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

Editors

Selected Topics on Constrained and Nonlinear Control

Textbook

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: 978–80–968627–4–0

Preface

This textbook 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 textbook 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

Acknowledgements

The authors and editors are pleased to acknowledge the financial support the grant No. NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This book is also co-financed from the state budget of the Slovak Republic.

Contents

List of Contributors

Ľuboš Čirka
Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, e-mail: lubos.cirka@stuba.sk

Miroslav Fikar
Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, e-mail: miroslav.fikar@stuba.sk

Martin Herceg
Swiss Federal Institute of Technology, Zurich, e-mail: herceg@control.ee.ethz.ch

Morten Hovd
Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: morten.hovd@itk.ntnu.no

Mikuláš Huba
Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, e-mail: mikulas.huba@stuba.sk

Johannes Jäschke
Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: jaschke@chemeng.ntnu.no

Tor A. Johansen
Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: tor.arne.johansen@itk.ntnu.no

Michal Kvasnica
Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, e-mail: michal.kvasnica@stuba.sk

Martin Lauko
Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering Slovak University of Technology, Bratislava, Slovakia., e-mail: martin.lauko@stuba.sk

Tomáš Polóni
Institute of Measurement, Automation and Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, e-mail: tomas.poloni@stuba.sk

Boris Rohal'-Ilkiv
Institute of Measurement, Automation and Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, e-mail: boris.rohal-ilkiv@stuba.sk

Selvanathan Sivalingam
Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: selvanathan.sivalingam@itk.ntnu.no

Sigurd Skogestad
Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: skoge@chemeng.ntnu.no

Gergely Takács
Institute of Measurement, Automation and Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, e-mail: gergely.takacs@stuba.sk

Chapter 1
Measurement Polynomials as Controlled Variables

Johannes Jäschke, Sigurd Skogestad

Abstract In this chapter we present a method for finding controlled variables, which are nonlinear combinations of measurements. The procedure extends the concept of the null-space method (Alstad and Skogestad, 2007) to processes described by polynomial equations. The method consists of three main steps. First, the active constraints are determined. If the disturbance causes the set of active constraints to change, regions of constant active constraints are defined in the disturbance space. Second, optimally invariant variable combinations are determined for the remaining unconstrained degrees of freedom in each region. Third, unknown internal variables (states) and disturbances are eliminated to obtain new invariant variable combinations containing only known variables (measurements). Furthermore we show that if the disturbance causes the active constraints to change, the invariants may be used to identify, and switch to the right region. This makes the method applicable over a wide disturbance range with changing active sets. The procedure is applied to a case study, a four component isothermal CSTR.

1.1 Introduction

For continuous processes, which are operated in steady state most of the time, an established method to achieve optimal operation in spite of varying disturbances is real-time optimization (RTO) (Marlin and Hrymak, 1997). The real-time optimizer generally uses a nonlinear steady state model in

Johannes Jäschke
Department of Chemical Engineering, NTNU Trondheim, Norway, e-mail: jaschke@chemeng.ntnu.no

Sigurd Skogestad
Department of Chemical Engineering, NTNU Trondheim, Norway, e-mail: skoge@chemeng.ntnu.no

order to recompute new optimal setpoints for the controlled variables in the control layer below. This concept has gained acceptance in industry and is increasingly used for improving plant performance. However, installing an RTO system and maintaining it generally entails large costs.

A second approach for optimizing plant performance is to use a process model off-line to find a self-optimizing control structure. The basic concept of self-optimizing control was conceived by Morari et al (1980), who write that we "want to find a function c of the process variables which when held constant leads automatically to the optimal adjustments of the manipulated variables", but they did not provide any method for identifying this function. The idea is to use this function as a controlled variable and keep it at a constant setpoint by simple control structures, e.g. PID controllers, or by more complex model predictive controllers (MPC). Using this kind of controlled variables disburdens the real-time optimizer, or may even make it unnecessary (Jäschke and Skogestad, 2010).

The term "self-optimizing control" was defined in the context of controlled variable selection with the purpose of describing the practical goal of finding "smart" controlled variables $\boldsymbol{c}$. Skogestad (2000) writes:

> Self-optimizing control is achieved if a constant setpoint policy results in an acceptable loss L (without the need to re-optimize when disturbances occur).

Many industrial processes are operated using self-optimizing control, although it is not always called that. For example, optimally active constraints may be viewed as self-optimizing variables, e.g. maximum cooling of an air stream before entering a compressor. However, the more difficult problem is to identify self-optimizing control variables associated with unconstrained degrees of freedom. In most cases, engineering insight and experience leads to the choice of self-optimizing controlled variables, and the optimization problem is not formulated explicitly. An example for the unconstrained case is is controlling the air/fuel ratio entering a combustion engine at a constant value.

It has been noted previously (Halvorsen and Skogestad, 1997; Bonvin et al, 2001; Cao, 2003; Halvorsen et al, 2003), that the gradient of the cost function with respect to the degrees of freedom $\boldsymbol{u}$ would be the ideal controlled variable, $\boldsymbol{c} = J_{\boldsymbol{u}}$. However, the gradient $J_{\boldsymbol{u}}$ is usually not directly measurable, and analytical expressions for the gradient generally contain unknown disturbances. Therefore, the methods in self-optimizing control theory can be thought of as an approximation (in some "best" way) of the gradient using a measurement model.

In the last decade, several contributions have been made on the systematic search of controlled variables which have self-optimizing properties (Halvorsen et al, 2003; Alstad and Skogestad, 2007; Kariwala et al, 2008; Alstad et al, 2009; Heldt, 2009), but to the authors knowledge, self-optimizing control has only been considered locally, that is, using linear process models and a quadratic approximation of the cost function. This results in linear

measurement combinations $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ as controlled variables. In cases where a strong curvature is present at the optimum, the loss imposed by using linear measurement combinations may not be acceptable any more, and the controlled variables are not self-optimizing.

The main contribution of this work is to extend the ideas of self-optimizing control, in particular the concept of the null-space method (Alstad and Skogestad, 2007) to constrained systems described by multivariable polynomials. This results in controlled variables which are polynomials in the measurements, $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{y})$.

Second, we show that under some assumptions, the controlled variables can be used to determine when the set of active constraints changes and which set to change to.

1.2 Overview

The procedure for achieving optimal operation is summarized in Fig. 1.1. In steps 1 and 2 we formulate the optimization problem and determine regions of constant active constraints, also called critical regions. This is done by offline calculations, for example, by gridding the disturbance space with a sufficiently fine grid and optimizing the process for each grid point.

In step 3, for each critical region, the optimality conditions are formulated, and the Lagrangian multipliers are eliminated. Then the unknown variables, i.e. the disturbances and the internal state variables are eliminated from the optimality conditions to obtain an invariant variable combination which contains only measured variables and known parameters.

Optimal operation is achieved in each critical region by controlling the active constraints and the invariant measurement combinations.

Finally, we monitor the active constraints and the invariants of the neighbouring regions to determine when to switch to a new region.

1.3 Achieving Optimal Operation using Measurement Invariants

1.3.1 Problem Formulation

Optimal operation is defined as minimizing a scalar cost index $J(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{d})$ subject to satisfying the model equations, $g = 0$, and operational constraints, $h \leq 0$:

$$\min_{\boldsymbol{u},\boldsymbol{x}} J(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{d}) \quad \text{s.t} \quad \begin{cases} g(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{d}) = 0 \\ h(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{d}) \leq 0 \end{cases} \tag{1.1}$$

1. Formulate optimization problem
2. For the expected set of disturbances, find all regions with different sets of active constraints $\mathcal{A}_i$
3. For each region of active constraints $\mathcal{A}_i$
   - i Formulate optimality conditions
   - ii Eliminate:
     - a Lagrangian multipliers λ from optimality conditions to obtain invariants $J_{\boldsymbol{z},red}$ (reduced gradient)
     - b Unknown variables to obtain measurement invariants $\boldsymbol{c}(\boldsymbol{y})$, such that

$$\boldsymbol{c}(\boldsymbol{y}) = 0 \Longleftrightarrow J_{\boldsymbol{z},red} = 0$$

   - iii In each region $\mathcal{A}_i$, control the
     - a Active constraints
     - b Invariants $\boldsymbol{c}(\boldsymbol{y})$
4. Use controlled variables and measured constraints for changing regions

Fig. 1.1 Procedure for finding nonlinear invariants as controlled variables

Here $\boldsymbol{u}$, $\boldsymbol{x}$, $\boldsymbol{d}$ denote the manipulated input variables, the internal state variables, and the unmeasured disturbance variables, respectively. We assume that, in addition, we have measurements $\boldsymbol{y}$ satisfying the relation, $m(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) = 0$, which provide information about internal states, inputs, and disturbances. To simplify notation, we combine state and input variables in a vector $\boldsymbol{z} = [\boldsymbol{u}\,, \boldsymbol{x}]^{\boldsymbol{T}}$.

This is the same problem which is solved online at given sample times when using RTO. In this work, however, we do not wish to solve the optimization problem online, instead, we analyse the problem using offline calculations, in order to find good controlled variables which yield optimal operation when controlled at their setpoints.

1.3.1.1 Optimality Conditions

Let $\boldsymbol{z}^*$ be a feasible point of the optimization problem (1.1), and assume that all gradient vectors $\nabla_{\boldsymbol{z}} g_i(\boldsymbol{z}^*, \boldsymbol{d})$ and $\nabla_{\boldsymbol{z}} h_i(\boldsymbol{z}^*, \boldsymbol{d})$ associated with $g_i(\boldsymbol{z}^*, \boldsymbol{d}) = 0$ and the active constraints, $h_i(\boldsymbol{z}^*, \boldsymbol{d}) = 0$, are linearly independent. Then $\boldsymbol{z}^*$ is locally optimal if there exist Lagrangian multiplier vectors λ and ν, such that the following conditions, known as the KKT conditions are satisfied (Nocedal and Wright, 2006):

$$\begin{aligned} \nabla_{\boldsymbol{z}} J(\boldsymbol{z}^*, \boldsymbol{d}) + [\nabla_{\boldsymbol{z}} g(\boldsymbol{z}^*, \boldsymbol{d})]^{\boldsymbol{T}} \lambda + [\nabla_{\boldsymbol{z}} h(\boldsymbol{z}^*, \boldsymbol{d})]^{\boldsymbol{T}} \nu &= 0 \\ g(\boldsymbol{z}^*, \boldsymbol{d}) &= 0 \\ h(\boldsymbol{z}^*, \boldsymbol{d}) &\leq 0 \\ [h(\boldsymbol{z}^*, \boldsymbol{d})] \nu^{\boldsymbol{T}} &= 0 \\ \lambda, \nu &\leq 0 \end{aligned} \tag{1.2}$$

The condition that the Jacobian of the active constraints has independent rows (has full rank) is called the linear independence constraint qualification (LICQ) and guarantees that the Lagrangian multipliers λ and ν are uniquely defined at the optimum $\boldsymbol{z}^*$.

When optimizing nonlinear systems, such as polynomial systems, there are several complications which may arise. The optimality conditions, (1.2), will in general not have a unique solution. There may be multiple maxima, minima and saddle points, so finding the global minimum is not an easy task in itself. When a solution to (1.2) is found, it has to be checked whether it indeed is the desired solution (minimum). In addition, there may be solutions which are not physical (complex). So before controlling $\boldsymbol{c}(\boldsymbol{y})$ to zero, it has to be assured that the process actually is at the desired optimum.

This and other issues from nonlinear and polynomial optimization are not addressed in this work, the focus of this paper is rather to present a method which gives a controlled variable $\boldsymbol{c}(\boldsymbol{y})$ which is zero for all points that satisfy the KKT conditions, and which is nonzero whenever the KKT conditions are not satisfied.

1.3.2 Partitioning into Sets of Active Constraints

Generally, the set of inequality constraints $h_i(\boldsymbol{z}, \boldsymbol{d}) \leq 0$ that are active varies with the value of the elements in $\boldsymbol{d}$. The disturbance space can hence be partitioned into regions which are characterized by their individual set of active constraints. These regions will be called critical regions.

The concept of critical regions allows one to decompose the original optimization problem (1.1) into a sequence of equality constrained optimization problems, which are valid in the corresponding critical region. This idea is also applied in multi-parametric programming (Pistikopoulos et al, 2007). However, we do not search for an explicit expression for the inputs $\boldsymbol{u}^*$, as in multi-parametric programming. We rather use each subproblem to find good controlled variables $\boldsymbol{c}$ for the corresponding critical region.

In order to obtain a fully specified system in each region,

1. the active constraints need to be controlled, and
2. a controlled variable has to be controlled for each unconstrained degree of freedom, $n_c = n_{DOF}$.

The number unconstrained degrees of freedom, n_{DOF} is calculated according to

$$n_{DOF} = n_z - n_g - n_{h,active} \tag{1.3}$$

where $n_z, n_g, n_{h,active}$ denote the number of variables $\boldsymbol{z}$, the number of model equations, g, and the number of constraints from h which are active ($h_i = 0$).

In the rest of the paper, by abuse of notation, all active constraints $h_i(\boldsymbol{z}, \boldsymbol{d}) = 0$ are included in the equality constraint vector $g(\boldsymbol{z}, \boldsymbol{d}) = 0$. Then in every critical region, the optimization problem (1.1) can be written as:

$$\begin{aligned} \min\, & J(\boldsymbol{z}, \boldsymbol{d}) \\ \text{s.t.} \quad & \\ & g(\boldsymbol{z}, \boldsymbol{d}) = 0 \end{aligned} \tag{1.4}$$

The KKT first order optimality conditions, (1.2), simplify for problem (1.4) in each critical region, to

$$\begin{aligned} \nabla_{\boldsymbol{z}} J(\boldsymbol{z}, \boldsymbol{d}) + [\nabla_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{d})]^{\boldsymbol{T}} \lambda = 0, \\ g(\boldsymbol{z}, \boldsymbol{d}) = 0. \end{aligned} \tag{1.5}$$

These expressions cannot be used for control yet, because they still contain unknown variables, in particular $\boldsymbol{x}$ (in $\boldsymbol{z} = [\boldsymbol{u}, \boldsymbol{x}]$), $\boldsymbol{d}$, and the Lagrangian multipliers λ, which have to be eliminated.

1.3.3 Eliminating the Lagrangian Multipliers $\boldsymbol{\lambda}$

We consider one equality constrained sub-problem (1.4) at a time. Every control structure that gives optimal operation has to satisfy (1.5). Recall that the LICQ holds, i.e. $\nabla_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{d})$ has full row rank for every value of $\boldsymbol{d}$ within the critical region. In addition, we assume that we have strict complementarity (either the constraint is active, or the corresponding value in λ is zero).

Proposition 1.1. *Let $\boldsymbol{N}(\boldsymbol{z}, \boldsymbol{d}) \in \mathbb{R}^{(n_z - n_g) \times n_g}$ be a basis for the null space of $\nabla_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{d})$. Controlling the active constraints $g(\boldsymbol{z}, \boldsymbol{d}) = 0$, and the variable combination $[\boldsymbol{N}(\boldsymbol{z}, \boldsymbol{d})]^{\boldsymbol{T}} \nabla_{\boldsymbol{z}} J(\boldsymbol{z}, \boldsymbol{d}) = 0$ results in optimal steady state operation.*

Proof. Select $\boldsymbol{N}(\boldsymbol{z}, \boldsymbol{d})$ such that $\boldsymbol{N}(\boldsymbol{z}, \boldsymbol{d}) \nabla_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{d}) = 0$. Since the LICQ are satisfied, the constraint Jacobian $\nabla_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{d})$ has full row rank and $\boldsymbol{N}(\boldsymbol{z}, \boldsymbol{d})$ is well defined and does not change dimension within the region. The first equation in (1.5) is premultiplied by $[\boldsymbol{N}(\boldsymbol{z}, \boldsymbol{d})]^{\boldsymbol{T}}$ to get:

$$[\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})]^{\boldsymbol{T}}\left(\nabla_{\boldsymbol{z}}J(\boldsymbol{z},\boldsymbol{d})+[\nabla_{\boldsymbol{z}}g(\boldsymbol{z},\boldsymbol{d})]^{\boldsymbol{T}}\lambda\right)=[\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})]^{\boldsymbol{T}}\nabla_{\boldsymbol{z}}J(\boldsymbol{z}.\boldsymbol{d})+\underline{0}\lambda$$
$$=[\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})]^{\boldsymbol{T}}\nabla_{\boldsymbol{z}}J(\boldsymbol{z},\boldsymbol{d}) \tag{1.6}$$

Since $\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})$ and has full rank, we have that (1.5) are satisfied whenever $g(\boldsymbol{z},\boldsymbol{d})=0$ and $\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})\nabla_{\boldsymbol{z}}J(\boldsymbol{z},\boldsymbol{d})=0$.

We introduce $J_{\boldsymbol{z},red}=\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})\nabla_{\boldsymbol{z}}J(\boldsymbol{z},\boldsymbol{d})$, and call $J_{\boldsymbol{z},red}$ the reduced gradient. By construction, the reduced gradient has $n_{DOF}=n_z-n_g$ elements. Controlling

$$J_{\boldsymbol{z},red}=[\boldsymbol{N}(\boldsymbol{z},\boldsymbol{d})]^{\boldsymbol{T}}\nabla_{\boldsymbol{z}}J(\boldsymbol{z},\boldsymbol{d})=0 \tag{1.7}$$

together with the active constraints, $g(\boldsymbol{z},\boldsymbol{d})=0$, fully specifies the system at the optimum and is equivalent to controlling the first order optimality conditions (1.5). However, $J_{\boldsymbol{z},red}$ cannot be used for control directly because it still depends on unknown variables, $\boldsymbol{d}$ and $\boldsymbol{x}$ ($\boldsymbol{x}$ enters through $\boldsymbol{z}=[\boldsymbol{u}\,,\boldsymbol{x}]^{\boldsymbol{T}}$). For this purpose, the disturbance and the internal states have to be eliminated from the expression (1.7).

A first naive approach would be to solve the measurement model equations $m(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d},\boldsymbol{y})=0$ and the active constraints $g(\boldsymbol{z},\boldsymbol{d})=0$ for the unknowns $\boldsymbol{d}$ and $\boldsymbol{x}$, and substitute the solution into $J_{\boldsymbol{z},red}$. As we show, this is straightforward in case of linear equations, but it becomes significantly more complicated when working with polynomials of higher degree.

1.4 Elimination for Systems of Linear Equations (Zero Loss Method)

In this section we describe the basic concept of how the unknowns are eliminated form $J_{\boldsymbol{z},red}$. Our procedure is demonstrated step by step for minimizing a quadratic cost function subject to linear constraints and a linear measurement model. Solving the model and measurement equations for the unknowns and substituting into $J_{\boldsymbol{z},red}$ is avoided, because this is difficult to extend to the polynomial case. Instead, we search for necessary and sufficient conditions which guarantee that the measurement model $m(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d},\boldsymbol{y})=0$, the active constraints and the model $g(\boldsymbol{z},\boldsymbol{d})=0$, and the reduced gradient $J_{\boldsymbol{z},red}=0$ are are satisfied at the same time. We require that the necessary and sufficient condition is a function of measurements $\boldsymbol{y}$ and known parameters, only.

The optimization problem we consider is

$$\begin{gathered}J(\boldsymbol{z},\boldsymbol{d})=\min\left[\,\boldsymbol{z}^{\boldsymbol{T}}\;\boldsymbol{d}^{\boldsymbol{T}}\,\right]\begin{bmatrix}\boldsymbol{J_{zz}} & \boldsymbol{J_{zd}}\\ \boldsymbol{J_{zd}}^{\boldsymbol{T}} & \boldsymbol{J_{dd}}\end{bmatrix}\begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{d}\end{bmatrix}\\ \text{s.t.}\\ \boldsymbol{A}\boldsymbol{z}-\boldsymbol{b}=0,\end{gathered} \tag{1.8}$$

and the linear measurement model is

$$\begin{aligned} m(\boldsymbol{z},\boldsymbol{d},\boldsymbol{y}) &= \boldsymbol{y} - [\boldsymbol{G}^y\ \boldsymbol{G}_d^y] \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{d} \end{bmatrix} = 0 \\ &= \boldsymbol{y} - \tilde{\boldsymbol{G}}^y \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{d} \end{bmatrix} = 0. \end{aligned} \tag{1.9}$$

We consider $[\boldsymbol{z}\,,\boldsymbol{d}]^{\boldsymbol{T}}$ as unknown and we assume that (1.8) has a solution, $\boldsymbol{J_{zz}}$ is positive definite, and $\boldsymbol{A}$ has full row rank. In addition, we assume that the measurements are linearly independent, and $\tilde{\boldsymbol{G}}^y = [\boldsymbol{G}^y\ \ \boldsymbol{G}_d^y]$ invertible.

The null space of the constraint gradient, $\boldsymbol{N}$, is a constant matrix which is independent of $\boldsymbol{z}$, such that $\boldsymbol{AN} = 0$. The first order necessary optimality conditions require that at the optimum

$$J_{\boldsymbol{z},red} = \boldsymbol{N^T}\nabla_{\boldsymbol{z}} J(\boldsymbol{z},\boldsymbol{d}) = \boldsymbol{N^T} \begin{bmatrix} \boldsymbol{J_{zz}}\ \boldsymbol{J_{zd}} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{d} \end{bmatrix} = 0. \tag{1.10}$$

If the number of independent measurements (n_y) is equal to the number of unknown variables variables ($n_z + n_d$), the measurement relations (1.9) can be solved for the unknowns and substituted into the gradient expression (1.10) to obtain

$$\boldsymbol{c}(\boldsymbol{y}) = \boldsymbol{N^T} \begin{bmatrix} \boldsymbol{J_{zz}}\ \boldsymbol{J_{zd}} \end{bmatrix} \left[\tilde{\boldsymbol{G}}^y\right]^{-1} \boldsymbol{y}. \tag{1.11}$$

Controlling $\boldsymbol{c}(\boldsymbol{y}) =$ and $\boldsymbol{Az} - \boldsymbol{b}$ to zero, then results in optimal operation.

In the case of polynomial equations of higher degrees, it is generally not possible to solve for the unknown variables. Therefore, we consider the problem from a slightly different perspective. Suppose $n_y = n_z + n_d$, then for any output and disturbance pair $(\boldsymbol{y},\boldsymbol{d})$, there exist a unique $\boldsymbol{z}$, which satisfies the measurement equations (1.9). However, an arbitrary pair ($\boldsymbol{y}$,$\boldsymbol{d}$) will fail to satisfy the first order optimality condition (1.10). More accurately, since $\boldsymbol{J_{zz}} > 0$, only one pair $(\boldsymbol{y},\boldsymbol{d})$ satisfies the first order optimality conditions.

Consider the elements of the reduced gradient vector (1.10), one at a time, together with all the measurement equations (1.9). Let the superscript (i) denote the i-th row of a matrix or a vector. We write the reduced gradient together with the measurements as a sequence of square linear systems

$$\underbrace{\begin{bmatrix} [\boldsymbol{NJ_{zz}}]^{(i)} & [\boldsymbol{NJ_{zd}}]^{(i)} & 0 \\ \boldsymbol{G}^y & \boldsymbol{G}_d^y & y \end{bmatrix}}_{\boldsymbol{M}^{(i)}} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{d} \\ -1 \end{bmatrix} = 0 \tag{1.12}$$

Here $\boldsymbol{M}^{(i)}$, $i = 1..n_{DOF}$ are square matrices of size $(n_y + 1) \times (n_y + 1)$.

We want to find a solution $[\boldsymbol{z},\boldsymbol{d}]$ which satisfies (1.12). For this system to have a solution for $[\boldsymbol{z},\boldsymbol{d}]^{\boldsymbol{T}}$, we must have $\text{rank}(\boldsymbol{M^{(i)}}) = n_y = n_z + n_d$.

The submatrix $[\boldsymbol{G}^y\ \boldsymbol{G}_d^y\ \boldsymbol{y}]$ already has rank n_y, irrespective of the value of $\boldsymbol{y}$ (or the control policy that generates the input $\boldsymbol{u}$ which in turn generates

$\boldsymbol{y}$). This follows because $[\boldsymbol{G}^y\ \boldsymbol{G}_d^y\ \boldsymbol{y}]$ has more columns than rows, and because $\text{rank}([\boldsymbol{G}^y\ \boldsymbol{G}_d^y]) = n_y$. Therefore, the condition for a common solution is:

$$\det(\boldsymbol{M}^{(i)}) = 0 \quad \text{for all } i = 1..n_{DOF}. \tag{1.13}$$

This condition guarantees that a common solution to (1.12) exists, so the elements of the controlled variable $\boldsymbol{c}$ are selected as $c_i = \det(\boldsymbol{M}^{(i)})$.

It remains to show that controlling the determinants $c_i = \det(\boldsymbol{M}^{(i)})$ gives the inputs which lead to the optimum. Since the system is linear and the rank of the measurement equations is n_y, there is a unique linear invertible mapping between the measurements $\boldsymbol{y}$ and the vector $[\boldsymbol{z}\,\boldsymbol{d}]^{\boldsymbol{T}}$. Therefore every value of $\boldsymbol{y}$ corresponds uniquely to some value in $\boldsymbol{z}$.

In the case with more measurements, $n_y > n_z + n_d$, any subset of $n_z + n_d$ measurements may be chosen such that $\text{rank}([\boldsymbol{G}^y\ \boldsymbol{G}_d^y]) = n_z + n_d$.

Remark 1.1. Actually, in the case of (1.8), we can use the constraint equations to eliminate the unmeasured internal states $\boldsymbol{x}$. Then we only $n_y = n_u + n_d$ measurements, and the matrices (1.12) become:

$$\boldsymbol{M}^{(i)} = \begin{bmatrix} (\boldsymbol{N}\boldsymbol{J}_{\boldsymbol{zz}})^{(i)} & (\boldsymbol{N}^{\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{zd}})^{(i)} & 0 \\ \boldsymbol{A} & 0 & \boldsymbol{b} \\ \boldsymbol{G}^y & \boldsymbol{G}_d^y & \boldsymbol{y} \end{bmatrix}, \tag{1.14}$$

and we must require, that

$$\text{rank}\left(\begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{G}^y & \boldsymbol{G}_d^y \end{bmatrix}\right) = n_z + n_d \tag{1.15}$$

Remark 1.2. When there are no constraints, we have that $\boldsymbol{z} = \boldsymbol{u}$, and this method results in the null space method (Alstad and Skogestad, 2007). In this case, $\boldsymbol{N}$ may be set to any nonsingular matrix, for example the identity matrix $\boldsymbol{N} = \text{i}$. Then we have that

$$\boldsymbol{c}_{\text{Nullspace}} = [\boldsymbol{J}_{\boldsymbol{uu}}\ \ \boldsymbol{J}_{\boldsymbol{ud}}][\tilde{\boldsymbol{G}}^y]^{-1}\boldsymbol{y}, \tag{1.16}$$

as has been derived in Alstad et al (2009).

The null space method was originally derived by Alstad and Skogestad (2007) using the optimal sensitivity matrix $\boldsymbol{F} = \frac{\partial \boldsymbol{y}^{opt}}{\partial \boldsymbol{d}}$. However, it escaped the authors notice then, that controlling $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ with $\boldsymbol{H}$ selected such that $\boldsymbol{H}\boldsymbol{F} = 0$, is indeed the same as controlling the gradient to zero.

Example 1.1. Consider a system from Alstad (2005). The cost to minimize is

$$J = (u - d)^2, \tag{1.17}$$

and the measurement relations (model equations) are

$$\begin{aligned} y_1 &= G_1^y u + G_{d,1} d \\ y_2 &= G_2^y u + G_{d,2}^y d \end{aligned} \tag{1.18}$$

where the variables u, d, y denote the input, the disturbance and the measurements, respectively. The values of the gains are given in table 1.1. We are

Variable	Value
G_1^y	0.9
$G_{d,1}^y$	0.1
G_2^y	0.5
$G_{d,2}^y$	-1.0

Table 1.1 Gain values for small example

searching for a condition on y_1 and y_2 such that the optimality condition is satisfied. The gradient is $\nabla_u J = 2(u-d)$ and $J_{uu} = 2$, $J_{ud} = -2$. It is easily verified that measurements are independent. This gives an equation system of 3 equations in 2 unknowns:

$$\boldsymbol{M} \begin{bmatrix} u \\ d \\ -1 \end{bmatrix} = 0 \tag{1.19}$$

where

$$\boldsymbol{M} = \begin{bmatrix} J_{uu} & J_{ud} & 0 \\ G_1^y & G_{d,1}^y & y_1 \\ G_2^y & G_{d,2}^y & y_2 \end{bmatrix} \tag{1.20}$$

Equation (1.19) has a solution $\begin{bmatrix} u \\ d \\ -1 \end{bmatrix}$ if and only if

$$\det(\boldsymbol{M}) = 0 \tag{1.21}$$

Therefore the necessary and sufficient condition for the existence of a non-trivial solution is

$$\begin{aligned} \det\left(\begin{bmatrix} J_{uu} & J_{ud} & 0 \\ G_1^y & G_{d,1}^y & y_1 \\ G_2^y & G_{d,2}^y & y_2 \end{bmatrix}\right) &= -y_1(J_{uu}G_{d,2}^y - G_2^y J_{ud}) + y_2(J_{uu}G_{d,1}^y - G_1^y J_{ud}) \\ &= 0 \end{aligned} \tag{1.22}$$

On inserting the parameter values from table 1.1, we obtain

$$c = \det(\boldsymbol{M}) = y_1 + 2y_2. \tag{1.23}$$

Controlling $c = y_1 + 2y_2$ to zero therefore yields optimal operation. This the same variable combination as found by applying the null-space method in Alstad (2005).

Even though obtaining the invariants via the determinant may seem cumbersome, it eliminates the necessity of inverting the measurements and solving for the unknowns. While this is of little advantage for systems of linear equations, the concept can be extended to systems of polynomial equations which cannot easily be solved for the right set of unknowns.

1.5 Elimination for Systems of Polynomial Equations

Let $\hat{\boldsymbol{d}}$ now denote the vector of all unmeasured variables, $\hat{\boldsymbol{d}} = [\boldsymbol{x}, \boldsymbol{d}]$, not only including disturbances, but also unknown states and unknown inputs, and let $\boldsymbol{y}$ include all measurements and known inputs. Thus, every variable belongs either to $\hat{\boldsymbol{d}}$ or $\boldsymbol{y}$, and we write the optimality conditions as

$$\begin{aligned} J_{\boldsymbol{z},red}(\boldsymbol{y}, \hat{\boldsymbol{d}}) &= 0 \\ g(\boldsymbol{y}, \hat{\boldsymbol{d}}) &= 0, \end{aligned} \tag{1.24}$$

and the measurement relations as

$$m(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0. \tag{1.25}$$

To be able to use the reduced gradient $J_{\boldsymbol{z},red}$ for control, all unknown variables $\hat{\boldsymbol{d}}$ have to be eliminated from it. For polynomial equations, this is not as straightforward as in the linear case. Even for the case of a univariable polynomial of degree 5 and higher, for example $d^5 - d + 1 = 0$, there exist no general analytic solution formulas, as was proven by Abel (1826). Therefore we are interested in finding a way to eliminate the unknown variables $\hat{\boldsymbol{d}}$ from $J_{\boldsymbol{z},red}(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$ without solving g and m for them first. This is exactly what was done in the previous section, where we used the determinant of a carefully constructed coefficient matrix, which characterizes the existence of a common solution in $\boldsymbol{d}$, to replace $J_{\boldsymbol{z},red}$. The determinant is a function of the known variables only, that is, the measurements $\boldsymbol{y}$ and the parameters $\tilde{\boldsymbol{G}}^y$, $\boldsymbol{J}_{\boldsymbol{zz}}$ and $\boldsymbol{J}_{\boldsymbol{zd}}$.

The generalization of the determinant to systems of polynomial equations is called resultant. According to Emiris and Mourrain (1999), "the resultant of an overconstrained polynomial system characterizes the existence of common roots as a condition on the input coefficients".

