loss of optimality, or, in the worse case, even to constraint violation and instability, with possible catastrophic consequences on the controlled plant. Therefore MPC was traditionally used only for slow processes where the time frame for the optimization to terminate is large enough.

If the sampling time decreases, or if less powerful control platforms are employed to perform the optimization online, additional care has to be taken to respect the hard real-time constraints. One approach to decrease the computational burden involved in obtaining the optimal control action u^* for a particular value of x is to "pre-compute" the optimal solution to a given optimization problem for all possible values of the state x using *parametric programming*. If all constraints of the optimization problem are linear, it can be shown Bemporad et al. (2002) that the optimal control can be found as an explicit function $u^*(x)$ mapping the states to the control inputs. This calculation is performed off-line and just once. Since the solution can be interpreted as a lookup table, the optimal control action to be obtained, on-line for various values of x, by a simple table lookup, which can be implemented efficiently even with low computational resources.

A successful application of such an approach to design of real-time control systems is determined by three main factors: (i) whether it is possible to construct the table off-line in an automated fashion; (ii) whether the table is reasonably large as not to exceed the memory capabilities of the control device; (iii) whether one can traverse the table for a particular value of x within one sampling instance; and (iv) whether the table lookup can be implemented using programming instructions which the control device understands.

In this paper we present a new version of the Multi-Parametric Toolbox (MPT) Kvasnica et al. (2004), and illustrate how it can be used for designing MPC controllers in lookup-table form in a user-friendly fashion. We discuss several methods which the toolbox implements in order to reduce the table size. These approaches allow one to export the lookup tables to target control devices with low memory storage. In addition, we present an overview of novel algorithms developed to speed up the table traversal and hence allow MPC to be applied to processes with fast dynamics. Finally, it is illustrated how the toolbox can be used to deploy MPC-based controllers by directly generating a real-time executable code. Control

[1] Corresponding author, e-mail: michal.kvasnica@stuba.sk

platforms supported by the toolbox range from DSP processors to programmable logic controllers (PLC). The primary objective of the toolbox is to make MPC control design and deployment an intuitive process which can be used by control engineers who do not necessarily posses full theoretical MPC expertise. By automating heavy computation and code generation tasks, the toolbox allows for a considerably simpler control design for real-life applications.

2. PARAMETRIC APPROACH TO MPC

We consider the class of discrete-time, stabilizable linear time-invariant systems given by the state-space representation

$$x(t+1) = Ax(t) + Bu(t), \tag{1a}$$

$$y(t) = Cx(t). \tag{1b}$$

Here, $x(t)$ is the state vector at time t, $x(t+1)$ is the successor state, $u(t)$ denotes the control input, and $y(t)$ is the system output. It is assumed that the variables are subject to lower/upper limits

$$\underline{x} \le x(t) \le \overline{x}, \quad \underline{u} \le u(t) \le \overline{u}, \quad \underline{y} \le y(t) \le \overline{y}. \tag{2}$$

For the system (1) consider now the constrained finite-time optimal control problem

$$\min_{U_N} \; \ell_N(x_N) + \sum_{k=0}^{N-1} \ell_k(x_k, y_k, u_k) \tag{3a}$$

$$\text{s.t. } x_0 = x(t), \tag{3b}$$

$$x_{k+1} = Ax_k + Bu_k, \tag{3c}$$

$$y_k = Cx_k, \tag{3d}$$

$$\underline{x} \le x_k \le \overline{x}, \; \underline{u} \le u_k \le \overline{u}, \; \underline{y} \le y_k \le \overline{y}, \tag{3e}$$

where x_k, u_k and y_k denote, respectively, the state, input, and output predictions at time instance $t+k$, initialized by the measurements of the current state $x(t)$. The prediction is carried out over a finite prediction horizon N. The aim is to find the vector $U_N := [u_0^T, u_1^T, \dots, u_{N-1}^T]^T$ of optimal control inputs which minimizes the cost function (3a). The form of the terminal state penalty ℓ_N and a stage cost $\ell_k(\cdot)$ depends on the particular control objective.

Different setups can be considered in practice Mikleš and Fikar (2007). *Regulation* problems have the aim to control the system such a way that the distance between the predicted states x_k (or the predicted outputs y_k) and the origin of the system is minimized, while simultaneously minimizing the control effort. This requirement can be translated to using $\ell_N = \|Q_N x_N\|_p$ and $\ell_k = \|Q_x x_k\|_p + \|Q_u u_k\|_p$ for state regulation, or to $\ell_N = 0$ and $\ell_k = \|Q_y y_k\|_p + \|Q_u u_k\|_p$ for output regulation. Here, Q_N, Q_x, Q_y and Q_u are penalty matrices used to tune for performance. Moreover, $\|\cdot\|_p$ represents a standard weighted vector p-norm, i.e. $\|Pz\|_1 = \sum_i |P_i z_i|$, $\|z\|_2 = z^T P z$, and $\|Pz\|_\infty = \max_i |P_i z_i|$ for some vector z and a matrix P.

In *tracking* problems the aim is to control the system such a way that the distance between the predicted states x_k (or outputs y_k) and a pre-scribed reference point is minimized, while simultaneously minimizing the increments of the control effort, i.e. $\ell_N = \|Q_N(x_N - x_{\text{ref}})\|_p$ and $\ell_k = \|Q_x(x_k - x_{\text{ref}})\|_p + \|Q_u \Delta u_k\|_p$ for state tracking, or $\ell_N = 0$ and $\ell_k = \|Q_y(y_k - y_{\text{ref}}\|_p + \|Q_u \Delta u_k\|_p$ for output tracking where $\Delta u_k = u_k - u_{k-1}$. Minimizing over this quantity is often recommended because it introduces integral action and hence mitigates the steady-state offset.

In RHMPC, the optimal sequence U_N^* is calculated by solving (3) for a given value of $x(t)$. Subsequently, only u_0^* is extracted from U_N^* and it is applied to the plant. At the next time instance the procedure is repeated again for a fresh measurements $x(t)$, hence introducing feedback into the MPC scheme. Since only u_0^* is required at each time step, the RHMPC feedback is given by

$$u_0^*(x(t)) = [I_{n_u} \; 0_{n_u} \; \cdots \; 0_{n_u}] U_N. \tag{4}$$

Since all constraints in (3) are linear, the optimization problem can be translated (see e.g. Bemporad et al. (2002)) into Quadratic Program (QP) if $p = 2$ or to a Linear Program (LP) if $p = 1$ or $p = \infty$. In order to speed up the task of obtaining $u_0^*(x(t))$ for a given value of the measurements $x(t)$, it is nowadays a standard practice to pre-compute the optimal solution *for all possible* initial conditions $x(t)$ by solving problem (3) as a parametric QP (pQP) or a parametric LP (pLP).By exploiting the fact that $x(t)$ enters the constraints linearly, the optimal solution $u_0^*(x(t))$ is a lookup table, which encodes a piecewise affine (PWA) dependence of the optimal control input on the state $x(t)$:

Theorem 2.1. (Bemporad et al. (2002)). The RHMPC feedback $u_0^*(x(t))$ for problem (3) is a continuous PWA function

$$u_0^*(x(t)) = F_i x(t) + G_i, \quad \text{if} \quad x(t) \in \mathcal{R}_i, \tag{5}$$

which consists of a finite number R of affine feedback laws. The i-th law is valid if $x(t)$ is contained in a convex polyhedral region $\mathcal{R}_i = \{x \mid H_i x(t) \le K_i\}$.

Theorem 2.1 suggests that the RHMPC feedback $u_0^*(x(t))$ can be constructed off-line as a lookup table and will henceforth be called the *explicit RHMPC feedback law.* The advantage of such an approach is that value of u_0^* for a particular value of $x(t)$ can be obtained by simply evaluating the table, as captured by Algorithm 1 The algorithm traverses through the regions sequentially, stopping once it finds a region which contains $x(t)$. In such case the optimal control action is calculated by evaluating the corresponding control law and returned back. Clearly, in the worst case, the algorithm has to search through all regions. But since the operations performed in Steps 2 and 3 only consist of simple matrix multiplications and additions, for a large class of problems running Algorithm 1 is faster compared to obtaining $u_0^*(x(t))$ by solving the optimization problem (3) on-line as an LP or a QP. The second benefit is

Algorithm 1 Sequential table traversal

```
INPUT: Regions $\mathcal{R}_i$, feedback laws $F_i$, $G_i$, number of
    regions $R$, state measurement $x(t)$
OUTPUT: Optimal RHMPC control input $u_0^*(x(t))$
1: for $r = 1, \dots, R$ do
2:   if $H_r x(t) \le K_r$ then
3:     $u_0^*(x(t)) = F_r x(t) + G_r$
4:     return
5:   end if
6: end for
```

that the number of floating point operations performed by the sequential search is known exactly, regardless of $x(t)$. Therefore given a table with R regions, one can a-priori say whether the given computational device will execute Algorithm 1 within of one sampling instance. For generic LP and QP problems such bounds could also be obtained, but they are very often too conservative to actually have a practical meaning.

3. MPC DESIGN WITH THE MULTI-PARAMETRIC TOOLBOX

MPT is a MATLAB-based toolbox which allows for user-friendly design and synthesis of MPC controllers using parametric programming. To illustrate capabilities of the toolbox, we consider a laboratory apparatus consisting of two liquid storage tanks placed one on top of the other Honc and Dušek (2006). Liquid inflow to the upper tank is governed by a pump, throughput of which is to be controlled. The liquid then accumulates in the tank and flows out through an opening to the lower tank. The outflow of the lower tank is then collected and recycled. Such a system can be described by a linearized model with two states (the deviations of liquid levels from chosen linearization points), one measured output (liquid level in the bottom tank), and one control signal (deviation of the pump volumetric flowrate from a stationary point). The control objective is to operate the pump such that the output reaches a given reference, while respecting input and state constraints.

MPC synthesis in MPT consists of several intuitive steps. First, the plant model is defined and converted into the discrete-time domain using a sampling time T_s:

```
>> A = [-0.0315, 0; 0.0315, -0.0315];
>> B = [0.0769; 0]; C = [0, 1]; D = 0;
>> tanks = ss(A, B, C, D);
>> Ts = 5;
>> model = mpt_sys(tanks, Ts);
```

Next, constraints on states, inputs and outputs are defined:

```
>> model.umin = -16.9811; model.umax = 3.0189;
>> model.xmin = [-21.44; -21.44];
>> model.xmax = [3.56; 3.56];
>> model.ymin = -21.44; model.ymax = 3.56;
```

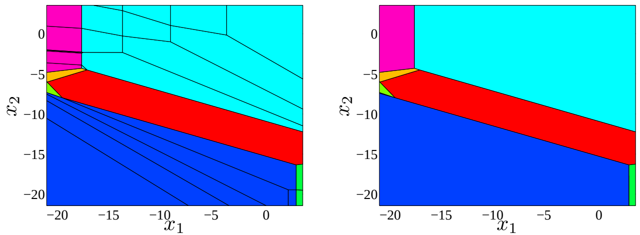
Notice that the constraints are imposed on the deviation variables (i.e. $x_1 = h_1 - h_1^s$, $x_2 = h_2 = h_2^s$, $u = q - q^s$, where h_1, h_2 are physical liquid levels, q is the liquid flowrate, and h_1^s, h_2^s and q^s are linearization points). The toolbox also allows to define generic linear constraints of the form

$$L_x x_k + L_u u_k + L_y y_k \leq M,$$

and/or to use constraint softening. We refer the reader to the MPT manual Kvasnica et al. (2006) for a detailed description of these advanced features.

Subsequently, parameters of the MPC optimization problem (3) have to be specified. These include the prediction horizon N, penalty matrices Q_y and Q_u, norm p, and reference signals.

```
>> mpc.N    = 6;   % prediction horizon
>> mpc.yref = 0;   % output reference
>> mpc.Qy   = 100; % penalty on (y_k - y_ref)
```



(a) Regions of the lookup table consisting of 31 regions. Regions which share the same expression for the optimizer are depicted with identical colors.

(b) Simplified version of the lookup table where the ORM method merges together regions which have the same expression of the optimizer.

Fig. 1. Regions of the lookup table for the two tanks example.

```
>> mpc.R    = 1;   % penalty on u_k
>> mpc.norm = 2;   % use quadratic cost
```

Finally, the parametric solution to problem (3) can be calculated as a lookup table using the following simple command:

```
>> controller = mpt_control(model, mpc)
```

Result of the computation is, in this case, a lookup table consisting of 31 regions in a 2D state-space. One can inspect the table graphically by plotting its regions by calling

```
>> plot(controller)
```

which produces the output as shown in Figure 1(a). In order to obtain the optimal control action for a particular value of $x(t)$, MPT provides the following easy-to-use notation:

```
>> x = [3; -4.5];
>> u = controller(x)
```

which will run Algorithm 1 using the provided initial state and the data stored in the `controller` variable. In subsequent sections we show how to generate real-time executable code and deploy it to selected control platforms. Having real-time implementation in mind, by executing

```
>> info(controller)
```

the toolbox provides a detailed information about the size of the table (expressed in bytes) and the worst-case number of operations the sequential search would perform. This information helps the control engineer to decide whether or not further post-processing of the table is necessary using some of the methods described in the next section. For the controller with 31 regions, the total memory footprint of the lookup table, as reported by the `info` function, is 3988 bytes, and the worst-case number of floating points operations Algorithm 1 performs is equal to 653 FLOPS.

4. CONTROLLER POST-PROCESSING

Since typical DSP processors can perform tens to hundreds of millions of FLOPS per second and posses several megabytes of memory, the computational burden of implementing MPC in a lookup table form in real-time is minute. However, moving to less powerful platforms, such as PLCs, poses a challenge on whether the memory footprint of the table (which is proportional to the number of its regions) can be further reduced and whether the table can be traversed using less operations. To highlight importance of these issues, take into account that low-cost PLCs only contain 2-8 kilobytes of memory and their computational power is also restricted.

Several methods have been developed to address these challenges. In general, they can be divided into two categories: reduction of the table size (e.g. Johansen and Grancharova (2003); Grieder et al. (2004); Cychowski and O'Mahony (2005); Scibilia et al. (2009)), and development of faster algorithms for table evaluation (e.g. Jones et al. (2005); Christophersen et al. (2007); Kvasnica et al. (2008)). The MPT toolbox implements two of such strategies which are generally useful from practical point of view: the *optimal region merging* (ORM) method of Geyer et al. (2008) for reduction of the table size, and the *binary search tree* (BST) algorithm of Tøndel et al. (2003) for faster table traversal.

To illustrate the ORM method, it should first be noted that it is a frequent case in parametric MPC that there are multiple regions of the table in which the expression for u^* is the same (i.e. $F_i = F_j$ and $G_i = G_j$ for some index i and j), cf. Fig. 1(a). The idea of ORM is to merge the regions which share the same expression for the optimizer into larger convex objects. While there are multiple ways of achieving this goal, the method of Geyer et al. (2008) utilizes boolean minimization to achieve a table defined over the least possible number of regions. In MPT, the optimal merging can be calculated by calling the following command:

```
>> simple = mpt_simplify(controller, 'optimal')
```

which produces a new lookup table stored in the variable called `simple`. The new object is still a valid parametric MPC controller, and therefore it can be processed as described in the previous section. For instance, calling `u = simple(x)` will return the optimal control action for state `x`, and `plot(simple)` will plot the regions of the simplified controller, cf. Fig. 1(b). For the two tanks example, the optimal merging was able to decrease the number of regions from 31 to 9, which reflects to a decrease of required memory storage from 4 kB to 1 kB. Simplification of the table also directly influences its evaluation time. Therefore the worst-case number of FLOPS which Algorithm 1 needs to perform to traverse the simplified table is now only 198 operations (compared to 653 FLOPS for the original table).

The BST algorithm allows the table to be traversed faster compared to the sequential approach of Algorithm 1. The basic idea is to hierarchically organize the controller regions into a tree structure where, at level of the tree, the number of regions to consider is decreased by a factor of two. Therefore the table traversal can be performed in time logarithmic in the number of regions (note that complexity of the sequential search is linear in R). The tree is constructed in an iterative fashion. At each iteration an optimal separating hyperplane $h_i x(t) \leq k_i$ is selected such that the set of all regions processed at the i-th iteration is divided into two smaller subsets: regions $\mathcal{R}_i^+$ residing on one side of the hyperplane, and $\mathcal{R}_i^-$ on the other side.

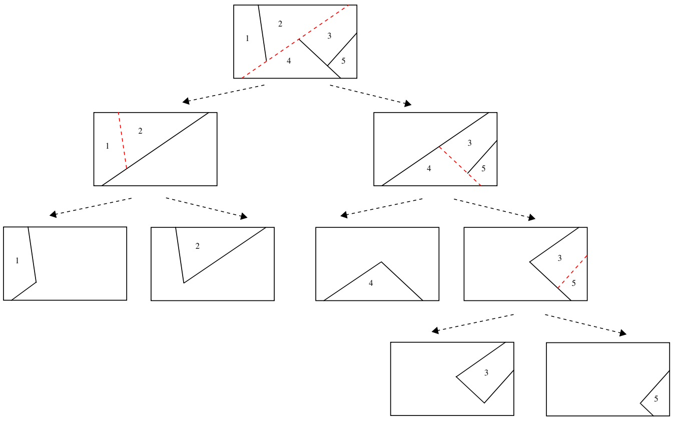


Fig. 2. Illustration of how a binary search tree is built for a table consisting of 5 regions. The optimal separating hyperplanes are depicted by red dashed lines.

A new node in the tree is then created which contains information about the hyperplane and two pointers to child nodes. The left child is created by recursively calling the algorithm for regions $\mathcal{R}_i^+$, and the right child for regions $\mathcal{R}_i^-$. The exploration of a given tree branch stops when no further subdivision can be achieved. In such a case a leaf node is created which points to the region which contains $x(t)$. The resulting tree is then composed of the set of separating hyperplanes linked to the actual regions through a set of pointers. The BST algorithm is illustrated graphically in Figure 2. BST trees can be constructed for arbitrary lookup tables using the following MPT command:

```
>> tree = mpt_searchTree(controller)
```

The output of the function is a new controller object, which behaves the same way as previously. This implies that obtaining u^* for a particular value of x reduces to simply calling `u = tree(x)`, which traverses the BST tree using Algorithm 2.

Algorithm 2 Table traversal via binary search tree

INPUT: BST tree composed of linked nodes and separating hyperplanes $h_i x(t) \leq k_i$, $i = 1, \ldots, M$, state measurements $x(t)$
OUTPUT: Optimal RHMPC control input $u^*(x)$

```
1: $r \leftarrow 1$
2: repeat
3:     if $h_r x(t) \leq k_r$ then
4:         $r \leftarrow$ index of the left child node
5:     else
6:         $r \leftarrow$ index of the right child node
7:     end if
8: until $r$ is a leaf node
9: $u_0^*(x(t)) = F_r x(t) + G_r$
```

If the `mpt_searchTree` function applied to the lookup table with 31 regions, it generates 14 separating hyperplanes and organizes them into a tree with 29 nodes at 5 levels. Therefore, in the worst case, Algorithm 2 would need to go through Steps 2–8 at most 5 times before terminating. The only calculations are actually carried out on Steps 3 and 9, the remaining steps just perform a simple memory transfer. This constitutes that the total effort is equal to 28

FLOPS, a significant reduction from 653 FLOPS needed by Algorithm 1.

5. CODE GENERATION AND DEPLOYMENT

As outlined in the previous sections, MPC can be implemented in real time by first constructing a suitable lookup table off-line, and then traversing the table on-line for a particular value of the state measurements. Specifically, we have illustrated how to do the traversal either sequentially (Algorithm 1) or by evaluating a binary search tree (Algorithm 2). As both algorithms only execute trivial arithmetic operations and do not involve any high-level optimization, they can be easily implemented, by hand, using any high- or low-level programming language and linked with the custom control application.

The MPT toolbox goes even further as it allows real-time executable code to be generated automatically. The first option is to let the toolbox generate plain C-code implementation of Algorithms 1 and 2 together with the necessary data (matrices H_i and K_i defining the table regions and the feedback laws F_i and G_i). A C-code version of Alg. 1 can be generated by calling

```
>> mpt_exportC(controller, 'target_filename')
```

The function generates the files `target_filename.c` and `target_filename.h` which contain, respectively, the table evaluation code and the table data. The code can be subsequently linked to any application written in the C language. Code and data for the binary search tree representation of the lookup table can be obtained by executing

```
>> mpt_exportST(tree, 'target_filename')
```

which again generates a plain C-code implementation which can be compiled directly for most typical control platforms.

Another option is to use the Real-Time Workshop (RTW), which is a de-facto standard code generation tool supporting different types of digital signal processors and other CPUs. MPT provides a RTW-compatible C-code implementation of the sequential search algorithm, which makes code generation and deployment a single click operation. To use this option, the controller is first embedded into the Simulink environment using a provided block and it is subsequently connected to signals from A/D and D/A convertors to close the control loop as shown in Figure 3. Then, one simply clicks on the respective icon in the Simulink window to initiate RTW, which compiles the overall control scheme including the lookup table and then automatically downloads it to any supported target CPU. We have applied this procedure to deploy the MPC controller for the two tanks system of Section 3 to the dSPACE DS 1104 control platform, which then took care of signal conversion, state estimation and table traversal. Closed-loop profiles of states and control inputs obtained during the experiment are shown in Figure 4.

Finally, the new of MPT also supports code generation and deployment specifically tailored to Siemens Simatic programmable logic controllers. The export, which can be initiated by calling

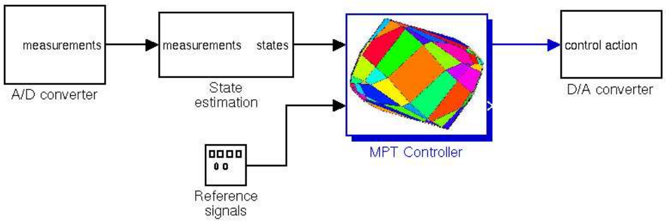


Fig. 3. Control loop setup in Simulink using MPT control block which implements the lookup table.