To be more specific, we consider multivariate polynomials $f \in \mathbb{R}[\boldsymbol{y}, \hat{\boldsymbol{d}}]$, that is real polynomial functions with coefficients in $\mathbb{R}$, and variables $\boldsymbol{y} = [y_1, y_2, \ldots, y_{n_y}]$ and $\hat{\boldsymbol{d}} = [\boldsymbol{x}, \boldsymbol{d}] = [\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{n_{\hat{d}}}]$. Given a $n_{\hat{d}}$-tuple, $\alpha_{i,j} = \big(\alpha_{i,j}(1), \alpha_{i,j}(2), \ldots, \alpha_{i,j}(n_{\hat{d}})\big)$, we use the shorthand notation

$$\hat{\boldsymbol{d}}^{\alpha_{i,j}} = \hat{d}_1^{\alpha_{i,j}(1)} \hat{d}_2^{\alpha_{i,j}(2)} \dots \hat{d}_{n_{\hat{d}}}^{\alpha_{i,j}(n_{\hat{d}})}.$$

Then we can write a system of n polynomials in compact form

$$f_i(\boldsymbol{y}, \hat{\boldsymbol{d}}) = \sum_{j=0}^{k_i} a_{i,j}(\boldsymbol{y}) \hat{\boldsymbol{d}}^{\alpha_{i,j}}, \quad i = 1..n \tag{1.26}$$

where the coefficients $a_{ij}(\boldsymbol{y}) \neq 0$ are polynomials in $\mathbb{R}[\boldsymbol{y}]$, that is polynomials in $\boldsymbol{y}$ with coefficients in $\mathbb{R}$.

We consider the functions $a_{i,j}(\boldsymbol{y})$ as polynomial coefficients, and $\hat{\boldsymbol{d}}$ as variables. For every polynomial f_i, we collect the exponent vectors in the set $\mathcal{E}_i = \{\alpha_{i,1}, \dots, \alpha_{i,k_i}\}$. This set is called support of the polynomial f_i.

The support of the polynomial $f = d_1^2 + d_1 d_2 - 1$, for example, is $\mathcal{E} = \{(2,0), (1,1), (0,0)\}$. We denote as Q_i the convex hull of the support of a polynomial, $Q_i = \text{conv}(\mathcal{E}_i)$.

Further, we denote the set of complex numbers without zero as $\mathbb{C}^*$ (that is $\mathbb{C}^* = \mathbb{C} \setminus 0$).

Next present some basic concepts from algebraic geometry taken from Cox et al (2005).

Definition 1.1 (Affine variety). Consider $f_1, \dots, f_n$ polynomials in $\mathbb{C}[\hat{d}_1, \dots \hat{d}_{n_{\hat{d}}}]$. The affine variety defined by $f_1, \dots, f_n$ is the set

$$V(f_1, \cdots, f_n) = \left\{ (\hat{d}_1, \dots, \hat{d}_{n_{\hat{d}}}) \in \mathbb{C}^{n_{\hat{d}}} : f_i(\hat{d}_1, \dots \hat{d}_{n_{\hat{d}}}) = 0 \quad i = 1 \dots n \right\} \tag{1.27}$$

Casually speaking, the variety is the set of all solutions in $\mathbb{C}^{n_{\hat{d}}}$.

Definition 1.2 (Zariski closure). Given a subset $S \subset \mathbb{C}^m$, there is a smallest affine variety $\bar{S} \subset \mathbb{C}^m$ containing S. We call $\bar{S}$ the Zariski closure of S.

Let $L(\mathcal{E}_i)$ be the set of all polynomials whose terms all have exponents in the support $\mathcal{E}_i$:

$$L(\mathcal{E}_i) = \left\{ a_{i,1} \hat{\boldsymbol{d}}^{\alpha_{i1}} + \dots + a_{i,k_i} \hat{\boldsymbol{d}}^{\alpha_{ik_i}} : a_{i,j} \in \mathbb{C} \right\} \tag{1.28}$$

Then the coefficients $a_{i,j}$ of n polynomials define a point in $\mathbb{C}^{n \times k_i}$.

Now let

$$Z(\mathcal{E}_1, \dots \mathcal{E}_n) \subset L(\mathcal{E}_1) \times \dots \times L(\mathcal{E}_n) \tag{1.29}$$

be the Zariski closure of the set of all $(f_1, \dots f_n)$ for which (1.26) has a solution in $(\mathbb{C}^*)^{n_{\hat{d}}}$ (that is the Zariski closure of all coefficients $a_{i,j} \in \mathbb{C}$ for which (1.26) has a solution).

For an overdetermined system of $n_{\hat{d}} + 1$ polynomials in $n_{\hat{d}}$ variables we have following result:

Theorem 1.1 (Sparse resultant (Gelfand et al, 1994; Cox et al, 2005)). *Assume that* $Q_i = \mathrm{conv}\,(\mathcal{E}_\mathrm{i})$ *is an* $n_{\hat{d}}$*-dimensional polytope for* $i = 1, \ldots, n_{\hat{d}} + 1$*. Then there is an irreducible polynomial* $\mathcal{R}$ *in the coefficients of the* f_i *such that*

$$(f_1, \ldots, f_{n_{\hat{d}}+1}) \in Z(\mathcal{E}_1, \ldots, \mathcal{E}_{n_{\hat{d}+1}}) \Longleftrightarrow \mathcal{R}(f_1, \ldots, f_{n_{\hat{d}}+1}) = 0. \tag{1.30}$$

In particular, if

$$f_1(d_1 \ldots d_{n_{\hat{d}}}) = \cdots = f_{n_{\hat{d}}+1}(d_1 \ldots d_{n_{\hat{d}}}) = 0 \tag{1.31}$$

has a solution $(\hat{d}_1, \ldots, \hat{d}_{n_{\hat{d}}})$ *in* $(\mathbb{C}^*)^{n_{\hat{d}}}$*, then*

$$\mathcal{R}(f_1, \ldots, f_{n_{\hat{d}}+1}) = 0. \tag{1.32}$$

Remark 1.3. The requirement that Q_i has to be $n_{\hat{d}}$-dimensional is no restriction and can be relaxed, (Sturmfels, 1994). However, for simplicity, we chose to present this result here.

Depending on the allowed space for the roots, there are other resultant types (e.g. Bezout resultants and Dixon resultants for system of homogeneous polynomials), with different algorithms to generate the coefficient matrix. Generally they will be conditions for roots in the projective space with homogeneous (or homogenized) polynomials. For more details on different resultants, we refer to Gelfand et al (1994); Sturmfels (1994); Cox et al (2005). An overview of different matrix constructions in elimination theory is given in Emiris and Mourrain (1999).

We choose to use the sparse resultant, because most polynomial systems encountered in practice are sparse in the supports. That means, for example, a polynomial of degree 5 in two variables x, y will not contain all 21 possible combinations of monomials $x^5, y^5, x^4y, xy^4, \ldots, x^4, y^4, x^3y, \ldots, y, x, 1$. Just as in linear algebra, this sparseness can be exploited for calculating the resultant. Another reason for using the sparse resultant is that it gives the necessary and sufficient conditions for toric roots, that is roots in $(\mathbb{C}^*)^{n_{\hat{d}}}$, such that the input polynomials need not be homogeneous (or homogenized), as in other resultants.

Finally, the sparse resultant enables us to work with Laurent polynomials, that is polynomials with positive and negative integer exponents.

Generally, resultant algorithms set up a matrix in the coefficients of the system. The determinant of this matrix is then the resultant or a multiple of it. Generating the coefficient matrices and their determinants efficiently is a subject to ongoing research, but there are some useful algorithms freely available. In this work, we use the maple software package `multires` Busé and Mourrain (2003), which can be downloaded from the internet[1].

[1] `http://www-sop.inria.fr/galaad/logiciels/multires`

For more details on the theory of sparse resultants, we refer to Gelfand et al (1994); Emiris and Mourrain (1999); Sturmfels (2002); Dickenstein and Emiris (2005).

1.5.1 Finding Invariant Controlled Variables for Polynomial Systems

After introducing the concepts above, we are ready to apply them in the context of controlled variable selection and self-optimizing control. As in the linear case above, we assume that the active constraints and the model equations, $g(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$, and the measurement relations, $m(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$, are satisfied.

Let $J^{(i)}_{\boldsymbol{z},red}$ denote the i-th element in the reduced gradient expression. To obtain the n_c controlled variables needed for the unconstrained degrees of freedom we have:

Theorem 1.2 (Nonlinear measurement combinations as controlled variables). *Given $\hat{\boldsymbol{d}} \in (\mathbb{R}^*)^{n_{\hat{d}}}$, and $n_y + n_g = n_{\hat{d}}$, independent relations $g(\boldsymbol{y}, \hat{\boldsymbol{d}}) = m(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$ such that the system*

$$\begin{aligned} g(\boldsymbol{y}, \hat{\boldsymbol{d}}) &= 0 \\ m(\boldsymbol{y}, \hat{\boldsymbol{d}}) &= 0 \end{aligned} \tag{1.33}$$

has finitely many solutions for $\hat{\boldsymbol{d}} \in (\mathbb{C}^*)^{n_{\hat{d}}}$. Let $\mathcal{R}(J^{(i)}_{z,red}, g, m), \quad i = 1 \dots n_{\boldsymbol{c}}$ be the sparse resultants of the n_c polynomial systems composed of

$$J^{(i)}_{z,red}(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0, \quad g(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0, \quad m(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0 \quad i = 1 \cdots n_c, \tag{1.34}$$

then controlling the active constraints, $g(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$, and $c_i = \mathcal{R}(J^{(i)}, g, m) \quad i = 1, \dots, n_c$, yields optimal operation throughout the region.

Proof. The active constraints are controlled, so $g(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$ and $m(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$ are satisfied always, and there is no condition on the parameters for this part of the system.

The system $g(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0, m(\boldsymbol{y}, \hat{\boldsymbol{d}}) = 0$ has only finitely many solutions for $\hat{\boldsymbol{d}}$, so the set of possible $\hat{\boldsymbol{d}}$ is fixed. Moreover, we know that a real solution to the subsystem $g(\boldsymbol{y}, \boldsymbol{d}) = m(\boldsymbol{y}, \boldsymbol{d}) = 0$ exists, since it is the given disturbance.

From Theorem 1.1, the sparse resultant gives the necessary and sufficient conditions for the existence of a solution for (1.34) in $\boldsymbol{d} \in (\mathbb{C}^*)^{n_d}$. Therefore, whenever $J^{(i)}_{z,red} = 0$, the resultant is zero (necessary condition). On the other hand if $\mathcal{R}(J_{z,red}, g, m) = 0$ then the system (1.34) is satisfied, too (sufficient condition).

This holds for any solution $\hat{\boldsymbol{d}} \in (\mathbb{C}^*)^{n_{\hat{d}}}$, and in particular the "actual" values of $\hat{\boldsymbol{d}}$. Because there are as many resultants as unconstrained degrees of

freedom, controlling $\mathcal{R}(J^{(i)}_{z,red}, g, m)$ for $i = 1, \ldots, n_u$ satisfies the necessary conditions of optimality in the region.

Remark 1.4. In cases where the $\hat{\boldsymbol{d}} \notin (\mathbb{C}^*)^{n_{\hat{d}}}$, we may apply a variable transformation to formulate the problem such we get $\hat{\boldsymbol{d}} \in (\mathbb{C}^*)^{n_{\hat{d}}}$. For example a translation $d = \tilde{d} - 1$.

Remark 1.5. By partitioning the overall optimization problem into several regions of active constraints, we assume that we have obtained well behaving systems for each region. In particular it is assumed that there are no base points (values of $a_{i,j}(\boldsymbol{y})$, where a polynomial in g or m vanishes for all values of $\hat{\boldsymbol{d}}$).

Remark 1.6. In some cases, the matrix of coefficients may be singular, yielding an identically zero determinant. These cases can be handled by a perturbation of the system at that point. This is a standard method of handling degeneracies in resultants Canny (1990); Rojas (1999).

Example 1.2 (Case with one disturbance). Consider the system of two polynomials in one unknown variable d, with one measurement relation $m(y, d) = 0$. At the optimum we must have:

$$\begin{aligned} J_{z,red} = \boldsymbol{N}\nabla_{\boldsymbol{z}} J(y, d) = a_{1,1}(y) + a_{1,2}(y)d = 0 \\ m(y, d) = a_{2,1}(y) + a_{2,2}(y)d + a_{2,3}(y)d^2 = 0 \end{aligned} \tag{1.35}$$

This system of univariate polynomials in d is overdetermined, and does not have a common solution d^* for arbitrary coefficients $a_{1,1}, a_{1,2}, a_{2,1}, a_{2,2}, a_{2,3}$. The sparse resultant coincides in the case of univariate polynomials with the classical resultant, which is the determinant of the Sylvester matrix (Cox et al, 1992), and the vanishing of the resultant is the necessary and sufficient condition for the existence of a common root. We construct the Sylvester matrix

$$Syl = \begin{bmatrix} a_{1,2}(y) & a_{1,1}(y) & 0 \\ 0 & a_{1,2}(y) & a_{1,1}(y) \\ a_{2,3}(y) & a_{2,2}(y) & a_{2,1}(y) \end{bmatrix}, \tag{1.36}$$

and the resultant is (where we omit writing the dependence on y explicitly):

$$\mathcal{R}\left(J_{z,red}, m(y, d)\right) = \det(Syl) = a_{1,2}^2 a_{2,1} - a_{1,2}a_{1,1}a_{2,2} + a_{2,3}a_{1,1}^2 \tag{1.37}$$

For a common root d^* to exist, the polynomial in the coefficients, $\mathrm{Res}(f_1, f_2)$ must vanish. Since the model $m(y, d)$ is chosen such that it is always satisfied, $m(y, d) = 0$ for any disturbance $d \in \mathbb{R}$, controlling the resultant to zero is the condition for the reduced gradient $J_{\boldsymbol{z},red}$ to become zero. So for a particular given exogenous $d \in \mathbb{R}$, the optimality conditions will be satisfied, and operation will be optimal.

1.6 Switching Operating Regions

In this section, we present a pragmatic approach for detecting when to change the control structure, because of changes in the active set. This task is a research field in itself (Baotić et al (2008) has worked on linear systems with quadratic objectives), and an exhaustive study is outside the scope of this paper. However, we would like to present a procedure, which may be used as starting point for a more thorough investigation of this problem in future work.

From a pure optimization perspective, there is no difference between a constraint and a controlled variable $\boldsymbol{c}(\boldsymbol{y})$, as the controlled variable may be simply seen as an active constraint, and, similarly, an active constraint may be considered a variable which is controlled at its constant setpoint. From this perspective, there is no difference between an active constraint and the model equations, either.

However, from an implementation point of view, there are differences between the model, the active constraints, and the controlled variables $\boldsymbol{c}(\boldsymbol{y})$. First of all, the active constraints and the controlled variables $\boldsymbol{c}(\boldsymbol{y}) = 0$ are not satisfied automatically, that is one has to control them to their setpoints. Secondly, since their values are known (or calculated using known measurements) they may be used for detecting when to switch control structures. The basic idea is to monitor the controlled variables and the active constraints of all neighbouring regions.

The main assumptions are that the regions are adjacent, that the disturbance moves the system continuously from one region to another, and that the system cannot jump over regions. In addition, we assume that controlling $\boldsymbol{c}(\boldsymbol{y}) = 0$ is equivalent to controlling the gradient to zero, as shown in the previous section. To determine when the control structure should be switched, we propose two rules:

1. (A new constraint becomes active) When a new constraint becomes active, change the control structure to the corresponding region
2. (A constraint becomes inactive) As soon as the controlled variable $\boldsymbol{c}$ in one of the neighbouring regions becomes zero (reaches its optimal setpoint), change the control structure to the corresponding region.

However, since we are working with systems of polynomial equations, there are some potential pitfalls here. The first one is that we are assuming that the regions of active constraints are adjacent, and that a changing disturbance moves the system continuously to from one region into another. Although this is the case for many systems in practice, it has to be confirmed that this holds for the particular case in consideration.

The second pitfall is that since our controlled variables are derived from the optimality conditions, this method will give optimal operation (and switching), as long as the same optimality conditions cannot be satisfied at two

distinct $\boldsymbol{d}$. This will hold if the optimization problem is convex in the disturbance space of interest.

1.7 Case Study

We consider an isothermal CSTR with two parallel reactions, as depicted in Fig. 1.2, taken from Srinivasan et al (2008). The reactor is fed with two feed streams F_A and F_B which contain the reactants A and B in the concentrations c_A and c_B. In the main vessel, the two components react to the desired product C, and the undesired side product D. The reactants A and B are not consumed completely during the reaction, so the outflow contains all four products. The CSTR is operated isothermally, and we assume that perfect temperature control has been implemented.

The products C and D are formed by the reactions:

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_1} \mathrm{C} \\ 2\,\mathrm{B} &\xrightarrow{k_2} \mathrm{D} \end{aligned} \tag{1.38}$$

We wish to maximize the amount of desired product $(F_A+F_B)c_C$, weighted by a yield factor $(F_A+F_B)c_C/(F_A c_{A,in})$ (Srinivasan et al, 2008). The amount of heat to remove and the maximum flow rate are limited by the equipment, and we formulate the mathematical optimization problem as follows (Srinivasan et al, 2008):

$$\max_{F_A,F_B} \frac{(F_A+F_B)c_C}{F_A c_{A_{in}}}(F_A+F_B)c_C \tag{1.39}$$

subject to

$$\begin{aligned} F_A c_{A_{in}} - (F_A+F_B)c_A - k_1 c_A c_B V &= 0 \\ F_B c_{B_{in}} - (F_A+F_B)c_B - k_1 c_A c_B V - 2k_2 c_B^2 V &= 0 \\ -(F_A+F_B)c_C + k_1 c_A c_B V &= 0 \\ F_A + F_B &\le F_{max} \\ k_1 c_A c_B V(-\Delta H_1) + 2k_2 c_B V(-\Delta H_2) &\le q_{max} \end{aligned} \tag{1.40}$$

Here, k_1 and k_2 are the rate constants for the two reactions, $(-\Delta H_1)$ and $(-\Delta H_2)$ are the reaction enthalpies, q_{max} the maximum allowed heat, V the reactor volume, and F_{max} the maximum total flow rate. The measured variables ($\boldsymbol{y}$), the manipulated variables ($\boldsymbol{u}$), the disturbance variables ($\boldsymbol{d}$), and the internal states are given in table 1.2, and the parameter values of the system are listed in table 1.3.

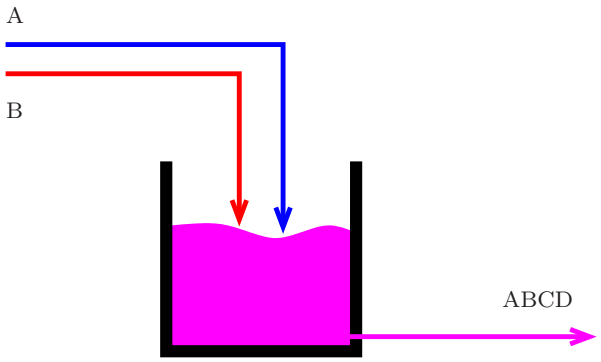


Fig. 1.2 CSTR with two reactions

Table 1.2 Overview of variables

Symbol	Description	Comment
F_A	Inflow stream A	measured input
F_B	Inflow stream B	measured input
F	total flow	measured variable
q	heat produced	measured variable
c_B	concentration of B	measured variable
c_A	concentration of A	unmeasured variable
c_C	concentration of C	unmeasured variable
k_1	rate constant reaction 1	unmeasured disturbance

Table 1.3 Parameters

Symbol	Unit	Value
k_1	l/(mol h)	0.3 - 1.5
k_2	l/(mol h)	0.0014
$(-\Delta H_1)$	J/mol	7×10^4
$(-\Delta H_2)$	J/mol	5×10^4
$c_{A,in}$	mol/l	2
$c_{B,in}$	mol/l	1.5
V	l	500
$F_{\max}$	l	22
q_{max}	kJ/h	1000

We write the combined vector of states $\boldsymbol{x} = [c_A, c_B, c_C]$ and manipulated variables $\boldsymbol{u} = [F_A, F_B]$ as

$$\boldsymbol{z} = \left[c_A, c_B, c_C, F_A, F_B \right]^{\boldsymbol{T}} . \tag{1.41}$$

1.7.1 Identifying Operational Regions

Following the procedure from Section 1.3, the system is optimized off-line for the range of possible disturbances $d = k_1$. This shows that the system can be partitioned into three adjacent critical regions, defined by their active constraints.

The critical regions are visualized in Fig. 1.3, where the normalized constraints are plotted over the disturbance range. In the first region, for disturbances below about $k_1 = 0.65\frac{\text{l}}{\text{mol h}}$, the flow constraint is the only active constraint. The second critical region for values between about $k_1 = 0.65\frac{\text{l}}{\text{mol h}}$ and $k_1 = 0.8\frac{\text{l}}{\text{mol h}}$ is characterized by two active constraints, i. e. both the flow constraint and the heat constraint are active. Finally, in the third region, above about $k_1 = 0.8\frac{\text{l}}{\text{mol h}}$ only the heat constraint remains.

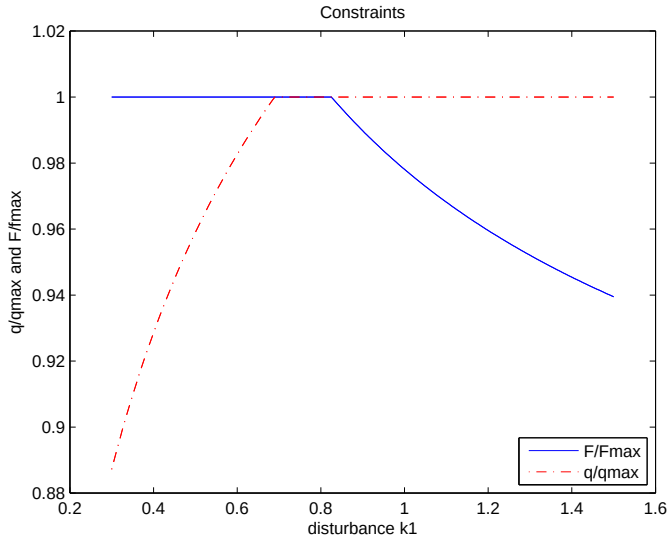


Fig. 1.3 Optimal values of the constrained variables

1.7.2 Eliminating λ

In each critical region, the set of controlled variables contains the active constraints (we know that they should be controlled at the optimum). This leaves the unconstrained degrees of freedom, which is the difference between the number of manipulated variables and the active constraints, $n_{DOF} =$

$n_{\boldsymbol{z}} - n_g$. For each of the unconstrained degrees of freedom one controlled variable is needed.

In the first critical region this gives $n_{DOF,1} = 5 - 4 = 1$ unconstrained degrees of freedom, so apart from the active constraint, which is the first controlled variable, we need to control one more variable (invariant).

To obtain the reduced gradient, we calculate the null space of Jacobian of the active set $\boldsymbol{N}_z^{\boldsymbol{T}}$ and multiply it with the gradient of the objective function $\nabla_{\boldsymbol{z}} J(\boldsymbol{z}, \boldsymbol{d})$ to obtain $J_{z,red,1} = \boldsymbol{N}_z^{\boldsymbol{T}} \nabla_{\boldsymbol{z}} J$. Depending on the algorithm to compute the null space, this may become a fractional expression, but since we want to control the process at the optimum, i. e. we control $J_{z,red,1}$ to zero, it is sufficient to consider only the numerator of $J_{z,red,1}$. This is possible because a fraction vanishes if the numerator is zero (provided the denominator is nonzero which is the case here because $\nabla_{\boldsymbol{z}} g$ has full rank). For the critical region 1, we obtain from (1.7) the invariant

$$\begin{aligned} J_{z,red,1} = -(F_A + F_B)^2 c_C \big[& -3c_C F_B^2 F_A - 3c_C F_A^2 F_B \\ & - 4c_C c_B F_A^2 k_2 V - 4c_C k_2 V^2 k_1 c_B^2 F_A - c_C F_A^3 \\ & - c_C F_B^3 - 4c_C k_2 V^2 k_1 c_B^2 F_B - c_C c_B F_A^2 k_1 V \\ & - 4c_C c_B F_B^2 k_2 V - c_C c_B F_B^2 k_1 V - c_C F_A^2 c_A k_1 V \\ & - c_C F_B^2 c_A k_1 V - 8c_C F_A c_B F_B k_2 V \\ & - 2c_C F_A c_B F_B k_1 V - 2c_C F_A F_B c_A k_1 V \\ & + 8F_A k_1 V^2 c_{A,in} k_2 c_B^2 + 2F_A^2 k_1 V c_B c_{A,in} \\ & + 2F_A k_1 V F_B c_B c_{A,in} - 2F_A^2 k_1 V c_{B,in} c_A \\ & -2F_A k_1 V F_B c_{B,in} c_A \big] \end{aligned} \tag{1.42}$$

which should be controlled to zero. This expression may be simplified slightly, since it is known that $(F_A + F_B)^2 c_C \neq 0$. It is therefore sufficient to control the factor in square brackets in (1.42) to zero.

Similarly, in the second critical region $n_{DOF,2} = 5 - 5 = 0$, and here we simply control the active constraints, keeping q at $q_{\max}$ and F at $F_{\max}$.

In the third critical region $n_{DOF,3} = 5 - 4 = 1$, and we use one of the manipulated variables control the active constraint ($q = q_{max}$) while the other one is used to control the invariant measurement combination $J_{z,red,3}$, which is an expression similar to (1.42).

1.7.3 Eliminating Unknown Variables

The invariant variable combinations for the first and the third critical region $J_{z,red,1}$ and $J_{z,red,3}$ still contain unknown variables, namely k_1, c_A and c_C, and cannot be used for feedback control directly.

To arrive at variable combinations which can be used for control, we include all known variables into $\boldsymbol{y}$, and all unknown variables into $\hat{\boldsymbol{d}}$, such that $\hat{\boldsymbol{d}} = [k_1, c_a, c_C]$. Then we write the necessary conditions for optimality as for each region as

$$\begin{aligned} J_{z,red}(\boldsymbol{y}, \hat{\boldsymbol{d}}) &= 0 \\ g(\boldsymbol{y}, \hat{\boldsymbol{d}}) &= 0. \end{aligned} \tag{1.43}$$

Considering the known variables $\boldsymbol{y}$ as parameters of the system, we want to find conditions on these parameters such that (1.43) is satisfied. The system has $n_{\hat{d}} = 3$ unknown variables, k_1, c_a and c_c, which we know that they are not zero. This corresponds to solutions $[k_1, c_A, c_C] \in (\mathbb{C}^*)^3$. According to section 1.5 we have that (1.43) is satisfied if and only if the sparse resultant is zero.

In critical region 1 and 3, the number of equations $n_{eq} = 5$ (model equations+active constrains+invariant), and the number of unknowns $n_{\hat{d}} = 3$. So we have more equations than necessary. Since we assume no measurement noise, all measurements are equally good, and we may select a subset of $n_{\hat{d}} + 1$ equations from (1.43) to compute the sparse resultant for the subset of equations. Obviously, the reduced gradient must be contained in this set of equations. Alternatively, as we do in the following, we can eliminate one more variable from the invariant.

For the first region, we use the sparse resultant of the system consisting of the invariant (1.42), the model equations (the first three equality constraints in (1.40)) and the first (active) inequality constraint in (1.40) to eliminate k_1, c_A, c_C and F_B and to calculate the controlled variable combination. The computations were performed using the `multires` software (Busé and Mourrain, 2003). After division by nonzero factors, the controlled variable for region 1 becomes:

$$\begin{aligned} c_1 = &-c_{b,in}^2 F_A^2 - F_A^2 c_{A,in} c_{b,in} + 6 F_A c_{A,in} k_2 c_b^2 V + 2 F_A c_{A,in} F_{max} c_b \\ &- F_A c_{A,in} F_{max} c_{b,in} + F_{max}^2 c_b^2 + c_{b,in}^2 F_{max}^2 + 4 V^2 k_2^2 c_b^4 \\ &- 2 c_{b,in} F_{max}^2 c_b - 4 V k_2 c_b^2 c_{b,in} F_{max} + 4 V k_2 c_b^3 F_{max} \end{aligned} \tag{1.44}$$

In the second critical region, control is simple; the two manipulated variables are used to control the two active constraints $F = F_{max}$ and $q = q_{max}$.

The third critical region is controlled similar to the first one. One input variable is used to control the active constraint, and the second input is used to control the resultant. The model equations (the first three equations) together with the energy constraint) in (1.40) and the reduced gradient are used to compute the resultant. Thus the unknown variables k_1, c_A, c_C, and F_B are eliminated from the reduced gradient. The controlled variable for region 3 is:

$$
\begin{aligned}
c_3 = &-4Vc_B^2k_2\Delta H_2F_Ac_{A,in}c_{B,in}q_{max}\Delta H_1 + F_Ac_{B,in}^2q_{max}^2\Delta H_1 \\
&+ 4V^2c_B^4k_2^2\Delta H_2F_Ac_{A,in}c_{B,in}\Delta H_1^2 - 4V^2c_B^4k_2^2\Delta H_2^2F_Ac_{A,in}c_{B,in}\Delta H_1 \\
&- 2Vc_B^2k_2F_Ac_{A,in}c_{B,in}\Delta H_1^2q_{max} - 4Vc_B^2k_2\Delta H_2F_Ac_{B,in}^2\Delta H_1q_{max} \\
&- 2Vc_B^2k_2\Delta H_2F_A^2c_{A,in}c_{B,in}^2\Delta H_1^2 + 8Vc_B^3k_2\Delta H_2\Delta H_1F_Ac_{A,in}q_{max} \\
&- 8V^2c_B^4k_2^2\Delta H_2c_{B,in}\Delta H_1q_{max} - 12V^2c_B^4k_2^2F_A\Delta H_2^2c_{B,in}^2\Delta H_1 \\
&- 8V^2c_B^5k_2^2\Delta H_2F_Ac_{A,in}\Delta H_1^2 + 8V^2c_B^5k_2^2\Delta H_2^2\Delta H_1F_Ac_{A,in} \\
&+ 8V^2c_B^5k_2^2F_A\Delta H_2^2c_{B,in}\Delta H_1 - q_{max}^3c_{B,in} + 2c_Bq_{max}^3 \\
&- \Delta H_1c_{B,in}F_Ac_{A,in}q_{max}^2 + 2c_BF_Ac_{A,in}q_{max}^2\Delta H_1 + F_A^2c_{A,in}c_{B,in}^2\Delta H_1^2q_{max} \\
&- 2c_BF_Ac_{B,in}q_{max}^2\Delta H_1 + 8Vc_B^3k_2\Delta H_2q_{max}^2 + 8V^2c_B^5k_2^2\Delta H_2^2q_{max} \\
&+ 8V^3c_B^6k_2^3\Delta H_2^3c_{B,in} - 2c_BF_A^2c_{A,in}c_{B,in}\Delta H_1^2q_{max} \\
&- 2Vc_B^2k_2\Delta H_1q_{max}^2c_{B,in} - 2Vc_B^2k_2\Delta H_2q_{max}^2c_{B,in} \\
&+ 4V^2c_B^4k_2^2\Delta H_2^2c_{B,in}q_{max} - 8V^3c_B^6k_2^3\Delta H_2^2c_{B,in}\Delta H_1
\end{aligned}
\tag{1.45}
$$

Although these expressions are quite involved, they contain only known quantities, and can be easily evaluated and used for control. Before actually using the measurement combinations for control, they are scaled so that the order of magnitude is similar. That is, c_1 is scaled (divided) by F_{max}, and c_2 is scaled by $\Delta H_1^2\Delta H_2F_AF_B$.

1.7.4 Using Measurement Invariants for Control and Region Identification

Having established the controlled variables for the three critical regions, it remains to determine, when to switch between the regions. Starting in the first critical region, the flow rate is controlled such that $F_A + F_B = F_{max}$, and the first measurement combination c_1 is controlled to zero. As the value of the disturbance k_1 rises, the reaction rate increases and the required cooling to keep the system isothermal, until maximum cooling is reached, Fig. 1.4. When the constraint is reached, the control structure is switched to the next critical region, where the inputs are used to control $q = q_{max}$ and $F_A + F_B = F_{max}$. While operating in the second region, the controlled variables of the neighbouring regions are monitored. As soon as one of the variables c_1 or c_3 reaches its optimal setpoint (i. e. 0) for its region the control structure is changed accordingly. Specifically, when k_1 is further increased, such that $c_3 = 0$ is reached, we must keep $F_A + F_B < F_{max}$ such to maintain the value $c_3 = 0$.

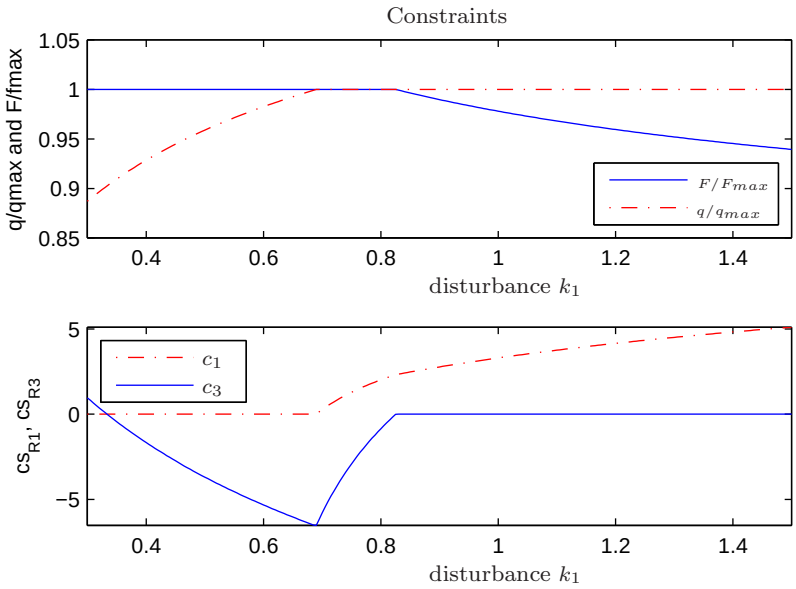


Fig. 1.4 Optimal values of controlled variables

1.8 Discussion

The presented method is based on the same idea as NCO tracking (François et al, 2005). However in contrast to NCO tracking, where the optimality conditions are solved for the optimizing *inputs*, this work focuses on finding the right *outputs* which express the optimality conditions. The corresponding inputs are generated by PI controllers and feedback control.

The method was developed as an alternative derivation and a generalization of the existing null space method (Alstad and Skogestad, 2007) for linear systems.

In the linear case, eliminating the constraints is straight forward, while this is not trivial in the polynomial case. However, by premultiplying ∇J by the null space of the constraints $\boldsymbol{N^T}$, we eliminate the Lagrangian multipliers from the equation set, and obtain the reduced gradient for the nonlinear case. The elimination of the Lagrangian multipliers can also be done by the resultant. Under the strict complementarity condition (either $\lambda = 0$ *or* the constraint is active), the solutions for λ lie in the toric variety, and therefore the sparse resultant gives necessary and sufficient conditions on the known variables so that the KKT system has a solution. We chose to apply the two-step procedure in this work, since this results in lower computational load when computing the resultants.

As an alternative to calculating resultants, the controlled variable combinations could be computed using Gröbner bases with an appropriate elimination ordering (Cox et al, 1992). One could use an appropriate monomial ordering which eliminates the unknown variables, and then use a polynomial from the elimination ideal as controlled variable. However, this Gröbner basis approach has some disadvantages, as it is not straightforward to find an elimination order which eliminates the unknown variables from the equation system while not yielding the "trivial solution" (i. e. the invariant is always zero when the constraints are satisfied). Another problem is that the selected invariant might give rise to "artificial solutions" which are not solutions of the original optimality conditions.

A similar approach is to calculate a Göbner basis for the ideal generated by the active constraints $g(\boldsymbol{y}, \hat{\boldsymbol{d}})$ and $m(\boldsymbol{y}, \hat{\boldsymbol{d}})$, and to reduce the $\boldsymbol{N}\nabla_{\boldsymbol{z}} J$ modulo the ideal. This avoids the trivial solution, however, the problem of choosing a monomial ordering which eliminates all unknown variables, remains.

Another argument against using a Gröbner basis for calculation the invariant, is, that it can yield very large and complicated expressions.

Since also the sparse resultants can give large expressions, in practice the method is best suited for small systems, with few constraints and equations. This is further emphasized by the fact that calculating the analytical determinant for large matrices is computationally demanding and that the construction of the resultant matrices is based on mixed subdivision, which is a hard enumeration problem (Cox et al, 2005).

In many cases (and in our case study) there are more polynomial equations than unknowns. Then the engineer has to chose which model polynomials to use in the resultant calculations in addition to the reduced gradient. From a purely mathematical view, this does not make any difference, as long as the set of model equations has finitely many solutions for $\boldsymbol{d}$. However the controlled variables will look quite different for different choices. The best (in terms of simplicity) choice depends on the structure of the equations, and is thus specific to the problem. However, as a general guideline, it would be advisable to keep the degrees of the polynomials low in the unknown variables. This leads to simpler resultants.

The resultant method, as presented above, does not take into account errors in the model and measurements. This is beyond the scope of this work. Our goal was to extend the idea of the null-space method (Alstad and Skogestad, 2007) and to demonstrate on a small example that the concept of finding variables which remain constant at optimal operation is possible also for polynomial systems.

Apart from handling noisy measurements and model mismatch, a further subject for future research is to find methods which reproduce not all solutions of the optimality conditions, but only a certain set of interest. This could be all the real solutions or solutions which reside in some further specified semialgebraic set.

1.9 Conclusions

In this chapter we have presented an approach to obtain optimal steady state operation which does not require online calculations. We have shown that, after identifying the critical regions, there exist optimally invariant variable combination for each region. If there are enough measurement/model relations ($n_g + n_m \geq n_{\hat{d}}$), the unknown variables can be eliminated by measurements and system equations, and the invariant combinations can be used for control using a decentralized self-optimizing control structure.

Further, we have shown that the measurement invariants can be used for detecting changes in the active set and for finding the right region to switch to.

Using methods form elimination theory, we have shown that, in principle, the idea of using polynomials in the measurements as self-optimizing control variables can be used to control the process optimally.

References

Abel NH (1826) Beweis der Unmöglichkeit, algebraische Gleichungen von höheren Graden als dem vierten allgemein aufzulösen. Journal für die reine und angewandte Mathematik pp 65–84

Alstad V (2005) Studies on selection of controlled variables. PhD thesis, Norwegian University of Science and Technology, Department of Chemical Engineering

Alstad V, Skogestad S (2007) Null space method for selecting optimal measurement combinations as controlled variables. Ind Eng Chem Res 46:846–853

Alstad V, Skogestad S, Hori E (2009) Optimal measurement combinations as controlled variables. Journal of Process Control 19(1):138–148

Baotić M, Borrelli F, Bemporad A, Morari M (2008) Efficient on-line computation of constrained optimal control. SIAM Journal on Control and Optimization 47:2470–2489

Bonvin D, Srinivasan B, Ruppen D (2001) Dynamic Optimization in the Batch Chemical Industry. in: Chemical Process Control-VI

Busé L, Mourrain B (2003) Using the maple multires package. URL http://www-sop.inria.fr/galaad/software/multires

Canny JF (1990) Generalised characteristic polynomials. J Symbolic Computation 9:241–250

Cao Y (2003) Self-optimizing control structure selection via differentiation. In: Proceedings of the European Control Conference

Cox D, Little J, O'Shea D (1992) Ideals, Varieties, and Algorithms. Springer-Verlag

Cox D, Little J, O'Shea D (2005) Using Algebraic Geometry. Springer

Dickenstein A, Emiris IZ (2005) Solving Systems of Polynomial Equations. Springer Berlin Heidelberg

Emiris IZ, Mourrain B (1999) Matrices in elimination theory,. Journal of Symbolic Computation 28(1-2):3–43, DOI DOI:10.1006/jsco.1998.0266

François G, Srinivasan B, Bonvin D (2005) Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. Journal of Process Control 15(6):701 – 712, DOI DOI:10.1016/j.jprocont.2004.11.006

Gelfand IM, Kapranov MM, Zelevinsky AV (1994) Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Boston, MA

Halvorsen IJ, Skogestad S (1997) Indirect on-line optimization through setpoint control. AIChE 1997 Annual Meeting, Los Angeles; paper 194h

Halvorsen IJ, Skogestad S, Morud JC, Alstad V (2003) Optimal selection of controlled variablesâĂă. Industrial & Engineering Chemistry Research 42(14):3273–3284

Heldt S (2009) On a new approach for self-optimizing control structure design. In: ADCHEM 2009, July 12-15, Istanbul, pp 807–811

Jäschke J, Skogestad S (2010) Self-optimizing control and NCO tracking in the context or real-time optimization. In: Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems, DYCOPS 2010, July 5-7, Leuven, Belgium, pp 593–598

Kariwala V, Cao Y, Janardhanan S (2008) Local self-optimizing control with average loss minimization. Ind Eng Chem Res 47:1150–1158

Marlin T, Hrymak A (1997) Real-time operations optimization of continuous processes. In: Proceedings of CPC V, AIChE Symposium Series vol. 93., pp 156–164

Morari M, Stephanopoulos G, Arkun Y (1980) Studies in the synthesis of control structures for chemical processes. part i: formulation of the problem. process decomposition and the classification of the control task. analsysis of the optimizing control structures. AIChE Journal 26(2):220–232

Nocedal J, Wright SJ (2006) Numerical Optimization. Springer

Pistikopoulos E, Georgiadis M, Dua V (2007) Multiparametric programming. Wiley-VCH

Rojas JM (1999) Solving degenerate sparse polynomial systems faster. J Symbolic Computation 28:155–186

Skogestad S (2000) Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10:487–507

Srinivasan B, Biegler LT, Bonvin D (2008) Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. Computers and Chemical Engineering 32:572–279

Sturmfels B (1994) On the newton polytope of the resultant. Journal of Algebraic Combinatorics 3:207–236

Sturmfels B (2002) Solving systems of polynomial equations

Chapter 2
Controller Performance Monitoring and Assessment

Selvanathan Sivalingam and Morten Hovd

Abstract The area of controller performance assessment is concerned with the analysis of existing controllers, for the purpose of locating areas where the control performance is inadequate. Thus, as opposed to most areas of control engineering that focus on controller *design*, controller performance assessment aims to provide tools for control system *maintenance*. Preferably, the controller performance monitoring should work with routine operating data, both to avoid disturbing the plant and because the sheer size of most chemical plants make active experimentation on a plant-wide scale unrealistic and in most cases unacceptable. Thus, the field of controller performance assessment gives value to the immense volumes of process data that are routinely logged and archived. The field has matured to the point where several commercial algorithms and/or vendor services are available for process performance auditing or monitoring.

2.1 Introduction

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). Recent research and development efforts in the area of primary control loop performance assessment have been targeted

Selvanathan Sivalingam
Department of Engineering Cybernetics, Norwegian University of Science and Technology, e-mail: selvanathan.sivalingam@itk.ntnu.no

Morten Hovd
Department of Engineering Cybernetics, Norwegian University of Science and Technology, e-mail: morten.hovd@itk.ntnu.no

at reducing the maintenance burden. The performance of a process controller often changes during plant operation. An initially well-tuned controller may become undesirably sluggish or aggressive due to equipment wear or changes in operating conditions causing changes in process dynamics (including the 'notorious' valve stiction) or operating constraints. A controller with poor performance increases operating costs, lowers product quality and even risks process safety. In practice, several poorly performing controllers often exist in plants and remain unnoticed for a quite long time before being detected and hopefully corrected. Having an automated means of detecting and then diagnosing the control loop degradation is essential to maintain/improve product quality, safety and also plant economy for any plant of non-trivial size. Even a 1% improvement either in energy efficiency or reduced product variability saves hundreds of millions of dollars for process industries (Bialkowski, 1993).

The term *control loop performance monitoring* means the action of watching out for changes in a statistic that reflects the control performance over time. The term *control loop performance assessment* refers to the action of evaluating such a statistic at a certain point in time. However, the two terms are used somewhat interchangeably in the literature.

The deployment of distributed control system (DCS), advanced control applications, and information management systems have become commonplace in the process industry. This has led to detailed information about the plant being archived on a daily basis. Competitive pressure and tighter environmental regulations have encouraged control engineers and managers to look at the archived information to identify potential areas of improvement and identify trends and problems in an incipient fashion for preventive maintenance. A spate of surveys on the performance of control loops reveal that a majority of control loops in processing industries perform poorly. Performance demographics of 26000 PID controllers collected across a wide variety of processing industries in a two year time span indicate that the performance of 16% of the loops can be classified as excellent, 16% as acceptable, 22% as fair, and 10% as poor, and the remaining 36% are in open loop (Desborough and Miller, 2002). Since PID controllers constitute 97% of all industrial controllers (Srinivasan et al, 2005), poorly performing loops pose a significant problem with huge financial implications.

In common industrial practice, only overall measures of process and control performance are monitored. The most commonly used measure of performance is the variance or standard deviation of key process variables. If the control strategies do not work well, the standard deviations can be very large. The reason that the standard deviation is used for monitoring is its direct relationship to process performance and profit. Optimal plant operation often lies at an operational constraint (or intersection of multiple constraints). A reduction in standard deviation therefore provides the ability to operate closer to the optimal point without increasing the risk of violating constraints. Furthermore, if no constraints are active at the optimal operating point, re-

duction in standard deviation implies that the plant operates more of the time close to that optimum.

To be practical, monitoring and diagnostic methods must be tailored to the needs of industrial plants. Thus, for controller performance assessment it is important to have an efficient tool available. Such a tool should be ideally equipped with a good user-friendly interface, readily understandable report generation, diagnosis information in text form, having a single composite index ranking the loop performance, and employ reliable computational software. Most large plants have many poorly performing loops, and instead of overwhelming operators and maintenance personnel the monitoring tool should advise on what performance problems should be tackled first, giving a clear ranking of the control loops according to the severity of the performance problems. The monitor should not disturb routine plant operation, and it should hence use only the routine plant operation data.

A primary difficulty of controller performance monitoring is the shear number of loops to be monitored – a typical large processing operation consists of hundreds of control loops, often operating under varying conditions. The majority of the controllers use the PID algorithm, but there may also be advanced multivariate model-based controllers and other application specific controllers. Maintenance of these loops is generally the responsibility of either a control engineer or an instrument technician, but other responsibilities, coupled with the tediousness of consistently monitoring a large number of loops, often results in control problems being overlooked for long periods of time.

Real-time controller performance monitoring to identify poorly or underperforming loops has become an integral part of preventative maintenance. Among others, rising energy costs and increasing demand for improved product quality are driving forces. Automatic process control solutions that incorporate realtime monitoring and performance analysis are fulfilling this market need. While many software solutions display performance metrics, however, it is important to understand the purpose and limitations of the various performance assessment techniques since each metric signifies very specific information about the nature of the process.

Controller performance assessment has been an area of active research for the last two decades, and several advanced algorithms are available in commercial software packages. These packages enable control engineers to easily and accurately obtain controller quality metrics without performing plant tests, and to monitor all aspects of their control loops.

Fig. 2.1 illustrates a typical control loop. Controller performance assessment techniques address questions such as

1. What set of performance measures should be chosen for a given process so that the scope for improvement in performance is correctly highlighted?
2. How can one arrange a set of performance measures to figure out the improvement potential without disturbing the running system? Is the controller 'healthy'?

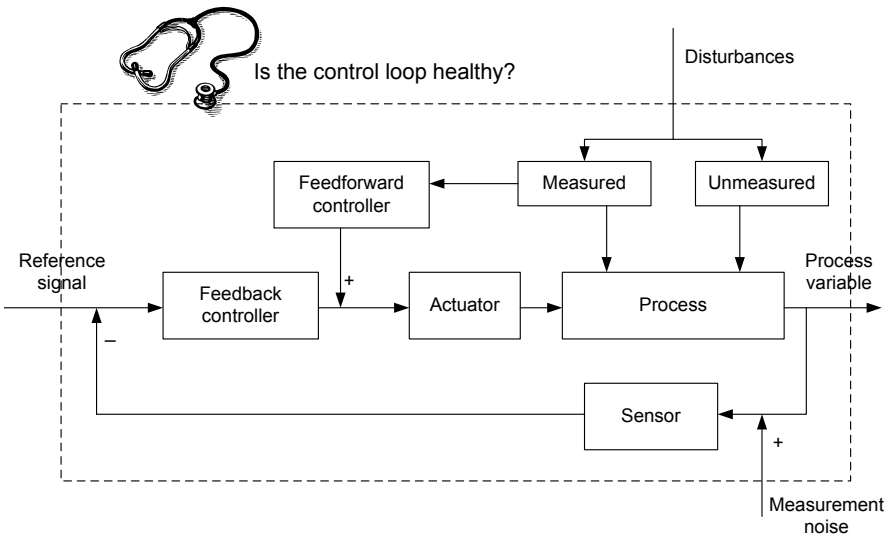


Fig. 2.1 A simple figure illustrating control performance assessment problems, inspired by Jelali (2005).

3. Is the controller doing its job satisfactorily? If not, what leads to loop degradation?
4. How can the growing data available from the process industries be exploited for this task?

A recent survey by Jelali (2005) summarizes answers to some of these questions which are described in the following section.

2.2 Sources of Poor Control Loop Performance

There are several possible causes for poor control performance, requiring different remedies. One way of categorizing the causes of poor control performance is the following four groups:

1. Improper and inadequate controller tuning and lack of maintenance
2. Equipment malfunction or poor design
3. Poor or missing feedforward compensation
4. Inappropriate control structure

In the following, each of these causes for poor performance is discussed in more detail.

2.2.1 Improper and Inadequate Controller Tuning and Lack of Maintenance

The ability of proportional integral (PI) and proportional integral derivative (PID) controllers to compensate most practical industrial processes has led to their wide acceptance in industrial applications. As per literature, there are perhaps only 5-10% of control loops that cannot be controlled adequately by single input, single output (SISO) PI or PID controllers; in particular, these controllers are believed to perform well for processes with benign dynamics and modest performance requirements. It has been stated that 98% of control loops in the pulp and paper industries are controlled by SISO PI controllers (Bialkowski, 1996) and that in process control applications, more than 95% of the controllers are of PID type.

PI or PID controller implementation has been recommended for the control of processes of low to medium order, with small time delays, when parameter setting must be done using tuning rules and when controller synthesis is performed either once or more often. However, despite decades of development work, surveys indicating the state of the art of control in industrial practice report sobering results.

For example, Ender (1993) states that in his testing of thousands of control loops in hundreds of plants, it has been found that more than 30% of installed controllers are operating in manual mode and 65% of loops operating in automatic mode produce less variance in manual than in automatic, (*i.e.*, the automatic controllers actually degrade operational performance rather than help improving it). The situation does not appear to have improved more recently, as per latest report (Overschee and Moor, 2000) that 80% of PID controllers are badly tuned; 30% of PID controllers operate in manual with another 30% of the controlled loops increasing the short term variability of the process to be controlled (typically due to too strong integral action). Also, it is stated that 25% of all PID controller loops use default factory settings, implying that they have not been tuned at all.

One clearly cannot expect good performance from a controller that has never been tuned. For controllers that have been tuned, but still show poor performance, inappropriate tuning parameters may be the result of tuning based on a poor plant model or inadequate mastery of control engineering by the person performing the tuning. The most common cause of poor control performance is, however, that controllers are normally designed and tuned at the commissioning stage, but left unchanged after that for years (or decades), although the performance of many control loops decays over time owing to changes in the characteristics of the material/product being used, modifications of operating points/ranges and changes in the status of the plant equipment (wear, plant modifications).

In industrial practice, the main reasons quoted for lack of tuning and maintenance are:

- The commissioning engineers tune the controllers until they are considered "good enough". They do not have time to optimise the control. Most controllers are tuned once they are installed, and then never again.
- Often, the controllers are conservatively tuned (*i.e.*, for the "worst case") to retain stability when operating conditions change in non-linear systems.
- There are only a few people responsible for maintenance of automation systems and all are fully busy with keeping the control systems in operation, *i.e.*, they have no or very little time for improving controllers. Typically a remarkable number of controllers have to be maintained by a very small number of control engineers.
- Operators and engineers often do not have the necessary education and understanding of process control to be able to know what can be expected of the control or what the causes of poor performance are. Sometimes, the poor control performance becomes the norm and production people accept it as normal. Various studies indicate that the 'half-life' of good control loop performance is about 6 months (Bialkowski, 1993).

2.2.2 Equipment Malfunction or Poor Design

Control loop performance degradation may be the result of failing or malfunctioning sensors or actuators (*e.g.*, due to stiction, hysteresis and deadband). Sensors are critical components in almost all modern engineering systems and are used to not only obtain basic plant operational information but also to compute control actions. A fault in a sensor is typically characterized by a change in its operational characteristics, and severe sensor faults will typically cause measurements to misrepresent the operation condition of plant; mislead control actions and consequently cause energy waste, an increase in operation costs, and/or unacceptable quality. Failing sensors may also pose a risk to the overall safety of the plant. Reliable sensors are essential for reliable monitoring and control of automation systems. The detection and diagnosis of these changes, or sensor fault diagnosis, plays an important role in the operation of many engineering systems. Sensor fault detection falls within the well-established field of fault detection, for which there is a vast specialized literature. Although accurate sensors is a prerequisite for good control performance, the topic will not be addressed further here.

Oscillations in control loops raise particular concerns because they increase variability in product quality, accelerate equipment wear, and may cause other issues that could potentially disrupt the operation. Therefore, detecting and diagnosing oscillations yield commercial benefits and are important activities in control loop supervision and maintenance.

Generally, oscillations are caused by any one or a combination of the following reasons:

1. limit cycles caused by valve stiction or other process nonlinearities

2. poor controller tuning
3. poor process and control system design
4. external oscillatory disturbances

More serious is the problem when the process or a process component is not appropriately designed. The relation between process design and control can be succinctly summarised by the following quotation from a paper by Ziegler and Nichols (1943): "In the application of automatic controllers, it is important to realize that controller and process form a unit; credit or discredit for results obtained are attributable to one as much as the other. A poor controller is often able to perform acceptably on a process, which is easily controlled. The finest controller made, when applied to a miserably designed process, may not deliver the desired performance. True, on badly designed processes, advanced controllers are able to eke out better results than older models, but on these processes there is a definite end-point which can be approached by the instrumentation and it falls short of perfection." Thus, the problems mentioned in this item cannot be overcome by retuning the controller.

2.2.3 Poor or Missing Feedforward Compensation

In an industrial control system there are often many measured signals available in addition to the the measurement of the process variables that are actively controlled. The question is to find a way to select the feedforward variable out of the available measurements. Petersson et al (2003) has made a comparison of two different methods for feedforward control structure assessment. The first method is taken from Desborough and Harris (1993), and is based on comparing the actual variance with the minimum achievable variance, and the contribution of the disturbance to the overall variance. The second is a method for evaluating deterministic additive disturbances and estimate where they enter in the process. The methods complement each other and show that assessment methods should be handled with care, and their use should reflect the disturbance scenario affecting the plant. In the case of frequent deterministic disturbances, rejection time speed of response, etc., may be appropriate performance measures, and the second assessment method appears more appropriate. For minimizing yield and quality variations in the presence of stochastic disturbances, the first assessment method should be chosen.

The first assessment method can be applied using only routine operating data, whereas the second method requires knowledge of a plant and disturbance model or experimentation on the plant. However, although feedforward control is generally sensitive to model errors (Skogestad and Postlethwaite, 2005), the second assessment method is not particularly sensitive to model accuarcy (Petersson et al, 2003).

Clearly, the decision whether to use feedforward control depends on whether the degree of improvement in the response to the measured disturbance justifies the costs of implementation and maintenance.

Feedforward control is always used along with feedback control, even for open loop stable systems, because feedback is required to track setpoint changes and to suppress unmeasured disturbances that are always present in any real process.

2.2.4 Inappropriate Control Structure

Inadequate input/output pairing, ignoring mutual interactions between the system variables, competing controllers, insufficient degrees of freedom, the presence of strong nonlinearities and the lack of time-delay compensation in the system are frequently found as sources for control problems.

A proper coverage of these issues is far beyond the scope of this chapter. Instead, we refer to, e.g., Larsson and Skogestad (2000) and Skogestad and Postlethwaite (2005) for information on issues such as

- Systematic approaches to control system design for entire plants, including the determination of the available degrees of freedom for plant operation.
- Interaction analysis, in particular the use of the Relative Gain Array (RGA).

Occasionally, strong nonlinearities are well known, are essentially static, and are easily described based on readily available plant information. In such cases, gain scheduling may possibly be designed without much difficulty, and be able to counteract much of the plant nonlinearity, yielding a controller response that does not vary much over the desired operating region. In other cases, feedback may be utilized to obtain a more linear response. For example, a local flow controller may 'linearize' a non-linear valve, yielding an essentially linear response from setpoint change to observed flow in cases where the response from change in valve stem position to observed flow would be highly non-linear.

There exists a vast literature on non-linear control that may be consulted when the simple approaches above do not suffice, see e.g., Khalil (1996). However, in contrast to simple linear controllers like the PI controller, which may often be tuned on-line, most non-linear controller design approaches requires the plant model to be known, which is actually a major complicating factor in many cases.

Time delay compensation, such as the use of a Smith predictor, may drastically improve control performance. However, one should be aware that in many applications the time delay will depend strongly on operating conditions. This may cause robustness problems unless properly accounted for in the controller design.

2.3 Detection and Diagnosis of Oscillations in Control Loops

Poorly performing control loops often display oscillations. Looking for oscillating loops is therefore a reasonable approach to identifying (some of the) poorly performing control loops in the plant. In most cases, oscillating loops are easily identified by manual inspection. The task of oscillation detection is therefore primarily focussed on detection methods that can be easily and reliably automated for either on-line or archival data, as low manpower and the sheer number of control loops rule out manual inspection of all the loops.

Due to interactions in a process plant, oscillations originating in one loop will usually spread to other loops. Similarly, external oscillating disturbances can spread widely across the plant. Following detection of the oscillating loops in a plant, it is therefore desirable determine where the oscillations arise, and preferably also diagnose the cause of the oscillations.

2.3.1 Detection of Oscillating Control Loops

For the trained human eye, detection of oscillations may seem a trivial task. However, it is far from trivial to define and describe oscillations in a typical signal from a process plant in such a way that it can reliably be automated (in either on-line or off-line tools). The following are some of the properties of oscillating signals that may be exploited to identify the periodicity present in the signals. It is assumed that the signals under study are stable, or at least only marginally unstable, as otherwise the control loops in question will have to be taken out of service (and it should then be apparent that the control loop needs attention). Any exponentially growing signal will eventually hit some system constraint or cause some malfunction. It is to be noted that control loops are here classified as oscillatory if they show an unacceptable tendency to oscillate, a perfect limit cycle is not a requirement. Stable loops with insufficient damping will also be classified as oscillatory in this context.

A number of oscillation detection methods are suggested in the literature which fall into the following four main categories:

1. Methods based on the auto-covariance function
2. Spectral peak detection
3. Methods based on the integrated absolute error
4. Method based on wavelet plots

Before going into details of different oscillation detection methods, it is necessary to know some important statistical tools which are defined as follows:

2.3.1.1 The Auto-covariance Function (ACF)

The auto-covariance function of a stationary process is essentially a measure of how closely the values of a variable (the statistical dependence of time-series data), when measured at different times, are correlated. For a variable x and a data set of N data-points, the auto-covariance function is defined as

$$r_{xx}[l] = \frac{1}{N}\sum_{k=1}^{N-1}(x[k] - \bar{x})(x[k+l] - \bar{x}) \tag{2.1}$$

where k denotes the sample index and $\bar{x}$ is the mean value of the series. The ACF of a time-series is symmetric about the lag $l = 0$. For time series from stable systems (like the ones we are considering here), it clearly takes the largest value at lag $l = 0$. Often, a normalized auto-covariance function, known as the autocorrelation function is used. It is defined as

$$\rho_{xx}[l] = \frac{r_{xx}[l]}{r_{xx}[0]} \tag{2.2}$$

$\rho_{xx}[l]$ lies between -1 and 1 for stable systems, and the autocorrelation function of an oscillatory signal is also oscillatory. For stable signals, it generally decays with increasing lags, whereas it will oscillate for systematically oscillating signals, and a periodic signal will have a periodic autocorrelation function. In principle, one should be able to detect oscillations directly from the autocorrelation function. However, it might not be the case if the signal contains multiple frequencies, measurement noise, assymmetric oscillations, etc. Nonlinear effects may also introduce oscillations at frequencies that are multiples of the base oscillation frequency. Nevertheless, Moiso and Piiponen (1998) propose an oscillation index calculated from the roots of a second order AR model fitted to the autocorrelation function. The method of Miao and Seborg, which is described below, is also based on the autocorrelation function.

2.3.1.2 The Power Spectrum

Then power spectral density (PSD), or simply the power spectrum is a positive real function of a frequency variable associated with a stationary random process. The power spectrum results from a Fourier transform of the auto-covariance function, and in essence it is the frequency domain equivalent of the auto-covariance function.

$$\phi_{xx}(\omega) = \sum_{l=-(N-1)}^{N-1} r_{xx}[l]e^{-il\omega} \tag{2.3}$$

If the signal exhibits a purely sinusoidal oscillation at a particular frequency, the power spectrum will have a peak at that frequency. An oscillation that does not decay with time, will have a very large peak at that frequency in the power spectrum. The problems of using the power spectrum for oscillation detection are similar to those of using the autocorrelation function. Instead of the power spectrum having a single spike at the oscillating frequency, the signal may be corrupted by noise and nonlinear effects such that the power spectrum is blurred or contains numerous spikes.

2.3.2 The Oscillation Detection Method of Miao and Seborg (ACF Based)

Miao and Seborg (1999) uses the autocorrelation function to detect oscillations. It calculates a somewhat non-standard 'decay ratio', as illustrated in Fig. 2.2.

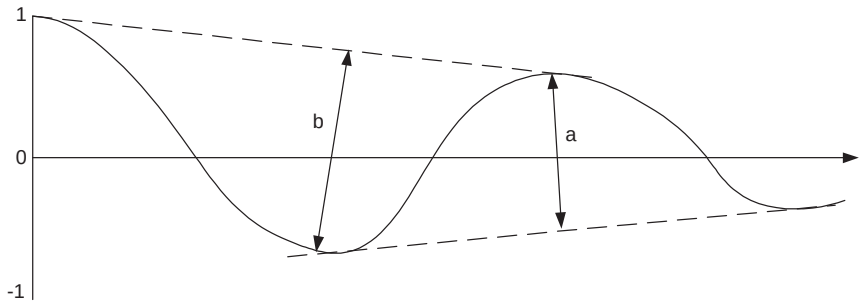


Fig. 2.2 Calculation of the Miao-Seborg oscillation index from the autocorrelation function.

The Miao-Seborg oscillation index is simply the ratio given by $R = a/b$. A threshold value of $R = 0.5$ is proposed, a larger value will indicate (unacceptable) oscillations. Little justification is provided for this measure. In particular, it is not explained why this measure is better than simply comparing the magnitude of neighbouring peaks in the autocorrelation function.

Nevertheless, industrial experience appears to be favourable, and oscillations are detected with reasonable reliability. Some drawbacks are

- it is relatively complicated for on-line oscillation detection and hence, it is better suited for offline analysis of batches of data.

- it does not take the amplitude of oscillations directly into account. Some oscillations of small amplitude may be acceptable, but this method will classify also loops with acceptable oscillation as oscillatory.
- it assumes that the oscillations are the main cause of variability in the measured variable. If a control loop experiences frequent (and irregular) setpoint changes of magnitude larger than the amplitude of the oscillations, it may fail to detect the oscillations.