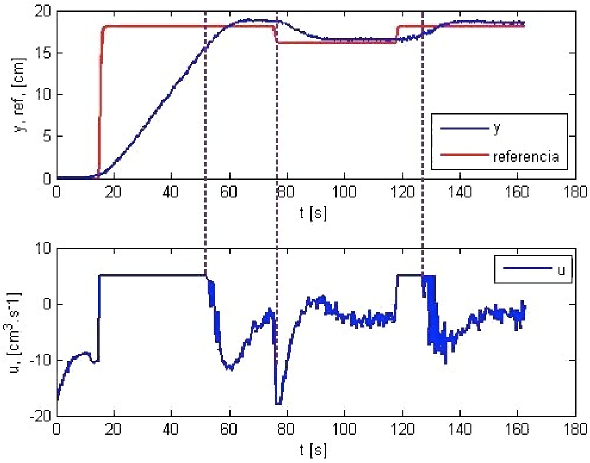
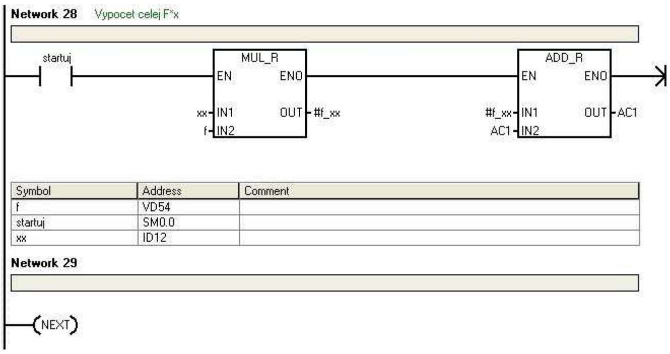


Fig. 4. Signals acquired during real-time control of the two tanks system.



Symbol	Address	Comment
f	VD54	
startuj	SM0.0	
xx	ID12	

Fig. 5. Short excerpt of the LAD implementation of Algorithm 2.

```
>> mpt_exportPLC(controller, 'target_filename')
```

produces two files – the PLC version of Algorithm 2 and a data file containing description of the binary search tree which encodes a given lookup table. The tree traversal code, implemented using the Ladder Logic (LAD) programming language, is universal and can process any kind of lookup tables described by binary search trees. If needed, a corresponding Statement List version of the algorithm can be generated as well. The code and the data are subsequently downloaded to the PLC, which then executes the table traversal at every sampling instances based on the measurements of the states. As the LAD code only uses standard blocks like multiplications, additions, comparisons and memory access, it can be easily adapted to other types of PLCs as well. A short excerpt of the LAD code is shown in Figure 5.

6. CONCLUSIONS

In this paper we have shown how MPC can be implemented using low computational resources. The approach is based on pre-calculating the solution to a given MPC optimization problem just once, for all possible initial conditions. The result is then given as a lookup table, which can be easily implemented in C or PLC languages and deployed to standard control platforms. We have presented a new version of the Multi-Parametric Toolbox, which automates the control design and code generation, and provides efficient means for reducing the table complexity. The new version will be available shortly at the project's web site `http://control.ee.ethz.ch/~mpt`.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/4055/07, and 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Christophersen, F.J., Kvasnica, M., Jones, C.N., and Morari, M. (2007). Efficient evaluation of piecewise control laws defined over a large number of polyhedra. In P.J.A. S. G. Tzafestas (ed.), *Proceedings of the European Control Conference ECC '07*, 2360–2367. URL http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=390.

Cychowski, M. and O'Mahony, T. (2005). Efficient off-line solutions to robust model predictive control using orthogonal partitioning. In *Proceedings of the 16th IFAC world congress*.

Geyer, T., Torrisi, F., and Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7), 1728–1740. URL http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=3079.

Grieder, P., Wan, Z., Kothare, M., and Morari, M. (2004). Two level model predictive control for the maximum control invariant set. In *American Control Conference*. Boston, Massachusetts.

Honc, D. and Dušek, F. (2006). Laboratory plant for control experiments. In *Proc. 7. International Scientific-Technical Conf. Process Control 2006*, 219. Kouty nad Desnou, Czech Republic.

Johansen, T. and Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Trans. on Automatic Control*, 48, 810–815.

Jones, C.N., Grieder, P., and Raković, S.V. (2005). A Logarithmic Solution to the Point Location Problem for Closed-Form Linear MPC. In *IFAC World Congress*. Prague, Czech Republic.

Kvasnica, M., Christophersen, F.J., Herceg, M., and Fikar, M. (2008). Polynomial approximation of closed-form MPC for piecewise affine systems. In *Proceedings of the 17th IFAC World Congress*, 3877–3882. Seoul, Korea. URL http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=709.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/~mpt/.

Kvasnica, M., Grieder, P., Baotić, M., and Christophersen, F.J. (2006). *Multi-Parametric Toolbox (MPT) User's Manual*. Automatic Control Laboratory, ETH Zurich.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall.

Mikleš, J. and Fikar, M. (2007). *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg. URL http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=366.

Qin, S. and Badgewell, T. (1997). An overview of industrial model predictive control technology. *Chemical Process Control*, 93(316), 232–256.

Scibilia, F., Olaru, S., and M, H. (2009). Approximate explicit linear mpc via delaunay tessellation. In *Proceedings of the 10th European Control Conference*. Budapest, Hungary.

Tøndel, P., Johansen, T., and Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5), 945–950.

Comments – Remarks

Automated Piecewise Affine Approximation of Nonlinear Systems

Michal Kvasnica [*,1], **Alexander Szücs** [*], and **Miroslav Fikar** [*]

[*] *Institute of Information Engineering, Automation, and Mathematics,*
Slovak University of Technology in Bratislava,
81237 Bratislava, Slovakia
e-mail : {michal.kvasnica,alexander.szucs,miroslav.fikar}@stuba.sk

Abstract: : Successful application of model-based control synthesis and analysis methods usually depends on availability of a good process model. Recently, the concept of hybrid models was introduced as a good compromise between accuracy and complexity. Hybrid models feature a collection of local linear models accompanied with logic conditions which enforce switching of the local dynamics. The problem which we address in this paper is the following: given a nonlinear dynamical model and a fixed complexity of its hybrid approximation, how should one design the hybrid model of maximal accuracy? The answer is first provided in terms of optimally approximating nonlinear functions in one variable. The best approximation is found by solving a nonlinear optimization problem. Subsequently, the procedure is extended to approximation of functions in n variables and we show that this problem boils down to solving n problems of the former kind.

Keywords: hybrid systems, approximation, nonlinear optimization

1. INTRODUCTION

Mathematical models of physical plants play a vital role in many areas, such as in rigorous simulations, analysis, or control synthesis. Typically, high model accuracy is usually desired while keeping the model complexity on an acceptable level. Traditionally, nonlinear models were preferred from simulations, while most of available control techniques are based on a local approximation around a single operating point. The concept of hybrid systems (Branicky, 1995) can be viewed as a compromise solution between accuracy of the model and its complexity. Hybrid models feature a collection of local models accompanied with logic IF-THEN conditions which enforce switching of the local dynamics. When all local models are linear (or affine), such systems are referred to as *linear* hybrid systems. Although still nonlinear due to the presence of switches, the underlying piecewise linearity allows for somewhat easier control synthesis and analysis compared to using full nonlinear models. Several mathematical frameworks capable of capturing the relation between logic rules and linear dynamics can be used: Piecewise Affine (PWA) models (Sontag, 1981), Mixed Logical Dynamical (MLD) systems (Bemporad and Morari, 1999), Linear Complementarity systems (Heemels et al., 2000) and max-min-plus-scaling models (De Schutter and Van den Boom, 2001). Under mild assumptions, all these frameworks are equivalent to each other and it is possible to transform e.g. the MLD system into a PWA model and vice-versa (Heemels et al., 2001). For the purpose of this work we consider PWA models, which use the concept of multiple linearization to approximate a given nonlinear system with arbitrary accuracy.

The problem which we address in this paper is the following: given a nonlinear dynamical model $x^+ = f(x, u)$ and a fixed complexity of its PWA approximation $\widetilde{f}(x, u) \approx f(x, u)$, how should one design $\widetilde{f}$ which minimizes the approximation error $\int (f(x, u) - \widetilde{f}(x, u))^2$? The answer is non-trivial even putting optimality of the approximation aside. Traditionally, two distinct approaches for deriving PWA approximations are used. When the mathematical formulation of the original nonlinear system is known, one can design the approximation by hand. This is usually done by employing human knowledge and experience to devise several linearization points around which the original nonlinear model should be linearized. Needless to say, placement of such points has a crucial impact on the accuracy of the approximation. The HYSDEL (Hybrid Systems Description Language) tool (Torrisi and Bemporad, 2004; Kvasnica and Herceg, 2010) can be used to accelerate this line of development. Formally, HYSDEL transforms a linguistic description of a hybrid system into the corresponding MLD model, which can then be converted into the PWA form. The language allows to define IF-THEN switching rules which, based on whether some logic condition is satisfied or not, enforce certain continuous dynamics. Another option is to use hybrid identification techniques (Ferrari-Trecate et al., 2001; Roll et al., 2004; Ferrari-Trecate, 2005) to construct the PWA approximation from the input-output measurements. The crucial advantage is that the model of the original nonlinear system is not required to be fully available. The downside, however, is that the approximation is only accurate in the interval captured by the identification data. Moreover, the procedure is computationally expensive and suited mainly to low-dimensional problems.

In this work we propose to use an optimization-based approach to derive PWA approximations of nonlinear systems whose vector field is an a-priori known function of multiple variables. After formally stating the problem in Section 2, we show in Section 3 that an optimal PWA approximation of generic nonlinear functions in one variable can be formulated and solved as a nonlinear programming problem. Several non-trivial illustrative cases are discussed to show that the approach is both efficient and computational tractable. Subsequently, the approach is extended to deriving PWA approximations of multivariable functions in Section 4. We show that, under a certain assumption, the problem boils down to solving a series of one-dimensional approximations. The algorithmic and software implementation of the approximation procedure are then discussed in Section 5. Specifically, we introduce a new software tool which is capable of exporting the obtained optimal PWA approximations into the HYSDEL language. This brings two crucial advantages. First, the HYSDEL compiler can be used to convert the PWA approximation into a mathematical form, which is then suitable e.g. for control design. Second, since the exported approximation is described in a human-readable format, it can be further fine-tuned by hand. Finally, in Section 6 we illustrate the procedure on a case study involving a highly non-linear chemical reactor.

2. PROBLEM STATEMENT

We consider generic dynamic systems in discrete-time

$$x^+ = f(x, u), \tag{1}$$

where the vector field $f(\cdot, \cdot)$ is assumed to be continuous in the state variables $x \in \mathbb{R}^{n_x}$ and in the inputs $u \in \mathbb{R}^{n_u}$. System states and inputs are assumed to be constrained to connected and closed domains $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$, respectively.

The objective is to approximate (1) by a different dynamic system $x^+ = \widetilde{f}(x, u)$ whose vector field $\widetilde{f}(x, u)$ is a PWA function which consists of a pre-specified number N of local linear dynamics:

$$\widetilde{f}(x, u) = \begin{cases} A_1 x + B_1 u + c_1 & \text{if } \left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \mathcal{R}_1 \\ \quad\vdots & \quad\vdots \\ A_N x + B_N u + c_N & \text{if } \left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \mathcal{R}_N. \end{cases} \tag{2}$$

Here, $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $c_i \in \mathbb{R}^{n_x}$, are the state-update matrices of the i-th local linear approximation, and $\mathcal{R}_i \subset \mathbb{R}^{n_x \times n_u}$ is the region of validity of the i-th local model satisfying $\mathcal{R}_i \neq \emptyset$, $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$, $\forall i \neq j$, and $\cup_i \mathcal{R}_i = \mathcal{X} \times \mathcal{U}$.

Formally, the problem which we aim at solving can be stated as follows:

Problem 2.1. Given a nonlinear vector field $f(x, u)$ of system (1), find the PWA approximation (2) of pre-specified complexity which minimizes the approximation error

$$e_{\text{aprx}} := \int (f(x, u) - \widetilde{f}(x, u))^2 \, dx du, \tag{3}$$

where the integral is evaluated over the whole region of validity of (1), i.e. over $\mathcal{X} \times \mathcal{U}$.

In the sequel we show how to solve Problem 2.1 provided that the vector field $f(z)$, $z = [x,\ u]^T$ satisfies the following assumption.

Assumption 2.2. The function $f(z_1, \ldots, z_n)$ can be written as $\sum_{i=1}^{n} \alpha_i \Big(\prod_{j=p_i}^{q_i} f_j(z_j) \Big)$.

As an example, the function $z_1 e^{z_2}$ satisfies such an assumption, while the function $e^{z_1 z_2}$ does not. Although the assumption is somewhat restrictive, the gained advantage is that approximating any multivariable function $f(z_1, \ldots, z_n)$ boils down to solving a series of 1D problems, as evidenced in the following two sections.

Remark 2.3. Since the approximation procedure discussed in the sequel considers only the vector field in the right-hand-side of (1), continuous-time systems $\dot{x} = f(x, u)$ can be treated as well.

3. FUNCTIONS IN ONE VARIABLE

First, we consider the one-dimensional case, i.e. approximating a nonlinear function $f(z) : \mathbb{R} \mapsto \mathbb{R}$, with domain $\mathcal{Z} \subset \mathbb{R}$, by a PWA function $\widetilde{f}(z) = a_i z + c_i$ if $z \in \mathcal{R}_i$. Since $\mathcal{Z}$ is assumed to be connected and closed, it is a line segment $[\underline{z},\ \overline{z}]$. Regions $\mathcal{R}_i$ define the partition of such a line into N non-overlapping parts, i.e. $\mathcal{R}_1 = [\underline{z},\ r_1]$, $\mathcal{R}_2 = [r_1,\ r_2]$, …, $\mathcal{R}_{N-1} = [r_{N-2},\ r_{N-1}]$, $\mathcal{R}_N = [r_{N-1},\ \overline{z}]$ with $\cup_i \mathcal{R}_i = [\underline{z},\ \overline{z}]$. Solving Problem 2.1 then becomes to find the slopes a_i, offsets c_i and breakpoints r_i such that the approximation error is minimized, i.e.

$$\min_{a_i, c_i, r_i} \int_{\underline{z}}^{\overline{z}} (f(z) - \widetilde{f}(z))^2 \, dz \tag{4a}$$

$$\text{s.t. } \widetilde{f}(z) = \begin{cases} a_1 z + c_1 & \text{if } z \in [\underline{z},\ r_1] \\ \quad\vdots & \quad\vdots \\ a_N z + c_N & \text{if } z \in [r_{N-1},\ \overline{z}] \end{cases} \tag{4b}$$

$$\underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \tag{4c}$$

$$a_i r_i + c_i = a_{i+1} r_i + c_{i+1},\ i = 1, \ldots, N-1, \tag{4d}$$

where (4d) enforces continuity of $\widetilde{f}(z)$ along the breakpoints r_i. The IF-THEN based nonlinear constraint (4b) can be eliminated by observing that, by definition, regions $\mathcal{R}_i$ are non-overlapping, and the integral in (4a) can hence be written as

$$\int_{\underline{z}}^{\overline{z}} \big(f(z) - \widetilde{f}(z)\big)^2 = \sum_{i=1}^{N} \Big(\int_{r_{i-1}}^{r_i} \big(f(z) - (a_i z + c_i)\big)^2 \Big), \tag{5}$$

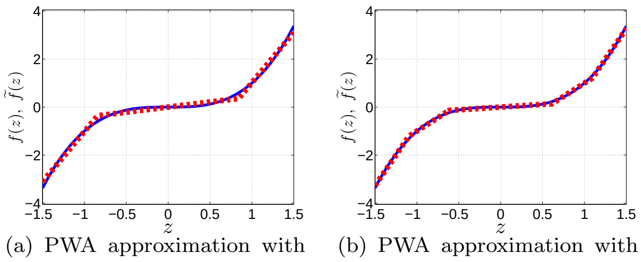
with $r_0 = \underline{z}$ and $r_N = \overline{z}$. The NLP (4) can therefore be written as

$$\min_{a_i, c_i, r_i} \sum_{i=1}^{N} \Big(\int_{r_{i-1}}^{r_i} \big(f(z) - (a_i z + c_i)\big)^2 \, dz \Big) \tag{6a}$$

$$\text{s.t. } \underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \tag{6b}$$

$$a_i r_i + c_i = a_{i+1} r_i + c_{i+1},\ i = 1, \ldots, N-1. \tag{6c}$$

For simple functions $f(z)$, the integral in (6a) can be expressed in an analytical form in unknowns a_i, c_i, r_i, along with the corresponding gradients. For more complex expressions, the integrals can be evaluated numerically, e.g. by using the trapezoidal rule. In either case, problem (6) can be solved to a local optimality e.g. by using



(a) PWA approximation with 3 regions.

(b) PWA approximation with 5 regions.

Fig. 1. Graph of $f(z) = z^3$ (blue line) and the PWA approximations $\widetilde{f}(z)$ (red dashed lines).

Table 1. Value of the objective (6a) and runtime of the NLP (6) as a function of N for Example 3.1.

N	Value of (6a)	Runtime of (6)
3	$3.1 \cdot 10^{-2}$	0.05 sec
4	$1.9 \cdot 10^{-2}$	0.20 sec
5	$5.2 \cdot 10^{-3}$	0.21 sec
6	$3.5 \cdot 10^{-3}$	0.33 sec
7	$1.6 \cdot 10^{-3}$	0.37 sec

the `fmincon` solver of MATLAB. Alternatively, one can use global optimization methods (Adjiman et al., 1996; Papamichail and Adjiman, 2004; Chachuat et al., 2006) which guarantee that an ϵ-neighborhood of the global optimum can be found.

Example 3.1. Consider the function $f(z) = z^3$ on domain $-1.5 \leq z \leq 1.5$. The analytic form of the integral (6a) is

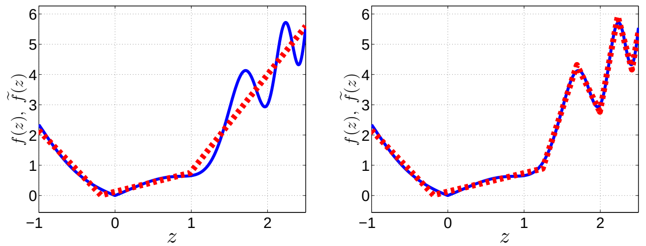
$$\sum_{i=1}^{N}\Big(c_i^2(r_i+r_{i-1})+a_ic_i(r_i^2-r_i^2)+\frac{a_i^2}{3}(r_i^3-r_{i-1}^3)- \\ -\frac{c_i}{2}(r_i^4-r_{i-1}^4)-\frac{2a_i}{5}(r_i^5-r_{i-1}^5)+\frac{1}{7}(r_i^7-r_{i-1}^7)\Big),$$

with $r_0 = -1.5$ and $r_N = 1.5$. The PWA approximation of $f(z)$ with $N = 3$ regions was obtained by solving the NLP (6) using `fmincon`, which only took 0.05 seconds on a 2.4 GHz CPU running MATLAB 2009b. The obtained PWA approximation is then given by

$$\widetilde{f}(z) = \begin{cases} 4.1797z + 3.1621 & \text{if } -1.5 \leq z \leq -0.8423 \\ 0.4257z & \text{if } -0.8423 \leq z \leq 0.8423 \\ 4.1797z - 3.1621 & \text{if } \ 0.8423 \leq z \leq 1.5 \end{cases}$$

Naturally, quality of the approximation can be improved by increasing the complexity of the PWA function, i.e. by enlarging N, as documented in Table 1. As can be seen from the reported results, accuracy of the approximation increases by roughly a factor of two for each additional degree of freedom. Two PWA approximations with $N = 3$ and $N = 5$ are shown, respectively, in Figures 1(a) and 1(b).

Example 3.2. Consider the function $f(z) = |z| + 0.5z^2 - \sin(z^3)$ on domain $-1 \leq z \leq 2.5$, graph of which is shown in Figure 2(a). Since no analytic expression of the integral in (6a) could be obtained, we have opted for numeric integration of the cost while solving the NLP problem (6) by `fmincon`. Again, we have investigated PWA approximations with $N = 3, \ldots, 7$ regions. Accuracy of the approximation, along with the runtime of the associated NLP problem (6), are reported in Table 2. The PWA approximations for $N = 3$ and $N = 7$ are shown in Figures 2(a) and 2(b).



(a) PWA approximation with $N = 3$ regions.

(b) PWA approximation with $N = 7$ regions.

Fig. 2. Graph of $f(z) = |z| + 0.5z^2 - \sin(z^3)$ (blue line) and the PWA approximations $\widetilde{f}(z)$ (red dashed lines).

Table 2. Value of the objective (6a) and runtime of the NLP (6) as a function of N for Example 3.2.

N	Value of (6a)	Runtime of (6)
3	0.753	1.2 sec
4	0.652	1.6 sec
5	0.504	1.4 sec
6	0.470	2.8 sec
7	0.022	4.9 sec

4. MULTIVARIABLE FUNCTIONS

The task is to approximate a given multivariable function $f(z_1, \ldots, z_n) : \mathbb{R}^n \mapsto \mathbb{R}$ with domain $\mathcal{Z} \subset \mathbb{R}^n$ by a PWA function $\widetilde{f}(z_1, \ldots, z_n)$, defined over the same domain, such that the approximation error (3) is minimized.

Definition 4.1. (Williams (1993)). Function $f(z_1, \ldots, z_n)$ is called *separable* if it can be expressed as the sum of functions of a single variable, i.e. $f(z_1, \ldots, z_n) = f_1(z_1) + \cdots + f_n(z_n)$.

If $f(z_1, \ldots, z_n)$ is readily separable (e.g. when $f(z_1, z_2) = e^{z_1} + \sin(z_2)$), its optimal PWA approximation can be obtained by applying the 1D scenario of Section 3 to the individual components of the function, i.e. $\widetilde{f}(z_1, \ldots, z_n) = \widetilde{f}_1(z_1) + \cdots + \widetilde{f}_n(z_n)$. The total number of regions over which the PWA approximation $\widetilde{f}(\cdot)$ is defined is hence given by $\sum_{j=1}^{n} N_j$, where N_j is the pre-specified complexity of the j-th approximation $\widetilde{f}_j(z_j)$.