These comments also apply to the ACF-based detection method of Moiso and Piiponen (1998) described above.

2.3.2.1 Spectral Peak Detection Method

This method is based on looking for peaks in the power spectrum. Visual inspection of spectra is helpful in this method because strong peaks can be easily seen – although as previously noted it is desirable to fully automate the oscillation detection method. Automated oscillation detection based on spectral analysis becomes difficult if the oscillation is intermittent and periods vary every cycle, as the power spectrum may then have blurred or multiple peaks.

2.3.2.2 IAE Based Detection Methods

There are several oscillation detection methods that are based on the Integral Absolute Error (IAE). Hägglund's method of oscillation detection Hägglund (1995) falls in this category, although Hägglund's measure may be said to be a more general measure of control performance rather than exclusively an oscillation detection method. The basic idea behind the measure is that the controlled variable in a well-functioning control loop should fluctuate around the setpoint, and that long periods on one side of the setpoint is a sign of poor tuning.

Hägglund's performance monitor looks at the control error $e(t) = r(t) - y(t)$, and integrates the absolute value of $e(t)$ for the period between each time this signal crosses zero:

$$\text{IAE} = \int_{t_{i-1}}^{t_i} |e(t)| \, \mathrm{d}t$$

where t_{i-1} and t_i are the times of two consecutive zero crossings. Whenever this measure increases beyond a threshold value, a counter is incremented, and an alarm is raised when the counter passes some critical value. It is shown in Hägglund (1995) how a forgetting factor can be used to avoid alarms from well-functioning loops which are exposed to infrequent, large disturbances (or setpoint changes).

Critical tuning parameters for this monitoring method are the IAE threshold value and the counter alarm limit. Typical choices for the IAE threshold value are

$$\text{IAE}_{\text{lim}} = 2a/\omega_u$$
$$\text{IAE}_{\text{lim}} = aT_I/\pi$$

where a is an acceptable oscillation magnitude, ω_u is the ultimate frequency (the oscillation frequency found in a closel loop Ziegler Nichols experiment), and T_I is the integral time in a PI(D) controller. The more rigorous of the two treshold values is the first, and ω_u would be available if the loop was tuned with e.g. Hägglund's relay-based auto tuning procedure. However, often ω_u will not be available, and the second expression for IAE_{lim} will then have to be used – this expression is intended to work as a reasonable approximation of the first expression for IAE_{lim} for a reasonably tuned loop. Naturally, this may be misleading if the cause of poor control performance is poor choice of controller tuning parameters.

The counter alarm limit is simply a tradeoff between the sensitivity of the monitoring method and the rate of "unnecessary" alarms. This monitoring method is

- Simple and appliccable for on-line implementation.
- It takes oscillation amplitude into account – it is ignores small oscillations unless the oscillation period is very long.
- Some tuning of the monitoring method must be expected. The guidelines for choosing IAE_{lim} is based on knowledge of the ultimate frequency of the control loop – which typically is not known unless a Ziegler-Nichols type tuning experiment or a Hägglund type autotuner is used. Alternatively, it is proposed to base IAE_{lim} on the controller integral time – which is only reasonable if the loop is well tuned.

2.3.2.3 The Regularity Index

Hägglund's monitoring method is extended in Thornhill and Hägglund (1997) for off-line oscillation detection, resulting in a new oscillation measure called the regularity index.

To calculate the regularity index, the integral absolute error is calculated, and when the control error crosses zero, the measure

$$\frac{\text{IAE}_i}{\Delta T_i \sigma} \tag{2.4}$$

is plotted together with the time t_{i+1} for the most recent zero crossing. Here IAE_i is the integral absolute error between the two most recent zero crossings, ΔT_i is the time between the zero crossings, and σ is an estimate of the r.m.s.

value of the noise. It is recommended to filter the measurements by estimating an AR model for the measurement, and to base the analysis (calculation of IAE) based on a one step ahead prediction from the AR model rather than the raw measurement. This will reduce the influence of measurment noise, and the AR model estimation can also give an estimate of the measurement noise, from which σ can be calculated.

Next, a treshold value ξ is chosen, and a *regularity factor* is derived from the time intervals Δk_i between each time the measure in Eq. (2.4) crosses the treshold value. Thus,

$$R_i = \frac{\Delta k_{i+1}}{\Delta k_i}; \qquad q(\xi) = \frac{\text{Mean value of } R}{\text{Standard deviation of } R} \tag{2.5}$$

The regularity index is then

$$q = \max_{\xi} q(\xi) \tag{2.6}$$

The period of oscillation is estimated from the number of times the measure in Eq. (2.4) crosses the treshold ξ between the first and last instance of crossing the treshold.

2.3.2.4 The Method of Forsman and Stattin

This method also looks at the control error $e(t) = r(t) - y(t)$, but it is strictly an oscillation detection method and not a general performance measure. Forsman and Stattin (1999) proposes comparing both the areas between the control error and zero and the time span that the error has the same sign. However, the resulting area and time span is not compared with the immediately previous area/timespan (when the control error had opposite sign), rather the comparison is made with the preceding period when the control offset had the same sign. This is illustrated in Fig. 2.3.

The method uses two tuning constants α and γ, that both should be in the range between 0 and 1, and simply counts the number of times h_A in a data set that

$$\alpha < \frac{A_{i+1}}{A_i} < \frac{1}{\alpha} \text{ and/or } \gamma < \frac{\delta_{i+1}}{\delta_i} < \frac{1}{\gamma}$$

and the number of times h_B that

$$\alpha < \frac{B_{i+1}}{B_i} < \frac{1}{\alpha} \text{ and/or } \gamma < \frac{\varepsilon_{i+1}}{\varepsilon_i} < \frac{1}{\gamma}$$

where A_i, B_i, δ_i and ε_i are defined in Figure 2.3. The oscillation index is then given by $h = (h_A + h_B)/N$, where N is the number of times in the data set that the control offset crosses zero.

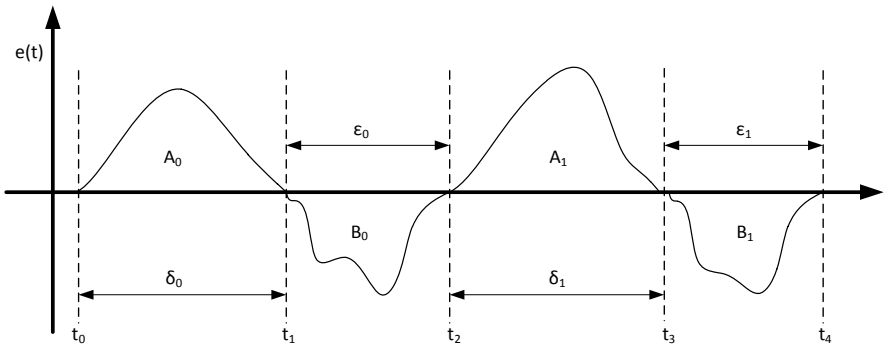


Fig. 2.3 The oscillation detection method of Forsman and Stattin.

Forsman and Stattin recommend closer examination of loops having $h > 0.4$, and if $h > 0.8$ a very clear oscillative pattern can be expected.

2.3.2.5 Wavelet Based Methods

When there is a persistent oscillation with little variation on the oscillation frequency, the power spectrum gives a clear signature for the oscillation. This is because, the signal has a sharp peak of large magnitude at the frequency of oscillation. However, in process industries, there are some cases where the oscillation is intermittent, *i.e.* non-persistent. In such scenario where the nature of the signal changes over time, the Fourier transform is used on subsets of the data to observe the time-varying frequency content. At this point, it should be emphasised that the decision of dividing data into different segments is done heuristically by visual inspection and there appears to be no rigorous method available to perform such segmentation. Wavelet analysis plays a crucial role here in treating time and frequency simultaneously in time-frequency plane. This provides signal amplitude as a function of frequency of oscillation (the resolution) and time of occurrence.

A typical wavelet spectrum indicates time on the horizontal axis and period (or frequency) on the vertical axis, and amplitude is represented by hues in the contour lines on the time-frequency plane (Torrence and Compo, 1998). It is then possible to visualize the relation between the timing of frequency emerging and disappearing in the signal, thus providing more precise and deeper insights into the process behaviour. Wavelet analysis has been successfully applied to plant-wide disturbance (oscillation) detection or diagnosis by Matsuo et al (2004).

2.3.2.6 Pre-filtering Data

All methods presented above may be ineffective for noisy data, and both Miao and Seborg (1999) and Forsman and Stattin (1999) discuss pre-filtering the data with a low pass filter to reduce the noise. Thornhill and Hägglund (1997) propose filtering through using the one-step-ahead prediction from an AR model, as described previously. Clearly, the filter should be designed to give a reasonable tradeoff between noise and oscillation detection in the frequency range of interest. The interested reader should consult the original references for a more comprehensive treatment of this issue.

2.3.2.7 Brief Conclusions on Oscillation Detection

Oscillation detection is a very easy task if the signal is pure sinusoidal with a single dominant frequency, without any noise and disturbances. In practice, however, the measurements of process variables are corrupted by instrument noise, and unknown disturbances. Further, the presence of multiple oscillations is common, caused by different kinds of faults occurring simultaneously. In some cases, intermittent oscillations are also a possibility. These effects often destroy the regularity of oscillations, which make oscillations harder to detect.

A good oscillation-detection method should be robust to such kinds of difficult scenarios to accurately detect the presence of oscillations in the time series. A good oscillation-detection methodology for industrial applications should have the following features: (a) usage of only time-series information of process variables with limited or no additional process knowledge, (b) robustness to the high-frequency measurement noise and disturbances, (c) Ability to handle the presence of multiple and intermittent oscillations, and (d) Amenability to complete automation without human intervention.

2.3.3 Oscillation Diagnosis

Once an oscillating control loop has been detected, it is naturally of interest to find the cause of the oscillations, in order to come up with some effective remedy. There is no general solution to the diagnosis problem, the proposed methods can at best handle parts of the problem. We will present diagnosis procedures proposed by Hägglund (1995). Valve stiction is a very common cause for oscillations. A separate section (the next section) is therefore devoted to diagnosing this cause of oscillations.

2.3.3.1 Manual Oscillation Diagnosis

Hägglund (1995) proposes the manual oscillation diagnosis procedure presented in Fig. 2.4

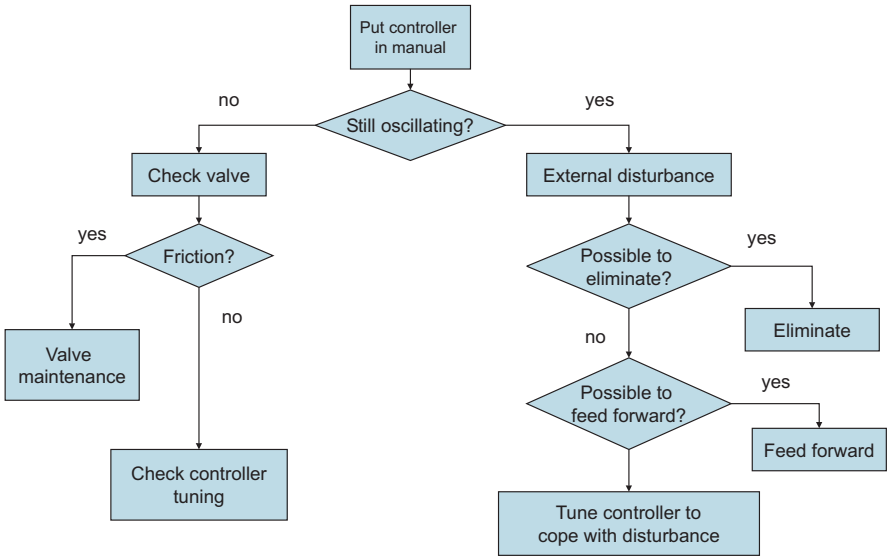


Fig. 2.4 Hägglund's method for manual oscillation diagnosis.

The main problem with this procedure is the assumption that if the oscillation (in the controlled variable) stops when the controller in a particular loop is put in manual, then the oscillation is caused by that loop. Often, oscillations arise from multivariable interactions between loops, and the oscillation will then stop when any one of these loops are put in manual. The first loop to be put in manual will then receive the "blame" for the oscillations, and will consequently be detuned (made slower). Therefore, the results of this procedure will depend on the order in which the loops are examined. If several loops show a similar oscillation pattern, one should therefore first examine the loop for which slow control is more acceptable.

The procedure is also a little short on examining other instrumentation problems than valve friction (stiction), *e.g.*, valve hysteresis, measurement problems, etc. Furthermore, the procedure gives no proposals for how to eliminate external disturbances. Clearly, the solution will be very dependent on the particular process, but typically it will involve modifying the process or the control in other parts of the process.

Additional flowcharts for oscillation diagnosis are presented in Thornhill and Hägglund (1997). Some of those flowcharts do not require putting the controller in manual. They also show how useful diagnostic information can

be derived from plotting the controlled variable (pv) vs. the setpoint (sp). Idealized plots for actuators with deadband, static friction in actuator, oversized valve as manipulated variable, and a linear loop with phase lag are shown. The use of such sp-pv plots is clearly limited to loops with frequent setpoint changes, otherwise setpoint changes have to be introduced purely for diagnostic purposes (*i.e.*, the plant has to be disturbed).

Thornhill and Hägglund (1997) also address nonlinearity detection (without further classifying the non-linearity) using the regularity index and the power spectrum for the controlled variable.

2.4 Diagnosis of Valve Stiction: Issues and Directions

A key challenge in controller performance monitoring is to find the root cause of distributed oscillations in chemical plants. Valves are the most common manipulated variables in chemical plants, and oscillations can cause a valve to wear out prematurely. It has been found that about 30% of the loops are oscillatory due to control valve problems (Bialkowski, 1993). Usually, the cause for such control valve problems is some undesirable valve nonlinearity, such as stiction or deadband. Among the many types of nonlinearities in control valves, stiction (short for *static friction*) is the most common and leads to more serious oscillations. Many studies have been carried out to develop methods to detect stiction.

2.4.1 Definitions: Stiction, Deadzone and Backlash

This section gives a brief definition of some important valve nonlinearities. Fisher (1999) uses the following definitions:

Friction is a force that tends to oppose the relative motion of two surfaces that are in contact with each other [1]. Friction has two components: static and dynamic friction. *Static friction* is the force that needs to be overcome before there is any relative motion between the surfaces. *Dynamic friction* is the force that needs to be overcome to maintain (already existing) relative motion between two surfaces.

Stiction is a colloquial term for static friction.

Backlash is a form of deadband that results from a temporary discontinuity between the input and the output of a device when the input of the device changes direction.

Hysteresis is the maximum difference in output value for any single input value during a calibration cycle, excluding errors due to deadband.

[1] We note that in this context we do not need to consider friction between fluids or between a fluid and a solid.

Deadband is a range through which an input can be varied, upon reversal of direction, without initiating observable response in the output. Deadband is typically expressed as a percentage of the input span.

The above definitions do not provide a very good distinction between *backlash* and *deadband*, and in the following these terms will be used interchangeably.

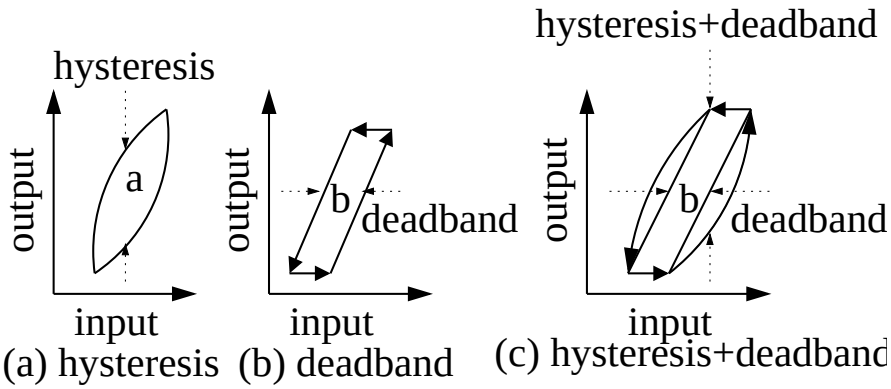


Fig. 2.5 Valve Nonlinearities – Hysteresis and deadband

2.4.2 Stiction Phenomenon in a Control Valve

A control valve consists of two main parts: a valve and an actuator that forces the stem to move. Additionally, it may contain a positioner, which is actually an embedded controller that controls the valve stem so that its position corresponds to the control signal. Fig. 2.6 displays a simple schematic of a control valve with a pneumatic actuator. The following discussion will assume the actuator to be of the pneumatic type, although the same stiction phenomena may occur with hydraulic or electrical actuators.

In process operation, a control valve is subjected to the following forces: the valve stem driving force caused by air pressure, spring force associated with the valve travel, seal-friction of the seals sealing the process fluid, and stem thrust originating in the process fluid passing through the valve body. Stiction in control valves is thought to occur due to seal degradation, lubricant depletion, inclusion of foreign matter, activation at metal sliding surfaces at high temperatures, and/or tight packing around the stem. The resistance offered from the stem packing is often cited as the main cause of stiction, although the stiction may arise wherever solid surfaces are in contact.

One very common cause of stiction is indirectly due to regulations on volatile organic compound (VOC) emissions. In many plants, a team monitors each valve periodically for VOC emissions, usually between the packing and the stem. If any minute leakage is detected, the packing in the valve body is tightened, often far more than is necessary. This causes the valve to stick, resulting in poor control performance and thereby degrading overall plant operation. Stiction often varies over time and operating regimes. Since wear is also nonuniform along the body, frictional forces are different at different stem positions.

2.5 Modelling of Valve Stiction

In this section, a few simple models for valve stiction are presented. The first of these models is motivated by physical considerations, and since 90% of actuators are air-operated, a pneumatic configuration is considered here. However, to modify the physical model for other types of actuators is very simple – only requiring the modification of how the actuator forces are calculated (*i.e.*, modifying F_a and possibly F_r below).

Following the presentation of the physical model, empirical models using one or two parameters are described. These models make no attempt at explaining the physical phenomena observed, but focus on reproducing the effects of stiction on the commonly logged signals in the control loop, i.e., the measured process variable (PV) and the controller output (OP).

2.5.1 Physical Model

The general structure of a pneumatic control valve is illustrated in Fig. 2.6. In the case illustrated by the figure, the valve is closed by the elastic force of the spring and opened by air pressure. Flow rate is changed according to the plug position, which is determined by the balance of the forces acting on the valve. The plug is connected to the valve stem. The stem is moved against static or dynamic frictional force caused by packing, which is a seal used to prevent leakage of process fluid. Smooth movement of the stem is impeded by excessive static friction. The valve position cannot be changed until the controller output overcomes static friction. The dynamic friction is often considerably smaller than the static friction. When the difference between elastic force and air pressure exceeds the maximum static friction force, the valve stem will start to move. The friction force is then reduced from the static friction level to the level of the dynamic friction, causing a large reduction in the force opposing the movement. This causes the valve stem to suddenly ‘jump’ to another position.

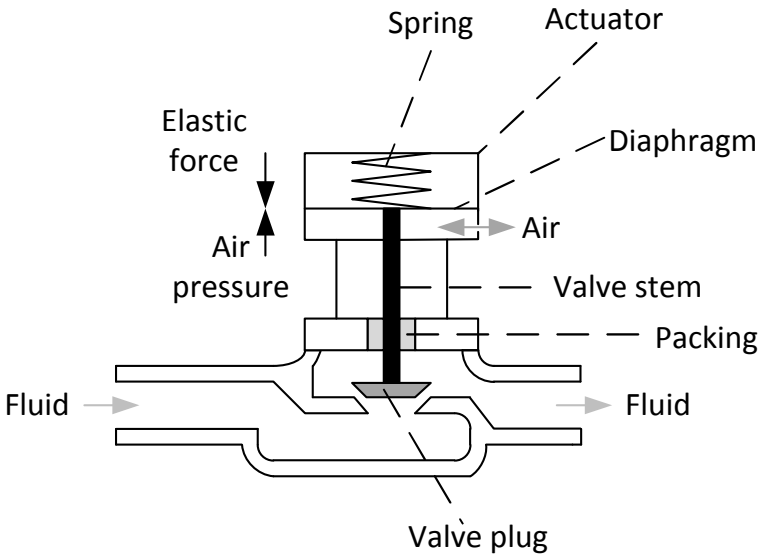


Fig. 2.6 Structure of a pneumatic control valve

Mathematically, for a sliding stem valve, the force balance equation based on Newton's second law can be written as

$$M\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = \sum \text{Forces} = F_a + F_r + F_f + F_p + F_i \tag{2.7}$$

where M is the mass of the moving parts, x is the relative stem position, $F_a = Au$ is the force applied by pneumatic actuator where A is the area of the diaphragm and u is the actuator air pressure or the valve input signal, $F_r = -kx$ is the spring force where k is the spring constant, $F_p = -\alpha\Delta P$ is the force due to fluid pressure drop where α is the plug unbalance area and ΔP is the fluid pressure drop across the valve, and F_f is the friction force. F_i and F_p are assumed to be zero because of their negligible contribution to the model Kayihan and Doyle III (2000). We will use a simple friction model from Olsson (1996):

$$F_f = \begin{cases} -F_c\text{sgn}(v) - vF_v & \text{if } v \neq 0 \\ -(F_a + F_r) & \text{if } |F_a + F_r| \leq F_s \text{ and } v = 0 \\ -F_s\text{sgn}(F_a + F_r) & \text{if } |F_a + F_r| > F_s \text{ and } v = 0 \end{cases} \tag{2.8}$$

Here v is the velocity of the stem movement. This friction model includes static and dynamic friction. The expression for the dynamic friction is in the first line of Eq. 2.8 and comprises a velocity-independent term F_c known as Coulomb friction and a viscous friction term vF_v that depends linearly upon velocity. Both act in opposition to the velocity, as shown by the negative

signs. The second line in Eq. 2.8 is the case when the valve is stuck. F_s is the maximum static friction. The velocity of the stuck valve is zero and not changing, therefore the acceleration is also zero. The third line of the model represents the situation at the instant of break-away. At that instant, the sum of forces is $(F_a + F_r) - F_s\text{sgn}(F_a + F_r)$, which is not zero if $|(F_a + F_r)| > F_s$. Therefore, the acceleration becomes non-zero and the valve starts to move.

The simple friction model above is able to create stick-slip behavior in simulations of control loops. However, due to numerical issues, it is in simulations necessary to apply the static friction in a small 'deadband' around $v = 0$. That is, in the first line of Eq. 2.8, use $\text{abs}(v) > \delta$ instead of $v \neq 0$, and in the second and third rows use $\text{abs}(v) < \delta$ instead of $v = 0$.

There also exists more sophisticated models of static and dynamic friction, the interested reader is referred to Olsson (1996). Despite its relative simplicity, a disadvantage of the physical model (2.7-2.8) is that it is practically impossible to get information on all the parameters of all the sticky valves used in a typical chemical plant (a problem that would clearly be exacerbated by the use of more sophisticated models). Hence, simple empirical models are preferred to first-principle models in modelling valve stiction. Such empirical models are often called "parametric" models, as they use a low number of parameters to describe the effects of valve stiction. The famous ones are one parameter model and two parameter model which are explained below.

2.5.2 Two Parameter Model

To model the relationship between the controller output and the valve position of a pneumatic control valve, the balance among elastic force, air pressure, and frictional force needs to be taken into account. The relationship can be described (Kano et al, 2004) as shown in Fig. 2.7. The dashed line (diagonal) denotes the states where elastic force and air pressure are balanced. The controller output and the valve position change along this line in an ideal situation without any friction. The ideal relationship is disturbed when friction arises. For example, the valve is resting at (a) where elastic force and air pressure are balanced. The valve position cannot be changed due to static friction even if the controller output, *i.e.*, air pressure, is increased. The valve begins to open at (b) where the difference between air pressure and elastic force exceeds the maximum static frictional force. Since the frictional force changes from static f_S to kinetic f_D when the valve starts to move at (b), a slip-jump of the size

$$J = f_S - f_D \tag{2.9}$$

happens and the valve state changes from (b) to (c). Thereafter, the valve state changes along the line l_2 which deviates from the ideal line by f_D because the difference between air pressure and elastic force is equal to f_D.

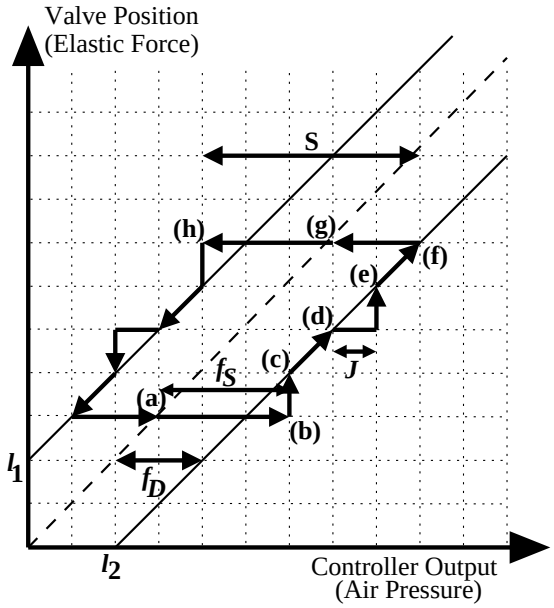


Fig. 2.7 Two parameter valve stiction model: relationship between controller output and valve position under valve stiction.

When the valve stops at (d), the difference between air pressure and elastic force needs to exceed f_S again for the valve to open further. Since the difference between them is f_D at (d), air pressure must increase by J to open the valve. Once air pressure exceeds elastic force by f_D, the valve state changes to (e) and then follows l_2.

Air pressure begins to decrease when the controller orders the valve to close at (f). At this moment, the valve changes its direction and comes to rest momentarily. The valve position does not change until the difference between elastic force and air pressure exceeds the maximum static frictional force f_S. The valve state (h) is just point-symmetric to (b). The difference of air pressure between (f) and (h) is given by

$$S = f_S + f_D \tag{2.10}$$

The valve state follows the line l_1 while the valve position decreases. The above-mentioned phenomena can be modelled as a flowchart shown in Fig. 2.7 (b). The input and output of this valve stiction model are the controller output u and the valve position y, respectively. Here, the controller output is transformed to the range corresponding to the valve position in advance. The first two branches check if the upper and the lower bounds of the controller output are satisfied. In this model, two states of the valve are explicitly

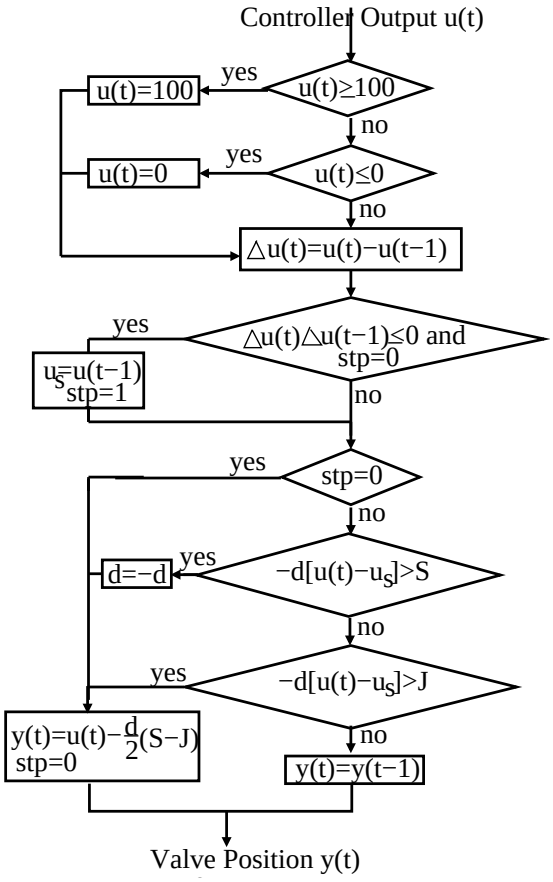


Fig. 2.8 Flowchart of two parameter valve stiction model.

distinguished: 1) a moving state ($stp = 0$), and 2) a resting state ($stp = 1$). In addition, the controller output at the moment the valve state changes from moving to resting is defined as u_S. u_S is updated and the state is changed to the resting state ($stp = 1$) only when the valve stops or changes its direction ($\Delta u(t)\Delta u(t-1) \leq 0$) while its state is moving ($stp = 0$). Then, the following two conditions concerning the difference between $u(t)$ and u_S are checked unless the valve is in a moving state. The first condition judges whether the valve changes its direction and overcomes the maximum static friction (corresponding to (b) and (h) in Fig. 2.7). Here, $d = \pm 1$ denotes the direction of frictional force. The second condition judges whether the valve moves in the same direction and overcomes friction. If one of these two conditions is satisfied or the valve is in a moving state, the valve position is updated via the following equation.

$$y(t) = u(t) - d \; f_D = u(t) - \frac{d}{2}(S - J) \tag{2.11}$$

On the other hand, the valve position is unchanged if the valve remains in a resting state. This model just requires two parameters S and J to model valve stiction.

2.5.3 One Parameter Model

This is a simple model characterized by just one parameter "d" (Stenman et al, 2003).

$$x(t) = \begin{cases} x(t-1) \text{ if } & |u(t) - x(t-1)| \leq d \\ u(t) & \text{otherwise} \end{cases} \tag{2.12}$$

Here $x(t)$ and $x(t-1)$ are past and present stem movements, $u(t)$ is the present controller output and 'd' is the valve stiction band. The stem movement of a sticky valve for a ramp input using this model is shown in Fig. 2.9.

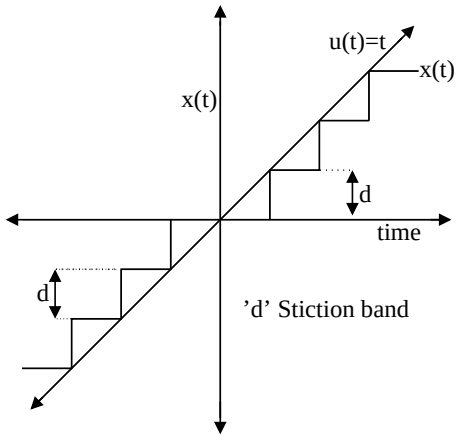


Fig. 2.9 Simulated stiction non-linearity for a ramp input using a one parameter model

$'d'$ is expressed in terms of the percentage or fraction of valve movement corresponding to the amount of stiction present in the valve. For instance, if 100 units of force are required to open the valve completely from completely closed position and 10 units of force is required to overcome the amount of static friction in the valve, stiction band is 10% or 0.1.

2.5.4 Discussion on Various Stiction Models

The physical model (including its more advanced cousins) has a clear disadvantage that it required the knowledge of several physical parameters that are not readily available, both design parameters of the equipment (mass, spring constant, diaphragm area) and parameters describing the static and dynamic friction. Naturally, the friction parameters may vary with time.

It should be noted that the two parameter model is developed specifically to study the consequences of deadband and stiction in control valves – it is not intended as a general friction model. Parameters of the two-parameter empirical stiction model can be related directly to plant data, and also such a model produces the similar open- and closed-loop behaviour as the physical model. The model only requires the controller output signal and the specification of deadband plus stickband (S) and slip-jump (J). It overcomes the main disadvantages of physical modelling of a control valve, i.e, it does not require the knowledge of the mass of the moving parts of the actuator, spring constant and the friction forces. The parameters of the two-parameter model are easy to choose and the effects of parameter changes on loop behaviour are easy to understand.

The basic difference between the two parameter model and one parameter model is that the force to overcome the deadband f_D in the one parameter model is assumed to be negligible. If f_D is taken to be zero, Eq. 2.9 and Eq. 2.10 imply that $S = J = f_S$. If $S = J$, Fig. 2.7 takes the shape of Fig. 2.9 which is the characteristic of one parameter model. Looking at Fig. 2.9 it can be concluded that it looks more like a quantizer than stick-slip behaviour, since intermediate values on the $x(t)$ axis cannot be achieved.