A surprisingly large number of non-separable functions can be converted into the separable form by applying a simple trick, elaborated in more details e.g. in Williams (1993). To introduce the procedure, consider a non-separable function $f(z_1, z_2) = z_1 z_2$ with domain $\mathcal{Z} := [\underline{z}_1, \ \overline{z}_1] \times [\underline{z}_2, \ \overline{z}_2]$. Define two new variables

$$y_1 = (z_1 + z_2), \quad y_2 = (z_1 - z_2). \tag{7}$$

Then it is easy to verify that $1/4(y_1^2 - y_2^2) = z_1 z_2$. The coordinate transformation therefore transforms the original function into a separable form, where both terms (y_1^2 and y_2^2) are now functions of a single variable. The procedure of Section 3 can thus be applied to compute PWA approximations of $f_{y_1}(y_1) := y_1^2$ and $f_{y_2}(y_2) := y_2^2$, where the function arguments relate to z_1 and z_2

via (7). Important to notice is that $f_{y_1}(\cdot)$ and $f_{y_2}(\cdot)$ have different domains, therefore their PWA approximations $\widetilde{f}_{y_1}(y_1) \approx y_1^2$ and $\widetilde{f}_{y_2}(y_2) \approx y_2^2$ will, in general, be different. Specifically, the domain of $f_{y_1}(\cdot)$ is $[\underline{y}_1,\ \overline{y}_1]$ with $\underline{y}_1 = \min\{z_1 + z_2 \mid \underline{z}_1 \le z_1 \le \overline{z}_1,\ \underline{z}_2 \le z_2 \le \overline{z}_2\}$ and $\overline{y}_1 = \max\{z_1 + z_2 \mid \underline{z}_1 \le z_1 \le \overline{z}_1,\ \underline{z}_2 \le z_2 \le \overline{z}_2\}$. Similarly, the domain of $f_{y_2}(\cdot)$ is $[\underline{y}_2,\ \overline{y}_2]$, whose boundaries can be computed by respectively minimizing and maximizing $z_1 - z_2$ subject to the constraint $[z_1,\ z_2]^T \in \mathcal{Z}$. The overall PWA approximation $\widetilde{f}(z_1, z_2) \approx z_1 z_2$ then becomes

$$\widetilde{f}(z_1, z_2) = 1/4(\widetilde{f}_{y_1}(z_1 + z_2) - \widetilde{f}_{y_2}(z_1 - z_2)). \tag{8}$$

The value of $\widetilde{f}(z_1, z_2)$ for any points z_1, z_2 is obtained by subtracting the value of the PWA function $\widetilde{f}_{y_2}(\cdot)$ evaluated at the point $z_1 - z_2$ from the function value of $\widetilde{f}_{y_1}(\cdot)$ evaluated at $z_1 + z_2$, followed by a linear scaling.

The procedure naturally extends to multivariable functions represented by the product of two nonlinear functions of a single variable, i.e. $f(z_1, z_2) = f_1(z_1) f_2(z_2)$. Here, the transformation (7) becomes

$$y_1 = f_1(z_1) + f_2(z_2), \quad y_2 = f_1(z_1) - f_2(z_2). \tag{9}$$

Therefore, $1/4(y_1^2 - y_2^2) = f(z_1, z_2)$ still holds. Let $f_{y_1}(y_1) := y_1^2$ and $f_{y_2}(y_2) := y_2^2$. The domain of $f_{y_1}(\cdot)$ is $[\underline{y}_1,\ \overline{y}_1]$ and $\operatorname{dom} f_{y_2}(\cdot) = [\underline{y}_2,\ \overline{y}_2]$ with

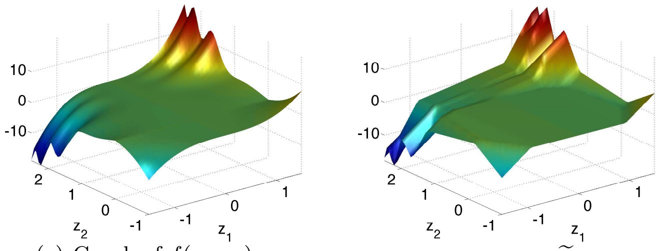
$$\underline{y}_1 = \min\{f_1(z_1) + f_2(z_2) \mid [z_1,\ z_2]^T \in \mathcal{Z}\}, \tag{10a}$$
$$\overline{y}_1 = \max\{f_1(z_1) + f_2(z_2) \mid [z_1,\ z_2]^T \in \mathcal{Z}\}, \tag{10b}$$
$$\underline{y}_2 = \min\{f_1(z_1) - f_2(z_2) \mid [z_1,\ z_2]^T \in \mathcal{Z}\}, \tag{10c}$$
$$\overline{y}_2 = \max\{f_1(z_1) - f_2(z_2) \mid [z_1,\ z_2]^T \in \mathcal{Z}\}, \tag{10d}$$

which can be computed by solving four NLP problems. Finally, since all expressions are now functions of a single variable, the PWA approximations $\widetilde{f}_1(z_1) \approx f_1(z_1)$, $\widetilde{f}_2(z_2) \approx f_2(z_2)$, $\widetilde{f}_{y_1}(y_1) \approx f_{y_1}(y_1)$, and $\widetilde{f}_{y_2}(y_2) \approx f_{y_2}(y_2)$ can be computed by solving the NLP (6). The overall optimal PWA approximation $\widetilde{f}(z_1, z_2) \approx f(z_1, z_2)$ then becomes

$$\widetilde{f}(z_1, z_2) = 1/4\Big(\widetilde{f}_{y_1}\big(\widetilde{f}_1(z_1) + \widetilde{f}_2(z_2)\big) - \widetilde{f}_{y_2}\big(\widetilde{f}_1(z_1) - \widetilde{f}_2(z_2)\big)\Big). \tag{11}$$

The evaluation procedure is similar as above. I.e., given the arguments z_1 and z_2, one first evaluates $\widetilde{z}_1 = \widetilde{f}_1(z_1)$ and $\widetilde{z}_2 = \widetilde{f}_2(z_2)$. Subsequently, one evaluates $\widetilde{y}_1 = \widetilde{f}_{y_1}(\cdot)$ with the argument $\widetilde{z}_1 + \widetilde{z}_2$, then $\widetilde{y}_2 = \widetilde{f}_{y_2}(\cdot)$ at the point $\widetilde{z}_1 - \widetilde{z}_2$. Finally, $\widetilde{f}(z_1, z_2) = 1/4(\widetilde{y}_1 - \widetilde{y}_2)$.

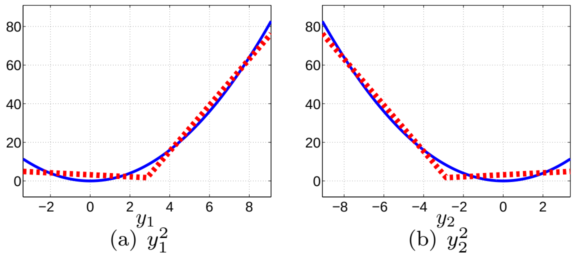
Example 4.2. Consider a non-separable function given as the product of the two functions discussed in Examples 3.1 and 3.2, i.e. $f(z_1, z_2) = f_1(z_1) f_2(z_2)$ with $f_1(z_1) = z_1^3$, $f_2(z_2) = |z_2| + 0.5 z_2^2 - \sin(z_2)^3$ on domain $[-1.5,\ 1.5] \times [-1,\ 2.5]$. Graph of the function is shown in Figure 3(a). In order to convert $f(z_1, z_2)$ into a separable form, we introduce variables y_1 and y_2 as per (9). The PWA approximation $\widetilde{f}(z_1, z_2) \approx f(z_1, z_2)$ is then given by (11). Here, $\widetilde{f}_1(z_1)$ was obtained by approximating $f_1(z_1)$ by a PWA function with 3 regions as shown in Figure 1(a), while $\widetilde{f}_2(z_2) \approx f_2(z_2)$ was approximated by 7 regions as depicted in Figure 2(b). Subsequently, the domains $[\underline{y}_1,\ \overline{y}_1]$ and $[\underline{y}_2,\ \overline{y}_2]$ were computed via (10), which resulted into $\operatorname{dom} y_1 = [-3.374,\ 9.095]$ and $\operatorname{dom} y_2 = [-9.095,\ 3.374]$. Finally, the PWA approximations $\widetilde{f}_{y_1}(y_1) \approx y_1^2$ and $\widetilde{f}_{y_2}(y_2) \approx y_2^2$ were obtained by solving the NLP (6) with $N = 2$. Graphs of y_1^2, y_2^2 and their respective PWA approximations are presented in Figure 4. The overall approximation $\widetilde{f}(z_1, z_2)$ therefore consists of 14 regions. Despite a rather crude approximation of the square functions, the combined PWA function (11), shown in Figure 3(b), features only a minor average approximation error of 3% and a worst-case error of 15%. By increasing the number of linearizations for y_1^2 and y_2^2 from $N = 2$ to $N = 4$ (hence increasing the complexity of $\widetilde{f}(z_1, z_2)$ from 14 to 18 regions), the average and worst-case errors can be further reduced to 1% and 8%, respectively.



(a) Graph of $f(z_1, z_2)$.

(b) Approximation $\widetilde{f}(z_1, z_2)$

Fig. 3. Graph of $f(z_1, z_2)$ and its PWA approximation (11) in Example 4.2.



(a) y_1^2

(b) y_2^2

Fig. 4. Functions y_i^2 (blue) and their PWA approximation $\widetilde{f}_{y_i}(y_i)$ (red dashed lines) in Example 4.2.

Separation of multivariable functions with more than two terms can be performed in an inductive manner. Consider $f(z_1, z_2, z_3) = f_1(z_1) f_2(z_2) f_3(z_3)$. First, approximate the product $f_1(z_1) f_2(z_2)$ by a PWA function of the form of (11), which requires four PWA approximations

$$\widetilde{f}_1(\cdot) \approx f_1(\cdot),\ \widetilde{f}_2(\cdot) \approx f_2(\cdot),\ \widetilde{f}_{y_1}(\cdot) \approx y_1^2,\ \widetilde{f}_{y_2}(\cdot) \approx y_2^2,$$

with y_1 and y_2 as in (9). Let $f_a(z_1, z_2) := f_1(z_1) f_2(z_2)$. Then $f(z_1, z_2, z_3) = f_a(z_1, z_2) f_3(z_3)$, which can again be approximated as a product of two functions. Specifically, define

$$y_3 = f_a(\cdot) + f_3(z_3), \quad y_4 = f_a(\cdot) - f_3(z_3), \tag{12}$$

and hence $f_a(z_1, z_2) f_3(z_3) = 1/4(y_3^2 - y_4^2)$. The domains over which y_3^2 and y_4^2 need to be approximated are, respectively, $[\underline{y}_3,\ \overline{y}_3]$ and $[\underline{y}_4,\ \overline{y}_4]$ with

$$\underline{y}_3 = \min\{f_1(z_1)f_2(z_2) + f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13a}$$
$$\overline{y}_3 = \max\{f_1(z_1)f_2(z_2) + f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13b}$$
$$\underline{y}_4 = \min\{f_1(z_1)f_2(z_2) - f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13c}$$
$$\overline{y}_4 = \max\{f_1(z_1)f_2(z_2) - f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13d}$$

and $z = [z_1,\ z_2,\ z_3]^T$. Subsequently, three additional PWA approximations

$$\widetilde{f}_{y_3}(y_3) \approx y_3^2,\ \widetilde{f}_{y_4}(y_4) \approx y_4^2,\ \widetilde{f}_3(z_3) \approx f_3(z_3)$$

need to be computed over the corresponding domains. The aggregated optimal PWA approximation $\widetilde{f}(z_1, z_2, z_3) \approx f(z_1)f(z_2)f(z_3)$ consists of 7 individual approximations and is given by

$$\widetilde{f}(\cdot) = 1/4\Big(\underbrace{\widetilde{f}_{y_3}(\widehat{f}_a + \widetilde{f}_3(z_3))}_{\widehat{y}_3} - \underbrace{\widetilde{f}_{y_4}(\widehat{f}_a - \widetilde{f}_4(z_3))}_{\widehat{y}_4}\Big). \tag{14}$$

Here, $\widehat{f}_a$ is the function value of $\widetilde{f}_a(z_1, z_2) \approx f_1(z_1)f_2(z_2)$ at z_1 and z_2, where $\widetilde{f}_a(\cdot)$ is obtained from (11), i.e.:

$$\widehat{f}_a = 1/4\Big(\underbrace{\widetilde{f}_{y_1}(\widetilde{f}_1(z_1) + \widetilde{f}_2(z_2))}_{\widehat{y}_1} - \underbrace{\widetilde{f}_{y_2}(\widetilde{f}_1(z_1) - \widetilde{f}_2(z_2))}_{\widehat{y}_2}\Big). \tag{15}$$

The overall PWA approximation $\widetilde{f}(z_1, z_2, z_3)$ can then be evaluated, for any $z_1,\ z_2,\ z_3 \in \mathcal{Z}$, by computing the function values of the respective approximations in the following order:

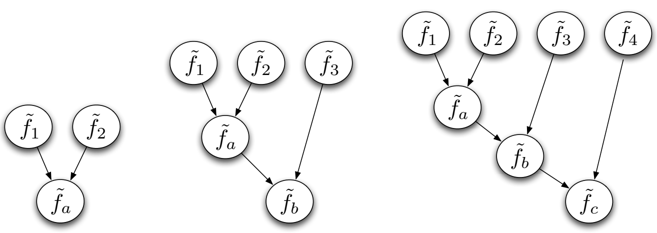
Step 1: $\widehat{y}_1 = \widetilde{f}_{y_1}(\widetilde{f}_1(z_1) + \widetilde{f}_2(z_2))$,
Step 2: $\widehat{y}_2 = \widetilde{f}_{y_2}(\widetilde{f}_1(z_1) - \widetilde{f}_2(z_2)$,
Step 3: $\widehat{y}_3 = \widetilde{f}_{y_3}(1/4(\widehat{y}_1 - \widehat{y}_2) + \widetilde{f}_3(z_3))$,
Step 4: $\widehat{y}_4 = \widetilde{f}_{y_4}(1/4(\widehat{y}_1 - \widehat{y}_2) - \widetilde{f}_3(z_3))$,
Step 5: $\widetilde{f}(z_1, z_2, z_3) = 1/4(\widehat{y}_3 - \widehat{y}_4)$.

Such an inductive procedure can be repeated *ad-infinitum* to derive PWA approximations of any multivariable function which satisfies Assumption 2.2. In general, the PWA approximation will consists of $2p + n$ individual PWA functions, where n is the number of variables in $f(z_1, \dots, z_n)$ and p is the number of products between individual subfunctions $f_j(z_j)$. As an example, for $f(\cdot) := \alpha_1 f_1(z_1)f_2(z_2)f_4(z_4) + \alpha_2 f_3(z_3)f_5(z_5)$ we have $p = 3$. We remark that inclusion of scalar multipliers α_j into the PWA description of the form (14)–(15) is straightforward and only requires linear scaling of the corresponding terms.

5. SOFTWARE IMPLEMENTATION

An algorithmic implementation of the inductive separation procedure of Section 4 is discussed next, provided that all functions are given in their symbolic representation. The procedure relies on two basic building blocks. The first one, represented by Algorithm 1, constructs the PWA approximation of a product of two functions, i.e. computes $\widetilde{f}(z_i, z_j) \approx f_i(z_i)f_j(z_j)$. Strictly speaking, the algorithm differentiates between two scenarios. If either f_i or f_j are PWA functions which approximate the product of some other functions (say $f_i \approx f_p f_q$), then $\widetilde{f} \approx f_i f_j$ is computed as shown in (12)–(15). Otherwise the procedure evidenced by (7)–(11) is followed.

Algorithm 2 then utilizes this block to construct a parse tree which defines the PWA approximation of the product of multiple functions, i.e. $\prod_{i=1}^n f_i(z_i)$. To illustrate the procedure, consider $f(z_1, z_2, z_3, z_4) = f_1(z_1)f_2(z_2)f_3(z_3)f_4(z_4)$. First, the stack of "unexplored" functions $\mathcal{S} = \{f_4, f_3, f_2, f_1\}$ is formed. In the first pass of the `while` cycle, f_1 and f_2 are popped from the stack and the PWA approximation $\widetilde{f}_a \approx f_1 f_2$ is computed by Algorithm 1. Subsequently, $\widetilde{f}_a$ is pushed back to $\mathcal{S}$ (which then becomes $\mathcal{S} = \{f_4, f_3, \widetilde{f}_a\}$), and new nodes of the parse tree $\mathcal{T}$ are created as shown in Figure 5(a). The procedure then repeats from Step 4. I.e., f_3 and $\widetilde{f}_a$ are popped from $\mathcal{S}$, $\widetilde{f}_b \approx f_3 f_a$ is computed, and the parse tree is updated as illustrated in Figure 5(b). Due to Step 6, $\mathcal{S} = \{f_4, \widetilde{f}_b\}$, and the algorithm therefore performs one more pass at which $\widetilde{f}_c \approx f_4 f_b$ is created and inserted into the tree, which finally looks like in Figure 5(c). The algorithm thereupon terminates since $\mathcal{S} = \{\widetilde{f}_c\}$ contains a single element.



(a) Step 1. (b) Step 2. (c) Step 3.

Fig. 5. Parsing tree $\mathcal{T}$ built by Algorithm 2.

If the function to be approximated contains sums of products, e.g. when $f(z_1, z_2, z_3, z_4) = \alpha_1 f_1(z_1)f_2(z_2) + \alpha_2 f_3(z_3)f_4(z_4)$, separate parsing trees have to be built by Algorithm 2 for each component of the summation. We remark that treating the scaling factors α_i only involves scaling the bottom-most node of the corresponding tree by the respective α_i.

Algorithm 1 PWA approximation of $f_i(z_i)f_j(z_j)$

```
INPUT: Functions $f_i(z_i)$, $f_j(z_j)$.
OUTPUT: Approximation $\widetilde{f}(z_i, z_j) \approx f_i(z_i)f_j(z_j)$.
Obtain the PWA approximations $\widetilde{f}_i(z_i) \approx f_i(z_i)$ and $\widetilde{f}_j(z_j) \approx f_j(z_j)$ by solving two NLPs (6).
Get $\underline{y}_i$, $\overline{y}_i$, $\underline{y}_j$, and $\overline{y}_j$ from (10) or (13).
Compute the PWA approximations $\widetilde{f}_{y_i}(y_i) \approx y_i^2$ and $\widetilde{f}_{y_j}(y_j) \approx y_j^2$ on domains $[\underline{y}_i,\ \overline{y}_i]$ and $[\underline{y}_j,\ \overline{y}_j]$ by solving two NLPs (6).
return $\widetilde{f}_i(z_i)$, $\widetilde{f}_j(z_j)$, and the symbolic representation of $\widetilde{f}(z_i, z_j)$.
```

The parsing tree generated by Algorithm 2 can be readily used to convert the PWA approximation $\widetilde{f}(z_1, \dots, z_n) \approx \sum_i \alpha_i \prod_j f_j(z_j)$ into a suitable mathematical model, which can subsequently be used for simulations, analysis, or control synthesis. Therefore we have created a software tool which takes a parsing tree $\mathcal{T}$ (or several such trees to accommodate for sums of products of functions), and *automatically* generates the corresponding HYSDEL representation of such a PWA approximation.

Algorithm 2 PWA approximation of $\prod_{i=1}^{n} f_i(z_i)$

```
INPUT: Functions $f_i(z_i)$.
OUTPUT: $\widetilde{f}(z_1, \dots, z_n) \approx \prod_{i=1}^{n} f_i(z_i)$.
Create an empty last-in-first-out stack $\mathcal{S}$ and an empty tree $\mathcal{T}$.
Push $f_i(z_i)$, $i = n, \dots, 1$ to the stack $\mathcal{S}$.
while $\mathcal{S}$ has more than one element do
    Pop two elements $f_j(z_j)$ and $f_k(z_k)$ from $\mathcal{S}$.
    Obtain $\widetilde{f}_j(z_j)$, $\widetilde{f}_j(z_j)$, and $\widetilde{f}(z_j, z_k) \approx f_j(z_j) f_k(z_k)$ by calling Algorithm 1.
    Push $\widetilde{f}(z_j, z_k)$ to $\mathcal{S}$.
    Create nodes $\widetilde{f}_j(z_j)$, $\widetilde{f}_k(z_k)$ and insert them to $\mathcal{T}$.
    Create a node $\widetilde{f}(z_j, z_k)$ and append it as a child of nodes $\widetilde{f}_j(z_j)$ and $\widetilde{f}_k(z_k)$.
end while
return Tree $\mathcal{T}$ representing $\widetilde{f}(z_1, \dots, z_n) \approx \prod_{i=1}^{n} f_i(z_i)$.
```

Formally, HYSDEL is a tool which transform a linguistic description of a hybrid system into the corresponding mathematical form. The language allows to define IF-THEN switching rules which, based on whether some logic condition is satisfied or not, set the value of certain continuous variables. This is achieved by modeling the rules as mixed-integer inequalities. This mechanism therefore allows to encode the IF-THEN rules of a PWA function (2), as illustrated on the following example. Consider a PWA function $f(z) : \mathbb{R} \mapsto \mathbb{R}$ with 3 regions:

$$f(z) = \begin{cases} a_1 z + c_1 & \text{if } z \leq r_1, \\ a_2 z + c_2 & \text{if } r_1 < z \leq r_2, \\ a_3 z + c_3 & \text{if } z > r_2. \end{cases} \tag{16}$$

Then the automatically generated HYSDEL representation of $f(z)$ is as follows.

```
 1 SYSTEM f {
 2  INTERFACE {
 3   INPUT  { REAL z; }
 4   OUTPUT { REAL f; }
 5   PARAMETER {
 6    REAL a1, a2, a3, c1, c2, c3, r1, r2;
 7   }
 8  }
 9  IMPLEMENTATION {
10   AUX { BOOL d1, d2; REAL f1, f2, f3; }
11   AD {
12    d1 = (z <= r1);
13    d2 = (z <= r2);
14   }
15   DA {
16    f1 = { IF d1       THEN a1*z + c1 };
17    f2 = { IF ~d1 & d2 THEN a2*z + c2 };
18    f3 = { IF ~d2      THEN a3*z + c3 };
19   }
20   OUTPUT { f = f1 + f2 + f3; }
21  }
22 }
```

Here, the INTERFACE section defines the input and output variables of the function, along with symbolic parameters a_i, c_i, and r_i. The AUX section then defined additional variables necessary to model the IF-THEN rules. The binary threshold detectors d1 and d2 used in the AD section will be set to true if and only if the analog condition on the right-hand-side of lines 12 and 13 is satisfied. The translation of these two lines into the corresponding mixed-integer inequalities will therefore ensure that d1=1 and d2=1 if $z \leq r_1$, then. For some z satisfying $r_1 < z \leq r_2$, the model will set d1=0 and d2=1. Finally, if $z > r_2$ holds, HYSDEL will transform the AD statements into mixed-integer inequalities which will enforce d1=0 and d2=0.

The binary indicators are subsequently used in the DA section which relates their truth value into the corresponding assignment. Here, we exploit the fact that the domain of $f(z)$ is connected, i.e. $(-\infty,\ r_1] \cup (r_1,\ r_2] \cup (r_2,\ \infty) = \mathbb{R}$, therefore each z will belong to exactly one region. If d1=1, then we know that $z \leq r_1$, and therefore $f_1 = a_1 z + c_1$. Implicitly, if the logic condition does not hold, $f_1 = 0$. Similarly, if d1=0 & d2=1 is true, then z belongs to the second region and hence $f_2 = a_2 z + c_2$. Again, if d1=0 & d2=1 does not hold, then $f_2 = 0$ is the implicit value. Therefore, for any value of z, exactly one of the variables f_1, f_2, f_3 will take a non-zero value. Therefore $f_1 + f_2 + f_3$ always gives the correct value of the PWA function $f(z)$. This relation is encoded on line 20.

The HYSDEL compiler will then take the source code and convert it into the Mixed Logical Dynamical (MLD) mathematical model. The MLD model can then be converted into the form of (2) e.g. by using the Multi-Parametric Toolbox (Kvasnica et al., 2004). The same toolbox can also be used to synthesize model predictive control strategies based either on MLD or PWA prediction models.

6. CASE STUDY

Consider a continuous stirred tank reactor (CSTR) where the reaction $A \rightarrow B$ takes place. The source compound is pumped into the reactor at a constant inflow with a constant concentration. The chemical reaction is exothermic and a coolant liquid is therefore pumped into the reactor's jacket to prevent overheating. The input temperature of the coolant is constant, while its flow rate q_c can be manipulated and is considered an exogenous input. Concentration of the reactant c_A inside of the reactor, temperature of the reactor mixture ϑ, and temperature of the cooling liquid in the jacket ϑ_c are the state variables of the CSTR. The normalized material and energy balances of such a reactor are then given by

$$\begin{aligned} \dot{c}_A &= \alpha_1 - \alpha_2 c_A - \alpha_3 c_A e^{-\beta/\vartheta}, \\ \dot{\vartheta} &= \alpha_4 - \alpha_5 \alpha_2 c_A e^{-\beta/\vartheta} + \alpha_6 \vartheta + \alpha_7 \vartheta_c, \\ \dot{\vartheta}_c &= \alpha_8 q_c + \alpha_9 (\vartheta - \vartheta_c) - \alpha_{10} \vartheta_c q_c, \end{aligned} \tag{17}$$

with constants α_i and β. The state and input variables are considered to belong to intervals $c_A \in [4,\ 4.2]$ mol $\cdot$ m^{-3}, $\vartheta \in [300,\ 320]$ K, $\vartheta_c \in [290,\ 310]$ K, and $q_c \in [0.002,\ 0.02]$ m$^3 \cdot$ h^{-1}.

The model features two nonlinearities: $\vartheta_c q_c$ and $c_A e^{-\beta/\vartheta}$, both of which satisfy Assumption 2.2. Since the first one involves a direct product of two variables, its PWA approximation $\widetilde{f}_a \approx \vartheta_c q_c$ can be obtained as in (8) by first defining $y_1 = \vartheta_c + q_c$, $y_2 = \vartheta_c - q_c$, followed by approximating the functions y_1^2 and y_2^2 by $\widetilde{f}_{y_1}(y_1)$ and $\widetilde{f}_{y_2}(y_2)$, respectively. Hence, the approximation $\widetilde{f}_1(\vartheta_c, q_c) \approx \vartheta_c q_c$ is represented by

$$\widetilde{f}_1(\vartheta_c, q_c) = 1/4\big(\widetilde{f}_{y_1}(\vartheta_c + q_c) - \widetilde{f}_{y_2}(\vartheta_c - q_c)\big). \tag{18}$$

The second nonlinearity can be approximated as in (11). First, the PWA approximation $\widetilde{g}(\vartheta) \approx e^{-\beta/\vartheta}$ is computed by solving (6). Then, $y_3 = c_A + e^{-\beta/\vartheta}$, $y_4 = c_A - e^{-\beta/\vartheta}$ are defined, followed by computing the respective PWA approximations $\widetilde{f}_{y_3}(y_3) \approx y_3^2$ and $\widetilde{f}_{y_4}(y_4) \approx y_4^2$. $\widetilde{f}_2(c_A, \vartheta) \approx c_A e^{-\beta/\vartheta}$ is thus given by

$$\widetilde{f}_2(c_A, \vartheta) = 1/4\Big(\widetilde{f}_{y_3}\big(c_A + \widetilde{g}(\vartheta)\big) - \widetilde{f}_{y_4}\big(c_A - \widetilde{g}(\vartheta)\big)\Big) \tag{19}$$

The overall PWA approximation of the original nonlinear system $\dot{x} = f(x, u)$ with $x = [c_A,\ \vartheta,\ \vartheta_c]^T$ and $u = q_c$ is thus

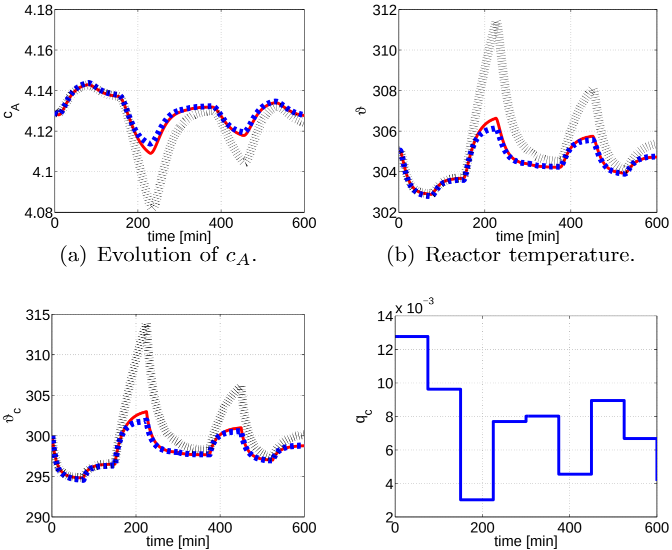
$$\begin{aligned}\dot{c}_A &\approx \alpha_1 - \alpha_2 c_A - \alpha_3 \widetilde{f}_2(c_A, \vartheta)\big),\\ \dot{\vartheta} &\approx \alpha_4 - \alpha_5 \alpha_2 \widetilde{f}_2(c_A, \vartheta) + \alpha_6 \vartheta + \alpha_7 \vartheta_c + \vartheta,\\ \dot{\vartheta}_c &\approx \alpha_8 q_c + \alpha_9(\vartheta - \vartheta_c) - \alpha_{10} \widetilde{f}_1(\vartheta_c, q_c),\end{aligned} \tag{20}$$

which can be easily converted into the general PWA form (2) as described in the previous section.

To assess approximation accuracy, we have investigated the open-loop evolution of the original nonlinear model (17) and compared it to the behavior of its PWA approximation (20). To derive the PWA model, we have chosen 3 regions for $\widetilde{f}_{y_1}(\cdot)$, $\widetilde{f}_{y_2}(\cdot)$ in (18) and $\widetilde{f}_{y_3}(\cdot)$, $\widetilde{f}_{y_4}(\cdot)$ in (19), and $N = 2$ for $\widetilde{g}(\theta) \approx e^{-\beta/\vartheta}$. The simulation results are shown in Figure 6. To better illustrate advantages of the PWA approximation, the simulation scenario also shows evolution of linearized version of (17) around the nominal steady state $c_A^s = 4.13$, $\vartheta^s = 304$, $\vartheta_c^s = 297$, and $q_c^s = 0.006$. As can be seen from the results, the PWA approximation clearly outperforms the model based on a single linearization. Specifically, the model (20) provides a 15 times more accurate tracking of the nonlinear profile compared to the linear model. Important to notice is that the PWA model consists of 14 local linear models. By increasing N to 7 when approximating $\widetilde{f}_{y_1}(\cdot)$, $\widetilde{f}_{y_2}(\cdot)$ in (18) and $\widetilde{f}_{y_3}(\cdot)$, $\widetilde{f}_{y_4}(\cdot)$ in (19), the approximation accuracy is 60 times better compared to the linear model. The cost to be paid is the increased model complexity, which would then consist of 30 regions.

The HYSDEL version of the PWA model (20) is provided next. Since HYSDEL only allows modeling of dynamical system in discrete time, we have used the Euler approximation of the continuous-time model (17), i.e. $x^+ \approx \Delta_t f(x, u) + x$ with sampling time $\Delta_t = 1$ minute. Simulations have shown that, due to a relatively slow dynamics of (17), evolution of the discrete-time approximation is almost undistinguishable from its continuous-time counterpart. The comments wrapped in `/* */` have been manually included to increase readability.

```
SYSTEM cstr {
 INTERFACE {
  STATE  {
   /* state variables with bounds */
   REAL ca [4.0, 4.2]; /* concentration of A */
   REAL v  [300, 320]; /* reactor temperature */
   REAL vc [290, 310]; /* jacket temperature */
```



(a) Evolution of c_A.

(b) Reactor temperature.

(c) Jacket temperature.

(d) Randomly varying coolant flowrate.

Fig. 6. Simulation results for the CSTR. Red line: nonlinear model (17), blue dashed line: PWA model (20), black dotted line: linear approximation.

```
  }
  INPUT  {
   /* input signal with bounds */
   REAL qc [0.002, 0.02]; /* coolant flowrate */
  }
  PARAMETER {
   /* approximation of exp(-beta/v) by 2 regions */
   REAL g_a1, g_a2, g_c1, g_c2, g_r;
   /* approximation of y_1^2 with 3 regions */
   REAL y1_a1, y1_a2, y1_a3;
   REAL y1_c1, y1_c2, y1_c3;
   REAL y1_r1, y1_r2;
   /* approximation of y_2^2 with 3 regions */
   REAL y2_a1, y2_a2, y2_a3;
   REAL y2_c1, y2_c2, y2_c3;
   REAL y2_r1, y2_r2;
   /* approximation of y_3^2 with 3 regions */
   REAL y3_a1, y3_a2, y3_a3;
   REAL y3_c1, y3_c2, y3_c3;
   REAL y3_r1, y3_r2;
   /* approximation of y_4^2 with 3 regions */
   REAL y4_a1, y4_a2, y4_a3;
   REAL y4_c1, y4_c2, y4_c3;
   REAL y4_r1, y4_r2;
   /* parameters alpha_i */
   REAL a1, a2, a3, a4, a5;
   REAL a6, a7, a8, a9, a10, beta;
   /* sampling time */
   REAL Ts;
  }
 }
 IMPLEMENTATION {
  AUX {
   /* logic threshold detectors */
   BOOL g_d, y1_d1, y1_d2, y2_d1, y2_d2;
   BOOL y3_d1, y3_d2, y4_d1, y4_d2;
   /* auxiliary continuous variables */
   REAL g, f1, f2;
   REAL y1, y1_f1, y1_f2, y1_f3;
   REAL y2, y2_f1, y2_f2, y2_f3;
   REAL y3, y3_f1, y3_f2, y3_f3;
   REAL y4, y4_f1, y4_f2, y4_f3;
  }
  /* approximation of exp(-beta/v) */
  AD {
   g_d   = (v <= g_r);
  }
  DA {
   /* special case for 2 regions */
   g = { IF g_d THEN g_a1*v + g_c1
                ELSE g_a2*v + g_c2 };
  }
  /* approximation of vc*qc */
  AD {
   y1_d1 = ( vc + qc ) <= y1_r1;
   y1_d2 = ( vc + qc ) <= y1_r2;
   y2_d1 = ( vc - qc ) <= y2_r1;
   y2_d2 = ( vc - qc ) <= y2_r2;
  }
  DA {
   y1_f1 = { IF y1_d1          THEN y1_a1*v + y1_c1 };
   y1_f2 = { IF ~y1_d1 & y1_d2 THEN y1_a2*v + y1_c2 };
   y1_f3 = { IF ~y1_d2         THEN y1_a3*v + y1_c3 };
   y2_f1 = { IF y2_d1          THEN y2_a1*v + y2_c1 };
   y2_f2 = { IF ~y2_d1 & y2_d2 THEN y2_a2*v + y2_c2 };
   y2_f3 = { IF ~y2_d2         THEN y2_a3*v + y2_c3 };
  }
  LINEAR {
   /* function value of \tilde{f}_{y_1} */
   y1 = y1_f1 + y1_f2 + y1_f3;
   /* function value of \tilde{f}_{y_2} */
   y2 = y2_f1 + y2_f2 + y2_f3;
   /* function value of \tilde{f}_1(y_1, y_2) */
   f1 = 1/4*(y1 - y2);
  }
  /* approximation of ca*exp(-beta/v) */
  AD {
   y3_d1 = ( ca + g )  <= y3_r1;
   y3_d2 = ( ca + g )  <= y3_r2;
   y4_d1 = ( ca - g )  <= y4_r1;
   y4_d2 = ( ca - g )  <= y4_r2;
  }
  DA {
   y3_f1 = { IF y3_d1          THEN y3_a1*v + y3_c1 };
   y3_f2 = { IF ~y3_d1 & y3_d2 THEN y3_a2*v + y3_c2 };
   y3_f3 = { IF ~y3_d2         THEN y3_a3*v + y3_c3 };
   y4_f1 = { IF y4_d1          THEN y4_a1*v + y4_c1 };
   y4_f2 = { IF ~y4_d1 & y4_d2 THEN y4_a2*v + y4_c2 };
   y4_f3 = { IF ~y4_d2         THEN y4_a3*v + y4_c3 };
  }
  LINEAR {
   /* function value of \tilde{f}_{y_3} */
   y3 = y3_f1 + y3_f2 + y3_f3;
   /* function value of \tilde{f}_{y_4} */
   y4 = y4_f1 + y4_f2 + y4_f3;
   /* function value of \tilde{f}_2(y_3, y_4) */
   f2 = 1/4*(y3 - y4);
  }
  CONTINUOUS {
   /* discrete-time state-update equations */
   ca = (a1 - a2*ca - a3*f2)*Ts + ca;
   v  = (a4 - a5*a2*f2 + a6*v + a7*vc)*Ts + v;
   vc = (a8*qc + a9*(v-vc) - a10*f1)*Ts + vc;
  }
 }
}
```

7. CONCLUSIONS

We have shown that a large class of dynamical systems with nonlinear vector fields can be approximated by PWA systems of fixed complexity in an optimal manner. The procedure boils down to solving a series of one-dimensional problems for which efficient solution methods exist. Derivation of the approximation can be easily automated and the HYSDEL variant of the hybrid approximation can be generated, hence allowing for subsequent control synthesis based on the hybrid model. An experimental version of the automatic translator is available for free download at http://www.kirp.chtf.stuba.sk/~sw/.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/4055/07, and 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Adjiman, C.S., Androulakis, I.P., Maranas, C.D., and Floudas, C.A. (1996). A global optimization method, αBB for process design. *Computers and Chemical Engineering*, 20, 419–424.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Branicky, M. (1995). *Studies in hybrid systems: modeling, analysis, and control.* Ph.D. thesis, LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA.

Chachuat, B., Singer, A.B., and Barton, P.I. (2006). Global methods for dynamic optimization and mixed-integer dynamic optimization. *Ind. Eng. Chem. Res.*, 45(25), 8373–8392.

De Schutter, B. and Van den Boom, T. (2001). On model predictive control for max-min-plus-scaling discrete event systems. *Automatica*, 37(7), 1049–1056.

Ferrari-Trecate, G. (2005). Hybrid Identification Toolbox (HIT). Available from http://www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/HIT_toolbox.html.

Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2001). Identification of Piecewise Affine and Hybrid Systems. In *Proc. on the American Control Conference*, 3521–3526. Arlington (VA), USA.

Heemels, W.P.M., De Schutter, B., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.

Heemels, W., Schumacher, J., and Weiland, S. (2000). Linear complementarity systems. *SIAM Journal on Applied Mathematics*, 60(4), 1234–1269.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/~mpt/.

Kvasnica, M. and Herceg, M. (2010). HYSDEL 3.0. Available from http://kirp.chtf.stuba.sk/~kvasnica/hysdel3/.

Papamichail, I. and Adjiman, C.S. (2004). Global optimization of dynamic systems. *Computers and Chemical Engineering*, 28, 403–415.

Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40, 37–50.

Sontag, E.D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2), 346–358.

Torrisi, F. and Bemporad, A. (2004). HYSDEL — A tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, 12, 235–249.

Williams, H. (1993). *Model Building in Mathematical Programming.* John Wiley & Sons, Third Edition.

Comments – Remarks

Dynamic Optimization of Diafiltration Process for Albumin Production ⋆

Radoslav Paulen * **Miroslav Fikar** * **Zoltán Kovács** **
Peter Czermak **,***

* *Institute of Information Engineering, Automation and Mathematics FCFT, Slovak University of Technology in Bratislava, Slovakia, (e-mail: {radoslav.paulen,miroslav.fikar}@stuba.sk)*
** *Institute of Biopharmaceutical Technology, University of Applied Sciences Giessen-Friedberg, Giessen, Germany, (e-mail: {peter.czermak,kovacs.zoltan}@tg.fh-giessen.de)*
*** *Department of Chemical Engineering, Kansas State University, Manhattan, Kansas, USA*

Abstract: This paper is devoted to finding an optimal control of a diafiltration process which is designed to purify and to concentrate human albumin in given albumin-ethanol solution. This process can be controlled by time-dependent adding of diluant (water). We address the control of diluant utilization in a more general and fundamental form than any previous attempts. Instead of considering arbitrarily constructed schemes, we determine the optimal time-dependent profile of the diluant flow for the entire process by employing dynamic optimization methods. The paper addresses two problems of optimal process operation: (1) the minimization of process time, and (2) the minimization of applied diluant volume. Control vector parametrization approach is applied considering different parametrized forms of the control function such as constant, linear, and piece-wise constant schemes.

Keywords: Optimal Control, Albumin Separation, Diafiltration.

1. INTRODUCTION

Albumin has a great pharmaceutical value and its production from human plasma is realized in industrial-scale. In the production line, a process stream is generated that is a ternary system containing ethanol, water, and albumin. This stream is then to be further processed to meet certain technological requirements. Namely, the albumin has to be concentrated, and simultaneously to that, the ethanol level has to be reduced. These dual objectives can be achieved employing diafiltration.

Diafiltration is known as an effective membrane filtration technique to separate a macro-solute from a micro-solute on the basis of their molecular size differences. It usually consists of a sequence of three operational modes. These are the *concentration mode (C)*, the *constant-volume dilution mode (CVD)*, and the *variable-volume dilution mode (VVD)*. They differ from each other in the utilisation of diluant. The actual operational regime then combines these three modes.

The traditional diafiltration (TD) technique consists of three consecutive steps including a pre-concentration, a constant-volume dilution mode, and a post-concentration step. Conventionally, as it is also described in detail by Jaffrin and Charrier (1994), this strategy is applied in albumin production.

In order to minimize diafiltration process time, a number of optimization strategies have recently been introduced (Asbi and Cheryan, 1992, Foley, 1999, Yazdanshenas et al., 2005) that usually consider traditional diafiltration and optimize switching times between steps. For this particular problem of ethanol/albumin separation, Jaffrin and Charrier (1994) have proposed VVD process. It utilizes the diluant at a rate that is less than the permeate flow rate, thus, this process enables a simultaneous concentration of macro-solute and removal of micro-solute in one single step. Most recently, Takači et al. (2009) have introduced another diafiltration approach. Instead of a stepwise water utilization strategy, the authors have considered the ratio of diluant flow to permeate flow as a continuous function of the operational time. Linear, logarithmic, and exponential functions have been studied, and their impact on the required diafiltration time was simulated. It has been found that the diafiltration process time improved around 10 percent in comparison with the strategy suggested by Jaffrin and Charrier (1994).

⋆ The first author and the second author gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/0537/10, 1/0095/11, and the Slovak Research and Development Agency under the project APVV-0029-07. This work was also financed by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. The third author would like to thank the Hessen State Ministry of Higher Education, Research and the Arts for the financial support within the Hessen initiative for scientific and economic excellence (LOEWE-Program).

It should be mentioned that the previous attempts have considered arbitrarily pre-defined schemes of water utilization. It has been indicated in previous works (Foley, 2006, Takači et al., 2009) that the problem of finding the optimal wash-water utilization strategy is still open. In this paper, we apply dynamic optimization methods that can significantly improve solution of the time minimization problem. Moreover, we employ the same methods to analyze a diluant minimization problem that can attract the attention of industrial sector as well.

2. ENGINEERING PROBLEM STATEMENT

The schematic representation of membrane diafiltration process is shown in Fig. 1. In a batch operation, the retentate stream is recirculated to the feed tank, and the permeate stream $q(t)$ is collected separately. During the operation, fresh solute-free diluent stream $u(t)$ (i.e. wash-water) can be added into the feed tank to replace solvent losses.

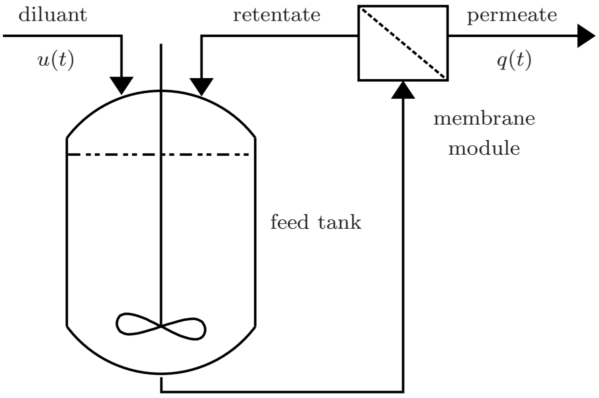


Fig. 1. Schematic representation of diafiltration process.

We consider an ultrafiltration (UF) application that is experimentally investigated by Jaffrin and Charrier (1994). The UF apparatus, the experimental conditions, the membrane properties, and the sample analysis have been described in detail in their work. In brief, a cellulose acetate membrane with 20 kDa cut-off was used for albumin ultrafiltration. The permeate flux was monitored at various albumin and ethanol concentrations using a recirculation flowrate of 300 L/h and a transmembrane pressure of 4 bar. It was found that the membrane is freely permeable to ethanol, but no albumin can pass through it. The experimental investigation has also shown that both albumin concentration c_1 and ethanol concentration c_2 considerably affects the permeate flow q. In the following, we use the relation computed and validated in their study given as

$$q(c_1, c_2) = \frac{1}{b_1 + b_2c_1 + b_3c_2 + b_4c_1c_2 + b_5c_1^2 + b_6c_2^2} \qquad (1)$$

where the constants b_i can be found in Table 1. In this study, normalized process time is being used, that enables straightforward scale-up calculations. Normalized process time is defined as the time necessary to process an initial feed solution of $0.0666\,\mathrm{m}^3$ which corresponds to $1\,\mathrm{kg}$ of albumin (initial albumin concentration is $15\,\mathrm{kg\,m^{-3}}$) being separated with $1\,\mathrm{m}^2$ membrane. We investigate different cases of initial concentration of ethanol ($c_{2,0}$) and of restriction for final concentration of albumin ($c_{1,f}$). These are summarized in Table 2.