2.6 Diagnosis of Valve Stiction

A number of researchers have studied the valve stiction problem and suggested methods for detecting it. Horch and Isaksson (1998a) presented a fairly complex method for detecting stiction by calculating log-likelihood ratios for multiple models. Their method requires knowledge of the nonlinear plant and stiction models and extended Kalman filtering. Stenman et al (2003) also proposed a complicated method based on "multi-model mode estimation" and change detection. Apart from the method's conceptual complexity, the more significant drawback with this method is the use of the one-parameter stiction model, which was argued above to be too simplistic.

In the following, we will briefly explain some simple stiction detection methods that only require the use of routine operating data.

2.6.1 Shape-based Stiction Detection

In addition to causing the "slip jump", the detection of which is the basis of most stiction detection methods, stiction is also a major cause for deadband in the relationship between controller output (OP) and manipulated variable (MV) Fisher (1999).

Plots of OP vs. PV are used in formulating the shape based stiction detection algorithms. When a deadband is present (here assumed to be caused by valve stiction), the shape between OP and MV turns out to be a parallelogram. The shape-based methods predominantly use only routine operation data for detecting stiction.

There are three different methods (Kano et al, 2004; Hiroshi et al, 2004; Yamashita, 2006) in literature to detect valve stiction in control loops using OP-PV plots. They are all based on the presence of the following characteristics: (1) There are sections where the valve position does not change even though OP changes. Here, stiction is stronger as such sections are longer. (2) The relationship between OP and MV takes the shape of a parallelogram if slip jump is neglected. Stiction is stronger as the distance between two ends of the parallelogram is longer.

The general advantages of shape-based stiction-detection methods are: (1) they can quantify the stiction, and (2) they are applicable also to situations without periodic oscillation.

These methods require OP data and MV data to find suspicious movement of valves, because they aim to find particular shapes or relationships between OP and MV from their data. Such particular shapes or relationships represent a mismatch between OP and MV signals. If a fast flow measurement is available, flowrate can be used as MV when valve position data are not available. The difficulty associated with this is (i) noise, and (ii) the flow loop has dynamics that can distort the shape of the stiction pattern.

2.6.2 Method Based on Cross-correlation Function

In general, the OP-MV plot is a straight line at 45° for a healthy linear valve (assuming valve position dynamics are fast compared to the changes in the OP value), and any deviations such as deadband can be diagnosed by visual inspection. Automated analysis of the OP-MV plot can be problematical, however, due to the presence of noise, varying set point and the difficulty of maintaining a data base of all possible patterns for a match. In practice, the flow through the control valve is frequently not measured unless it is in a flow control loop. Similarly, the position, while it may be measured on a modern valve with a positioner, is quite often not available in the data historian.

The challenge in analysis of valve problems, then, is to determine and quantify the type of fault present using OP and PV data only. The major

difficulty is that the process dynamics (integration in the case of a level loop) greatly interfere with the analysis. Stiction in a loop with an integrating process can be detected by examination of the probability density function of the pv signal or of its derivatives.

Horch (1999) has developed a method for detecting stiction, based on measurements of the controlled variable and the controller output. The method assumes that the controller has integral action, which is typically required in order for the presence of stiction to cause sustained oscillations – as explained above.

Horch found that the cross-correlation function between controller output and controlled variable typically is an odd function[2] for a system oscillating due to stiction. On the other hand, if the oscillation is due to external disturbances, the cross-correlation function is normally close to an even function. Unstable loops oscillating with constant amplitude (due to input saturation) also have an even cross-correlation function.

For a data set with N data points, the cross-correlation function between u and y for lag τ (where τ is an integer) is given by

$$r_{uy}(\tau) = \frac{\sum_{k=k_0}^{k_1} u(k)y(k+\tau)}{\sum_{k=1}^{N} u(k)y(k)} \tag{2.13}$$

where

$$\begin{aligned} k_0 &= 1 \text{ for } \tau \geq 0 \\ k_0 &= \tau + 1 \text{ for } \tau < 0 \\ k_1 &= N - \tau \text{ for } \tau \geq 0 \\ k_1 &= N \text{ for } \tau < 0 \end{aligned}$$

Note that the denominator in Eq. (2.13) is merely a normalization, giving $r_{uy}(0) = 1$. It is not necessary for the stiction detection method.

Horch's stiction detection method has been found to work well in many cases. However, it fails to detect stiction in cases where the dominant time constant of the (open loop) process is large compared to the observed period of oscillation. In such cases the cross-correlation function will be approximately even also for cases with stiction. This problem is most common with integrating processes (*e.g.*, level control loops), but may also occur for other processes with slow dynamics.

[2] Reflecting the 90° phase shift due to the integral action in the controller.

2.6.3 Stiction Detection Based on Curve Fitting

The curve-fitting method (He et al, 2007) is based on qualitative analysis of the control signals *i.e.*, in the case of control-loop oscillations caused by controller tuning or external oscillating disturbances, the OP and PV typically follow sinusoidal waves for both self-regulating and integrating processes. In the case of stiction, the valve-position signal usually takes the form of a rectangular wave. Because the valve position signal is usually unmeasured, instead of looking at the valve position signal, the measured output of the first integrating element after the valve is examined, which is either OP or PV. The integrating element converts the rectangular valve position moves into a triangular wave. For self-regulating processes, the PI-controller acts as the first integrator and the OP's move follows a triangular wave, whereas for integrating processes such as level control, the integrator in the process integrates the rectangular waves and the PV signal follows a triangular wave.

The above analysis answers the questions of which signal to look after and why it takes a triangular shape in the presence of valve stiction. The basic idea of the new detection method is to fit two different functions, triangular wave and sinusoidal wave, to the measured oscillating signal of the first control-loop component containing an integrator after the valve (*i.e.* OP for self-regulating processes or PV for integrating processes). The data set is first divided into segments according to zero crossings, and for each segment a half-period of a triangular and a sinusoidal wave is fitted. A better fit to a triangular wave indicates valve stiction, while a better fit to a sinusoidal wave indicates the absence of stiction.

There are several advantages associated with the developed curve-fit method. One advantage is that it is applicable to both self-regulating and integrating processes, because for both type of processes, the same idea applies, *i.e.* after one integration, the rectangular wave (valve position) becomes a triangular wave, while the only difference is where the first integration component after the valve is located in the control loop. Another advantage is its industrial practicability due to the following reasons: (i) the methodology is straightforward and easy to implement. Fundamentally, the curve fit is a simple least-squares regression problem, which also provides its robustness against noise and outliers, and (ii) the detection is easily automated and does not require user interaction, and (iii) it can handle asymmetric or damped oscillations, as well as intermittent oscillations.

Clearly, measurement noise can lead to a number of 'spurious' zero crossings, and some thought is therefore required to make the method for detecting zeros crossings robust. Furthermore, the method assumes that the ('pure') integration is the only dynamics affecting the data set used. For cases where the oscillations are in the frequency range of other dynamic elements in the loop – which may easily happen if a valve positioner is used – the response after the first integrating element in the loop may well differ from the ideal-

ized triangular ('sawtooth') wave. The reliability of the curve fitting method in this case is not known.

2.6.4 Stiction Detection using an OP-PV Plot

The method involves plotting the controller output (OP, manipulated variable) vs. the controlled variable (PV). If these two variables tend to move in a closed path around an area where the curve seldom enters, this is a sign of an oscillating control loop, where there is a phase lag (different from $n \cdot 180°$) between input and output. If the OP-PV plot shows sharp 'corners', this is considered to be an indication of significant stiction. Without the sharp corners, there is no cause for suspecting non-linearity (*i.e.*, stiction) to be the cause of the oscillations, since they may just as well be caused by poor tuning and random noise or oscillating disturbances. The use of an OP-PV plot is illustrated in Fig. 2.10, where the blue curve shows a case with stiction, and the red curve shows the same system without stiction. The use of this method is apparently widespread in industrial practice, although its origin is not known to these authors. In the example illustrated in Fig. 2.10, this method would correctly identify stiction in a case with some measurement noise.

However, numerical experience and intuition would suggest that this method may fail in cases with severe measurement noise, especially when there is a phase difference of close to $n \cdot 180°$ at the dominant frequency of oscillation. Filtering may reduce the sensitivity to noise, but may also reduce the sharp corners in the OP-PV curve that are necessary to distinguish stiction from other causes of oscillation (which may occur also for linear systems).

In some cases, the problems related to interpreting OP-PV plots may be reduced by introducing a time shift between the OP and PV time series. This is illustrated in Figures 2.11 and 2.12. In the left part of Fig. 2.11 we see an OP-PV plot for a linear system with an oscillatory disturbance (and some measurement noise). The OP-PV plot has an elliptical shape, but since the disturbance is at a low frequency there is very little phase shift between the OP and PV time series – which means that we are looking at the ellipse 'from the side'. The result is that it is hard to conclude whether there are sharp corners that would indicate nonlinear effects. In the right part of the figure, we se the same plot – but this time with the PV time series shifted by 100 samples. The elliptical shape of the plot is now clear (the shifting of the time series has 'turned the ellipsoid to face us').

In Fig. 2.12, we have OP-PV plots for a system with oscillations due to stiction. Again, in the left part of the figure, it is hard to conclude whether there are sharp corners in the plot that indicate nonlinearity. In the right

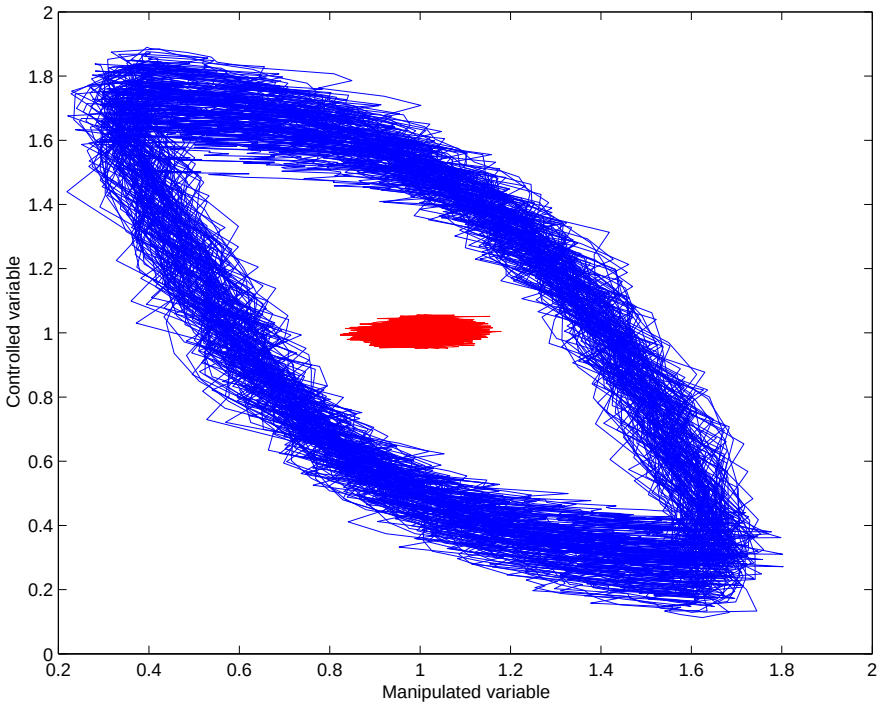


Fig. 2.10 Use of OP-PV-plot to detect stiction. The blue curve shows a system with stiction, the red curve shows the same system without stiction.

part of the figure, we have again shifted the PV time series, and the sharp corners are now evident, indicating nonlinearity (*i.e.*, stiction).

2.6.5 Stiction Detection using Higher Order Statistics

Using higher order statistical methods to detect the presence of nonlinearity has been used for almost three decades (Hinich, 1982). A method based on higher order statistics (HOS), which can detect whether a time series is nonlinear or not, has been developed in Choudhury et al (2004b). A HOS measure, *bispectrum*, is used to detect non-linearity. The bispectrum measures interaction between two frequencies, and is defined as

$$B(f_1, f_2) = E\left[X(f_1)X(f_2)X^*(f_1 + f_2)\right] \tag{2.14}$$

where $B(f_1, f_2)$ is the bispectrum at the frequency pair $f_1, f_2)$, $X(f)$ is the discrete Fourier transform of the time series $x(k)$, $'*'$ denotes the complex

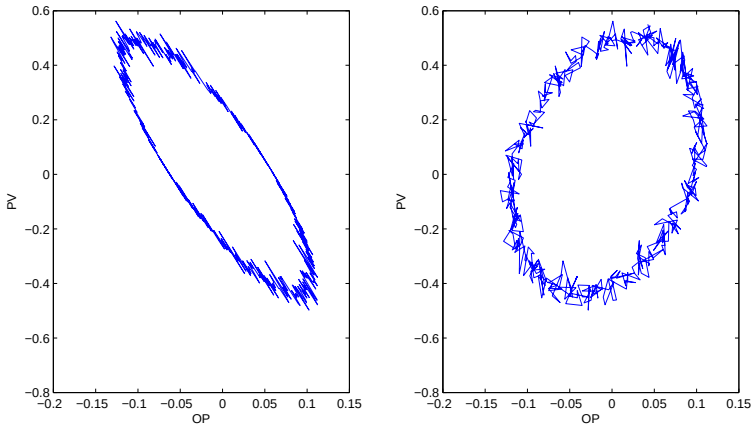


Fig. 2.11 OP-PV-plot for a linear system with oscillatory disturbance. Left: original OP-PV plot. Right: OP-PV plot, with PV time series shifted in time.

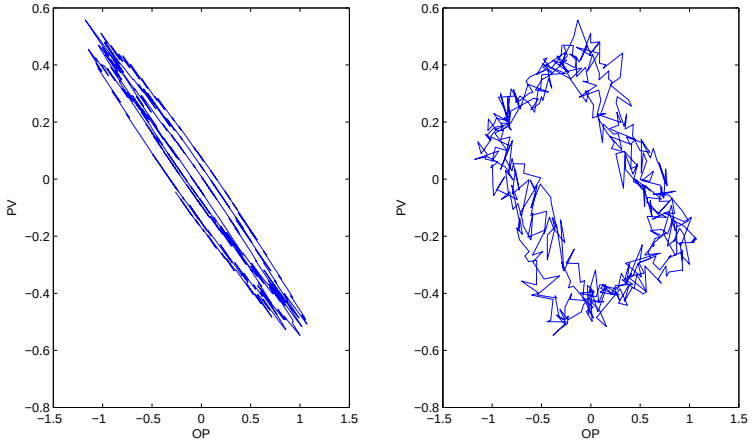


Fig. 2.12 OP-PV-plot for a system with stiction with significant phase shift between the OP and PV time series at the oscillation frequency. Left: original OP-PV plot. Right: OP-PV plot, with PV time series shifted in time.

conjugate, and E is the expectation operator. In practice, the expectation operation is approximated by calculating the Fourier transform for a number of segments of a long data series, and averaging over these transforms. For more detail of the data treatment and signal processing, consult Choudhury et al (2004a) and the references therein. It is clear from (2.14) that $B(f_1, f_2)$

can be plotted in a 3D plot with two frequency axes and the corresponding value of the bispectrum (real part, imaginary part, or absolute value) on the third axis.

In order to simplify interpretation, the bispectrum can be normalized to be real valued and between 0 and 1, resulting in the socalled *bicoherence* function $\text{bic}(f_1, f_2)$:

$$\text{bic}^2(f_1, f_2) = \frac{|B(f_1, f_2)|^2}{E\left[|X(f_1)X(f2)|^2\right] E\left[|X(f_1 + f_2)|^2\right]} \tag{2.15}$$

The bicoherence is expected to be flat for a linear signal. Significant peaks and troughs in the bicoherence is therefore an indication of non-linearity. A discrete ergodic[3] time series $x(k)$ is called linear if it can be represented by a random variable $e(k)$ passed through finite impulse response dynamics h, that is:

$$x(k) = \sum_{i=0}^{n} h(i)e(k-i) \tag{2.16}$$

where the random variable $e(k)$ is independent and identically distributed. In Choudhury et al (2004a) it is shown that if $e(k)$ has zero mean and a Gaussian (normal) distribution, then the bicoherence function is exactly zero. The authors of Choudhury et al (2004a) therefore propose a 'Non-Gaussianity Index' NGI based on a statistical test of whether the bicoherence is significantly different from zero, and a 'Non-linearity index' NLI based on whether the squared maximum of the bicoherence deviates much from the mean value of the squared bicoherence. Theoretically, NGI > 0 should indicate a non-gaussian signal, and NLI > 0 should indicate a non-linear signal. In practical implementation it is recommended to set the thresholds a little higher, with NGI > 0.001 indicating a non-Gaussian signal, and NLI > 0.01 indicating a non-linear signal.

It is recommended to use the NGI first and then use the NLI only for signals that have been found to be non-Gaussian. If *both* the NGI and NLI exceed their thresholds, one should look for a non-linear cause of the poor performance, *e.g.*, valve stiction or backlash, or other non-linear phenomena. Otherwise, the cause for the poor performance is likely to be "linear", *e.g.*, a linear external disturbance or an excessively tightly tuned controller.

[3] Roughly speaking, a time series is called *ergodic* if the time average of the signal value over a significant segment of the time series can be expected to be the same irrespective of where in the time series the segment is located.

2.6.6 Stiction Detection using Hammerstein Model Based Approach

Here, the fundamental idea is to convert the stiction-detection and quantification problem into a low-order Hammerstein-type system-identification problem, followed by a global optimisation search for the stiction parameters.

This idea focuses on finding a noninvasive method to determine if there exists the presence of stiction in a control valve. It approximates process dynamics by a low-order transfer-function model while estimating parameters for the static stiction model to account for the non-linearity induced by the stiction. It is an underlying assumption here to consider that most industrial processes can be approximated as the first- or second-order-plus-time-delay process.

It is necessary to select an appropriate stiction-model before proceeding. The basic steps to follow in any Hammerstein approach are:

1. Given a stiction-model structure and OP data, effectively bound a search space of unknown stiction-model parameters.
2. Choose stiction-model parameters from the bounded stiction-model space, and a series of manipulated variable (MV) data is calculated from controller output data according to the given valve-stiction model.
3. With MV and PV data, the process model is identified such that a MSE is minimised. By varying stiction-model parameters, different process models are obtained.
4. Find the stiction model that describes the characteristics of the control valve behaviour the best. Find the minimum model error and get the corresponding process-model and stiction-model parameters.

In (Srinivasan et al, 2005), Hammerstein model-based approach has been proposed for linear processes for quantifying valve stiction through a joint identification procedure. In this approach, identification of linear plant dynamics is decoupled from the nonlinear element which is achieved by an iterative procedure. A similar approach using the two-parameter model presented above to quantify stiction is discussed in Choudhury et al (2008). Another work using a Hammerstein-based identification approach with the two-parameter model can be found in Jelali (2008). The difference between these two approaches seem to be mainly in how the optimization is performed in order to identify the stiction model parameters.

2.6.7 Stiction Compensation

There are a number of papers looking at using the controller to compensate for stiction, not only in process control, but also in other areas like robotics. There

are many models for stiction – that all share the common trait that none of them can be expected to be a perfect representation of the phenomenon.

The compensation schemes are typically rather complex, finely tuned to the specifics of the stiction model used, and not very surprisingly they often work well for the same stiction model. What is lacking is the demonstration of any sort of robustness for the compensation scheme. In a simulation study one could at least use a different model for the 'system' than the stiction model used in designing the controller. The practical usefulness of such stiction compensation schemes are therefore at best not proven.

Industrial practitioners report that use of derivative action often has some positive effect on stiction. However, derivative control action may not be suitable for all control loops, and there is also the question whether it should be placed in the main controller or in the valve positioner. Some further work in this area may therefore be warranted.

Other practical approaches to managing control problems due to stiction, include changing the controller to a pure P controller, or introducing a deadband in the integrating term (only integrate when the offset is larger than the deadband). This may reduce or remove the oscillations, but have their own detrimental effects on control performance. These approaches are therefore mainly short-term modifications until valve maintenance can be performed.

2.6.8 Detection of Backlash

It was noted above that the terms *backlash* and *deadband* are used interchangeably. Although stiction is an important cause for backlash, this section focuses on the case when the deadband is the dominant valve non-linearity, and there is little or no slip-jump. The same situation was addressed in subsection 2.6.1, where it was assumed that the actual valve position is available. The method in this section addresses the problem when the valve position is not available, and the backlash detection must be based on the controlled variable (PV).

In a recent paper, Hägglund (2007) proposes a method for on-line estimation of the deadband. Using describing function analysis, it is shown that an integrating system controlled with an integrating controller will exhibit oscillations in the presence of backlash. These oscillations are typically quite fast and of significant amplitude, and will therefore be detected by an appropriate oscillation detection method.

Asymptotically stable processes with integrating controllers, on the other hand, will typically not show pronounced oscillations, but rather drift relatively slowly around the setpoint. This results in slow, low amplitude oscillations that often will not be detected by oscillation detection methods. Hägglund's deadband estimation method is developed for this kind of systems. It uses the control loop measurement, filtered by a second order low

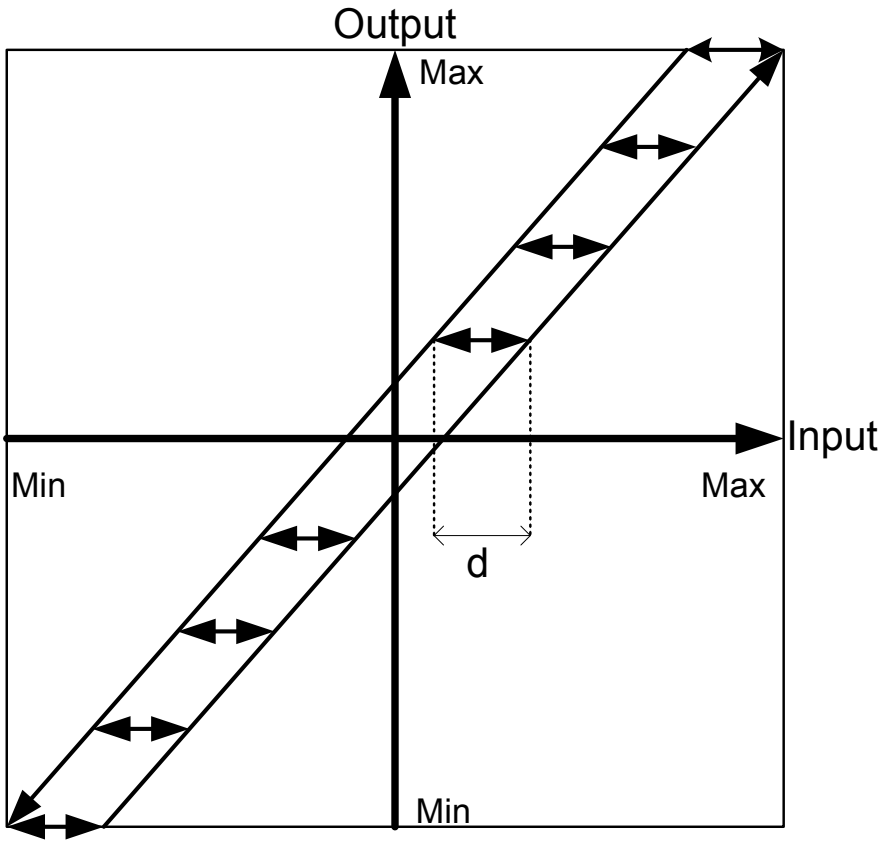


Fig. 2.13 Illustration of backlash with deadband of width d.

pass filter to reduce the effect of measurement noise. The filtered loop measurement is denoted y_f. The slow oscillations are typically at a frequency lower than the plant dynamics, and hence the plant model is represented by the steady state gain K_p. The controller is assumed to be a PI controller with proportional gain K and integral time T_i. The plant gain K_p and the controller gain K are assumed to be given in compatible units (such that their product is dimensionless).

The filtered control error is given as $e = y_{sp} - y_f$, where y_{sp} is the setpoint (or reference) for the control loop. Let t_i be the times when the filtered control error e changes sign. Correspondingly, $\Delta t = t_{i+1} - t_i$ denotes the time between successive zero crossings of the filtered control error. The deadband estimation is executed only when the time between these zero crossings is large, *i.e.*, when $\Delta t \geq 5T_i$. We also define

$$\Delta y = \int_{t_i}^{t_{i+1}} |e| \mathrm{d}t / \Delta t \tag{2.17}$$

Δy may thus be seen as the 'average' control error between the zero crossings. The deadband is then estimated as

$$\hat{d} = K \left(\frac{\Delta t}{T_i} - \frac{1}{K K_p} \right) \Delta y \tag{2.18}$$

This deadband estimation suffers from the fact that the steady state gain needs to be known. In many cases this will be available (although not necessarily easily available) from steady state plant simulations – even if dynamic simulation models are not available. Instead, Hägglund takes a more practical approach and argue that the deadband estimate is relatively insensitive to the value of K_p for the majority of plants. This stems from the fact that the estimation is performed only when $\Delta t \geq 5T_i$, and the observation that the product KK_p is normally larger than 0.5 (assuming a reasonable controller tuning in the absence of backlash, and that the controller tuning is not dominated by pure time delay).

For more details of implementation of the deadband estimation, the reader is referred to the original publication by Hägglund (2007).

2.6.9 Backlash Compensation

It is possible to compensate for backlash by adding an extra term to the calculation of the manipulated variable

$$u = u_{FB} + u_{BC} \tag{2.19}$$

where u_{FB} is the ordinary controller output[4], and u_{BC} is an additional term added to compensate for backlash. The ideal backlash compensation would be

$$u_{BC} = \frac{d}{2} \mathrm{sgn} \left(\frac{\mathrm{d}u_{FB}}{\mathrm{d}t} \right) \tag{2.20}$$

Due to noise this ideal compensation is impractical, and some filtering is necessary. Hägglund (2007) proposes using the filtered control error e introduced in the subsection above, resulting in the backlash compensation

$$u_{BC} = \frac{\delta}{2} \mathrm{sgn}(e) \tag{2.21}$$

[4] the subscript FB implies the use of a feedback controller, but u_{FB} may also include disturbance feedforward components.

where $\delta \leq \hat{d}$. The motivation for basing the compensation on the filtered control error is that sign changes in this term corresponds to changes in the derivative of the integral term of the controller. The integral terms is less sensitive to noise than the proportional and derivative terms.

The use of filtered signals for backlash compensation introduces a delay in detecting the sign changes of the derivative of the manipulated variable, and this is further aggravated by considering only the integral term of the controller. Therefore the δ used in the backlash compensation should be somewhat reduced compared to the deadband d.

2.7 Benchmarking and Performance Measures

To understand how well complex processes are being managed, it is necessary to monitor and analyze a representative range of performance metrics. The specific type of metrics will be process dependent but to capture the state of a process, careful selection of performance indicators is important. A common classification is into Financial and Non-Financial performance measures. Typical examples of financial performance measures are profitability, sales, unit costs whilst non-financial indicators might include employee retention rates, customer satisfaction levels, and product defect rates. A second classification is the quantitative-qualitative divide into "hard" and "soft" performance measures. Hard performance metrics are those strictly computable quantities based on numerical measurable data; these range from financial measures like unit cost and sales per day to non-financial quantities and technical measures like process plant downtime, product defect rates and product physical properties (temperature, flow, dimensions, etc). In contrast "soft" performance indices are metrics of more difficult to measure quantities like customer satisfaction variables and are often captured by a set of linguistic variables such as very poor, poor, satisfactory, good, excellent.

2.7.1 Control Loop Performance Benchmarking

The performance of a control system relates to its ability to deal with the deviations between controlled variables and their set-points (or desired values). For benchmarking, the severity of these deviations should be quantified by a single number, the performance index (indicator/potential/measure/metric). In this section, we will first introduce some traditional performance measures that have been used for single-loop performance assessment in the case of frequent deterministic disturbances. Although these performance measures reflect (different aspects of) control performance, they lack a clear 'standard' against which the performance can be compared. This problem is avoided

by using *minimum (achievable) variance* as a performance measure, and this will be explained in the subsequent subsection. The final parts of this section will consider more advanced performance measures, and briefly address performance assessment for multivariable systems.

2.7.1.1 Univariate Performance Measures

For every process control application, there are

- Steady-state performance criteria.
- Dynamic response performance criteria

The principal steady-state performance criterion usually is zero error at steady state. It is well known that this performance criterion requires the use of integral,action, since a proportional controller can not achieve zero steady-state error. The evaluation of the dynamic performance of a closed-loop system is based on two types of commonly used criteria.

1. Criteria that use only a few points of the response.
2. Criteria that use the entire closed-loop response from time $t = 0$ until $t =$ very large.

Simple Performance Criteria

Several simple performance criteria are based on some characteristic features of the closed-loop response of a system to a step in the setpoint. The most often quoted are

- Overshoot – the maximum amount by which the response exceeds the new setpoint, divided by the magnitude of the setpoint change.
- Rise time – time needed for the response to reach 90% of its final value.
- Settling time – time needed for the response to settle within $\pm 5\%$ of the final value.
- Decay ratio – the ratio of the magnitudes of the second and first peak of the response, measured relative to the final value.

Each of the characteristics above could be used by a designer as a criterion for selecting the controller and to tune the controller parameters. Thus we could design the controller in order to have a specified overshoot, or specified settling time, and so on. It must be emphasized, though, that one simple characteristic does not suffice to describe the desired dynamic response.

Time-Integral Performance Criteria

These criteria are based on the entire response of the process. The most common are

- Integral of the square error (ISE), where

$$\text{ISE} = \int_0^\infty \epsilon^2(t)\mathrm{d}t$$

- Integral of the absolute of the error (IAE), where

$$\text{IAE} = \int_0^\infty |\epsilon(t)|\mathrm{d}t$$

- Integral of the time-weighted absolute error (ITAE), where

$$\text{ITAE} = \int_0^\infty t|\epsilon(t)|\mathrm{d}t$$

If we want to strongly suppress large errors, ISE is better than IAE because the errors are squared and thus contribute more to the value of the integral. For the suppression of small errors, IAE is better than ISE because when we square small numbers (less than one) they become even smaller. To suppress errors that persist for long times, the ITAE criterion be more appropriate because the presence of large t amplifies the effect of even small errors in the value of the integral.

The most popular benchmark for controller performance assessment, minimum variance controller was first introduced by Harris (1989a) for single loop feedback controllers. It provides a theoretical lower bound on the closed-loop process output variance. The calculation of the MVC assumes that the process can be represented adequately by a linear time-invariant (LTI) transfer function model with additive disturbances.

2.7.1.2 Minimum Variance Controller (MVC)

Minimum variance control is the best possible control in the sense that no controller can have a lower variance. Its implementation may not be desirable in practice because it may call for excessively aggressive control and may lack robustness to model errors. However, it provides a convenient 'hard' bound on achievable performance against which the performance of other controllers can be compared. Such a basis is especially important in deciding corrective steps. For instance, if the current performance were inadequate but were close to the minimum variance, corrective steps would have to be directed toward structural changes to the process. On the other hand, if the variance under the current control were substantially greater than the

calculated minimum, corrective steps could be directed toward improving the controller performance without making changes to the plant.

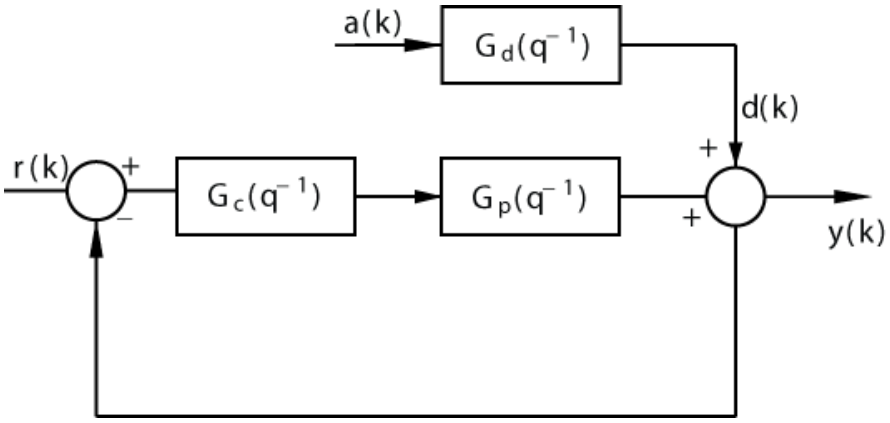


Fig. 2.14 Schematic diagram of a simple feedback control system.