Table 1. Permeate volumetric flowrate relation constants (Data taken from Takači et al. (2009)).

Constant	Value
b1	2.877
b2	1.698E-01
b3	1.874E-02
b4	5.708E-04
b5	-2.394E-04
b6	9.334E-05

Table 2. Initial and final conditions on macro-/micro-solute concentrations.

Case	$c_{1,0}$ [kg m^{-3}]	$c_{1,f}$ [kg m^{-3}]	$c_{2,0}$ [kg m^{-3}]	$c_{2,f}$ [kg m^{-3}]
1	15	80	98.35	0.1
2	15	80	146.3	0.1
3	15	80	194.3	0.1
4	15	120	98.35	0.1
5	15	120	146.3	0.1
6	15	120	194.3	0.1
7	15	240	98.35	0.1
8	15	240	146.3	0.1
9	15	240	194.3	0.1

3. PROCESS MODEL

In our previous study (Kovács et al., 2009), we have proposed a comprehensive mathematical model in a compact form for batch diafiltration processes. The process dynamics can be described by the following first-order ordinary differential equations (ODE) with their corresponding initial conditions

$$\dot{V} = u - q(c_1, c_2), \qquad V(t_0) = V_0 \qquad (2)$$

$$\dot{c}_1 = \frac{c_1}{V}\left[q(c_1, c_2)\mathcal{R}_1(c_1, c_2) - u\right], \qquad c_1(t_0) = c_{1,0} \qquad (3)$$

$$\dot{c}_2 = \frac{c_2}{V}\left[q(c_1, c_2)\mathcal{R}_2(c_1, c_2) - u\right], \qquad c_2(t_0) = c_{2,0} \qquad (4)$$

where V represents feed tank solution volume, u is fresh solute-free diluant volumetric flowrate, and q stands for permeate volumetric flow rate. $\mathcal{R}_1$ and $\mathcal{R}_2$ denote the rejection of macro-solute and micro-solute, respectively. It should be mentioned that, in a general case, the actual value of both q and $\mathcal{R}$ depends on the concentrations of macro-solute c_1 as well as on micro-solute concentration c_2. In this specific application, however, the rejection of the macro-solute (i.e. albumin) is found to be unity, while the rejection of the micro-solute (i.e. ethanol) to be zero. Thus, the model reduces to the following initial value problem

$$\dot{V} = u - q, \qquad V(t_0) = V_0 \qquad (5)$$

$$\dot{c}_1 = \frac{c_1}{V}(q - u), \qquad c_1(t_0) = c_{1,0} \qquad (6)$$

$$\dot{c}_2 = -\frac{c_2}{V}u, \qquad c_2(t_0) = c_{2,0} \qquad (7)$$

Control of diafiltration processes is traditionally operated using a dimensionless variable $\alpha(t)$ which is defined as a fraction between inflow and outflow

$$\alpha(t) = \frac{u(t)}{q(t)} \tag{8}$$

Introducing $\alpha(t)$ into Equations (5)–(7) yields

$$\dot{V} = (\alpha - 1)q, \qquad V(t_0) = V_0 \tag{9}$$

$$\dot{c}_1 = \frac{c_1}{V}(1-\alpha)q, \qquad c_1(t_0) = c_{1,0} \tag{10}$$

$$\dot{c}_2 = -\frac{c_2}{V}\alpha q, \qquad c_2(t_0) = c_{2,0} \tag{11}$$

4. PROCESS OPTIMIZATION

In this section, two different process optimization problems are introduced and solved. The first one represents a traditional minimum time problem. In this problem, optimal trajectories of function $\alpha(t)$ are computed in order to minimize running time of a batch diafiltration process. The second problem takes a different approach to enhance process performance. Here, the minimization of diluant consumption during the diafiltration process is considered. This is an important issue especially in pharmaceutical industry. In such applications, the diafiltration water has to meet strict quality regulations to avoid contamination of the process stream, and thus, is available at a considerably high price.

4.1 Minimum Time Problem

The objective of this optimization task is to find the time dependent function $\alpha(t)$ which uses minimum time to drive the process from initial state to a prescribed terminal state. Mathematical formulation of this dynamic optimization problem is as follows

$$J_1 = \min_{\alpha(t)} t_f \tag{12a}$$

s.t.

$$\dot{V} = (\alpha - 1)q, \quad V(t_0) = V_0, \quad V(t_f) = V_0\frac{c_{1,0}}{c_{1,f}} \tag{12b}$$

$$\dot{c}_1 = \frac{c_1}{V}(1-\alpha)q, \quad c_1(t_0) = c_{1,0}, \quad c_1(t_f) = c_{1,f} \tag{12c}$$

$$\dot{c}_2 = -\frac{c_2}{V}\alpha q, \quad c_2(t_0) = c_{2,0}, \quad c_2(t_f) = c_{2,f} \tag{12d}$$

Note that solution volume at the final time V_f is fully determined from the desired final macro-solute concentration value $c_{1,f}$ and follows from its material balance.

4.2 Minimum Diluant Problem

The second problem addresses minimization of total amount of diluant $u(t) = \alpha(t)q(t)$ used to drive the process from initial state to a prescribed terminal state assuming that the final time t_f is a free variable. Mathematical formulation (12) remains unchanged in this case except for the cost function

$$J_2 = \min_{\alpha(t)} \int_{t_0}^{t_f} \alpha(t)q(t)\mathrm{d}t \tag{13}$$

4.3 Optimization Strategy

Presented optimization problems can be solved using various approaches of dynamic optimization. Most popular are methods based on discretization of original infinite-dimensional problem to a finite-dimensional problem which can be then treated by a nonlinear programming (NLP) tools. Even if the solution is not truly optimal in the sense of the original formulation, methods are numerically robust and computed solutions converge to infinite-dimensional formulation.

According to the applied discretization, the methods can be based on discretization of control, e.g. control vector parametrization (CVP) method (Balsa-Canto et al., 2001, Fikar and Latifi, 2002, Goh and Teo, 1988, Vassiliadis et al., 1994) and on complete (controls and states) discretization, e.g. orthogonal collocation (OC) (Avraam et al., 1998, Cuthrell and Biegler, 1987, Logsdon and Biegler, 1989). Both groups possess certain advantages and drawbacks. In general, OC produces a large sparse NLP formulation and is of infeasible type, where solution is obtained only if optimum is found. On the other hand, CVP spends a large fraction of time in solution of differential equations even for a combination of optimized parameters that is far from the optimum. However, CVP methods can exploit robustness and efficiency of modern ODE solvers. Some of these are capable to provide sensitivity information used for evaluation of a more accurate gradient information (Hirmajer and Fikar, 2006).

There are numerous software packages (commercial or free) for solving dynamic optimization problems implemented in various programming environments. MATLAB packages such as OC based Dynopt (Čižniar et al., 2005) or CVP based DOTcvp (Hirmajer et al., 2008) are among those available freely.

In this paper we apply CVP approach where the control variable $\alpha(t)$ is considered to be parametrized as a combination of various time dependent terms. We consider function $\alpha(t)$ to be

(1) constant $\alpha(t) = \alpha_0$
(2) linear $\alpha(t) = \alpha_0 + \alpha_1 t$
(3) $N\times$piece-wise constant (PWC)

$$\alpha(t) = \sum_{k=1}^{N} \alpha_k \chi_k(t) \tag{14}$$

where α_k are values of $\alpha(t)$ at respective time intervals with

$$\chi_k(t) = \begin{cases} 1 & \text{if } t \in [t_{k-1}, t_k], \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

The constant $\alpha(t)$ formulation represents VVD strategy whereas linear $\alpha(t)$ formulation has been proposed in Takači et al. (2009).

If we assume that $\alpha(t)$ can be a function of time, we can approximate it by a trajectory that is piece-wise constant. Approximation with a low value of N (say 2–3) can produce results that are compatible with known diafiltration strategies as TD and VVD. As N becomes larger, we can decide whether traditional strategies are sufficient or if there is some room for improvement using more advanced $\alpha(t)$ trajectories.

5. RESULTS AND DISCUSSION

Optimization problems (12) and (13) were solved using CVP method with gradient information determined using sensitivity equations approach.

We found that proposed problems are strongly similar to each other in a certain sense, since the solutions are almost the same for both studied problems. At first we discuss results for constant and linear $\alpha(t)$.

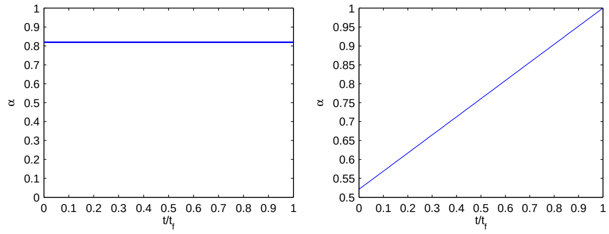
5.1 Constant and Linear Time Dependent $\alpha(t)$

The same optimum has been observed for both optimization problems. The improvement using linear compared to constant $\alpha(t)$ is in average 9.7% for the minimum time operation and 42.3% for the minimum diluant problem. The results obtained are given in Table 3 and optimal control profiles for Case 9 in Fig. 2.

Table 3. Minimum operation times and minimum diluant consumptions for constant and linear $\alpha(t)$.

Case	constant $\alpha(t)$		linear $\alpha(t)$	
	J_1^*[h]	J_2^*[m^3]	J_1^*[h]	J_2^*[m^3]
1	2.40	0.223	2.19	0.152
2	2.63	0.236	2.42	0.160
3	2.85	0.245	2.64	0.166
4	2.37	0.193	2.13	0.116
5	2.60	0.204	2.36	0.123
6	2.81	0.212	2.59	0.126
7	2.31	0.155	2.00	0.071
8	2.54	0.164	2.24	0.075
9	2.75	0.171	2.47	0.076

Note that if the target albumin concentration $c_1(t_f)$ increases, faster process operation is obtained. This confirms the finding of Takači et al. (2009). Similarly, it has been observed in Asbi and Cheryan (1992) that minimum time does not depend on concentration monotonously and can increase as well.



(a) $\alpha(t)$ profile

(b) $\alpha(t)$ profile

Fig. 2. Optimal trajectories of $\alpha(t)$, Case 3. Left: constant α, right: linear α.

5.2 Piece-wise Constant $\alpha(t)$ for Minimum Time Problem

We have chosen 2, 3, and 40 PWC segments in order to investigate the impact of choosing PWC control strategy to minimize the total time of process operation. Table 4 summarizes results obtained by several PWC functions $\alpha(t)$. Starred cost function J_1^* represents the minimum time attained, whereas the unstarred cost function J_2 means evaluation of the corresponding total diluant consumption.

Table 4. Minimum operation times and diluant consumptions for different $N\times$ PWC $\alpha(t)$.

Case	2×PWC $\alpha(t)$		3×PWC $\alpha(t)$		40×PWC $\alpha(t)$	
	J_1^*[h]	J_2[m^3]	J_1^*[h]	J_2[m^3]	J_1^*[h]	J_2[m^3]
1	2.04	0.086	2.04	0.086	2.04	0.088
2	2.30	0.104	2.30	0.104	2.29	0.103
3	2.54	0.118	2.54	0.123	2.54	0.124
4	1.98	0.058	1.98	0.058	1.98	0.059
5	2.24	0.075	2.24	0.075	2.24	0.076
6	2.49	0.088	2.48	0.095	2.48	0.096
7	1.84	0.030	1.84	0.030	1.84	0.030
8	2.11	0.044	2.11	0.044	2.11	0.047
9	2.36	0.055	2.35	0.067	2.35	0.063

Compared to constant or linear case, advantages of using PWC profiles are evident. The average gain is 64.8% for diluant problem and 14.3% for minimum time problem in comparison with constant α.

If treating $\alpha(t)$ as a PWC function, obtained results show that there is a similarity in trajectories of optimal $\alpha(t)$ for cases 1,4,7; cases 2,5,8; and for cases 3,6,9.

When cases 1,4,7 are considered, two and three PWC segments produce the same optimal operation characterised as the traditional diafiltration (TD) process with $\alpha(t) = \{0, 1\}$ for appropriate time lengths.

For other cases, two and three PWC segments produce similar solutions that are a combination of VVD and TD processes with $\alpha(t) = \{\alpha_0, 1\}$ where $\alpha_0 > 0$.

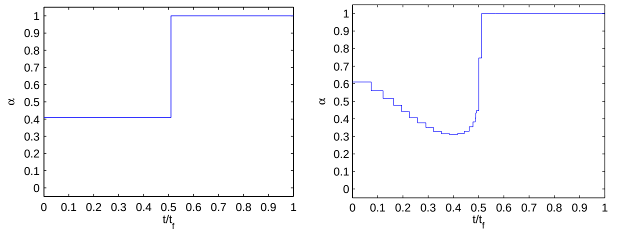
Although a finer PWC approximation (forty PWC segments) exhibits a different optimal $\alpha(t)$ trajectory, minimum operation time stays almost unchanged and differences in final time between 3 and 40 segments are negligible. Hence, minimum is in this case flat, i.e. final time value is not strongly influenced by shifting $\alpha(t)$ trajectory from the optimal to traditional diafiltration operation. Figure 3 shows optimal control trajectory and its differences for 2 and 40 segments in Case 9. Corresponding state trajectories for 40 segments are shown in Figure 4. Thus, some of the results support claims of Jaffrin and Charrier (1994) that VVD operation is time optimal for this process. However, if the concentration $c_{2,0}$ is smaller (Cases 1,4,7), TD operation can be used.

If our results are compared with those obtained in Takači et al. (2009), we can see that the proposed approach performs significantly better since we do not consider predefined functions of $\alpha(t)$ (linear, logarithmic, exponential) but we permit it to behave arbitrarily. This results in lower minimum operation times obtained.

5.3 Piece-wise Constant $\alpha(t)$ for Minimum Diluant Problem

We have again chosen 2, 3, and 40 PWC segments in order to investigate the impact of choosing PWC control strategy to minimize the diluant comsumption. Table 5 summarizes the results.

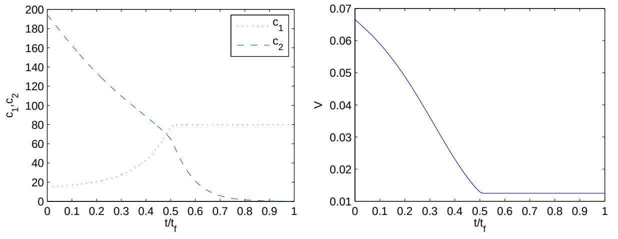
Starred cost function J_2^* represents the minimum diluant consumption attained, whereas the unstarred cost function J_1 means the corresponding total time.



(a) $\alpha(t)$ profile

(b) $\alpha(t)$ profile

Fig. 3. Optimal minimum time trajectories of $\alpha(t)$, Case 3. Left: 2 PWC segments, right: 40 PWC segments.



(a) $\alpha(t)$ profile

(b) $\alpha(t)$ profile

Fig. 4. Optimal minimum time state trajectories, Case 3, 40 PWC segments. Left: concentrations, right: volume.

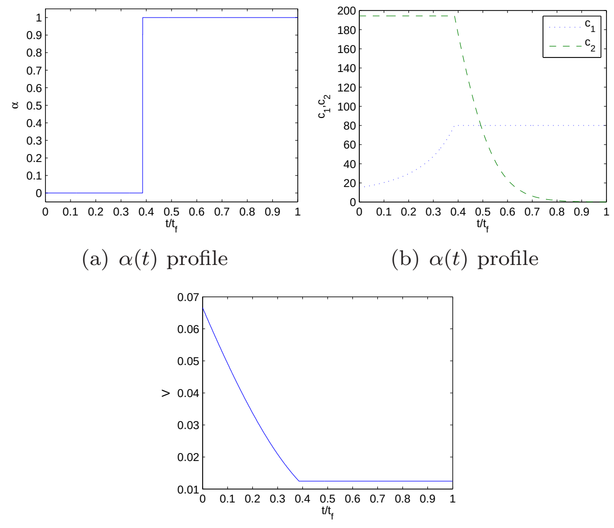
Table 5. Operation times and minimum diluant consumptions for different $N\times$ PWC $\alpha(t)$.

Case	2×PWC $\alpha(t)$		3×PWC $\alpha(t)$		40×PWC $\alpha(t)$	
	J_1[h]	J_2^*[m^3]	J_1[h]	J_2^*[m^3]	J_1[h]	J_2^*[m^3]
1	2.04	0.086	2.04	0.086	2.04	0.086
2	2.31	0.091	2.31	0.091	2.31	0.091
3	2.59	0.095	2.59	0.095	2.55	0.095
4	1.98	0.057	1.98	0.057	1.98	0.057
5	2.26	0.061	2.26	0.061	2.24	0.061
6	2.55	0.063	2.55	0.063	2.52	0.063
7	1.84	0.029	1.84	0.029	1.85	0.029
8	2.13	0.030	2.13	0.030	2.13	0.030
9	2.42	0.032	2.42	0.032	2.42	0.032

The average gain in the minimum diluant consumption is 69% comparing to constant $\alpha(t)$ case. TD process operation was proved to be optimal in all cases. For different number of PWC segments, obtained minimum final times differ only slightly from these computed in previous case. Thus, the minimum diluant consumption and minimum time operation is the same in Cases 1,4,7. If concentration $c_{2,0}$ is increased, it is possible to obtain optimal operation with substantially less diluant as in the minimum time problem but at the expense of longer processing times. Figure 5 shows optimal control and state trajectories in Case 9 with 40 PWC segments.

It is possible to compare minimum time and minimum diluant optimal trajectories. If minimum time is considered, $\alpha(t)$ forces both concentrations to be taken towards desired terminal state as soon as possible. If the minimum diluant case is considered, the volume is reduced to its terminal condition as soon as possible.

Albumin diafiltration has been considered as a time minimization problem in previous studies (Jaffrin and Charrier, 1994, Takači et al., 2009). We would like to highlight that the computed minimal time is relatively insensitive to changes of the optimized control profile. Such changes, however, might dramatically influence the diluant consumption. Table 4 and 5 indicate that operation time cannot be decreased without an increase in the diluant consumption.



(a) $\alpha(t)$ profile

(b) $\alpha(t)$ profile

(c) $\alpha(t)$ profile

Fig. 5. Optimal minimum diluant trajectories, Case 3, 40 PWC segments. α, concentrations, volume.

Obviously, decision makers face a trade-off problem between diluant consumption and operational time. In fact, the most economical process can be determined as minimization of a mixed objective involving the operational cost of the pump and the cost of the utilized dilution water. In such a complex cost function, the operational cost of the pump is a product of the operational time, the power consumption of the pump, and the electricity price. Furthermore, the cost of the diluant depends on the unit price of the utilized diafiltration water. Such cost-benefit analysis is out of the scope of this paper since it relies on information on these cost factors of the industrial settings. However, this type of complex problem can also be solved in a similar manner with dynamic optimization tools as shown by Fikar et al. (2010).

6. CONCLUSIONS

In this work, a diafiltration process for albumin purification is considered. To optimize the process, we have proposed dynamic optimization framework suitable to deal with different optimization strategies as well as different considered control trajectories. Control vector parametrization approach was used to obtain optimal constant, linear and piece–wise constant profiles of function $\alpha(t)$.

We have demonstrated the power of the proposed optimization method on two selected problems: minimum time and minimum diluant consumption problem, and investigated whether conventionally used diafiltration techniques can be considered as optimal. We found that two-step TD process, involving a pre-concentration and a constant-volume dilution mode step, is optimal in case of minimization of overall diluant consumption. However, obtained optimal trajectories can differ from the traditionally used operation in case of time minimization.

REFERENCES

Asbi, B.A. and Cheryan, M. (1992). Optimizing process time for utrafiltration and diafiltration. *Desalination*, 86, 49–62.

Avraam, M.P., Shah, N., and Pantelides, C.C. (1998). Modelling and optimisation of general hybrid systems in the continuous time domain. *Computers chem. Engng.*, 22, S221–S228.

Balsa-Canto, E., Banga, J.R., Alonso, A.A., and Vassiliadis, V.S. (2001). Dynamic optimization of chemical and biochemical processes using restricted second-order information. *Computers chem. Engng.*, 25(4–6), 539–546.

Cuthrell, J.E. and Biegler, L.T. (1987). On the optimization of differential-algebraic process systems. *AIChE Journal*, 33(8), 1257–1270.

Fikar, M., Kovács, Z., and Czermak, P. (2010). Dynamic optimization of batch diafiltration processes. *Journal of Membrane Science*, 355(1-2), 168–174.

Fikar, M. and Latifi, M.A. (2002). User's guide for FORTRAN dynamic optimisation code DYNO. Technical Report mf0201, LSGC CNRS, Nancy, France; STU Bratislava, Slovak Republic.

Foley, G. (1999). Minimisation of process time in ultrafiltration and continuous diafiltration: the effect of incomplete macrosolute rejection. *Journal of Membrane Science*, 163(1–2), 349–355.

Foley, G. (2006). Water usage in variable volume diafiltration: comparison with ultrafiltration and constant volume diafiltration. *Desalination*, 196, 160–163.

Goh, C.J. and Teo, K.L. (1988). Control parameterization : a unified approach to optimal control problems with general constraints. *Automatica*, 24(1), 3–18.

Hirmajer, T. and Fikar, M. (2006). Optimal control of a hybrid coupled tanks system. In R. Trappl (ed.), *Cybernetics and Systems 2006*, volume 1, 41–45. Autrian Society for Cybernetic Studies, Vienna, Austria.

Hirmajer, T., Čižniar, M., Fikar, M., Balsa-Canto, E., and Banga, J.R. (2008). Brief introduction to DOTcvp - dynamic optimization toolbox. In *Proceedings of the 8th International Scientific - Technical Conference Process Control 2008*. Kouty nad Desnou, Czech Republic.

Jaffrin, M. and Charrier, J. (1994). Optimization of ultrafiltration and diafiltration processes for albumin production. *Journal of Membrane Science*, 97, 71 – 81.

Kovács, Z., Fikar, M., and Czermak, P. (2009). Mathematical modeling of diafiltration. *Hungarian Journal of Industrial Chemistry*, 37(2), 159–164.

Logsdon, J.S. and Biegler, L.T. (1989). Accurate solution of differential-algebraic optimization problem. *Ind. Eng. Chem. Res.*, 28, 1628–1639.