Consider the block diagram of a feedback control system shown in Fig. 2.14. Assume that this single-loop system can be represented adequately by a linear time-invariant, discrete transfer function model and additive disturbance. The process transfer function is given by G_p, while G_c and G_d denotes the controller and disturbance transfer functions, respectively,

$$y(k) = G_p(q^{-1})u(k-b) + d(k) = \frac{\omega(q^{-1})}{\delta(q^{-1})}u(k-b) + d(k) \tag{2.22}$$

where q^{-1} is the backward shift operator and $y(k)$ and $u(k)$ are deviations of the measured process output and controller outputs, respectively, from their nominal operating values; $d(k)$ is a bounded disturbance; $\omega(q^{-1})$ and $\delta(q^{-1})$ are polynomials in the backward shift operator; and $b \geq 1$ is the number of whole periods of delay in the process. The term $d(k)$ is assumed to represent all unmeasured disturbances acting on $y(k)$ and it may be deterministic or stochastic. Let $d(k)$ be given as a linear function of past values of a statistically independent random sequence of variables, $\{a_{ij}\}$,

$$d(k) = G_d(q^{-1})a(k) = \frac{\theta(q^{-1})}{\phi(q^{-1})}a(k) \tag{2.23}$$

The terms $\theta(q^{-1})$ and $\phi(q^{-1})$ are assumed to be stable polynomials. It is also to be noted that the performance monitoring of unstable loops need not be required. For constant reference inputs, the deviations of the outputs from their steady-state values are given by

$$y(k) = \frac{\alpha(q^{-1})}{\beta(q^{-1})}a(k) = \Psi(q^{-1})a(k) \tag{2.24}$$

$$y(k) = [\psi_0 + \psi_1 q^{-1} + \cdots + \psi_b q^{-1} + \cdots +]a(k) \tag{2.25}$$

where Ψ_j is the jth impulse response coefficient from disturbance to measurement, when the control is active.. The series in Equation 2.25 is convergent if the closed loop between $y(k)$ and $d(k)$ is stable. Because of the delay term, $q^{-(b-1)}$, the first b terms in Equation 2.25 are identical to those computed from the disturbance transfer function $G_d(q^{-1})$ and can be interpreted as system invariant. Thus, only terms at lag b and beyond are affected by the current controller action. The reason for this is seen as follows. Once a disturbance appears at the output, it is feedback to the controller and the controller makes a correction. However, because of the delay, that corrective action has no effect on the output for b time intervals into the future. No disturbance compensation can occur at the output until the deadtime of the system has expired.

The variance of the controlled variable can be calculated by squaring Equation 2.25 and then applying the expectation operator $E\{.\}$,

$$\sigma_y^2 = E\{y(k)\} = [\psi_0^2 + \psi_1^2 + \cdots + \psi_b^2 + \psi_{b+1}^2 + \cdots +]\sigma_a^2 \tag{2.26}$$

where

$$\sigma_a^2 = E\{a(k)^2\} \tag{2.27}$$

Equations 2.25 and 2.26 show how the variance of the controlled variable is related to the variance of $\{a_{ij}\}$, the process dynamics, the disturbance model, and the controller – since ψ_k will depend on the controller for $k \geq b$. If the feedback controller is a MVC then the b-step ahead forecast (terms at and beyond b) equals zero and the output variance is given by

$$\sigma_{mv}^2 = [\psi_0^2 + \psi_1^2 + \cdots + \psi_{b-1}^2]\sigma_a^2 \tag{2.28}$$

This controller then rejects the predicted effect of the disturbance after the deadtime has elapsed. Thus, the controlled variable under minimum variance control will depend on only the most recent b past disturbances

$$y_{mv}(k) = [\psi_0 + \psi_1 q^{-1} + \cdots + \psi_b q^{-1}]a(k) \tag{2.29}$$

The finite stochastic process in equation 2.29 is called a moving average process of order b. It then follows that any controller that is not minimum variance must inflate the variance, that is

$$\sigma_y^2 = \sigma_{mv}^2 + \sigma_{\bar{y}}^2 \tag{2.30}$$

where $\sigma_{\bar{y}}^2$ is the increase in the output variance above the minimum obtainable variance. Note that in the derivation of the minimum variance above,

we have assumed that the plant G_p is stable and has a stable inverse. These assumptions are fulfilled for many control loops.

An important consideration in the calculation of the theoretical minimum variance, is that routine process data, with or without feedback control, are used rather than specially designed tests. The only requirement is that the number of observations is large and representative of the process. The ratio of the output variance to the theoretical variance under MVC is called the Harris index (HI) Harris (1989b) and given by

$$\mathrm{HI} = \frac{\sigma_y^2}{\sigma_{mv}^2} \tag{2.31}$$

From the Equation 2.31, it is clear that HI ≥ 1 and has no upper bound. When HI is significantly greater than one, further analysis must be done to ascertain the cause for the variance inflation. For practical purposes, the normalized Harris index (NHI) is defined as follows,

$$\mathrm{NHI} = 1 - \frac{\sigma_{mv}^2}{\sigma_y^2} = 1 - \frac{\psi_0^2 + \Psi_1^2 + \cdots + \Psi_{b-1}^2}{\psi_0^2 + \Psi_1^2 + \cdots + \Psi_{b-1}^2 + \cdots +} \tag{2.32}$$

This index represents the fractional increase in the variance of the output that arises from not implementing an MVC. Further, unlike HI, NHI is bounded to the interval [0, 1]. When NHI = 0, the controller is an MVC. The closer that NHI gets to one, the larger the variance of the process output, y; relative to its best possible performance, σ_{mv}^2.

Obtaining the Impulse Response Model

In order to identify a model for the effect of the unknown disturbance on the controlled variable, we must first select a model structure. We will here use an autoregressive (AR) model, where we assume that the disturbance a is a zero mean white noise:

$$y_k + \alpha_1 y_{k-1} + \alpha_2 y_{k-2} + \cdots = a_k$$

or, in terms of the *backwards shift operator* q^{-1}:

$$(1 + \alpha_1 q^{-1} + \alpha_2 q^{-2} + \alpha_3 q^{-3} + \cdots) y_k = A(q^{-1}) y_k = d_k$$

Now, the AR model is very simple, and one may therefore need a high order for the polynomial $A(z^{-1})$in order to obtain a reasonably good model. One therefore runs the risk of "fitting the noise" instead of modelling system dynamics. It is therefore necessary to use a data set that is much longer than the order of the polynomial $A(q^{-1})$. However, if a sufficiently large data set is used (in which there is significant variations in the controlled variable y), industrial experience indicate that acceptable models for the purpose of

control loop performance monitoring is often obtained when the order of the polynomial $A(q^{-1})$ is 15-20. The AR model has the advantage that a simple least squares calculation is all that is required for finding the model, and this calculation may even be performed recursively, *i.e.*, it is applicable for on-line implementation. We will here only consider off-line model identification. The expected value of the disturbance a is zero, and thus we have for a polynomial $A(q^{-1})$ of order p and a data set of length N with index k denoting the most recent sample

$$\begin{bmatrix} y_{k-1} & y_{k-2} & \cdots & y_{k-p+1} & y_{k-p} \\ y_{k-2} & y_{k-3} & \cdots & y_{k-p} & y_{k-p-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{k-N+p} & y_{k-N-1+p} & \cdots & y_{k-N+2} & y_{k-N+1} \\ y_{k-N-1+p} & y_{k-N-2+p} & \cdots & y_{k-N+1} & y_{k-N} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix}$$

$$= -\begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-N+p+1} \\ y_{k-N+p} \end{bmatrix} + \begin{bmatrix} a_k \\ a_{k-1} \\ \vdots \\ a_{k-N+p+1} \\ a_{k-N+p} \end{bmatrix}$$

$$\Updownarrow$$

$$Y\underline{\alpha} = -\underline{y} + \underline{a}$$

where the underbars are used to distinguish vector-valued variables from scalar elements. The expected value of the disturbance a is zero, and thus the model is found from a least squares solution after setting $\underline{a}= 0$:

$$\underline{\alpha} = -(Y^T Y)^{-1} Y^T \underline{y}$$

After finding $\underline{\alpha}$, an estimate of the noise sequence is simply found from $\underline{a}= Y\underline{\alpha}+\underline{y}$, from which an estimate of the disturbance variance σ_a^2 can be found. Having found the polynomial $A(q^{-1})$, the impulse response coefficients ψ_i are found from

$$y_k = \frac{1}{A(q^{-1})} a_k = \Psi(q^{-1}) a_k$$

using polynomial long division. Here $\Psi(q^{-1}) = 1+\psi_1 q^{-1}+\psi_2 q^{-2}+\psi_3 q^{-3}+ \cdots$.

2.7.1.3 Calculating the Harris Index

The Harris index is the ratio of the observed variance to the variance that would be obtained by MVC. The minimum achievable variance can be calcu-

lated from Eq. (2.29) above, using the identified impulse response coefficients and the estimated disturbance variance

$$\sigma_a^2 = \frac{1}{N-1}\sum_{i=1}^{N}(a_i - \overline{a})^2$$

where $\overline{a}$ is the mean value of the estimated disturbance, which is zero by construction.

The observed variance of the controlled variable can be computed similarly. However, if there is a persistent offset in the control loop, i.e., if the mean value of the controlled variable deviates from the reference, this should also be reflected in a measure of control quality. Hence, a modified variance should be used which accounts for this persistent offset

$$\sigma_{y,o}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - y_{ref})^2$$

If there is a persistent offset from the reference, the modified variance $\sigma_{y,o}^2$ will always be larger than the true variance σ_y^2, and the Harris index becomes

$$\mathrm{HI} = \frac{\sigma_{y,o}^2}{\sigma_{y,mv}^2}$$

while the Normalized Harris Index becomes

$$\mathrm{NHI} = 1 - \frac{\sigma_{y,mv}^2}{\sigma_{y,o}^2}$$

2.7.1.4 Obtaining the Deadtime

All the information required for calculating the NHI can be obtained from routine operating data, *provided* the deadtime is known. If the deadtime is not known, it may be estimated from online data – *provided* sufficiently informative data can be found, otherwise an identification experiment may be required in order to determine the deadtime. Further information on deadtime estimation can be found in Bjorklund and Ljung (2003).

2.7.2 Modifications to the Harris Index

Despite the theoretical elegance of the derivation of the minimum variance controller, the minimum variance controller may not be a realistic choice for a controller in a real application. This is because it is sensitive to model errors, and may use excessive moves in the manipulated variable. It *does* provide

an absolute lower bound on the theoretically achievable variance, but it is nevertheless of interest to have a control quality measure which compares the actual performance to something (hopefully) more realistic.

A simple modification to the Harris index is to simply use a too high value for the time delay, thus increasing the 'minimum' variance. This is discussed in Thornhill et al (1999) The resulting performance index will then no longer compare actual performance with a theoretically optimal performance. In Thornhill et al (1999), typical choices for the 'prediction horizons' are discussed for common control loop types in refineries (*e.g.*, pressure control, flow control, *etc.*). This modification is sometimes known as the *extended horizon performance index*.

Another modification is to assume that the 'ideal' controller does not totally remove the effect of disturbances after one deadtime has passed, but rather that the effect of the disturbance decays as a first order function after the deadtime has passed. If we assume that this decay is described by the parameter μ ($0 < \mu < 1$), so that the ideal response to disturbances against which performance is measured would be

$$y_{k,mod} = \sum_{i=0}^{\delta-1} \psi_i a_{k-i} + \sum_{i=\delta}^{\infty} \psi_{\delta-1} \mu^{i-\delta+1} a_{k-i}$$

which results in a modified 'benchmark variance'

$$\sigma^2_{y,mod} = \sigma^2_{y,mv} + \frac{\mu^2}{1-\mu^2}\sigma^2_a$$

The modified control performance index then simply becomes

$$H_{I,mod} = \frac{\sigma^2_{y,o}}{\sigma^2_{y,mod}}$$

This modified Harris index is proposed by Horch and Isaksson (1998b) and Kozub (1996). Horch and Isaksson also provide some guidelines for how to specify the tuning factor μ. They find that if one wishes to account for a possible error in the estimated deadtime of ± 1 sample interval, and still require a gain margin of 2 for the 'ideal closed loop', this corresponds to choosing $\mu > 0.5$. It is also recommended to have a realistic attitude to how much the dynamics of the closed loop system can be sped up, compared to the dynamics of the open loop process. Horch and Isaksson argue that it is unrealistic to speed up the system by a factor of more than 2-4[5]. If we denote

[5] While this argument is reasonable for many control loops, it is obviously incorrect for integrating processes (e.g., level control), where the open loop time constant is infinite. Ideally, one should base an estimate of the achievable bandwidth on more fundamental system properties like time delays, inverse response, or limitations in the manipulated variables.

the open loop dominant time constant τ_{ol}, and the desired closed loop time constant is τ_{ol}/v, then the parameter μ should be chosen as

$$\mu = \exp\left(-\frac{vT_s}{\tau_{ol}}\right)$$

where T_s is the sampling interval for the control system.

2.7.3 Assessing Feedforward Control

The time series analysis behind the Harris index can also be extended to cases with feedforward control from measured disturbances. In cases where disturbances are measurable, but not used for feedforward control, the analysis can be used to quantify the potential benefit (in terms of variance reduction) from implementing a feedforward controller. This is described by Desborough and Harris (1992). The analysis requires knowledge of the deadtimes from measured disturbances to controlled variable in addition to the deadtime from the manipulated variable to the controlled variable[6]. Their analysis results in an Analysis of Variance table, which shows how much of the observed variance is due to the unavoidable minimum variance, and what fractions of the excess variance is affected by feedback control alone, how much is affected by feedforward control alone, and how much is affected by both feedback and feedforward control.

In a related paper, Stanfelj et al (1993) address the analysis of the cause for poor performance, and show how to determine whether it is due to poor feedforward or feedback control. If the cause is poor feedback control, it is sometimes possible to determine whether it is due to poor tuning, or due to errors in the process model. This obvioulsy requires that a (nominal) process model is available, in contrast with the analysis of Desborough and Harris which only requires the knowledge of deadtimes. Reliable model quality assessment also requires some external excitation of the control loop, typically via controller setpoint changes.

2.7.4 Advanced Benchmarks

There exists more advanced benchmarks which can be used for practical purposes. They are

[6] The deadtime from measured disturbances to the controlled variables should be possible to identify from closed loop data, given a data segment with significant variations in the measured disturbance. If the identified deadtime is equal to or higher than the time delay from manipulated to controlled variable, the measured disturbance does not contribute to variance in the controlled variable under minimum variance control.

1. The linear quadratic Gaussian (LQG) regulator benchmarking
2. Generalised minimum variance (GMV) benchmarking
3. Restricted-structure or model-based benchmarking (RS)

These benchmarks are briefly explained below.
LQG benchmarking: It is proposed (Huang and Shah, 1999) as an alternative to minimum variance benchmarking. The advantage of the LQG benchmark is that it also accounts for the use of the manipulated variable, whereas the MVC assumes that arbitrarily large manipulated variable moves can be made 'for free'. The main disadvantage lies in the requirement to know the full model of the process. The use of an LQG benchmark for CPA is much more complicated than the traditional methods based on the MVC.

GMV benchmarking: The use of the Generalized Minimum Variance controller as a benchmark for performance monitoring was proposed by Grimble (2002). The Generalized Minimum Variance controller can be seen as an LQG controller for restricted choices of dynamic weights on the input and control error. However, the design restriction also allows for simpler calculation of the controller. More important in the CPA context is that the GMV benchmark can be calculated using plant data and knowledge of the deadtime only, without requiring the knowledge of the full model, see Grimble (2002) for details.

RS benchmarking: In contrast to MVC, the majority of practical controllers are of PID-type, and have a specific order and structure. Therefore, it has been argued that realistic performance indicators should be applied for their assessment, as proposed by Eriksson and Isaksson (1994) and Ko and Edgar (2001). These approaches calculate a lower bound of the variance by restricting the controller type to PID only (optimal PID benchmarking) and allow for more general disturbance models. The PID-achievable lower bound is generally larger than that calculated from MVC, but is designed to be achievable by a PID controller. That is, one is interested in determining how far the control performance is from the "best" achievable performance for the pre-specified controller. Like the LQG benchmark, RS benchmarking also requires knowledge of the full order model.

2.7.5 Multivariate Performance Measures

An extension to the derivation of HI of MIMO processes is possible by using multivariate spectral factorization and thereby solving a multivariate Diophantine identity. Also, the filtering and correlation algorithm developed by Huang and Shah (1999) for single loop performance assessment can also be extended to address MIMO controller performance by adding the concept of an interactor matrix or time-delay matrix.

However, the Harris index became popular because of its simplicity. When the approach is extended to multivariate systems, the requirements of *a priori* knowledge and calculation burden are unavoidably increased due to the interactive effects among different variables. The interactor matrix, which allows the feedback control-invariant term of the outputs to be extracted, is essential in the calculation of the MV for a multivariable 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). However, the above techniques require detailed knowledge which is normally challenging to obtain or estimate accurately. Plant tests introducing sufficient excitation, followed by considerable modeling effort has to be undertaken in order to get this information. This is the major difficulty for the application of multivariable MVC benchmark performance assessment algorithms. The method developed by Xia et al (2006) can estimate upper and lower bounds of the MIMO MV performance index from routine operating data if the I/O delay matrix is known. The lower bound can be estimated from routine operating data, while the estimation of the upper bound normally requires introducing additional delays to the controller. The method can be applied to evaluate the regulatory performance of MIMO industrial controllers.

2.8 Procedure for Controller Performance Assessment

Controller performance assessment is a challenging task in industrial process control. Ideally, any controller performance assessment technique should have the following attributes (Hugo, 2005):

1. Should be independent of disturbance or setpoint spectrums. Both the disturbances and setpoint changes can vary widely in a plant, and the assessment should be insensitive to the time period when the data was taken.
2. Should not require plant tests. This requirement is generally met, as the user is interested in the closed- loop behavior of the process. However, closed-loop data can be information poor, and any performance assessment technique should include tests of the accuracy of the results.
3. Able to be automated. The large numbers of loops in a plant necessitate that at least part of the controller performance assessment be done automatically.
4. Require minimum specification of process dynamics.
5. Absolute or non-arbitrary measure. The metric should compare the current quality of control to some universal standard.
6. Sensitive to detuning or process model mismatch or equipment problems only. The metric should give an indication of only those things that the control engineer can affect.

7. Indicative of why the controller is performing poorly. Ideally, the measures should indicate what should be done to improve control, whether the problem is due to poor tuning, valve sticking, or oscillations from an unknown source.
8. Measure the improvement in profit due to the controller. This may be separate from measuring reduction in variance, as a major profit contribution for some controllers is pushing the process to constraints.

According to Hugo, current software packages generally meet requirements 1-6 above. Requirement 7 is only partially met identifying new tuning parameters or a process model strictly from closed-loop data is the function of a self-tuning regulator (which has found very limited success in industry). The main difficulty in requirement 8 is defining a base case, which is an activity that is best done off-line. However, performance assessment techniques can indicate whether advanced control can reduce the variance over the current PID controllers.

It is not sufficient and sometimes dangerous to rely on a single statistic for performance monitoring and diagnosis, as each criterion has its merits and limitations. The best results are often obtained by the collective application of several methods that reflect control performance measures from different aspects. Sections 2.8.1 to 2.8.4 below propose a systematic procedure for CPA, based in the recommendations in the recent study by Jelali (2005).

2.8.1 Preliminary Analysis of Data

2.8.1.1 Data Pre-processing

Real world data are generally *(i)* incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), *(ii)* noisy (containing errors or outliers) and *(iii)* inconsistent (containing discrepancies in codes or names). The following necessary steps need to be considered before processing the data.

- Use raw data collected at a proper sampling frequency.
- Strictly avoid filtering/smoothing or compression of the data.
- Remove the outliers or bad data.
- Do mean centering and scaling.

Outliers are unusual data values that are not consistent with most observations. Commonly, outliers result from measurement errors, coding and recording errors, and, sometimes, are natural, abnormal values. Such non-representative samples can seriously affect the model produced later. There are two strategies for dealing with outliers: *(i)* detect and eventually remove outliers as a part of the preprocessing phase, or *(ii)* develop robust modeling methods that are insensitive to outliers.

Data compression is found to have detrimental effects on the reliability/-validity of control loop performance measures (Thornhill et al, 2004). When the controller performance indices estimated using the compressed data are used for performance assessment, we are inclined to make the errors in assessing the control loop performance. Hence, it becomes necessary to use the raw data (uncompressed) for controller performance monitoring. Also, it is common to centre and scale data such that each variable in the analysis have mean zero and unit variance. Subtracting the mean of the data is often called "mean centering". It results in a shift of the data towards the mean. The mean of the transformed data thereafter equals to zero.

2.8.1.2 Interaction Analysis when Dealing with MIMO Systems

Industrial control systems generally are designed by assuming that the multivariable control problem can be decomposed into a series of single input-single output problems. Often the individual input-output pairs are selected intuitively. However, this qualitative approach is sometimes not sufficient for control loop design. There is a need to place the interaction analysis for controls and outputs on a more quantitative basis, which accounts directly for dynamic properties of the system (Skogestad, 2004).

Multivariate CPA is only required when the loops are strongly coupled. This can be found out by applying standard interaction measures, such as relative gain array, which are simple to calculate and interpret provided a process model is available. Cross-correlation (coherence) analysis is also useful to assess the interaction between the control loops. Even in the case of significant interactions, one should apply CPA methods, which do not require the interactor matrix of the process.

2.8.1.3 Time Delay Estimation

In many applications, the time delay can be directly or indirectly estimated. When time varying, the delay should be continuously updated based on input/output measurements. Some methods for time delay estimation are discussed in Jelali (2005) and the references therein. When the time delay is completely unknown, or its determination/adaptation is costly, the use of the extended prediction horizon approach (Dumont et al, 1993) is highly recommended.

2.8.2 Detection of Specific Malfunctions

The correlation (covariance) analysis of the control error is simple and should be always carried as a first test before carrying out further performance analysis. The cross-correlation between measured disturbances and the control error can be used to qualitatively assess feedforward control.

Also spectral analysis of the closed-loop response, which allows one to detect oscillations, offsets, non-linearities, and measurements noises present in the process easily, should be performed. A common symptom of poor loop performance is the appearance of oscillations in process variables. The next step should be to evaluate how linear (or nonlinear) the closed loop is by applying one or several tests for detecting non-linearity as possible root-cause for control loop performance problems. This is of particular relevance for loops that are found to be oscillating.

For loops with (stationary) offset from setpoint, one should check whether this is due to lacking integral action or saturation of the manipulated variable. In the latter case, one should consider whether the saturation is due to an undersized manipulated variable, or the result of an inappropriate control structure causing competition with other loops.

For loops where persistent oscillations are the main problem, rather than input saturation or stationary offset, stiction and backlash detection should be performed. Some relevant techniques have been presented in preceding sections. Even if a specific malfunction cannot be identified, indications of non-linear effects in a signal can be used to guide the search for the cause of the performance problems.

2.8.3 Evaluation of Level of Control Performance

2.8.3.1 Apply the MVC-based Assessment

This should be the standard benchmark to be applied. Appropriately selected model orders (typically $N \geq 10\tau$), and a minimum length of data (typically $N \geq 150\tau$) are necessary for obtaining reliable results. Here, τ refers to time delay. When the Harris index signals that the loop is performing well, then further assessment is neither useful nor necessary. In the case, where a poor performance relative to MVC is detected, there is a potential to improve the control loop performance, but no guarantee that this will be attained by means of retuning the existing controller. Further, analysis is then warranted.

2.8.3.2 Apply User-specified or Advanced Control Performance Benchmarking

Baselines and thresholds (historical benchmark values) using data with "perfect" controller performance, or restricted structure (e.g., PI) performance benchmarking (preferably combined with IMC tuning) can be applied. Also, the use of more advanced linear quadratic gaussian/ generalized minimum variance benchmarking can be an option, particularly in cases where performance improvement cannot be achieved by retuning the running controller, and/or for supervisory control loops. Although restricted-structure benchmarking is quite demanding since it requires plant model to be known, a beneficial side-effect that it can provide information on how the controller can be retuned/designed to obtain optimal performance.

2.8.4 Improvement of Control Performance

2.8.4.1 Retune the Control Loop

Adjust some parameters of the control loop(s) found to be poorly performing. When retuning is not necessary, or does not improve the control performance, modifications of the instrumentation, control system structure or the process itself will be required, if the current operation is deemed unacceptable.

2.8.4.2 Modify the Control Structure

When retuning does not improve the control performance, modify some structural components. In some cases, this could mean a complete redesign of the control loop(s). Watching a controller performance metric over many operating regions, might help to discover opportunities for gain-scheduling or possibly the use of adaptive control.

2.8.4.3 Repair/Redesign System Components

In some situations, inspection and maintenance measures should be taken. This might follow directly from the findings in section 2.8.2, if specific malfunctions such as valve stiction has been identified. In other cases, redesign may be necessary if acceptable performance cannot be achieved with well-structured and well-tuned control system. This may involve modification of the feedback dynamics, such as reducing the time delay by changing the process flow (e.g., adding a bypass), or changing the sensor location. Also,

disturbance sources might be eliminated, or supplementary sensors installed to enable feedforward control.

In the procedure described above, many parameters have to be selected by the user. As an initial basis, default parameters for the performance index calculation, which were shown to be useful for various generic categories of refinery control loops by Thornhill et al (1999), may be used or determined in a similar way. That work substantially lowered the barrier to large scale implementation of performance-index-based monitoring. It is necessary and well-spent time to carefully test, inspect, and compare the CPA results using different parameter choices. Usually, similar parameter values may be used in the control performance assessment of control loop of the same category (such as flow control, pressure control, *etc.*).

2.9 Issues in Multivariate Systems

There is considerable incentive for extending the univariate controller performance measures to the multivariable case, both for maintaining these controllers and evaluating their economics, but the solutions available thus far are difficult to implement. The main disadvantage is that the user must specify or determine the plant interactor matrix, which depends not only on the time delays, but is a function of the all plant dynamics.

To date, commercial packages do not contain algorithms for assessing multivariable controllers. However, it is possible to use the single loop techniques on each of the outputs of a multivariable controller, although the results will be somewhat biased. Below we give a brief examination of some aspects controller performance assessment for multivariable model predictive control (MPC), and the applicability of single-loop performance assessment to the multivariable case.

Multivariable controllers in general use all the inputs to control all the outputs, but the control engineer is mainly interested in how well each output is controlled to its setpoint, and this is exactly what single loop performance indices measure. The results will however be biased somewhat as more than one input can affect each output, with the amount of biasing dependent on the amount of process coupling. Fortunately, many process are not tightly coupled, and it is often the case that each major output is controlled largely by one input.

Model predictive control is primarily used for multivariable systems where it is desirable to operate close to operational constraints (and where the set of active constraints may change with operating conditions). MPC controllers typically consist of two "layers":

1. An upper layer where the optimal (in some sense) operating conditions are identified. This is typically formulated as a steady state optimization problem, using a steady state model of the process.

2. A lower layer which attempts to control the process to the optimal conditions identified in the layer above. At this layer, a dynamic process model is used, with the same steady state gains as those used in the layer above.

In both these layers, an optimization problems are solved on-line. Reasons for poor MPC performance may therefore be:

- Inappropriate formulation of the optimization cost function at one or both layers. The variables affecting the cost function, and their (relative) weight, should obviously reflect the desired operation of the plant. This point might seem obvious, but will often require considerable process knowledge and experience from the designer.
- Model/plant mismatch, *i.e.*, the model used by the MPC contains significant errors.
- Inappropriate (usually too stringent) constraints.

Some MPC controllers apply only input constraints directly in the optimization problem formulation, and translate output constraints into setpoints when such constraints are active. Other MPC controllers handle also output constraints directly in the optimization formulation. In either case, an active constraint means that control quality will have to deteriorate for some other variable(s).

Monitoring what constraints are active is the most basic step in CPA for MPC. Examining the Lagrangian multipliers for the optimization problem (especially at the upper layer of the MPC – which is usually more tightly related to economic performance) can give information on the potential gain from modifying the constraint. Ordinary mono-variable CPA metrics can also give valuable insight into the quality of the MPC control, in particular if there is only moderate interactions in the plant. Most MPC controllers account for input usage (and not only output variance) in the optimization problem at the lower level, a GMV-type performance measure may therefore be more appropriate than the Harris index. One should, however, be aware that the CPA measure will be more reliable if the same set of constraints are active throughout the length of the data series, as changes in the set of active constraints essentially means that the system is time varying.

The work of Patwardhan and Shah (2002) focuses on the performance diagnostics of MPC controllers. An attempt has been made out of their work to quantify the effect of constraints, model uncertainty and nonlinearity on the performance of linear MPC. Recently, Chen and Wang (2009) have developed a statistic-based method for performance assessment and monitoring of the multivariate feedback control system where an integration of principal component analysis and the autoregressive moving average filter for building up minimum variance performance bounds.

Jiang et al (2006) proposed a new scheme to detect and isolate Model-Plant Mismatch (MPM) for multivariate dynamic systems where the MPM problem is formulated in the state-space domain, as is widely done in the design and implementation of MPCs. The specific issue addressed therein is to

identify which among the state-space matrices had to be re-estimated in order to account for significant plant deviations from its nominal state, and three MPM detection indices (MDIs) are proposed to detect the MPM for that purpose. A shortcoming of their work is that changes in state-space matrices cannot be directly translated to changes in process characteristics like gain, time-constant and delay. 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. Recently, Selvanathan and Tangirala (2010) proposed a method for the diagnosis of poor control loop performance due to model plant mismatch (MPM) in the internal model control framework. In particular, the objective here is to identify the mismatch in specific components of a transfer function model, namely, the gain, time-constant and delay from routine closed-loop data. 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 which shows that there exists a unique signature in PMR for each combination of mismatch in model parameters. This method can not be applied directly to the multivariable control due to the presence of multivariable interactions, but Selvanathan and Tangirala (2010) do provide indications of research directions that may provide solutions to this shortcoming.

Acknowledgements The authors are pleased to acknowledge the financial support by a grant No. NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism.