Takači, A., Žikić-Došenović, T., and Zavargó, Z. (2009). Mathematical model of variable volume diafiltration with time dependent water adding. *Engineering Computations: International Journal for Computer-Aided Engineering and Software*, 26(7), 857–867.

Vassiliadis, V.S., Sargent, R.W.H., and Pantelides, C.C. (1994). Solution of a class of multistage dynamic optimization problems. 1. problems without path constraints, 2. problems with path constraints. *Ind. Eng. Chem. Res.*, 33, 2111–2122, 2123–2133.

Čižniar, M., Salhi, D., Fikar, M., and Latifi, M. (2005). Dynopt - dynamic optimisation code for MATLAB. In *Technical Computing Prague 2005*.

Yazdanshenas, M., Tabatabaeenezhad, A., Roostaazad, R., and Khoshfetrat, A. (2005). Full scale analysis of apple juice ultrafiltration and optimization of diafiltration. *Separation and Purification Technology*, 47(1–2), 52–57.

Comments – Remarks

Run-to-run Optimisation of Batch Processes with In-batch Controller

Marián Podmajerský * **Benoît Chachuat** ** **Miroslav Fikar** *

* *Institute of Information Engineering, Automation and Mathematics, STU in Bratislava, Slovakia, {marian.podmajersky,miroslav.fikar}@stuba.sk*
** *Centre for Process System Engineering, Department of Chemical Engineering, Imperial College London, UK, b.chachuat@imperial.ac.uk*

Abstract: This work examines an optimal process control of batch processes with process uncertainty. Assuming that the batch process is repeated several times then the run-to-run framework can be used. We propose an intergated control approach that computes a new control input to meet terminal constraints, between batches, and that improves this control input for model-mismatch, within batches. The simulation results show better convergence with proposed control scheme over an individual schemes dealing with either run-to-run adaptation or with in-batch neighbouring-extremal (NE) control.

Keywords: dynamic optimisation, neighbouring-extremal control, optimal control, integrated control scheme, run-to-run adaptation

1. INTRODUCTION

Chemical processes are subject to large uncertainty during their operation. Common sources of uncertainty include measurement noise, inaccurate kinetic rate parameters, feed impurities, and fouling. This usually give rise to a lower production quality and quantity along with operational constraint violations. Optimal control has the ability to mitigate the effect of uncertainty on process performance, especially in the presence of constraints (Kadam and Marquardt, 2007).

A number of methodologies for dealing with uncertainty and disturbances in batch process can be found in the literature. Model predictive control (MPC) (Allgöwer and Zheng, 2000; Garcia et al., 1989) implements a re-optimisation strategy and uses measurements to update the current state of the model. This strategy suffers two important deficiencies: i) the presence of constraints may result in an infeasible solution; ii) the re-optimisations may not be tractable in real-time. Clearly, the time needed to re-optimise the system depends on both the problem complexity and the computing performance. Too large a computation time may lead to performance loss, or worse constraint violations, especially in chemical processes that exhibit fast process dynamic. In the so-called explicit MPC approach (Bemporad et al., 2002), multi-parametric programming is used to pre-compute off-line all possible control actions for a given range of the state variables. The control inputs are then adjusted by simply selecting the control law that corresponds to the actual state of the process, as given by the latest measurements. Although this method can accommodate fast sampling times, its foremost limitation comes from the curse of dimensionality and from the quality of the linearisations. This currently limits the application of explicit MPC to problems having no more than a few state variables as well as piece-wise linear dynamics.

This paper presents a two-time-scale approach, whereby a run-to-run adaptation strategy (Bonvin et al., 2006) is implemented at the slow time scale (outer loop) and is integrated with a (constrained) neighbouring-extremal (NE) controller (Bryson and Ho, 1975) that operates at the fast time scale (inner loop). More specifically, run-to-run adaptation of the terminal constraints (Marchetti et al., 2007) is considered for the outer loop. In its original form, this scheme proceeds by re-optimising the batch operation between each run and adapting the terminal constraints based on the mismatch between their predicted and measured values; but no adaptation is made within a run. In order to reject disturbances within each run and at the same time promote feasibility and optimality, a NE controller is here considered as the inner loop. The theory of NE control, which has been developed over the last 4-5 decades to avoid the costly re-optimisation of (fast) dynamic systems, is indeed well-suited for batch process control. The integration between the outer- and inner-loops occurs naturally since the NE controllers are recalculated after each run based on the solution to the outer-loop optimization problem. The resulting integrated two-time-scale optimization scheme thus offers promise to enhance performance and tractability.

The paper is organised as follows. Theoretical background on NE control and run-to-run optimization is provided in Section 2. The proposed integrated two-times-scale optimisation scheme is described in Section 3 and demonstrated on the case study of a semi-batch reactor example in Section 4. Finally, Section 5 concludes the paper.

2. THEORETICAL BACKGROUND

2.1 Problem Formulation

Throughout the paper, the following dynamic optimisation problem with control and terminal bound constraints is considered:

$$\min_{\boldsymbol{u}} \; J = \phi(\boldsymbol{x}(t_f)) + \int_0^{t_f} L(\boldsymbol{x}(t),\boldsymbol{u}(t))\mathrm{d}t \tag{1}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t),\boldsymbol{u}(t)), \quad 0 \le t \le t_f \tag{2}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{3}$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f),t_f) \le \boldsymbol{\psi}_{ref} \tag{4}$$

$$\boldsymbol{u}^L \le \boldsymbol{u}(t) \le \boldsymbol{u}_U. \tag{5}$$

In (1)–(5), $t \ge 0$ denotes the time variable, with t_f the final time; $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the control vector; $\boldsymbol{x} \in \mathbb{R}^{n_x}$ the state vector, with initial value $\boldsymbol{x}_0$; J, ϕ and L the scalar cost, terminal cost, and integral cost, respectively; and $\boldsymbol{\psi}$ the vector of n_ψ terminal constraints. All the functions in (1)–(5) are assumed to be continuously differentiable with respect to all their arguments.

2.2 Necessary Conditions for Optimality

Following Bryson and Ho (1975), the Hamiltonian function H is defined as follows:

$$H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{\mu}^L,\boldsymbol{\mu}^U) = L(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u})^T\boldsymbol{\lambda} + \tag{6}$$

$$+ \boldsymbol{\mu}^L(\boldsymbol{u}^L - \boldsymbol{u}) + \boldsymbol{\mu}^U(\boldsymbol{u} - \boldsymbol{u}^U), \tag{7}$$

$\boldsymbol{\lambda} \in \mathbb{R}^{n_x}$ denotes the so-called adjoint (or costate) vector which satisfies

$$\dot{\boldsymbol{\lambda}} = -\boldsymbol{H}_{\boldsymbol{x}} = -\boldsymbol{F}_{\boldsymbol{x}}^T\boldsymbol{\lambda} - \boldsymbol{L}_{\boldsymbol{x}}, \; 0 \le t \le t_f, \tag{8}$$

with the terminal conditions given by

$$\boldsymbol{\lambda}(t_f) = \left[\boldsymbol{\phi}_{\boldsymbol{x}} + \boldsymbol{\nu}^T\boldsymbol{\psi}_{\boldsymbol{x}}\right]_{t=t_f}, \tag{9}$$

$\boldsymbol{\mu}^L(t), \boldsymbol{\mu}^U(t) \in \mathbb{R}^{n_u}$ are Lagrange multiplier vector functions satisfying

$$\boldsymbol{\mu}^{L^T}(\boldsymbol{u}^L - \boldsymbol{u}) = \boldsymbol{0}; \quad \boldsymbol{\mu}^L \ge \boldsymbol{0} \tag{10}$$

$$\boldsymbol{\mu}^{U^T}(\boldsymbol{u} - \boldsymbol{u}^U) = \boldsymbol{0}; \quad \boldsymbol{\mu}^U \ge \boldsymbol{0}, \quad 0 \le t \le t_f. \tag{11}$$

and $\boldsymbol{\nu} \in \mathbb{R}^{n_\psi}$ are Lagrange multipliers for the terminal constraints such that

$$\boldsymbol{0} = \nu_k\psi_k, \quad \nu_k \ge 0, \quad \text{for each } k = 1,\ldots,n_\psi. \tag{12}$$

Provided that the optimal control problem is not abnormal, the first- and second- order necessary conditions for optimality (NCO) read:

$$H_{\boldsymbol{u}} = L_{\boldsymbol{u}} + \boldsymbol{F}_{\boldsymbol{u}}^T\boldsymbol{\lambda} - \boldsymbol{\mu}^L - \boldsymbol{\mu}^U = \boldsymbol{0} \tag{13}$$

$$H_{\boldsymbol{uu}} \ge 0 \tag{14}$$

This latter determines the set of active terminal constraints at the optimum, which is denoted by the vector $\bar{\boldsymbol{\psi}}$ of dimension $n_{\bar{\psi}}$ and by complementary multiplier $\bar{\boldsymbol{\nu}}^*$. The constraints are inactive when the crresponding Lagrange multiplier is equal to zero. (The subscript such as $_y$ for a given variable denotes partial derivatives of that variable with respect to y.)

2.3 Neighbouring-extremal Control

Let's assume that the optimal control trajectory $\boldsymbol{u}^*(t)$ for the optimisation problem (1)–(5) consists of a sequence of constrained and unconstrained arcs. The optimal solution then comprises $\boldsymbol{x}^*(t)$, $\boldsymbol{\lambda}^*(t), \bar{\boldsymbol{\nu}}^*, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U, 0 \le t \le t_f$. For the control sequence, it is also assumed that the uncertainty is sufficiently small for the perturbed optimal control to have the same sequence of constrained and unconstrained arcs as the nominal solution.

The constrained optimal control problem obtained with a small variation in the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$ and in active terminal constraints $\bar{\boldsymbol{\psi}}(\boldsymbol{x}(t_f),t_f) = \delta\bar{\boldsymbol{\psi}}$ produces variations in optimal control vector $\delta\boldsymbol{u}(t)$, state vector $\delta\boldsymbol{x}(t)$, adjoint vector $\delta\boldsymbol{\lambda}(t)$ and Lagrange multiplier vector $\delta\bar{\boldsymbol{\nu}}$ (for the active terminal constraints $\bar{\boldsymbol{\psi}}$). Along unconstrained arcs, these variations can be calculated from the linearisation of the first-order NCO (10)–(12) around the extremal path (Bryson and Ho, 1975):

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}_{\boldsymbol{x}}^*\delta\boldsymbol{x} + \boldsymbol{F}_{\boldsymbol{u}}^*\delta\boldsymbol{u} \tag{15}$$

$$\delta\dot{\boldsymbol{\lambda}} = -\boldsymbol{H}_{\boldsymbol{xx}}^*\delta\boldsymbol{x} - \boldsymbol{F}_{\boldsymbol{x}}^{*T}\delta\boldsymbol{\lambda} - \boldsymbol{H}_{\boldsymbol{xu}}^*\delta\boldsymbol{u} \tag{16}$$

$$\boldsymbol{0} = \boldsymbol{H}_{\boldsymbol{ux}}^*\delta\boldsymbol{x} + \boldsymbol{F}_{\boldsymbol{u}}^{*T}\delta\boldsymbol{\lambda} + \boldsymbol{H}_{\boldsymbol{uu}}^*\delta\boldsymbol{u} \tag{17}$$

$$\delta\boldsymbol{x}(0) = \delta\boldsymbol{x}_0 \tag{18}$$

with additional conditions:

$$\delta\boldsymbol{\lambda}(t_f) = \left[\left(\boldsymbol{\phi}_{\boldsymbol{xx}}^* + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}_{\boldsymbol{xx}}^*\right)\delta\boldsymbol{x} + \bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^{*T}\delta\bar{\boldsymbol{\nu}}\right]_{t=t_f} \tag{19}$$

$$\delta\bar{\boldsymbol{\psi}} = \left[\bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^*\delta\boldsymbol{x}\right]_{t=t_f}. \tag{20}$$

A superscript * indicates that the corresponding quantity is evaluated along the extremal path $\boldsymbol{u}^*(t)$, $0 \le t \le t_f$, and corresponding states, adjoints and Lagrange multipliers.

Let us assume that the Hamiltonian function is regular, so that $\boldsymbol{H}_{\boldsymbol{uu}}^*$ is invertible along $0 \le t \le t_f$. The control variation $\delta\boldsymbol{u}(t)$ for these unconstrained arcs $\boldsymbol{\mu}^L = \boldsymbol{\mu}^U = \boldsymbol{0}$ is then given from (17):

$$\delta\boldsymbol{u}(t) = -(\boldsymbol{H}_{\boldsymbol{uu}}^*)^{-1}\left[\boldsymbol{F}_{\boldsymbol{u}}^{*T}\delta\boldsymbol{\lambda}(t) + \boldsymbol{H}_{\boldsymbol{ux}}^*\delta\boldsymbol{x}(t)\right]. \tag{21}$$

Overall, along constrained arcs, the control variation is equal to zero $\delta\boldsymbol{u}(t) = 0$. Then, $\delta\boldsymbol{x}(t)$ and $\delta\boldsymbol{\lambda}(t)$ satisfy the following multi-point boundary value problem (MPBVP):

$$\begin{aligned}
&\begin{pmatrix}\delta\dot{\boldsymbol{x}}(t)\\ \delta\dot{\boldsymbol{\lambda}}(t)\end{pmatrix} = \boldsymbol{\Delta}(t)\begin{pmatrix}\delta\boldsymbol{x}(t)\\ \delta\boldsymbol{\lambda}(t)\end{pmatrix},\\
&\delta\boldsymbol{x}(0) = \delta\boldsymbol{x}_0, \quad \delta\bar{\boldsymbol{\psi}} = \left[\bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^*\delta\boldsymbol{x}\right]_{t=t_f},\\
&\delta\boldsymbol{\lambda}(t_f) = \left[\left(\boldsymbol{\phi}_{\boldsymbol{xx}}^* + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}_{\boldsymbol{xx}}^*\right)\delta\boldsymbol{x} + \bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^{*T}\delta\bar{\boldsymbol{\nu}}\right]_{t=t_f}
\end{aligned} \tag{22}$$

where:

$$\boldsymbol{\Delta}(t) = \begin{cases}\begin{pmatrix}\boldsymbol{\alpha}(t) & -\boldsymbol{\beta}(t)\\ -\boldsymbol{\gamma}(t) & -\boldsymbol{\alpha}(t)^T\end{pmatrix} & \text{along unconstrained arcs}\\[2ex] \begin{pmatrix}\boldsymbol{F}_{\boldsymbol{x}}^* & \boldsymbol{0}\\ -\boldsymbol{H}_{\boldsymbol{xx}}^* & -\boldsymbol{F}_{\boldsymbol{x}}^{*T}\end{pmatrix} & \text{along constrained arcs}\end{cases} \tag{23}$$

and

$$\boldsymbol{\alpha}(t) := \boldsymbol{F}_{\boldsymbol{x}}^* - \boldsymbol{F}_{\boldsymbol{u}}^*(\boldsymbol{H}_{\boldsymbol{uu}}^*)^{-1}\boldsymbol{H}_{\boldsymbol{ux}}^* \tag{24}$$

$$\boldsymbol{\beta}(t) := \boldsymbol{F}_{\boldsymbol{u}}^*(\boldsymbol{H}_{\boldsymbol{uu}}^*)^{-1}\boldsymbol{F}_{\boldsymbol{u}}^{*T} \tag{25}$$

$$\boldsymbol{\gamma}(t) := \boldsymbol{H}_{\boldsymbol{xx}}^* - \boldsymbol{H}_{\boldsymbol{xu}}^*(\boldsymbol{H}_{\boldsymbol{uu}}^*)^{-1}\boldsymbol{H}_{\boldsymbol{ux}}^*. \tag{26}$$

Clearly, at each switching point between an unconstrained and a constrained arcs, a continuity of control, state and adjoint profiles must be preserved. For example, at a switching point between a lower bound and an interior arc, the value of control on lower bound matches the value of control in the interior arc $\boldsymbol{u}^H = \boldsymbol{u}^L$. Here, $\boldsymbol{u}^H$ represents the control obtained from solving the condition $H_{\boldsymbol{u}} = 0$.

In addition, state and adjoint trajectories are continuous at this point, too:

$$\boldsymbol{x}^*(t_k^+) = \boldsymbol{x}^*(t_k^-), \quad \boldsymbol{\lambda}^*(t_k^+) = \boldsymbol{\lambda}^*(t_k^-) \qquad (27)$$

Variations in switching times are difficult to determine and complicate the calculation of the NE control. To make this implementable, it is considered that the switching points are constant at their nominal times. The control values are then updated only between the fixed times. In practice, performance loss is negligible for small variations of switching times.

2.4 Numerical Computation of Neighbouring Feedback Control

The linear MPBVP (22) can be used to calculate the neighboring-extremal control correction $\delta\boldsymbol{u}(t)$, $0 \le t \le t_f$, in either one of two situations:

i. The initial state and (active) terminal constraint variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$ are available at discrete time instants, in which case the discrete feedback control can be obtained by directly re-solving the MPBVP. This can be done via a shooting method as described in Pesch (1989);
ii. The variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$ are available continuously in time, in which case the backward sweep method (Bryson and Ho, 1975) can be used to derive an explicit feedback control law. This approach is closely explained by Bryson and Ho (1975).

In this paper, we consider the first approach.

2.5 Run-to-run Constraint Adaptation

The principle behind run-to-run optimization is similar to MPC. But instead of adapting the initial conditions and moving the control horizon as is done in MPC, the adaptation is performed on the optimization model (e.g., model parameters or constraint biases) before re-running the optimizer. In run-to-run constraint adaptation, more specifically, the terminal constraints (5) in the optimization model are adapted after each run as (Marchetti et al., 2007):

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f), t_f) \le \delta\boldsymbol{\psi}, \qquad (28)$$

where $\delta\boldsymbol{\psi}$ stands for the terminal constraint bias. Such a bias can be directly updated as the difference between the available terminal constraint measurements, $\boldsymbol{\psi}^{\text{meas}}$, at the end of each run and the predicted constraint values. This simple strategy may however lead to excessive correction when operating far away from the optimum, and it may also exacerbate the sensitivity of the adaptation scheme to measurement noise. A better strategy consists of filtering the bias, e.g., with a first-order exponential filter:

$$\delta\boldsymbol{\psi}_{k+1} = [\boldsymbol{I} - \boldsymbol{W}]\,\delta\boldsymbol{\psi}_k + \boldsymbol{W}\left[\boldsymbol{\psi}_k^{\text{meas}} - \boldsymbol{\psi}(\boldsymbol{x}_k(t_f), t_f)\right], \qquad (29)$$

with k the run index, and $\boldsymbol{W}$ a gain matrix—typically, a diagonal matrix.

The constrained dynamic optimisation problem uses the available nominal process model. It is solved between each run, using any numerical procedure, such as the sequential or the simultaneous approach of dynamic optimisation. The optimal control trajectory $\boldsymbol{u}_k^*(t)$, $0 \le t \le t_f$, is computed and applied to the plant during the kth run. The predicted optimal response is denoted by $\boldsymbol{x}_k^*(t)$. The discrepancy between the measured terminal constraint values $\boldsymbol{\psi}_k^{\text{meas}}$ and the optimizer predictions $\boldsymbol{\psi}(\boldsymbol{x}_k^*(t_f), t_f)$ is then used to adjust the constraint bias as described earlier, before re-running the optimizer for the next run.

Of course, optimal control trajectory calculated between runs is suboptimal as the real process is never known perfectly.

3. TWO-TIMES-SCALE OPTIMISATION SCHEME

Run-to-run constraint adaptation was shown to be a promising technology in Marchetti et al. (2007). This approach provides a natural framework for handling changes in active constraints in dynamic process systems and it is quite robust towards model mismatch and process disturbances. Moreover, its implementation is simple. Inherent limitations of this scheme, however, are that (i) it does not perform any control corrections during the runs, and (ii) it typically leads to suboptimal performance.

On the other hand, neighbouring-extremal control is able to correct small deviations around the nominal extremal path in order to deliver similar performance as with re-optimisation. Since no costly on-line re-optimisation is needed, this approach is especially suited for processes with fast dynamics. However, the performance of NE control typically decreases dramatically in the presence of large model mismatch and process disturbances, and it requires a full-state measurement. This leads to sub-optimality or, worse, infeasibility when constraints are present or limited measurements are available.

Our proposal is to combine the advantages of these two approaches: Run-to-run constraint adaptation is applied at a slow time scale (outer loop) to handle large model mismatch and changes in active constraints, based on run-end measurements only. Further, NE control is applied at a fast time scale (inner loop) and uses measurement information available within each run, in order to enhance convergence speed and mitigate sub-optimality. It need to be stated that full-state measurement is required even in case of integrated scheme. The proposed integrated two-time-scale optimization scheme is depicted in Figure 1.

The implementation procedure is as follows:

Initialisation:

(0) Initialise the constraint bias $\delta\boldsymbol{\psi} = \boldsymbol{0}$, select a gain matrix $\boldsymbol{W}$ and set the run index to $k = 1$

Outer Loop:

(1) Determine $\boldsymbol{u}_k^*$ by solving the optimal control problem (1)–(5), then obtain the corresponding states $\boldsymbol{x}_k^*$ and adjoints $\boldsymbol{\lambda}_k^*$, with the active terminal constraints $\bar{\boldsymbol{\psi}}$ and Lagrange multipliers $\bar{\boldsymbol{\nu}}_k^*$, and together with Lagrange multipliers for boundary constraints $\boldsymbol{\mu}^L$ and $\boldsymbol{\mu}^U$ that satisfy NCO (10)–(12).

(2) Design a NE controller around the extremal path $\boldsymbol{u}_k^*$, either by using the backward sweep approach (continuous measurements), or by applying the shooting method (discrete measurements).