References

Bialkowski W (1993) Dreams versus reality: a view from both sides of the gap. Pulp and Paper Canada 94:19–27

Bjorklund S, Ljung L (2003) A review of time-delay estimation techniques. In: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, vol 3, pp 2502–2507

Chen J, Wang WY (2009) Performance assessment of multivariable control systems using pca control charts. In: Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on, pp 936–941

Choudhury MAAS, Shah SL, Thornhill NF (2004a) Diagnosis of poor control loop performance using higher-order statistics. Automatica 40:1719–1728

Choudhury MAAS, Jain M, Shah SL (2008) Stiction-definition, modelling, detection and quantification. Process Control 18:232–243

Choudhury MS, Shah SL, Thornhill NF (2004b) Diagnosis of poor control-loop performance using higher-order statistics. Automatica 40:1719–1728

Desborough L, Harris T (1992) Performance assessment measure for univariate feedback control. Canadian Journal of Chemical Engineering 70

Desborough L, Miller R (2002) Increasing customer value of industrial control performance monitoring: Honeywell's experience. Proc AIChE Symp Ser 98:153–186

Desborough LD, Harris TJ (1993) Performance assessment measure for univariate feedforward/feedback control. Canadian Journal of Chemical Engineering 71

Dumont GA, Elnaggar A, Elshafei A (1993) Adaptive predictive control of systems with time-varying time delay. International Journal of Adaptive Control and Signal Processing 7(2):91–101

Ender D (1993) Process control performance: not as good as you think. Control Engineering 40:180–190

Eriksson PG, Isaksson A (1994) Some aspects of control loop performance monitoring. In: Control Applications, 1994., Proceedings of the Third IEEE Conference on, vol 2, pp 1029–1034

Fisher (1999) Control Valve Handbook. Fisher Controls International, Marshalltown, Iowa, USA

Forsman K, Stattin A (1999) A new criterion for detecting oscillations in control loops. In: Proceedings of the European Control Conference, Karlsruhe, Germany

Grimble MJ (2002) Controller performance benchmarking and tuning using generalized minimum varianve control. Automatica 38:2111–2119

Hägglund T (1995) A control-loop performance monitor. Control Eng Practice 3(11):1543–1551

Hägglund T (2007) Automatic on-line estimation of backlash in control loops. Journal of Process Control 17:489–499

Harris T (1989a) Assessment of control loop performance. Canadian Journal of Chemical Engineering 67:856–861

Harris TJ (1989b) Assessment of control loop performance. Can J Chem Eng 67:856–861

He QP, Wang J, Pottmann M, Qin SJ (2007) A curve fitting method for detecting valve stiction in oscillating control loops. Industrial & Engineering Chemistry Research 46(13):4549–4560, DOI 10.1021/ie061219a, URL http://pubs.acs.org/doi/abs/10.1021/ie061219a, http://pubs.acs.org/doi/pdf/10.1021/ie061219a

Hinich MJ (1982) Testing for gaussianity and linearity of a stationary time series. Time series analysis 3(3):169–176

Hiroshi M, Kano M, Hidekazu K (2004) Modeling and detection of stiction in pneumatic control valve. Transactions of the Society of Instrument and Control Engineers 40(8):825–833

Horch A (1999) A simple method for oscillation diagnosis in process control loops. Control Engineering Practice pp 1221–1231

Horch A, Isaksson A (1998a) A method for detection of stiction in control valves. In: Online fault detection and supervision in the chemical process industry, IFAC Workshop, Lyon, France, p 4B

Horch A, Isaksson AJ (1998b) A modified index for control performance assessment. In: Proceedings of the American Control Conference, pp 3430–3434

Huang B, Shah S (1999) Performance assessment of control loops. Advances in Industrial Control, Springer

Huang B, Shah S, Fujiii H (1997) The unitary interactor matrix and its estimation using closed-loop data. Journal of Process Control 7(3):195–207

Hugo AJ (2005) Process Controller Performance Monitoring and Assessment. Control Arts Inc.

Jelali M (2005) An overview of control performance assessment technology and industrial applications. Control Engineering Practice 14:441–466

Jelali M (2008) Estimation of valve stiction in control loops using separable least-squares and global search algorithms. Journal of Process Control 18:632–642

Jiang H, Li W, Shah S (2006) Detection and isolation of model-plant mismatch for multivariate dynamic systems. In: IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pp 1396–1401

Kano M, Maruta H, Kugemoto H, Shimizu K (2004) Practical model and detection algorithm for valve stiction. In: IFAC Symposium on Dynamics and Control of Process Systems

Kayihan A, Doyle III F (2000) Friction compensation for a process control valve. Control Engineering Practice 8(799-812)

Khalil HK (1996) Nonlinear Systems. Prentice Hall, Upper Saddle River, New Jersey, USA

Ko BS, Edgar T (2001) Performance assessment of multivariable feedback control systems. Automatica 37(899-905)

Kozub D (1996) Controller performance monitoring and diagnosis: Experiences and challenges. In: CPC-V, pp 83–95

Larsson T, Skogestad S (2000) Plantwide control - a review and a new design procedure. Modeling, Identification and Control 21:209–240

Matsuo T, Tadakuma I, Thornhill NF (2004) Diagnosis of unit-wide disturbance caused by saturation in a manipulated variable. In: IEEE Advanced Process Control Applications for Industry Workshop

Miao T, Seborg DE (1999) Automatic detection of excessively oscillating feedback control loops. In: IEEE Conference on Control Applications - Proceedings, Vol. 1., pp 359–364

Moiso M, Piiponen J (1998) Control loop performance evaluation. In: Proceedings of Control Systems, pp 251–258

Olsson H (1996) Control systems with friction. PhD thesis, Lund Institute of Technology, Sweden

Overschee PV, Moor BD (2000) Rapid: The end of heuristic pid tuning. In: Quevedo J, Escobet T (eds) Digital Control 2000: Past, Present and Future of PID control, IFAC Workshop, Pergamon Press, Terrassa, Spain

Patwardhan R, Shah S (2002) Issues in performance diagnostics of model-based controllers. Journal of Process Control 12:413–427

Petersson M, Årzén KE, Hägglund T (2003) A comparison of two feedforward control structure assessment methods. International Journal of Adaptive Control and Signal Processing 17:609–624

Rogozinski M, Paplinski A, Gibbard M (1987) An algorithm for calculation of a nilpotent interactor matrix for linear multivariable systems. IEEE Transactions on Automatic Control 32(3):234–237

Selvanathan S, Tangirala AK (2010) Diagnosis of poor control loop performance due to model plant mismatch. Industrial & Engineering Chemistry Research 49(9):4210–4229, DOI 10.1021/ie900769v, URL http://pubs.acs.org/doi/abs/10.1021/ie900769v, http://pubs.acs.org/doi/pdf/10.1021/ie900769v

Skogestad S (2004) Control structure design for complete chemical plants. Computers and Chemical Engineering 28:219–234

Skogestad S, Postlethwaite I (2005) Multivariable Feedback Control. Analysis and Design. John Wiley & Sons Ltd, Chichester, England

Srinivasan R, Rengaswamy R, Miller R (2005) Performance assessment. 1. a qualitative pattern matching approach for stiction diagnosis. Industrial & Engineering Chemistry Research 44(17):6708–6718

Stanfelj N, Marlin T, MaGregor JF (1993) Monitoring and diagnosing process control performance: the single-loop case. Industrial & Engineering Chemistry Research 32:301–314

Stenman A, Gustafsson F, Forsman K (2003) A segmentation-based method for detection of stiction in control valves. International Journal of Adaptive control and signal processing 17(625-634)

Thornhill N, Choudhury M, Shah S (2004) The impact of compression on data-driven process analyses. Journal of Process Control 14:389–398

Thornhill NF, Hägglund T (1997) Detection and diagnosis of oscillation in control loops. Control Eng Practice pp 1343–1354

Thornhill NF, Oettinger M, Fedenczuk P (1999) Refinery-wide control loop performance assessment. J of Process Control pp 109–124

Torrence C, Compo G (1998) A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79:61–78

Xia H, Majecki P, Ordys A, Grimble M (2006) Performance assessment of mimo systems based on i/o delay information. Journal of Process Control 16:373–383

Yamashita Y (2006) An automatic method for detection of valve stiction in process control loops. Control Engineering Practice 14:503–510

Chapter 3
Basic Notions of Robust Constrained PID Control

Mikuláš Huba

Abstract This chapter is aimed as introduction to the prepared textbook on the Robust constrained PID control. It makes you familiar with historical development of PID control, its basic components and structures, problems and motivations and with basic terminology used within this area. After studying this chapter you should be better able to describe phases in the technology development of the PID control, to characterize basic existing types of PID controllers and to explain, why the development of PID control cannot be considered as finished, to characterize different performance specifications and related terms as ε-nonovershooting, ε-nonundershooting, ε-monotonic and ε-n-pulse (nP) functions and their use in deriving the so called closed loop performance portrait, to explain notion of dynamical classes (DCs) of control and their relation to the Feldbaum's theorem about n-intervals of the relay minimum time control (MTC), to explain impact of DCs on performance and design of PID control, to explain notion of fundamental solutions for setpoint tracking and disturbance rejection and to explain and characterize basic elements of the extended table of fundamental PID controllers. Within this publication, this introduction is followed by the next chapter bringing the simplest structures of the DC0 and the new robust method based on the Performance Portrait used for the controller tuning. More advanced problems from higher dynamical classes are treated by separate papers in the associated workbook and in Preprints of the NIL workshop (Huba et al, 2011a,b)

Mikuláš Huba
Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, e-mail: mikulas.huba@stuba.sk

3.1 Introduction

This text is devoted to developing new approach to the robust constrained PID control of simple single-input-single output (SISO) plants. It starts by showing and arguing, where and why new alternative solutions were proposed to the traditional ones to extend mainstream of the contemporary development and how the traditional problems may be treated more efficiently. Despite the strong emphasis on comparing with the already existing works, the overview of references given is surely not complete. With respect to this, but also in other points, we will welcome any comments and proposals for improvements. For mathematically oriented reader the text may seem to be not sufficiently covered by proofs of basic conclusions. And conversely, for people from practice it may seem to be mathematically too demanding: We spent a lot of space by trying to fit the academic and engineering control methods together to get approaches matching optimally needs of real time control. Therefore, the text is frequently illustrated by examples and will be complemented by results of controlling physical plant models brought by other outputs (workbook, workshop preprints) created and presented within the NIL project.

PI controllers operate about 90% control systems. They represent the core module of PID controllers that cover about 95% Åström and Hägglund (1995), or even 98% (Datta et al, 2000) of all control systems in practice. By nearly century of its existence and by its impact on practice it is related to personal experience of huge amount of people. We are trying to address this experience by stressing importance of controlling simple plants and by comparing different approaches and results.

As each control design, also the design of PID controllers must be based on some model of the plant behavior and the resulting controllers will necessarily depend on information embedded into this model. Since the model represents just an abstract approximation of chosen features of real systems, it is never complete and always it is to some degree uncertain. Its uncertainty is expected to influence quality of achieved control results that usually depends on factors as:

- measurement noise in identification and control,
- disturbances acting on the plant during the identification and control,
- numerical errors and other imperfection of the methods used in the identification, controller design and control,
- plant nonlinearities relevant to larger deviations from fixed operating points,
- plant nonlinearities relevant to the vicinity of the operating point (as e.g. hysteresis),
- non-modeled (high frequency) plant dynamics, i.e. dynamics not considered in deriving controller equations/structure,
- time related changes of the plant dynamics.

From the beginning of control design, each method used in practice was somehow be able to cope with impact of all these factors. In the last decades, the robustness aspects related to model uncertainty are treated more rigorously and many works and publication on robust process control based e.g. on H_2 and H_∞ norms proposed, discovered and analyzed a lot of useful features and methods. However, the huge number of newly appearing papers devoted to the robust control of simple SISO systems that try to optimize traditional solutions, or propose new ones (Skogestad, 2003; Baños and Vidal, 2007; Johnson and Moradi, 2005; Keel et al, 2008; O'Dwyer, 2006; Seok et al, 2007) indicate that there still exist features of PID control that are expected to be improved. The high number of appearing publications has also drawbacks as that it is practically impossible to follow all streams of ideas and methods of the development and to offer a unifying presentation giving explanation of their internal relations. And it is not enough to deal just with the newest development. As it is documented by many examples from science history, not every time the mostly spread opinions guarantee further progress and it can happen that some already forgotten ideas finally show to play the key role in achieving new generation of solutions.

Newer approaches of robust process control trace their origins (Morari and Zafiriou, 1989) to the "analytical" design by Newton, Gould, and Kaiser (1957) based on optimizing the ISE (Integral Square Error) performance index. Morari and Zafiriou denoted the earlier approaches as the "Trial and Error" ones. It can be, however, shown that already the older approaches involved some features of the robust design. And, on the other side, also the modern "robust&analytical" approaches mostly require some iterative modifications until the best compromise between the usually conflicting objectives is reached. It may e.g. be caused by the fact that the ISE based design is not primarily motivated by practical requirements but by the mathematical convenience and it is known to lead to slightly oscillatory behavior. Therefore, in this text design based on minimal IAE (Integral of Absolute Error) values will be preferred (Shinskey, 1990). For practical use, requirements of the fastest possible transients giving minimal IAE values will be extended by requirements of nonovershooting (NO), or monotonic (MO) control responses of the output variable that should be achieved by a reasonable excurse of the manipulated (control) variable giving minimal Total Variance (TV) values (Skogestad, 2003). So, in the control design reported in this text we will try to get as fast as possible MO control responses by respecting both the plant model uncertainties, the control signal constraints and demand on the total excurse of the manipulated variable.

As the 2$^{\text{nd}}$ most important pillar of the robust process control give Morari and Zafiriou the work by Youla et al (1976) on parameterizing all stable controller transfer functions possible to given (linear) plant and specified problem. In this way their primarily aim, to search for a good controller, was greatly simplified. From this point of view it may be, however, noted that the same aim (to parameterize all stable controllers and so to simplify

search for a good controller) was partially achieved already by the pole placement (pole assignment) control design. This (when choosing stable closed loop poles) is also giving continuum of stable parameterized controllers. According to Åström and Wittenmark (1984) pole assignment approach was firstly treated by J. Bertram in 1959 and the first published solution was given by Rissanen (1960). Despite this (older) approach is not as general as the parameterization by Youla, it is broadly used within different "modern" approaches. The first design step, the determination of parameterized nominal controllers, was, however, up to now not sufficiently completed by the second step, the robustification of the controller. This requires choosing appropriate closed loop poles. Instead of working with the closed loop poles (that are specified by negative numbers) it may be simpler to use positive parameters denoted as bandwidth, or their reciprocal values having meaning of time constants. However, up to now there do not exist proven techniques for robust performance design relating simply given control specifications with the closed loop poles and with the uncertainty information, nonmodelled dynamics and measurement noise. This step is therefore still mostly done by the trial and error method, whereby the choice of the closed loop poles is not only influenced by the system uncertainty and the nonmodelled dynamics but also by the constraints put on the control and state variables. This text shows how the "trial and error" procedures can be automatized and replaced by a systematic computer based qualitative and quantitative analysis appropriately taking into account both role of constraints, plan-model mismatch and different performance specifications. At least for the simplest loops with dominant dynamics up to the second order the problem of the control signal constraints may be eliminated by the generalized constrained pole assignment control (Huba et al, 1999; Huba, 2006). In connection with the computer based analysis, all developed controllers can be used also for achieving specified robustness degree.

Another broadly accepted approach to general parametrized solutions related to the robust control and building on the sensitivity functions, or the complementary sensitivity functions, was introduced by Åström and Hägglund (1995). By trying to have clear-cut physical interpretation of the effect of such tuning parameters and clear picture of their appropriate default values, the tuning should be relatively easily adjusted (Skogestad, 2003) to a particular situation and so to be much simpler and reliable. However, from the point of view of the robust constrained pole assignment control the sensitivity and complementary sensitivity functions do not always represent an effective and efficient solution. They e.g. do not match the natural expectation that when requiring the fastest possible monotonic output transients by decreasing:

- range of possible parameter fluctuations,
- effect of the nonmodelled dynamics (parasitic delays) and
- amplitude of the measurement noise,

the achieved solutions should converge to the MTC. Using the pole assignment method, such a requirement was systematically followed by Glattfelder and Schaufelberger (2003). The anti-windup PI controllers they have analyzed were very close to give ideal control signal step reactions converging to one pulse of the MTC.

The other important handicap of the development – the gap between the classical state space approach and the newer robust control was formulated byMorari and Zafiriou (1989) as "*no smooth transition from the established proven techniques and tools (PID controllers, Smith Predictor) to the new ones*" - may only be eliminated by modifications done from both sides. We will try to resonse this comment by requiring smooth transition of the new robust approach to the PID control up to the Relay MTC. Such attempts have already been done e.g. by works of Glattfelder and Schaufelberger (2003) (who analyzed achieved PID solutions both from the point of view of robust control and MTC). Compatibility of different approaches and their relevance for practice was approached from different points of view also by many other authors. E.g. Rivera et al (1986), or Skogestad (2003) tried to combine theory with practice and stressed importance of the manipulated variable in evaluating achieved control performance. In this text we are going to look for compatibility and to explore different structures of PID control from the point of view of the state-space approach to controller design and to reconstruction and compensation of disturbances by using disturbance observer (DOB). Simultaneously, the achieved constrained loop dynamics will be confronted with result of the MTC. Thereby, it will not be related just to a fixed nominal operating point but to larger areas of loop parameters enabling to keep chosen loop dynamical properties under every time present uncertainties of the plant model. In order to introduce an effective controller classification, it is further important to introduce new notions like *n-pulse function, fundamental controllers* and *dynamical classes of control.* Before coming with these new definitions, let us briefly review basic notions of PID control.

Definition 3.1 (PID controller). Under the notion of PID control we will include all controllers for setpoint tracking and disturbance rejection in systems with the dominant dynamics up to the 2^{nd} order described by the transfer functions

$$S(s) = \frac{K_s(1+T_0 s)}{s^2 + a_1 s + a_0} e^{-T_d s} \tag{3.1}$$

An alternative previously used definition could speak about controllers dealing with the reference and with the output signal (control error), its derivative and integral given by the transfer function

$$R(s) = K_P \left(1 + \frac{1}{sT_I} + sT_D\right) \tag{3.2}$$

or by its realizable modifications characterized by following definitions.

Definition 3.2 (ISA PID controller). According to the ISA standard, the two-degree-of-freedom PID controllers can be described as

$$U(s) = k_P \left\{ bW(s) - Y(s) + \frac{1}{T_I s}[W(s) - Y(s)] + \frac{T_D s}{1 + sT_D/N}[cW(s) - Y(s)] \right\} \tag{3.3}$$

whereby

- $Y(s), W(s), U(s)$ represent Laplace transforms of the controller output, setpoint and process output variable,
- k_P is the controller gain,
- T_I and T_D the integral and derivative time constants,
- b and c are the weighting coefficients of the proportional and derivative action and N describes filtration of the derivative action.

By setting $T_I \to \infty$ one achieves PD controller, for $T_D = 0$ one gets PI controller and for $T_I \to \infty$ and $T_D = 0$ the P controller.

Definition 3.3 (Series PID controller). As an alternative to the previous description one can consider serial controller form:

$$U(s) = k' \left\{ \left[b + \frac{1}{sT'_I} \right] \frac{1 + scT'_D}{1 + T'_D/N} W(s) - \left[1 + \frac{1}{sT'_I} \right] \frac{1 + scT'_D}{1 + T'_D/N} Y(s) \right\} \tag{3.4}$$

Definition 3.4 (Parallel PID controller). The third basic controller form is given by equation

$$U(s) = K\,[bW(s) - Y(s)] + \frac{K_I}{s}\,[W(s) - Y(s)]\,\frac{K_D s}{1 + sK_D/(NK)}\,[cW(s) - Y(s)] \tag{3.5}$$

These 3 basic PID controllers can yet be completed by the I-controller:

Definition 3.5 (I-controller). I-controller may be defined as

$$U(s) = \frac{1}{T_I s}\,[W(s) - Y(s)] = \frac{K_I}{s}\,[W(s) - Y(s)] \tag{3.6}$$

I-controller can be simply derived just from the parallel form by setting $K = K_D = 0$.

For decades, these linear controllers represent building stones of the vast majority of solved problems. As a standard option they yet include a constant output signal (bias) and structures for switching from manual to automatic regime.

By analyzing their possibilities many authors have finally come to conclusion that it would be oversimplified to consider just controllers (3.2)–(3.6). This shift from transfer function (3.2) to more complex structures is not just

the invention of this publication. It results from a longer historical development reflecting needs of practice. Even when remaining within the scope of linear control, since 1960s controllers (3.2) characterized by a triple of parameters (K_P, T_I, T_D) are being replaced by more complex structures of the *two-degree-of-freedom* controller (3.3)–(3.5) with *setpoint weighting* (Horowitz, 1963; Åström and Hägglund, 2005). These are characterized by 5 or 6 parameters (with filters of the derivative action). Controllers used in practice take forms of more or less complex structures that e.g. always consider also constraints given on the controller output. Despite to the linear character of basic equations, the industrially produced controllers are usually equipped with control constraints and blocking of an abundant integration known as *anti-windup* (aw), or *anti-reset-windup* (arw), by the possibility of on-off (pulse width modulated) control, or with other nonlinear options (as e.g. error squared controllers). As we show later, even all these advanced possibilities are not enough to cover needs on a reliable and high quality control of systems (3.1) and despite to respect to traditions it shows to be necessary to extend this basis by new elements and to introduce internal differentiation of all existing solutions. It is also to note that despite speaking about PID control as being derived for the dominant second order dynamics this does not restrict its applications to controlling much more complicated systems.

3.2 Innovation versus Conservativeness

It is not easy to quote the first application of PID control. Integrated with other parts of controlled processes, controllers with proportional and integral action were used for ages. But as the first mathematical formulation of the proportional and integral action one can mention the analysis of the speed control of a steam engine "On Governors" by Maxwell (1868). As devices independent from sensors, actuators and controlled plants PI controllers (denoted as automatic reset) appeared at the end of the World War I. Controllers with the derivative (D) action (denoted as pre-act) came around 1935. The development in this area relates to the legendary firms as Bristol, Fisher, Foxboro, Honeywell, Leeds & Northrup, or Taylor Instruments. The first methods for an optimal tuning of controllers appeared few years later (see e.g. Ziegler and Nichols (1942); Oldenbourg and Sartorius (1951)). But, when now, after more than one century of study and development of the relatively simple concept of PI and PID control one can still find several open questions of their reliable use and tuning, it is necessary to ask "why"? What are the reasons for inflation of different forms and realizations (series, parallel, non-interactive – ISA and interactive – series (Åström and Hägglund, 1995) different quasi-continuous realizations, etc.)?

What are the reasons for inflation of "optimal" tuning rules? Just O'Dwyer (2000, 2006) reports in his works 154 tuning rules for PI control and practically each control conference devoted to the control design brings new ones.

Also some other points are not clearly explained: why it is e.g. sometime necessary to use setpoint weighting - it means to modify coefficients b and c in Eqs. (3.3)–(3.5) - and sometimes not?

Why it is sometime necessary to use anti-windup measures and sometimes not? Why do we have inflation of aw – circuitry, when just Glattfelder and Schaufelberger (2003) report and analyze 10 different schemes for PI control (see also Kothare et al (1994))? But, can we expect something else, when there does not exist a generally accepted unique definition of the windup phenomenon?

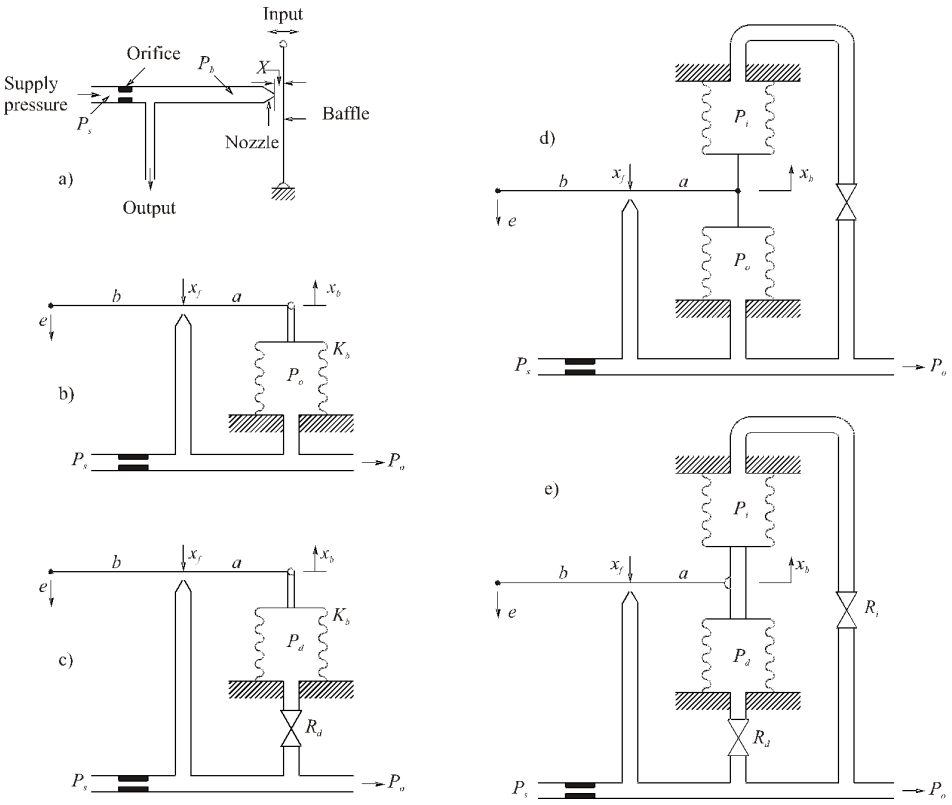


Fig. 3.1 Modular concept of pneumatic PID Control. a) Flapper-Nozzle high gain nonlinear amplifier, b) P controller, c) PD controller, d) PI controller and e) PID controller (see e.g. Ogata (1997); Van de Vegte (1994))

When we start to analyze reasons for this multi-dimensional inflation, we can identify several possible sources and points to discuss:

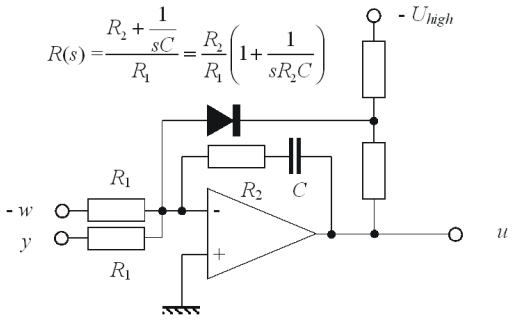


Fig. 3.2 Analog electronic PI controller

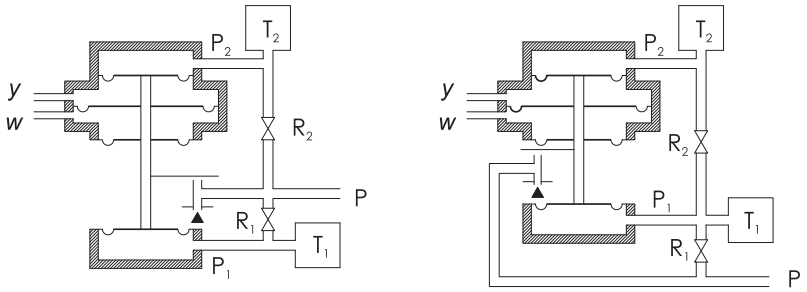


Fig. 3.3 Modifications of the pneumatic PID controller with a parallel (left) and a serious feedback (right) enabling to achieve different ranges of adjustable parameters and different tuning properties (Ogata, 1997)

- PID control is not a closed and unique solution, but result of a not yet finished development,
- conservatism of practice and tendency to work with older (may be out of data) solutions,
- existence of alternative solutions to the specified problems offering different performance,
- failure to analyze the physical essence of the solved problems,
- absence of reliable controllers and their tuning for some typical situations, e.g. for systems with large dead-time, or for unstable systems,
- absence of controllers respecting given control constraints for some typical situations,
- not yet finished development of methods for a reliable (self-) tuning of controllers.

Conservativeness of users is closely related to the historical development of the PID controller technology. In the initial period, large amount of different pneumatic, hydraulic electrical and electro-mechanical devices were spontaneously developed mostly on an experimental bases. Theoretical studies of derived controllers started just later, when practice required a deeper under-

standing of their optimal tuning and when it was necessary to replace older devices by newer electronic controllers (to the end of 1950s) and digital ones (since 1980s).

After 1960, due to the invention of transistors, the older pneumatic controllers started being replaced by newer electronic devices based on high gain operational amplifiers. Their dynamical properties, determined by the feedback impedances are much more transparent and can easily be mathematically described.

After 1960, new wave of digital controllers started. Around 1980 it is already to observe fast invasion of digital quasi-continuous microprocessor based controllers. They work with relatively short sampling periods that can frequently be neglected and the controllers may be considered as the continuous-time ones. However, it is to remember that by sampling a high frequency noise signals at the input of the analogue to digital (A/D) converters, a low frequency signals may appear. In the older analogue controllers the controller inertia naturally filtered these. The study of digital controllers brought to light necessity of introduction of new anti-aliasing filters.

Comparing with the analogue control, the fundamental and up to day not fully used feature of digital control is its flexibility and broad functionality. One of its exceptional features is an easy implementation of dead time that is very important for its compensation in control loops. It was required by solutions as the Smith predictor (Smith, 1957), or a bit older controller by Reswick (1956). For the analogue controllers, implementation of the dead time required in its compensation represented a serious technical problem. Due to this, practice has motivation to use simpler PI controllers instead of them. Reaction to the new situation still does not correspond to the well-known fact that the majority of processes can be approximated by the first order models with dead time!

Introduction of digital controllers gave birth to new phenomenon called *windup*. Why it was not recognized earlier? May this fact be explained so that in the digitalization phase the older solutions robust against windup were not described fully correctly? Effect of the waves of innovation on the anti-windup control circuitry is in a catching way described in the book "Control Systems with Input and Output Constraints" by Glattfelder and Schaufelberger (2003). They show several examples from the field of power control, by which they demonstrate problems arising by replacing older generations of controllers by newer. They show that it is sometimes simpler to imitate by new solutions the old pneumatic controllers than to invest into reengineering of the whole technological complex. Related back to the already existing solutions, this argumentation can be understood. However, it should not be acceptable for newly designed solutions, when the new controllers give much broader possibilities!

From the application point of view it has to be noted that the first generation of controllers was mostly designed to compensate effect of disturbances acting in the vicinity of fixed operating points. When a transition to a new

operating point was required, it was either done under manual control, or by special units. So, the first tuning rules and strategies were devoted to optimal compensation of disturbances. This category is e.g. represented by the most popular tuning rules by Ziegler and Nichols (1942). The optimal behavior corresponding to the setpoint changes was focused just later. Furthermore, the first pneumatic controllers were constructed in such way that they did not initiate excessive controller windup. This started to be dominating just for newer generation of digital controllers, what resulted also in corresponding research work (see e.g. Fertik and Ros (1967); Kramer and Jenkins (1971)), when more important results appear just around 1970.

Development of the technology of PID control has, of course, influenced also the development of the control theory. The today frequently ventilated gap between the theory and practice has several resources: e.g. the generally accepted internal classification of PID control does not reflect all basic situations occurring in practice. It seems that this gap reasonably increased after replacing the first generation of experimentally designed controllers by more transparent and easily describable electronic and digital ones, when some important construction details were neglected and forgotten. The other point is that the control theory developed into an independent discipline what has brought also several self-centered features and artificial problems that do not respond to real needs. Failures in solving real problems lead many researchers to leave the traditional analytical controllers and to look for new solutions based e.g. on fuzzy control, neural networks, genetic algorithms or optimization based predictive control. Although these new solutions bring many new interesting and useful features and options, it does not mean that a theory describing PID control becomes obsolete. Many fictitious advantages of the new approaches represent, in fact, just the not sufficient knowledge of the possibilities of the traditional ones. But, what should be improved in the traditional approach? At first, we should understand more deeply motivations that gave birth to these structures. This will also need to introduce performance specification that will be used for evaluating, if the controller design is meeting as close as possible practical requirements.

3.3 Advanced Modifications of PID Control

Next we will briefly show some newer modifications of the PI and PID controllers to illustrate broad spectrum of existing solutions that will be stepwise explained in this book. Going back to the first pneumatic PI controllers, their structure may be represented by feedback from the controller output through a low pass filter (Åström and Hägglund, 1995) in Fig. 3.4.

In the proportional zone of control, when the saturation limits are not active, the controller transfer function becomes

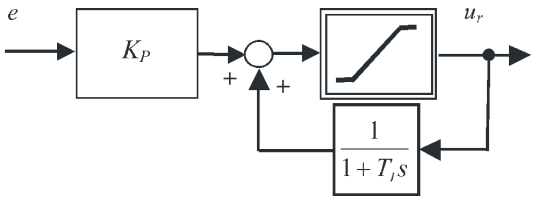


Fig. 3.4 Serial implementation of the PI controller considering control signal constraints

$$R(s) = K_P \frac{1}{1 - \frac{1}{1+T_I s}} = K_P \left(\frac{1}{1 + T_I s} \right) \tag{3.7}$$

Another broadly used modification of the PI controller denoted usually as the I-P controller uses the proportional feedback acting just on the plant output (Fig. 3.5 left). It may be shown to be a special case of setpoint weighting (with $b = 0$, $T_D = 0$ in (3.3), or to be equivalent to the PI controller with the input filter (prefilter) in Fig. 3.5 right.