(3) **Inner Loop:**
Implement the NE controller during the kth run in order to calculate the corrections $\delta\boldsymbol{u}_k(t)$ to $\boldsymbol{u}_k^*(t)$

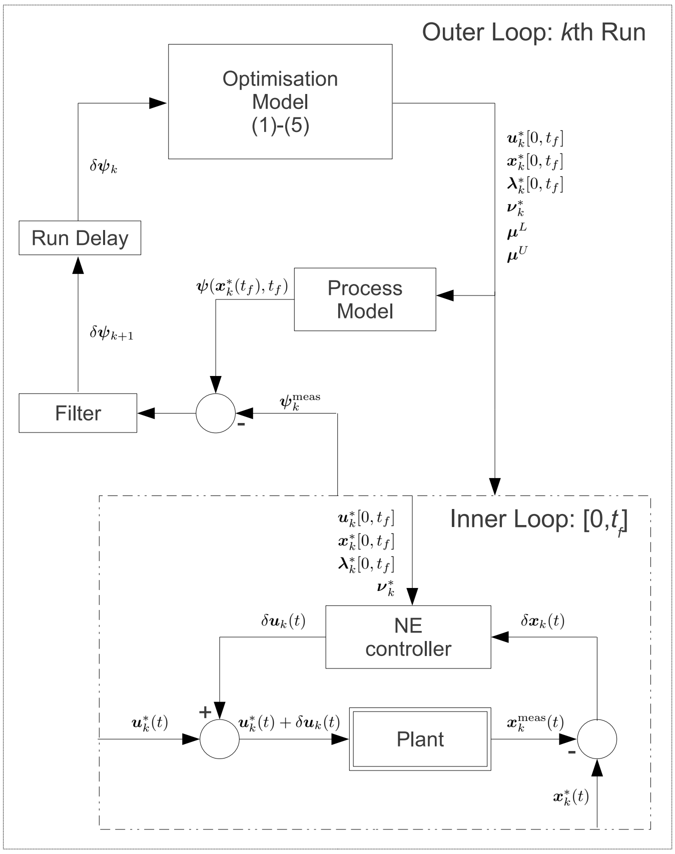


Fig. 1. Two-times-scale optimisation scheme employing NE control in the inner loop and run-to-run constraint adaptation in the outer loop.

based on the available (continuous or discrete) process measurements.

(4) Update the constraint bias $\delta\psi_{k+1}$ as the filtered difference between the measured values of the terminal constraints and their predicted counterparts.

(5) Increment the run index $k \leftarrow k+1$, and return to Step 1.

4. CASE STUDY

4.1 Semi-Batch reactor model

A semi-batch reactor example taken from Chen and Hwang (1990) is considered to illustrate the proposed integrated two-times-scale approach. The goal is to maximise the yield of ethanol using the feed rate $u(t)$ as the control variable, while keeping the liquid volume below some maximum threshold. Simple bound constraints are imposed on the feed rate. The mathematical formulation of this problem is:

$$\max_u J = c_{\mathrm{E}}(t_f)V(t_f) - 0.1\int_{t_0}^{t_f} u^2 \mathrm{d}t \tag{30}$$

s.t.

$$\dot{c}_{\mathrm{MS}}(t) = p_1(t)c_{\mathrm{MS}}(t) - u(t)\left(\frac{c_{\mathrm{MS}}(t)}{V(t)}\right) \tag{31}$$

$$\dot{c}_{\mathrm{S}}(t) = -10p_1(t)c_{\mathrm{MS}}(t) + u(t)\left(\frac{150 - c_{\mathrm{S}}(t)}{V(t)}\right) \tag{32}$$

$$\dot{c}_{\mathrm{E}}(t) = p_2(t)c_{\mathrm{MS}}(t) - u(t)\left(\frac{c_{\mathrm{E}}(t)}{V(t)}\right) \tag{33}$$

$$\dot{V}(t) = u(t) \tag{34}$$

where:

$$p_1(t) = \left(\frac{0.408}{1+\frac{c_{\mathrm{E}}}{16}}\right)\left(\frac{c_{\mathrm{S}}}{0.22+c_{\mathrm{S}}}\right) \tag{35}$$

$$p_2(t) = \left(\frac{1}{1+\frac{c_{\mathrm{E}}}{71.5}}\right)\left(\frac{c_{\mathrm{S}}}{0.44+c_{\mathrm{S}}}\right). \tag{36}$$

The state values $c_{\mathrm{MS}}(t)$, $c_{\mathrm{S}}(t)$, $c_{\mathrm{E}}(t)$ and $V(t)$ are the cell biomass, substrate, and ethanol concentrations [g/L], and the volume [L]. The final time is set to $t_f = 60\,\mathrm{h}$. The reactor container is initially fed by $V(0) = 10\,\mathrm{L}$ of reaction mixture with biomass and substrate concentrations $c_{\mathrm{MS}}(0) = 1\,\mathrm{g/L}$ and $c_{\mathrm{S}}(0) = 150\,\mathrm{g/L}$. No ethanol is initially present in the reaction mixture. The feed rate [L/h] is bounded as:

$$0 \le u \le 12\,[\mathrm{L/h}]. \tag{37}$$

The liquid volume is limited by $V^{\max} = 200\,\mathrm{L}$, so the terminal condition reads:

$$V(t_f) \le V^{\max}\,[\mathrm{L}]. \tag{38}$$

Note that the integral term $\int_{t_0}^{t_f} u^2\mathrm{d}t$ augments the original objective function in order to make the control problem non-singular. This way $H_{\boldsymbol{u}}$ depends on the control variable and Hamiltonian H is regular.

4.2 Open-loop optimal control

Solving the optimisation problem (30)–(38), with the sequential method (Edgar and Himmelblau, 1988; Guntern et al., 1998), the piecewise constant control profile (see Figure 2) shows the presence of one interior arc and two boundary arcs. Further analysis of this solution indicates that optimal control consists of a lower bound, an interior arc and another short lower bound. As the problem is regular, the control action along interior arc can be explicitly determined from the necessary conditions of optimality. Note that along boundary arcs, the control action is determined by a lower bound hence the control variations are simply $\delta\boldsymbol{u} = 0$. The switching times t_1 and t_2 between these arcs are not explicitly known and they need to be estimated, too. The switching times from piecewise constant control profile give good initial guess for these switching times. Overall, the optimal control solution is given as:

(1) $t \in (t_0, t_1)$, the control remains on its lower bound $u^*(t) = 0$

(2) $t \in (t_1, t_2)$, the control is given as a solution of a differential-algebraic system of equations:

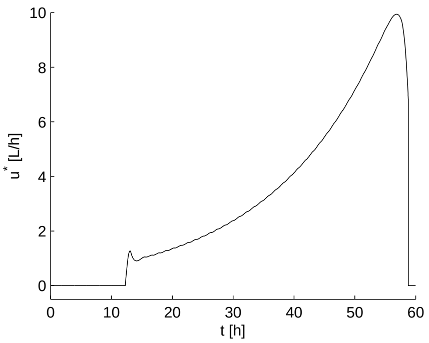


Fig. 2. Nominal control solution

$$\dot{\boldsymbol{x}} = \text{(31)–(36)}; \quad \delta\boldsymbol{x}(0) = \delta\boldsymbol{x}_0 \tag{39}$$

$$\dot{\boldsymbol{\lambda}} = -H_{\boldsymbol{x}}; \quad \boldsymbol{\lambda}(t_f) = \left[\boldsymbol{\phi}_{\boldsymbol{x}} + \boldsymbol{\nu}^T \boldsymbol{\psi}_{\boldsymbol{x}}\right]_{t=t_f} \tag{40}$$

$$0 = H_{\boldsymbol{u}} \tag{41}$$

(3) $t \in (t_1, t_f)$, the control remains on its lower bound $u^*(t) = 0$, again.

(39)–(41) is a non-linear multi-point boundary value problem. The optimal control profile is obtained by computing the switching times t_1 and t_2, the initial conditions for adjoints $\boldsymbol{\lambda}_0$, and the Lagrange multiplier for terminal constraint $\boldsymbol{\nu}$, according to the indirect shooting method (Bryson and Ho, 1975). It is checked that the performance index $J^* = 20689$ matches the objective value obtained with the sequential method $J = 20699$. Also see in Figure 3 that this optimal control profile satisfies the terminal constraint (38).

4.3 Closed-loop optimal control

The case study compares the performance of the two-time-scale integrated solution with a pure constraint adaptation control scheme and a pure neighbouring-extremal controller.

The integrated two-time-scale scheme is applied using a nominal model perturbed by varying the initial values and adding measurement noise. While the NE controller is designed using the nominal mathematical model, the simulations are performed for measured outputs from the perturbed model. It is also considered that full-state measurements are presented. The measured outputs are states with addition of white noise. Measurement noise is considered with the following variations in initial conditions $\delta\boldsymbol{x} = [0.17\ -6\ 0.9\ 0.8]$. These variations are chosen to cause a performance loss and terminal constraint violation, when applying the open-loop control profile. Run-to-run constraint adaptation is initialised with a constraint bias of $\delta\psi = 0$ and considers a filter gain of $W = 0.2$. The filter parameters were chosen so as to achieve the terminal constraint as fast as possible while avoiding oscillations during the adaptation.

Figure 4 compares the evolution of the performance during the first 20 batches. The evolution of the terminal constraint is presented in the left plot. Observe that the pure NE controller does not violate the terminal constraint, but on the other hand this constraint is inactive in all batches. In contrast, pure constraint adaptation violates the terminal constraint in the most of the batches. In last 5 batches, the method almost reaches terminal constraint. Note that this approach seems to be more sensitive to measurement noise then the other approaches. The integrated scheme starts in close proximity of terminal constraint. In the following batches, this result is slowly enhenced to meet the terminal constraint. Due to the fact that control corrections are applied during each batch as well, this approach is able to correct the control profile with lower sensitivity to measurement noise. The middle plot of Figure 4 shows the evolution of terminal constraint bias. This bias varies a little for the integrated scheme, because the NE controller in the inner loop is able to recover a large portion of optimality loss. In contrast, constraint adaptation requires heavier bias adaptation since no correction is made during the batch. The right plot shows evolution of the performance index. The worst average case is for pure NE control. In contrast, the pure constraint adaptation exhibits the highest values of performance indices. The cost function of the proposed integrated approach stays betwen these to extremes and is the closest to optimimal solution for perturbed system.

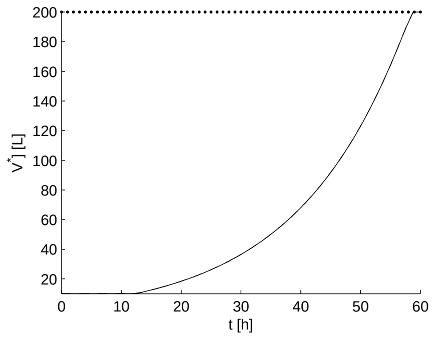


Fig. 3. Nominal state trajectory for liquid volume (solid line) and maximum value (dotted line)

The resulting control profile after adaptation within 20 batches is shown in right plot in Figure 5. The control still consists of the tree same arcs, but the switching times have changed compared to nominal solution displayed in Figure 2, as a result of the constrained adaptation. The corresponding measured output of liquid volume is presented in the left plot in Figure 5. It can be seen that the measured output of perturbed process is in good agreement with re-optimised solution.

5. CONCLUSIONS

In this paper, an integrated two-times-scale control scheme for batch processes has been proposed. It improves the performance of dynamic real-time optimisation applied to batch processes. The combination of two approaches, namely run-to-run adaptation and neighbouring-extremal control, allows to complement the benefits of each other, while mitigating some of their deficiencies. On the other hand, run-to-run adaptation allows to deal with large model mismatch and handles better terminal constraints. Advantages of the integrated scheme have been demonstrated on the case study for a semi-batch reactor. As part of future work, an extension of the current scheme to singular control problems is currently under investigation,

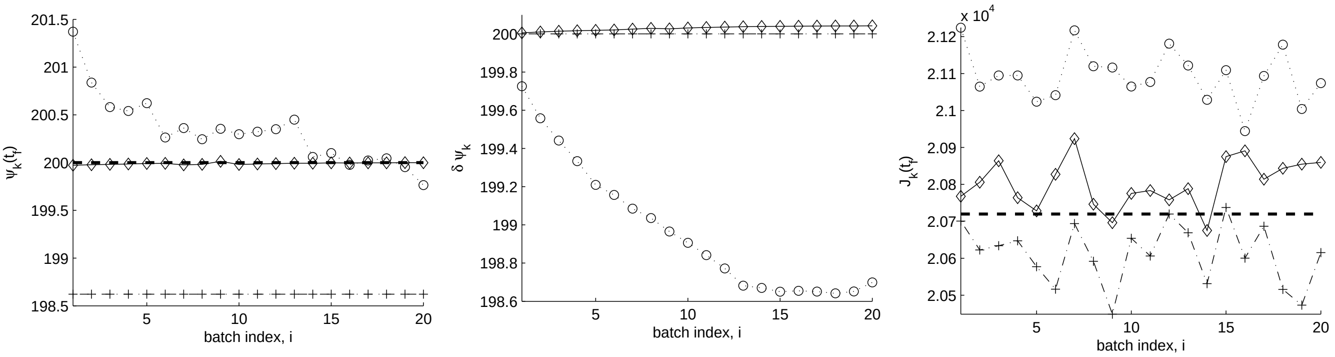


Fig. 4. **Dotted lines with circles**: constraint adaptation alone, **dash-dotted lines with crosses**: neighbouring extremal control alone, **dashed lines**: optimal solution for perturbed system, **solid lines with diamonds**: integrated two-time-scale scheme control. **Left plot**: Evolution of the terminal constraint; **Middle plot**: Evolution of the terminal constraint bias; **Right plot**: Evolution of the performance index.

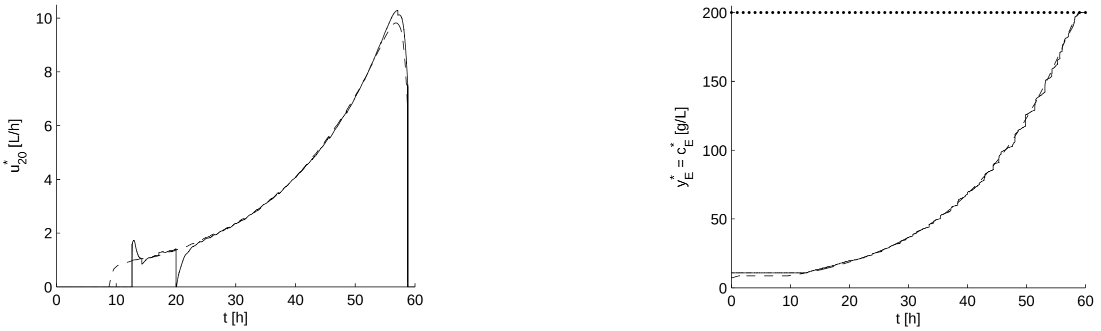


Fig. 5. Performance with perturbed initial conditions after 20 run of adaptation. **Left plot**: Converged solution of control trajectory; **Right plot**: Converged solution of measured output of liquid volume; **Solid line**: Perturbed system with two-time-scale integrated scheme; **Dashed line**: Optimal solution for perturbed system;

as well as the ability to handle problems with state path constraints.

ACKNOWLEDGEMENTS

Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Allgöwer, F. and Zheng, A. (2000). *Nonlinear Model Predictive Control*. Birkhäuser Verlag.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Bonvin, D., Srinivasan, B., and Hunkeler, D. (2006). Control and optimization of batch processes: Improvement of process operation in the production of specialty chemicals. *IEEE Control Systems Magazine*, 26(6), 34–45.

Bryson, A.E. and Ho, Y.C. (1975). *Applied Optimal Control – Optimization, Estimation and Control*. Hemisphere publishing corporation.

Chen, C.T. and Hwang, C. (1990). Optimal control computation for differential-algebraic process systems with general constraints. *Chemical Engineering Communications*, 1(97), 9–26.

Edgar, T.F. and Himmelblau, D.M. (1988). *Optimization of Chemical Processes*. McGraw-Hill, New York.

Garcia, C.E., Prett, D.M., and Morari, M. (1989). Model Predictive Control: Theory and Practice – A Survey. *Automatica*, 25(3), 335–348.

Guntern, C., Keller, A., and Hungerbuhler, K. (1998). Economic Optimization of an Industrial Semi-batch Reactor Applying Dynamic Programming. *Industrial and Engineering Chemistry Research*, 37(10), 4017–4022.

Kadam, J.V. and Marquardt, W. (2007). Integration of economical optimization and control for intentionally transient process operation. *Lecture Notes in Control and Information Sciences*, 358, 419–434.

Marchetti, A., Chachuat, B., and Bonvin, D. (2007). Batch process optimization via run-to-run constraints adaptation. In *European Control Conference*. Kos, Greece.

Pesch, H.J. (1989). Real-time computation of feedback controls for constrained optimal control problems. Part II: A correction method based on multiple shooting. *Optimal Control Applications & Methods*, 10, 147–171.

Comments – Remarks

Robust control of a CSTR

J. Závacká, M. Bakošová, K. Matejičková

Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation and Mathematics,
Radlinského 9, 812 37 Bratislava, Slovakia
fax : +421 2 52496469, e-mail: jana.zavacka@stuba.sk

Abstract: The paper presents a method for design of robust PI controllers for systems with interval uncertainty. The proposed method is combination of the method based on plotting the stability boundary locus in the (k_p,k_i)-plane with the pole-placement method. The method is implemented for design of robust PI controller for the continuous stirred tank reactor (CSTR) with hydrolysis of propylene oxide to propylene glycol. The reactor has three uncertain parameters: the reaction enthalpy, the pre-exponential factor and the overall heat transfer coefficient. The control input is the volumetric flow rate of the coolant and the controlled output is the temperature of the reacting mixture. Mathematical model of the reactor has been obtained in the form of the 4^{th} order transfer function with interval polynomials in the numerator and the denominator.

1. INTRODUCTION

Chemical reactors are ones of the most important plants in chemical industry, see e.g. Mikleš and Fikar (2007). Their operation, however, is corrupted with various uncertainties. Some of them arise from varying or not exactly known parameters, as e.g. reaction rate constants, reaction enthalpies or heat transfer coefficients. In others cases, operating points of reactors vary or reactor dynamics is affected by various changes of parameters of inlet streams. All these problems (uncertainties) can cause poor performance or even instability of closed-loop control systems. Application of robust control approach can be one of ways for overcoming all these problems, see e.g. Alvarez-Ramirez and Femat (1999), Gerhard et al. (2004), Bakošová et al. (2009).

In this paper, a simple method for design of robust PI controllers is presented (Tan and Kaya (2003)). The method is based on plotting the stability boundary locus in the (k_p,k_i)-plane and then parameters of a stabilizing PI controller are determined from the stability region, see e.g. Závacká et al. (2009). The PI controller stabilizes a controlled system with interval parametric uncertainties, when the stability region is found for sufficient number of Kharitonov plants (Barmish (1994)).

The pole-placement method is used to specify controller parameters in the robust stability area such that they assure certain quality of the control performance. The closed loop characteristic equation is created for controlled system and PI controller, and the closed loop characteristic equation is also obtained by the choice of the poles of the closed loop or the relative damping of the control response or the natural frequency of the control response.

The approach is used for design of a robust PI controller for the continuous stirred tank reactor with hydrolysis of propylene oxide to propylene glycol that can be modeled in the form of a transfer function with parametric interval uncertainty. The reactor has three uncertain parameters: the reaction enthalpy, the pre-exponential factor in the reaction rate constant and the overall heat transfer coefficient. The control input is the volumetric flow rate of the coolant and the controlled output is the temperature of the reacting mixture. The mathematical model of the reactor has been derived in the form of the 4^{th} order transfer function with interval polynomials in the numerator and the denominator.

2. ROBUST PI CONTROLLER DESIGN

Consider a single-input single-output (SISO) control system shown in Fig. 1, where

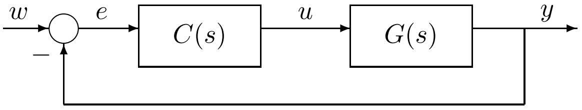


Fig. 1. Control system

$$G(s) = \frac{N(s)}{D(s)} \tag{1}$$

is the plant to be controlled and $C(s)$ is a PI controller in the form

$$C(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s} \tag{2}$$

The problem is to find the parameters of the PI controller (2) that stabilize the system in Figure 1, where w is the set point, e – the control error, u – the control input and y – the controlled output.

Decomposing the numerator and the denominator polynomials in (1) (Tan and Kaya (2003)) into their even and odd parts, and substituting, where $s = j\omega$ is the frequency, gives

$$G(j\omega) = \frac{N_e(-\omega^2) + j\omega N_o(-\omega^2)}{D_e(-\omega^2) + j\omega D_o(-\omega^2)} \tag{3}$$

The closed loop characteristic polynomial can be written as

$$\begin{aligned}\Delta(j\omega) &= [k_i N_e(-\omega^2) - k_p\omega^2 N_o(-\omega^2) - \\ &-\omega^2 D_o(-\omega^2)] + j[k_p\omega N_e(-\omega^2) + \\ &+k_i\omega N_o(-\omega^2) + \omega D_e(-\omega^2)] = 0\end{aligned} \tag{4}$$

Then, equating the real and imaginary parts of $\Delta(j\omega)$ to zero, one obtains

$$\begin{aligned}&k_p(-\omega^2 N_o(-\omega^2)) + k_i(N_e(-\omega^2)) = \\ &= \omega^2 D_o(-\omega^2)\end{aligned} \tag{5}$$

and

$$\begin{aligned}&k_p(N_e(-\omega^2)) + k_i(N_o(-\omega^2)) = \\ &= -D_e(-\omega^2)\end{aligned} \tag{6}$$

Let

$$\begin{aligned}F(\omega) &= -\omega^2 N_o(-\omega^2) \\ G(\omega) &= N_e(-\omega^2) \\ H(\omega) &= N_e(-\omega^2) \\ I(\omega) &= N_o(-\omega^2) \\ J(\omega) &= \omega^2 D_o(-\omega^2) \\ K(\omega) &= -D_e(-\omega^2)\end{aligned} \tag{7}$$

Then, (5) and (6) can be written as

$$\begin{aligned}k_p F(\omega) + k_i G(\omega) &= J(\omega) \\ k_p H(\omega) + k_i I(\omega) &= K(\omega)\end{aligned} \tag{8}$$

From (8), parameters of the PI controller (2) are

$$k_p = \frac{J(\omega)I(\omega) - K(\omega)G(\omega)}{F(\omega)I(\omega) - G(\omega)H(\omega)} \tag{9}$$

and

$$k_i = \frac{K(\omega)F(\omega) - J(\omega)H(\omega)}{F(\omega)I(\omega) - G(\omega)H(\omega)} \tag{10}$$

Solving these two equations simultaneously for $\omega \geq 0$, the set of parameters k_p and k_i is obtained. Then, it is possible to plot the dependence of k_i on k_p, and the stability boundary locus $l(k_p, k_i, \omega)$ in the (k_p, k_i)-plane is obtained. The stability boundary divides the parameter plane into stable and unstable regions. The stability region is found by the choice of testing points inside the regions.

The method is very fast and effective, however, frequency rating becomes important. An efficient approach to avoid frequency rating can be obtained by using the Nyquist plot. It is only necessary to find real values of ω that satisfy

$$\mathrm{Im}[G(s)] = 0 \tag{11}$$

Consider a feedback system (Figure 1) with the PI controller (2) and the interval plant

$$\begin{aligned}G(s,b,a) &= \frac{N(s,b)}{D(s,b)} = \\ &= \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_0}\end{aligned} \tag{12}$$

where $b_i \in [b_i^-, b_i^+]$, $i = 0, 1, 2, \ldots, m$ and $a_i \in [a_i^-, a_i^+]$, $j = 0, 1, 2, \ldots, n$. Let the Kharitonov polynomials associated with $N(s,b)$ and $D(s,a)$ are (Barmish (1994)):

$$\begin{aligned}N_1(s,b) &= b_0^- + b_1^- s + b_2^+ s^2 + b_3^+ s^3 + \ldots \\ N_2(s,b) &= b_0^+ + b_1^+ s + b_2^- s^2 + b_3^- s^3 + \ldots \\ N_3(s,b) &= b_0^+ + b_1^- s + b_2^- s^2 + b_3^+ s^3 + \ldots \\ N_4(s,b) &= b_0^- + b_1^+ s + b_2^+ s^2 + b_3^- s^3 + \ldots\end{aligned} \tag{13}$$

and

$$\begin{aligned}D_1(s,a) &= a_0^- + a_1^- s + a_2^+ s^2 + a_3^+ s^3 + \ldots \\ D_2(s,a) &= a_0^+ + a_1^+ s + a_2^- s^2 + a_3^- s^3 + \ldots \\ D_3(s,a) &= a_0^+ + a_1^- s + a_2^- s^2 + a_3^+ s^3 + \ldots \\ D_4(s,a) &= a_0^- + a_1^+ s + a_2^+ s^2 + a_3^- s^3 + \ldots\end{aligned} \tag{14}$$

By taking all combinations of the $N_i(s,b)$ and $D_j(s,a)$ for $i, j = 1, 2, 3, 4$, the following family of sixteen Kharitonov plants can be obtained

$$G_K(s) = G_{ij}(s) = \frac{N_i(s)}{D_j(s)} \tag{15}$$

where $i, j = 1, 2, 3, 4$; $K = 1, \ldots, 16$. Define the set $S_{ij}(C(s)G_{ij}(s))$, which contains all values of the parameters of the controller $C(s)$ which stabilizes $G_{ij}(s)$. Then the set of all the stabilizing values of parameters of a PI controller, which stabilizes the interval plant of (12) can be written as

$$\begin{aligned}&S(C(s)G_K(s)) = S(C(s)G_{11}(s)) \cap \\ &\cap S(C(s)G_{12}(s)) \cap \ldots \cap S(C(s)G_{44}(s))\end{aligned} \tag{16}$$

3. POLE-PLACEMENT METHOD

The pole-placement control design Mikleš and Fikar (2007) belongs to the class of well-known analytical methods where transfer function of the controlled process is known. In this method Vaneková et al. (2010), only the closed-loop denominator that assures stability is specified. The advantage of this approach is its usability for a broad range of systems. If the controller is of PID structure then the characteristic equation can be one of the following

$$s + \omega_0 = 0 \tag{17}$$

$$s^2 + 2\xi\omega_0 s + \omega_0^2 = 0 \tag{18}$$

$$(s + \alpha)(s^2 + 2\xi\omega_0 s + \omega_0^2) = 0 \tag{19}$$

or the combination of (18) and (19), where ξ is the relative damping, ω_0 the natural undamped frequency, and $-\alpha$ is a closed-loop pole. Specifying suitable values of parameters ξ, ω_0, α in (17) - (19) leads to the controller which can assure the desired quality of the control responses. To obtain unique solution, the system of equations for calculation of controller parameter has to be the system with zero degree of freedom. If higher order characteristic polynomial is considered, then any of parameters ξ, ω_0 or α can be added to known variables.

4. DESCRIPTION OF CONTROLLED PROCESS

The continuous stirred tank reactor for hydrolysis of propylene oxide to propylene glycol, see e.g. Molnár et al.

(2002), Vaneková et al. (2010) was chosen as a controlled process. The reaction is described as follows

$$C_3H_6O + H_2O \longrightarrow C_3H_8O_2 \tag{20}$$

The reactor is fed with propylene oxide, methanol and water. Methanol is added to improve the solubility of propylene oxide in water. The excess of water provides higher selectivity to propylene glycol and eliminates consecutive reactions of propylene oxide as a key component. Dependence of the reaction rate constant on the reacting mixture temperature is described by the Arrhenius equation

$$k = k_\infty e^{-\frac{E}{RT_r}} \tag{21}$$

where k_∞ is the pre-exponential factor, E is the activation energy, R is the universal gas constant, and T_r is the temperature of the reacting mixture.

Assuming ideal mixing in the reactor, constant reacting volume, and the same volumetric flow rates of the inlet and outlet streams, the mass balance for any species in the system is

$$V_r \frac{dc_j}{dt} = q_r(c_{j0} - c_j) + V_r \nu_j r \tag{22}$$

where $j = 1, 2, 3$ and V_r is the reacting volume, c_j is the molar concentration of the j-th component, c_{j0} is the feed molar concentration of the j-th component, q_r is the volumetric flow rate of the reacting mixture, ν_j is the stoichiometric coefficient of the j-th component, $r = kc_{C_3H_6O}$ is the molar rate of the chemical reaction.

It is assumed further that the specific heat capacities, densities and volumetric flow rates do not depend on temperature or mixture composition, and also the heat of mixing and the mixing volume can be neglected. The simplified enthalpy balance of the reacting mixture used as a standard in reactor design Ingham et al. (2007) is

$$\begin{aligned} V_r \rho_r c_{pr} \frac{dT_r}{dt} &= q_r \rho_r c_{pr}(T_{r0} - T_r) - \\ &-UA(T_r - T_c) + V_r(-\Delta_r H) r \end{aligned} \tag{23}$$

and the simplified enthalpy balance of the cooling medium is

$$\begin{aligned} V_c \rho_c c_{pc} \frac{dT_c}{dt} &= q_c \rho_c c_{pc}(T_{c0} - T_r) + \\ &+UA(T_r - T_c) \end{aligned} \tag{24}$$

where T is the temperature, ρ is the density, c_p is the specific heat capacity, $\Delta_r H$ is the reaction enthalpy, U is the overall heat transfer coefficient, A is the heat exchange area. The subscripts denote: 0 the feed, c the cooling medium, and r the reaction mixture. The values of constant parameters and steady-state inputs of the reactor are summarized in Table 1. Model uncertainties of the reactor follow from the fact that there are three physical parameters in this reactor: the pre-exponential factor, the reaction enthalpy and the overall heat transfer coefficient, the values of which vary within certain intervals (Table 2). Nominal values of these parameters are the mean values of the intervals and they are used to derive the reactor nominal model.

Table 1. Constant parameters and steady-state inputs of the chemical reactor

Parameter	**Value**
V_r $[m^3]$	2.407
V_c $[m^3]$	2.000
ρ_r $[kg.m^{-3}]$	974.19
ρ_c $[kg.m^{-3}]$	998
c_{pr} $[kJ.kg^{-1}.K^{-1}]$	3.7187
c_{pc} $[kJ.kg^{-1}.K^{-1}]$	4.182
A $[kJ.min^{-1}.K^{-1}]$	8.695
Steady-state input	**Value**
(E/R) $[K]$	10183
$q_r[m^3.min^{-1}]$	0.072
$q_c[m^3.min^{-1}]$	0.6307
$T_{rf}[K]$	299.05
$T_{cf}[K]$	288.15
$c_{f,C_3H_6O}[kmol.m^{-3}]$	0.0824
$c_{f,C_3H_8O_2}[kmol.m^{-3}]$	0

Table 2. Uncertain parameters in the CSTR

Parameter	Minimal	Maximal
$\Delta_r H$ $[kJ.mol^{-1}]$	-5.52×10^6	-5.40×10^6
k_∞ $[min^{-1}]$	2.6467×10^{11}	3.0067×10^{11}
U $[kJ.min^{-1}.K^{-1}]$	13.2	14.4

Parameter	Nominal
$\Delta_r H$ $[kJ.mol^{-1}]$	-5.46×10^6
k_∞ $[min^{-1}]$	2.8267×10^{11}
U $[kJ.min^{-1}.K^{-1}]$	13.8

5. APPLICATION OF ROBUST PI CONTROLLER AND POLE-PLACEMENT METHOD FOR CONTROL OF THE CSTR

For controller design, the mathematical model of the continuous stirred tank reactor with three uncertain parameters (Table 2) is obtained in the linearized form of a transfer function

$$G(s, b, a) = \frac{b_2 s^2 + b_1 s + b_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{25}$$

where coefficients in the numerator polynomial and the denominator one lie in following intervals: $b_2 \in [-0.0291, -0.0245]$, $b_1 \in [-0.0199, -0.0127]$, $b_0 \in [-0.0005740, -0.0003549]$, $a_3 \in [0.5801, 0.9030]$, $a_2 \in [0.1002, 0.2299]$, $a_1 \in [0.0062, 0.0142]$ and $a_0 \in [0.0001094, 0.0002412]$. Then sixteen Kharitonov plants are created for the reactor and described approach is used for robust PI controller design.

Equations for calculation of PI controller parameters 9, 10 lead to

$$k_i = \frac{-a_3\omega^4 + a_1\omega^2 + b_1\omega^2 \left(\frac{-a_4\omega^4 + a_2\omega^2 - a_0}{b_0 - b_2\omega^2} \right)}{-b_2\omega^2 + \frac{b_1^2\omega^2}{b_0 - b_2\omega^2} + b_0} \tag{26}$$

and

$$k_p = \frac{-a_4\omega^4 + a_2\omega^2 - a_0 - b_1 k_i}{b_0 - b_2\omega^2} \tag{27}$$

After a suitable choice of $\omega \in (0, 0.2407)$ (11), the stability boundary locus as the dependence of k_i on k_p is plotted. In Fig. 2 are shown the stability regions of sixteen Kharitonov plants, where intersection of these regions represents the stability region. Fig. 2 shows stability region for sixteen

Kharitonov plants and Fig. 3 represents zooming of the intersection of these regions.

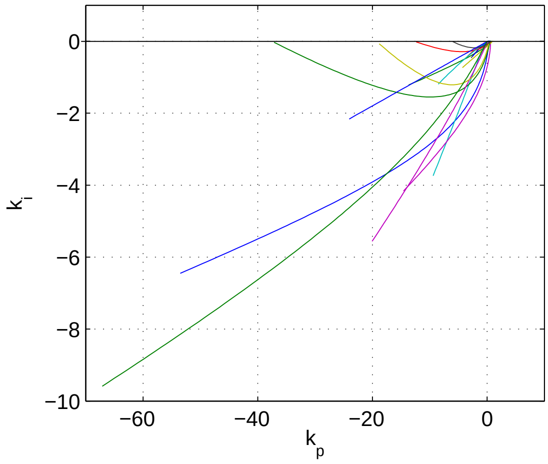


Fig. 2. Stability regions for sixteen Kharitonov plants

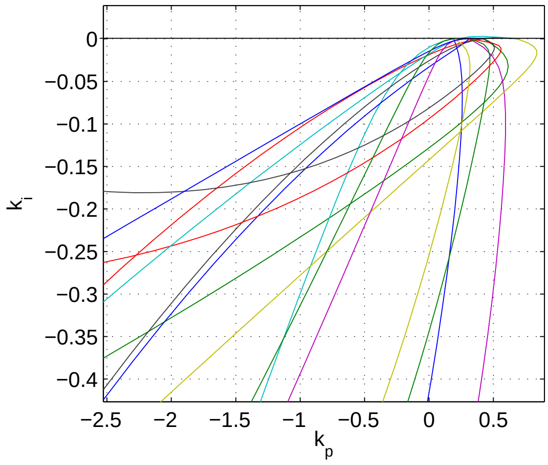


Fig. 3. Zooming of the intersection of stability regions

5.1 Pole-placement method for the system of the fourth order

Suppose that for controlled system (25) is necessary to find PI controller (2). The closed loop characteristic equation can be written as

$$\begin{aligned}(s+\alpha_1)\ (s+\alpha_2)(s+\alpha_3)\\ (s^2+2\xi\omega_0 s+\omega_0^2)=0\end{aligned} \tag{28}$$

The closed loop characteristic equation for the controlled system (25) and PI controller (2) has also the form

$$\begin{aligned}&s^3+\frac{a_3}{a_4}s^4+\frac{a_2+b_2k_p}{a_4}s^3+\\ &+\frac{a_1+b_2k_i+b_1k_p}{a_4}s^2+\\ &+\frac{a_0+b_1k_i+b_0k_p}{a_4}s+\frac{b_0k_i}{a_4}=0\end{aligned} \tag{29}$$

After comparison of coefficients in (28) and (29), parameters of the PI controller can be computed from (30) - (34)

$$0=2\xi\omega_0a_4+\alpha_3a_4+\alpha_2a_4+\alpha_1a_4-a_3 \tag{30}$$

Table 3. Controller parameters

	ξ	k_p	k_i
C1	1.3	−0.2980	−0.0105
C2	1.3	−1.1265	−0.0069
C3	0.85	−1.583	−0.0900
	$-\alpha_1$	$-\alpha_2$	$-\alpha_3$
C1	0.4929	0.3053	0.0294
C2	0.4929	0.3207	0.0294
C3	0.0297	0.4918	0.0619

$$\begin{aligned}b_2k_p=\omega_0^2a_4+2\xi\omega_0\alpha_3a_4+2\xi\omega_0\alpha_2a_4+\\ +\alpha_2\alpha_3a_4+2\xi\omega_0\alpha_1a_4+\\ +\alpha_1\alpha_3a_4+\alpha_1\alpha_2a_4-a_2\end{aligned} \tag{31}$$

$$\begin{aligned}b_2k_i+b_1k_p=\omega_0^2\alpha_3a_4+\omega_0^2\alpha_2a_4+\\ +2\xi\omega_0\alpha_2\alpha_3a_4+2\xi\omega_0\alpha_1\alpha_3a_4+\\ +\omega_0^2\alpha_1a_4+2\xi\omega_0\alpha_1\alpha_2a_4+\\ +\alpha_1\alpha_2\alpha_3a_4-a_1\end{aligned} \tag{32}$$

$$\begin{aligned}b_1k_i+b_0k_p=\alpha_2\alpha_3\omega_0^2a_4+\alpha_1\alpha_3\omega_0^2a_4+\\ +\alpha_1\alpha_2\omega_0^2a_4+2\xi\omega_0\alpha_1\alpha_2\alpha_3a_4-a_0\end{aligned} \tag{33}$$

$$b_0k_i=\alpha_1\alpha_2\alpha_3\omega_0^2a_4 \tag{34}$$

It is clear that unique solution k_p, k_i is obtained if only two of parameters ξ, ω_0 or α_1, α_2 and α_3 are chosen. Now is the pole-placement method used for reasonable choice of the controller from the robust stability region Fig. 3. The quality of the control response can be prescribed by the choice of the relative damping ξ and the natural undamped frequency ω_0 (28). The natural undamped frequency ω_0 is chosen in the same interval as the frequency ω. For plotting of curves system G_{max} with maximum values of interval coefficients (25) was used. The more important parameter from our point of view is the relative damping ξ and PI controllers were designed for $\xi = 1.3$ for the aperiodic and $\xi = 0.85$ for the periodic control response. After calculation of k_p and k_i from (30) - (34) for chosen ξ and ω, the curves representing the location of PI controllers are plotted and signed by stars (Fig. 4). Three PI controllers were chosen on the curves obtained using pole-placement method (Table 3).

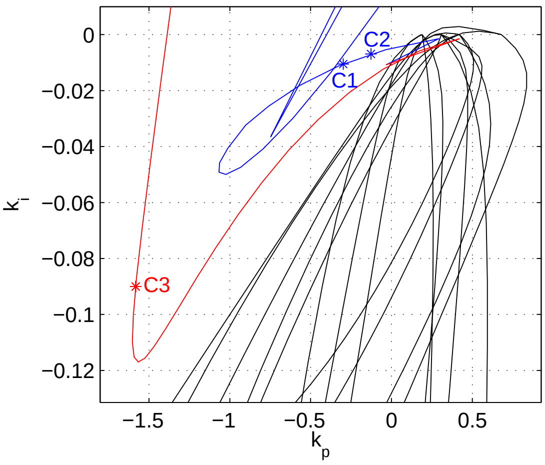


Fig. 4. Stability regions with curves ($\xi = 1.3$ blue, $\xi = 0.85$ red) from pole-placement method

6. SIMULATION RESULTS

The designed PI controllers (Table 3) for control CSTR (25) are verified by simulations.

Control responses of the closed loop with model (25) and different PI controllers (C1, C2, C3) are shown in Figs. 5, 6, 7 where w is the setpoint, y the controlled output and G_{min}, G_{max} are systems (25) with the minimum and maximum values of interval coefficients. Controllers C1, C2 ($\xi = 1.3$) were able to assure the aperiodic control responses . For controller C1 the setting time is smaller than for controller C2, because parameters of poles $\alpha_{1,2,3}$ for controller C1 are nearer to poles of the controlled system as poles $\alpha_{1,2,3}$ for controller C2. The controller C3 ($\xi = 0.85$) was able to assure the periodic control response. The setting time for G_{max} was even smaller then it was in the cases with controllers C1 and C2. But the control response for G_{min} is oscilating and it leads to the longer setting time in comparison with with using controllers C1 and C2.

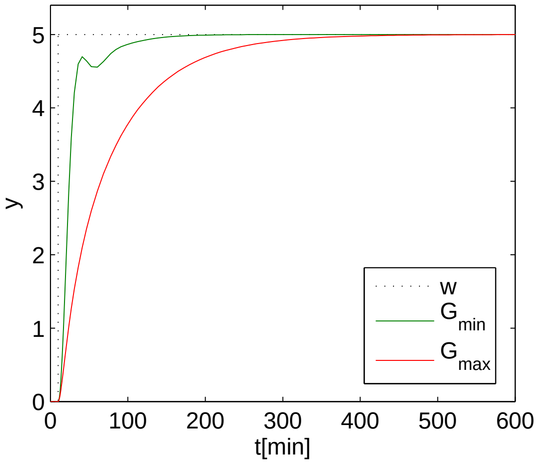


Fig. 5. Control responses for controller C1

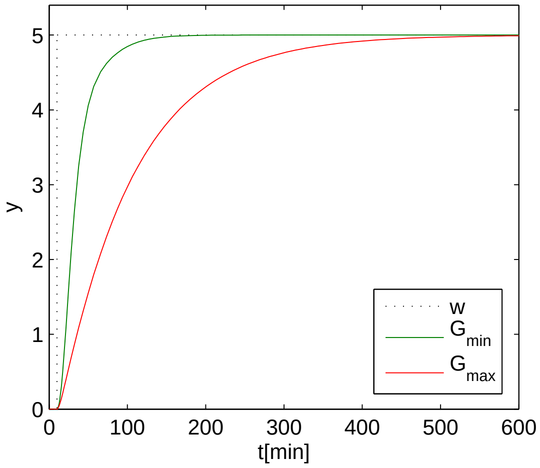


Fig. 6. Control responses for controller C2

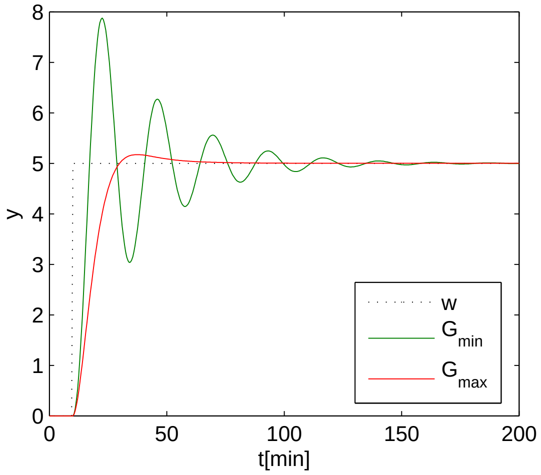


Fig. 7. Control responses for controller C3

7. CONCLUSION

In this paper, robust PI controllers were designed by combination of two methods. First is based on the plotting the stability boundary locus in the (k_p,k_i)-plane and second is pole-placement method for choice a specific controller. Further, PI controllers were designed for two values of ξ. These controllers were used for control CSTR with three uncertain parameters. Presented simulation experiments confirm that all designed robust PI controllers were able to control the reactor with three uncertainties.

ACKNOWLEDGMENTS

Supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.
The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/0537/10, 1/0095/11, and the Slovak Research and Development Agency under the project APVV-0029-07.

REFERENCES

Alvarez-Ramirez, J. and Femat, R. (1999). Robust PI stabilization of a class of chemical reactors. *Sytems Control Letter*, 38, 219–225.

Bakošová, M., Puna, D., Dostál, P., and Závacká, J. (2009). Robust stabilization of a chemical reactor. *Chemical Papers*, 65(5), 527–536.

Barmish, B. (1994). *New Tools for Robustness of Linear Systems.* Macmillan Publishing Company.

Gerhard, J., Monningmann, M., and Marquardt, W. (2004). Robust stable nonlinear control and design of a CSTR in a large operating range. In *Proc. 7th Int. Symp. Dynamics and Control of Process Systems*, Cambridge, Massachusset, USA, July 5 – 7. CD-ROM 92.

Ingham, J., Dunn, I., Heinzle, E., Přenosil, J., and Snape, J. (2007). *Chemical Engineering Dynamics: An Introduction to Modelling and Computer Simulation.* Weinheim: WILEY-VCH Verlag.

Mikleš, J. and Fikar, M. (2007). *Process modelling, Identification, and Control.* Springer Verlag, Berlin Heidelberg.

Molnár, A., Markoš, J., and Ľ. Jelemenský (2002). Accuracy of mathematical model with regard to safety analysis of chemical reactors. *Chemical Papers*, 56(6), 357–361.

Tan, N. and Kaya, I. (2003). Computation of stabilizing pi controllers for interval systems. In *Meditteranean Conference on Conntrol and Automation*. Rhodes, Greece.

Vaneková, K., Bakošová, M., and Závacká, J. (2010). Robust pi controllers for systems with transport delay. In K.A.R.D. Kozák Š. (ed.), *nternational Conference Cybernetics and Informatics*. Vydavateľstvo STU, Vazovova 5, 812 43 Bratislava, Slovakia.

Závacká, J., Bakošová, M., and Vaneková, K. (2009). Riadenie laboratórneho chemického reaktora robustným PI regulátorom. *Automatizace*, 52(6), 362–365.

Comments – Remarks

Author Index