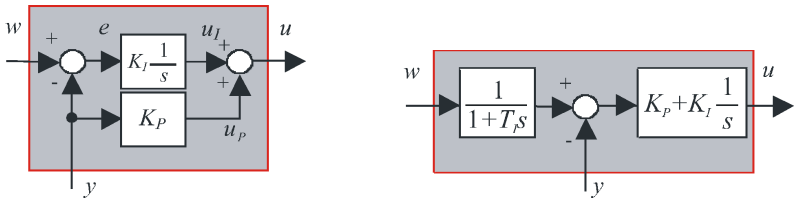


Fig. 3.5 I-P controller (left) and the equivalent PI controller with prefilter (right), $K_I = K_P/T_I$

Similarly, by using PD terms acting on the output only may give the I-PD, or PI-PD controllers.

For controlling stable first order systems with long dead time T_D, the Predictive PI controller (PPI, Hägglund (1996); Fig. 3.6 left) is used. This may be extended by a PD in the feedback path (Fig. 3.7 right) to the PID τd controllers (Shinskey, 2000). Both may be extended by the IMC filter, or by a prefilter, when one e.g. gets the structure of the Model Driven PID controller (Shigemasa et al, 2002; Yukitomo et al, 2002) with the IMC filter and the prefilter in Fig. 3.7 (above) that is equivalent to the PPI-PD controller in Fig. 3.7 (below)

All above mentioned structures may be covered by the 2 degree of freedom (2DOF) MD-PID controller with the 2nd order IMC filter and the prefilter in Fig. 3.8 (Shigemasa and Yukitomo, 2004; Yukitomo et al, 2004). They document that the PID control developed is far from to be finished and be interpreted just by the transfer function (3.2). Obviously, systems with long dead-time are being integrated as a part of the general PID control.

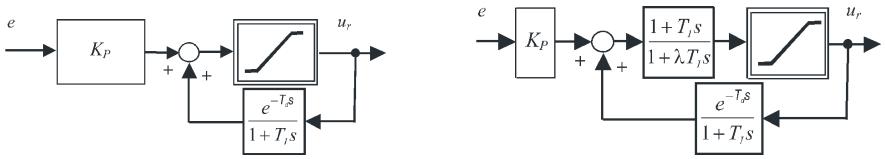


Fig. 3.6 Predictive PI controller (PPI) (left) and extended by the IMC controller (right)

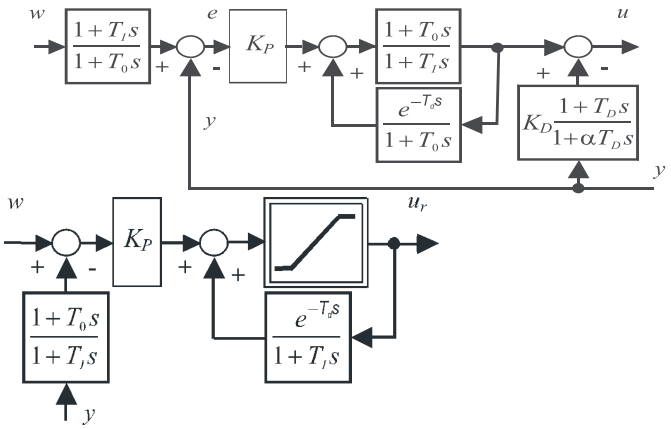


Fig. 3.7 Model Driven PID controller (above) (Shigemasa et al, 2002; Yukitomo et al, 2002) that is equivalent to the PPI-PD controller (below)

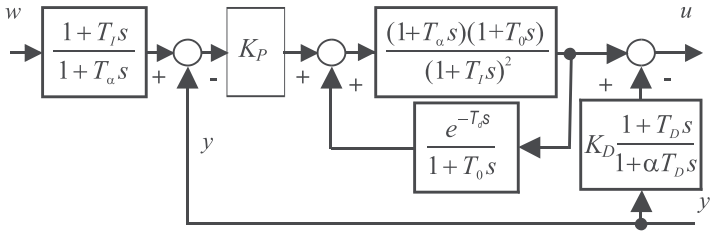


Fig. 3.8 2DOF MD-PID controller

As it is obvious from the title "Model Driven" PID controller, the plant model played an important role in derivation of previous controllers. One may speak about approaches using the plant model at least from late 1950s, when the first schemes for dead time compensation by Reswick (1956) and Smith (1957) appeared. Both were based on reconstruction of an output disturbance by a parallel plant model and the reconstructed disturbance was then used for compensation of the reference setpoint value. Use of the parallel plant model was later generalized within the Internal Model Control concept (Morari and Zafiriou, 1989) that developed its own structure and tuning approaches to

the PID control (Rivera et al, 1986; Skogestad, 2003) used frequently within the robust control.

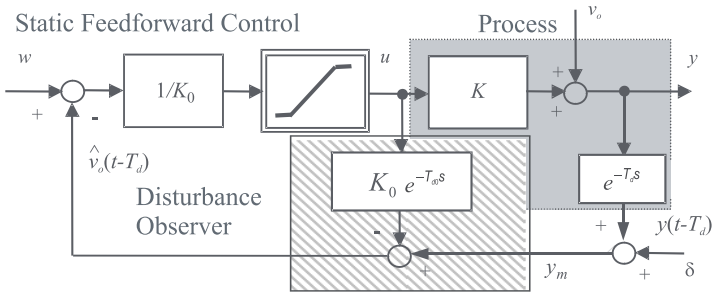


Fig. 3.9 Controller for dead time compensation by Reswick (1956)

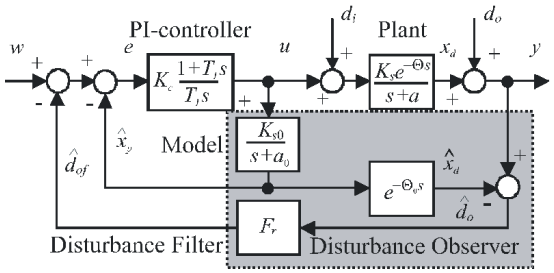


Fig. 3.10 Filtered Smith predictor for dead time compensation; Smith (1957) proposed this scheme with $F_r(s) = 1$

All above controllers may be considered as different modifications of the IMC control derived for reconstruction and compensation of output disturbances that were dominantly used in process control. This area is typically dealing with stable processes, whereby the measured signals may be rather poor. The remarkable progress in mini- and microcomputers and power electronics technology in the 1980's made it also possible to improve the performance of motion control, what consequently lead to testing of traditional and novel theories of control appropriate for mechatronic systems. In this area with dominant influence of the input (load) disturbances of frequently unstable, or marginally stable plants, but the relatively high quality measured signals, much more frequently the so called Disturbance Observe based servo systems (Ohnishi , 1987; Ohnishi et al, 1987; Umeno and Hori, 1991), or the Disturbance Observer based PID control (Zhao, 2004) are used. This approach that is based on inversion of the model dynamics was also extended to systems with long dead time (Zhong and Mirkin, 2002; Zhong and Normey-Rico, 2002).

The new textbook on Robust Constrained PID control tries to compare both approaches based on reconstruction and compensation of input and output disturbances and the traditional approach to the PID control more systematically, what requires to adopt also some terminology changes. Due to the fact that also the IMC control actually uses DO for reconstructing output disturbances and both the IMC and DO based PID control are internally using plant models, where appropriate, the PID structures for reconstruction and compensation of input disturbances will be denoted more eloquently as the PID-IM (Inverse Model) controllers and the structures for reconstruction and compensation of output disturbances as PID-PM (Parallel Model) controllers. Of course, the question is, if this was the best choice that will enable a modular terminology development appropriate to cover also possible modifications with different mixed forms of solutions, but answers to this question will bring just the future development. With respect to this, author of this chapter will be thankfull for any comments regarding these proposals.

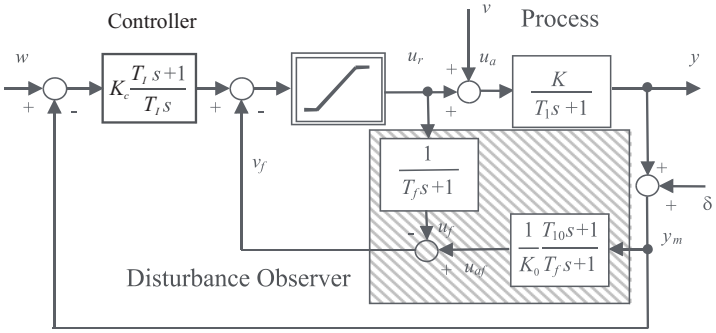


Fig. 3.11 Disturbance Observer (DO) based servo-system proposed by (Umeno and Hori, 1991) and interpreted by (Zhao, 2004) as the DO-PI controller

3.4 Performance of PID Control

Traditionally, PID control design may be carried out by using closed loop specifications in the time domain or in the frequency domain (Skogestad and Postlethwaite, 1996). In this book we will prefer the first ones, since their application in computer based design that would be based on exploiting information on the closed loop properties is extremely simple and straightforward. For characterizing the closed loop dynamics, we will use several qualitative and quantitative measures.

3.4.1 Settling Time t_s, IAE, TV, TV_0, TV_1 and TV_2

To characterize quality (speed) of control transients different performance indices are used as e.g. settling time, Integral of Absolute Error (IAE), Integral of Squared Error (ISE), or Total Variance (TV), whereby all these measures may be considered separately or in different logical combinations. Since we are always required to finish a control process in a limited time, it might seem that the basic performance index for process control should be defined as the *settling time*

$$y(t) - w = 0, \quad \forall t \geq t_s, \quad y_0 = y(0) \neq w \tag{3.8}$$

i.e. as time t_s required to reach by the output signal $y(t)$ starting from initial value $y_0 = y(0)$ a given setpoint value w. In general, however, the opposite is true. Here, we will exclusively deal with control problems that after a transient response to new reference state require maintaining system at its vicinity in steady state. In linear systems, transients to a constant setpoint value are theoretically infinitely long. So, finite settling time requires definition of certain neighborhood around it (Fig. 3.12). Such requirement also follows from the fact that all real control loops work with finite precision of measurement. To decide, when a transient finished by reaching steady state lying within defined neighborhood around reference value becomes yet more delicate problem in a noisy environment. Steady states can e.g. be indicated by fulfilling requirements put both on the plant input and output

$$|y(t) - w| \leq \varepsilon_y \cap |u(t) - u_w| \leq \varepsilon_u, \quad \forall t \geq t_s \tag{3.9}$$

whereby the control signal value u_w corresponds to maintaining output at the setpoint value w and parameters $\varepsilon_y, \varepsilon_u$ may follow from particular technology (measurement noise & required control precision). Alternatively, they should be chosen in such a way to prevent a premature indication of steady state at flat extreme points of oscillatory transients. In general, plant output and plant input (controller output) may achieve steady states at different time moments, a nearly fixed output value may be achieved by oscillation at the input (dynamical steady state), there may exist steady states with nonzero steady state error, etc.

Since the settling time indication (3.9) depends on definition of several parameters, in order to characterize speed and duration of transients at the plant output in a simpler way, IAE (Integral of Absolute Error) or ISE (Integral of Squared Error) performance indices are frequently used defined as

$$\mathrm{IAE} = \int_0^\infty |[e(t) - e(\infty)]| \, \mathrm{d}t, \quad \mathrm{ISE} = \int_0^\infty [e(t) - e(\infty)]^2 \, \mathrm{d}t \tag{3.10a}$$

$$\mathrm{IAE} = \int_0^\infty |[y(t) - w]| \, \mathrm{d}t, \quad \mathrm{ISE} = \int_0^\infty [y(t) - w]^2 \, \mathrm{d}t \tag{3.10b}$$

where $e(\infty) = \lim_{t \to \infty} e(t)$

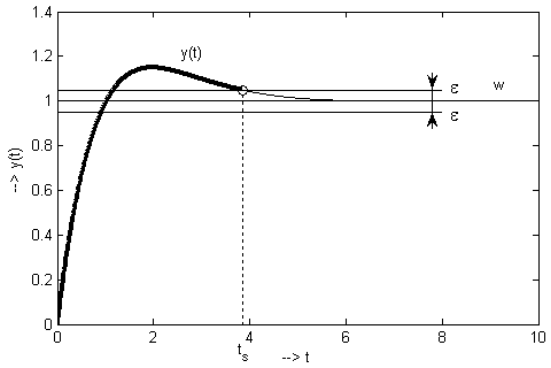


Fig. 3.12 Definition of settling time t_s (3.9) based only on the plant output $y(t)$ with $\varepsilon = \varepsilon_y$

The first definition is usually preferred in situations, when some permanent error is allowed, but it should not lead to a permanent increase of the integral values. The second definitions are appropriate for situations, where it is important to avoid permanent error.

With respect to problems with evaluating absolute value in analytical computations, i.e. due to the mathematical convenience, ISE is the criterion most frequently used by theoreticians for the analytical controller optimization. It is, however, also well known that such optimization underestimates small error values and leads to oscillatory transients. Therefore, with respect to practical requirements, in this book we are going to use dominantly IAE performance index, since. IAE is a good performance measure because the size and length of error is proportional to lost revenue (Shinskey, 1990). Because in optimizing controllers also minimal IAE values may correspond to transients with some overshooting, when aiming at monotonic transients, or transients without overshooting, it is not enough to look just for minimum of IAE, but one has to define also additional design constraints.

The required output behavior can generally be achieved by different transients of the manipulated variable at the controller output. Therefore, it is useful to evaluate also Total Variance (TV), a criterion (Skogestad, 2003) introduced for characterizing "smoothness" and total "energy consumptuion" at the controller output. This was defined as

$$\mathrm{TV} = \int_0^\infty \left| \frac{du}{dt} \right| \mathrm{d}t \approx \sum_i |u_{i+1} - u_i| \tag{3.11}$$

Also this is mostly difficult to be evaluated analytically and therefore it is usually computed experimentally after appropriate discretization with sampling period as small as possible.

3.4.2 Basic Qualitative Shapes of Transient Responses

To describe qualitative properties of transients of PID control that may be composed from several exponentials, or periodic functions, we will introduce following definitions:

Definition 3.6 (Nonovershooting (NO) and Nonundershooting (NU) functions). Function of time $f(t)$ with initial value $f(0) = \lim_{t\to 0^-} f(t)$ different from its final value $f(\infty) = \lim_{t\to\infty} f(t)$ fulfilling conditions

$$\begin{aligned} [f(t) - f(\infty)]\,\mathrm{sign}\,\{f(0) - f(\infty)\} \geq 0 \quad \forall t \geq 0 \\ [f(t) - f(0)]\,\mathrm{sign}\,\{f(\infty) - f(0)\} \geq 0 \quad \forall t \geq 0 \end{aligned} \tag{3.12}$$

will be denoted as NonOvershooting (NO) and NonUndershooting (NU) function.

NO output property may follow from safety and technology requirements. It is important in many technologies, as e.g. in controlling machine tools, in traffic and flight control tasks, etc. In controlling systems with dead-time, nonovershooting properties may become different from the monotonic ones.

Definition 3.7 (Monotonic (MO) function). Function of time $f(t)$ with initial value $f(0) = \lim_{t\to 0^-} f(t)$ different from its final value $f(\infty) = \lim_{t\to\infty} f(t)$ and preserving direction of changes

$$[f(t_2) - f(t_1)]\,\mathrm{sign}\,\{f(\infty) - f(0)\} \geq 0 \quad \forall t_2 > t_1 \geq 0 \tag{3.13}$$

will be denoted as MOnotonic (MO) function.

Obviously, MO function is also NO and NU function, but not conversely. Monotonic functions typical for PID control may e.g. be given as $f(t) = 1 - \mathrm{e}^{-t/T_1}$; $y(0) = 0$; $y(\infty) = 1$, whereby $T_1 > 0$ is the time constant describing how fast the signal approaches new steady state value $y(\infty)$. For $t = T_1$ it should be at 63% of $y(\infty)$. By limiting $T_1 \to 0$ one gets from this exponential *step function* that so may represent *limit case of MO functions.*

MO transients at the controller output (plant input) and at the plant output may be motivated by energy savings in actuators, by minimizing their wear, generated noise and vibrations, by comfort of passengers in traffic control, or by precision increase in controlling systems with actuator hysteresis. MO controller output will also be expected to yield the lowest possible TV values.

Definition 3.8 (One-Pulse (1P) function). Function of time $f(t)$ that is continuous for $t > 0$ (with possible discontinuity at the origin) with initial value $f(0) = \lim_{t\to 0^-} f(t)$ and having with respect to the finite steady state value $f(\infty) = \lim_{t\to\infty} f(t)$ just single extreme point $f_m = f(t_m) \neq f(0)$ at $t_m \geq 0$, whereby it fulfills conditions

$$\begin{array}{l}[f(t_2)-f(t_1)]\,\mathrm{sign}\,\{f(t_m)-f(0)\}\geq 0\\ [f(t_4)-f(t_3)]\,\mathrm{sign}\,\{f(\infty)-f(t_m)\}\geq 0\end{array},\ \text{for } 0\leq t_1<t_2\leq t_m\leq t_3<t_4<\infty \tag{3.14}$$

will be denoted as One-Pulse (1P) function.

Obviously, 1P function may be defined as function with one extreme point that is MO before and behind this extreme point. By allowing discontinuity of $f(t)$ at the origin, e.g. for $f(t)=\mathrm{e}^{-t}\mathbf{1}(t)$ the extreme point may also move to origin from the right, when $t_m=0^+$, whereby the interval before the extreme point shrinks to zero.

Examples of 1P functions may be represented by single exponential $f(t)=\mathrm{e}^{-t}\mathbf{1}(t)$ that has extreme point $f(0^+)=1$ and discontinuity at the origin, or by difference of two exponentials $f(t)=\left(\mathrm{e}^{-t}-\mathrm{e}^{-2t}\right)\mathbf{1}(t)$ having extreme $f_m=1/4$ at $t_m=\ln 2$

Definition 3.9 (Two-Pulse (2P) function). Function of time $f(t)$ that is continuous for $t>0$ (with possible discontinuity at the origin) with initial value $f(0)=\lim_{t\to 0^-}f(t)$ that is having two extreme points $f_{m1}=f(t_{m1})\neq f(0)$ and $f_{m2}=f(t_{m2})$ at $t_{m2}>t_{m1}>0$ with respect to the finite steady state value $f(\infty)=\lim_{t\to\infty}f(t)$ and fulfilling conditions

$$\begin{array}{c}[f(t_2)-f(t_1)]\,\mathrm{sign}\,\{f(t_{m1})-f(0)\}\geq 0\\ [f(t_4)-f(t_3)]\,\mathrm{sign}\,\{f(t_{m2})-f(t_{m1})\}\geq 0\\ [f(t_6)-f(t_5)]\,\mathrm{sign}\,\{f(\infty)-f(t_{m2})\}\geq 0\\ \text{for } 0\leq t_1<t_2\leq t_{m1}\leq t_3<t_4\leq t_{m2}\leq t_5<t_6<\infty\end{array} \tag{3.15}$$

will be denoted as Two-Pulse (2P) function.

Obviously, 2P function may be defined as function with two extreme points that is MO on each interval not including one of them. By allowing discontinuity of $f(t)$ at the origin, the first extreme point may also move to origin from the right, when $t_{m1}=0^+$, whereby the interval before this extreme point shrinks to zero. Example of such a function may again be given by difference of two exponentials $f(t)=\mathrm{e}^{-2t}-\mathrm{e}^{-t}$; $f(t)=0$ with $f_{m1}=1$ at $t_{m1}=0^+$ and $f_{m2}=1/8$ at $t_{m2}=\ln 4$.

By generalizing previous definitions to get a unique term for all above functions, we may come to notion of nP function. Within this text it will be mostly constraint to 0P, 1P and 2P functions.

Definition 3.10 (n-Pulse (nP) function). Function of time $f(t)$ that is continuous for $t>0$ (with possible discontinuity at the origin) with initial value $f(0^-)=\lim_{t\to 0^-}f(t)$ that is having with respect to the finite final value $f(\infty)=\lim_{t\to\infty}f(t)$ n extreme points $f_{mi}=f(t_{mi})$, $i=1,\ldots,n$ at $0<t_{m1}<\cdots<t_{mn}$ and is MO on each interval not including one of these extreme points will be denoted as n-Pulse (nP) function. Again, by allowing discontinuity of $f(t)$ at the origin, the first extreme point may also move to zero from the right, when $t_{m1}=0^+$, whereby the first MO interval before this extreme point shrinks to zero.

By introducing notion of nP function it is so possible to denote MO function as 0P one. Since by limiting values of nP function to any interval containing $f(\infty)$ one does not change number of extreme points, it can also be used in constrained control. After achieving saturation limits, by decreasing duration of MO intervals among particular saturation pahses, nP functions may approach rectangular (relay) n-pulses of discontinuous MTC, but for $t > 0$ they always remain continuous in time.

To cover whole spectrum of transients typical for PID control we should yet complete the above list by definition of periodic functions interpreted as nP function with $n \to \infty$. Then, after specifying the damping ratio (as e.g. by Ziegler and Nichols (1942)) we could treat also oscillatory loop behavior. But, with respect to available space, within this text we will deal just with finite values of n.

3.4.3 Quantifying Qualitative Measures

By identifying properties like loop stability, NO, NU, MO, or nP shape of particular variable one typically get binary (true/false) information. On the other hand, performance indices like IAE or TV (3.8)–(3.11) give quantitative information about the loop behavior that enables refined evaluation of its quality. However, in control engineering it is frequently required to quantify also the above mentioned binary information, e.g. by expressing how far the system from stability, nonovershooting, or monotonicity border is. In the frequency domain there are broadly used robust design methods based on assigning stability degree, gain, phase and stability margin (see e.g. Anderson and Moore (1969); Datta et al (2000); Skogestad (2003); Skogestad and Postlethwaite (1996). Similarly, it is possible to introduce such quantitative measures for stability, nonovershooting, nonundershooting and monotonicity also in the time domain.

Quantitative measures for stability: In the time domain, stability or more precisely instability degree can be indicated in different ways – e.g. by requiring limited output value

$$|y(t)| < T_{\max} < \infty; \quad \forall t > 0 \tag{3.16}$$

by limiting possible settling time, IAE, ISE or TV values, maximal overshooting, by decreasing damping ratio, etc. When these measures increase over some predefined values chosen e.g. as a multiple of optimal value, or with respect to technological constraints, transients may be denoted as unstable. Despite this step seems to be mathematically vague, in fact it matches requirements of practice much closer than the usually used stability definition based on closed loop poles position – one can easily find example of stable

system that is not usable in practice because of extremely high amplitudes of inner signals.

With respect to finite measurement precision and to quantization typical for digital signals, it is more realistic to relate NO and NU signal properties not to a precise final value, but to an error band specified symmetrically (having width 2ε) around supposed final, or initial value. Then, for increasing signals over- and undershooting is signalized just after crossing this band. By introducing several levels ε it is possible to replace the binary (true/false) information by more detailed quantitative information telling e.g. that under measurement (evaluation) precision 2% of the maximal output value our system response may be considered as 0.02-NO, but this already does not hold for precision defined as 1% of the maximal output signal value. Similarly, for increasing, or decreasing signals it is possible to weaken strict monotonicity by introducing final evaluation precision into the monotonicity tests (Fig. 3.13).

Definition 3.11 (ε-NO and ε-NU functions). A continuous signal $f(t)$ with the initial value $f_0 = f(0)$ and with the final value $f_\infty = f(\infty)$ will be denoted as ε-nonovershooting, or ε-nonundershooting, when it fulfills conditions

$$\begin{matrix}[f(t) - f(\infty)]\,\mathrm{sign}\,\{f(0) - f(\infty)\} \geq -\varepsilon \\ [f(t) - f(0)]\,\mathrm{sign}\,\{f(\infty) - f(0)\} \geq -\varepsilon\end{matrix} \quad \forall t \geq 0,\ \varepsilon > 0 \tag{3.17}$$

Definition 3.12 (ε-MO function). A continuous nearly MO signal $f(t)$ with the initial value $f_0 = f(0)$ and with the final value will be denoted as ε-monotonic when it fulfills condition

$$[f(t_2) - f(t_1)]\,\mathrm{sign}\,\{f(\infty) - f(0)\} \geq -\varepsilon;\ \forall t_2 > t_1 \geq 0;\ \varepsilon > 0 \tag{3.18}$$

To simplify program implementation, requirements (3.18) may be evaluated digitally by working with relatively small sampling period and by comparing just finite number of subsequent values $f(i)$ and $f(i+k)$, $k = 1, 2, \ldots, K$, whereby $K = T_h/T$, T being the sampling period, is chosen to cover at least one half-period T_h of possible high-frequency signal superimposed on the dominant monotonic signal

$$\begin{matrix}[f(i+1) - f(i)]\,\mathrm{sign}\,\{f(\infty) - f(0)\} \geq -\varepsilon; \cap \ldots \\ \ldots \cap ([f(i+K) - f(i)]\,\mathrm{sign}\,\{f(\infty) - f(0)\} \geq -\varepsilon) \\ \varepsilon > 0, \quad K \geq 1, \quad i = 1, 2, \ldots, \infty\end{matrix} \tag{3.19}$$

Whereas (3.17)–(3.19) characterize amplitudes of superimposed high-frequency signals, deviations from strict monotonicity may also be characterized by limiting new integral measure that gives total contribution of high-frequency deviations (proportional not just to the amplitude, but also to the number of peaks) denoted as TV_0

$$\mathrm{TV}_0 = \sum_i |u_{i+1} - u_i| - |u(\infty) - u(0)| < \varepsilon_0 \tag{3.20}$$

TV_0 takes zero values just for strictly MO control signal transients.

In this way, it may be interesting to apply this criterion both to the plant output and to the plant input signals. Testing of amplitude deviations according to (3.19) may be reasonably simplified due to the following Lemma.

Lemma 3.1. *Constrained continuous signal $f(t)$ having initial value $f_0 = f(0)$ and final value $f_\infty = f(\infty)$ with local extreme points $f_{lei} = f(t_{lei})$ is ε-monotonic, if all subsequent local extreme points f_{lei} fulfill condition*

$$|y_{le,i+1} - y_{le,i}|\, sign\,(y_\infty - y_0) \geq -\varepsilon_y, \quad i = 1, 2, 3, \ldots \tag{3.21}$$

Proof. Follows from the fact that the maximal signal increase in the direction opposite to $y_\infty - y_0$ in (3.18) will be constrained by two subsequent extreme points. interesting to apply this criterion both to the plant output and to the plant input signals.

Definition 3.13 (ε-nP function). 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

$$|f_{mni} - f(\infty)| > \varepsilon; \quad f_{mni} = f(t_{mni}),\ i = 1, \ldots, n \text{ at } 0 < t_{mn1} < \cdots < t_{mnn} \tag{3.22a}$$

$$[f_{mni} - f(\infty)]\,[f_{mn,i+1} - f(\infty)] < 0,\ i = 1, \ldots, n-1 \tag{3.22b}$$

that is ε-MO on each of $n+1$ intervals not including one of these extreme points will be denoted as the ε-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 $(0, t_{tmn1})$ before this extreme point shrinks to zero.

Nearly nP-dynamics may also be specified by limiting TV_n values that take zero value exactly for signal consisting of $n+1$ monotonic intervals devided by n extremes. E.g. for signals with 1P dominant control it is possible to work with limited TV_1 criterion defined according to

$$\mathrm{TV}_1 = \sum_i |u_{i+1} - u_i| - |2u_m - u(\infty) - u(0)| < \varepsilon_1 \tag{3.23}$$

that takes zero values just for strictly 1P control signal, whereby it does not depend on possible control signal constraints. For control signals with superimposed higher harmonics it takes positive values.

Similarly, for systems with dominant 2P control function the acceptable contribution of higher harmonics may be limited by

$$\mathrm{TV}_2 = \sum_i |u_{i+1} - u_i| - |2u_{m1} - 2u_{m2} - u(\infty) - u(0)| < \varepsilon_2 \qquad (3.24)$$

For ideal 2P control functions it yields value $\mathrm{TV}_2 = 0$

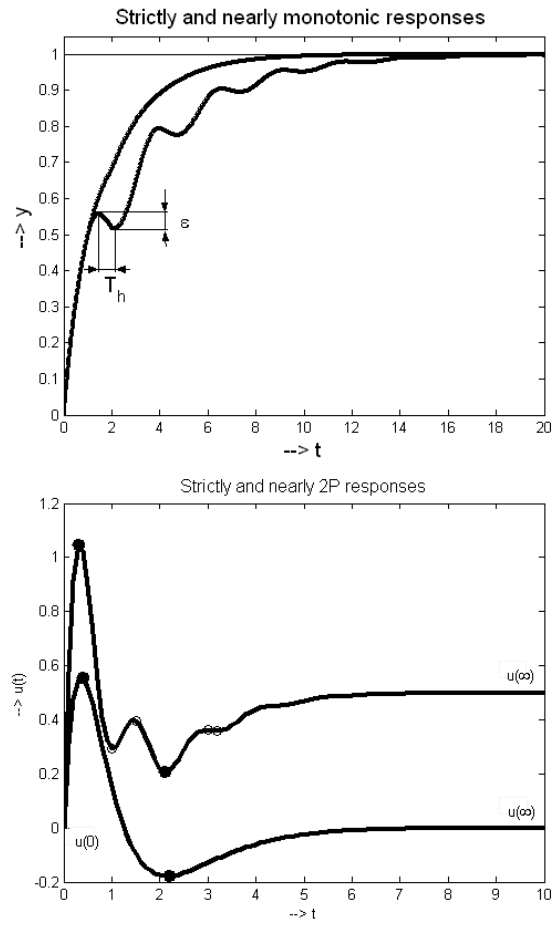


Fig. 3.13 Above: Strictly monotonic signal satisfying (1.12) and "nearly monotonic" signal satisfying (1.18) with $K = T_h/T$, T being the sampling period, is chosen to cover at least one quarter-period T_h of possible high-frequency signal superimposed on the dominant monotonic signal $\varepsilon = \varepsilon_y$; Below: Nearly and strictly 2P responses; local extreme points denoted by "o" and significant extreme points denoted by "•"

In applying weakened versions of ε-NO, ε-MO, or ε-nP properties it is, however, to remember that achieved information depends on ε – e.g. with acceptable overshooting 1% a transient may be classified as MO, but for acceptable overshooting 0.1% as 1P function. From one point of view it is quit

normal that under final measurement precision one is not able to distinguish these two properties, when the error is below the system resolving power. On the other hand, it is clear that these weakened versions should be applied carefully with tolerances not exceeding acceptable measurement (evaluation) precision, otherwise one get unexpected and unusable results.

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 plants the number of control pulses cannot decrease below the number of unstable poles.

NO specifications (not distinguishing between nonovershooting and monotonic control) exist also in the frequency domain (see e.g. Keel et al (2008)) but their application is extremely complicated, especially when speaking about dead time systems. Specific measure for deviations from monotonicity was also introduced by Åström and Hägglund (2004). Here, we have preferred new measures for deviations from NO, MO and nP function properties that may not only be tested numerically, by evaluating simulated or experimentally measured transients corresponding to the setpoint and disturbance step responses, but they represent a modular system and are also appropriate for constrained control. These specifications may be hierarchically organized into trees, whereby the closed loop stability will be considered as the root property, NO, NU, MO and nP as child properties. Graphically represented in the plane of loop parameters, together with quantitative measures, such properties will be giving *performance portrait* of particular control loop.

3.4.4 Performance Portrait (PP)

The closed loop PP represents information about the closed loop performance corresponding to setpoint and disturbance step responses expressed over a grid of (possibly normalized normalized) loop parameters. For a loop represented by a D-dimensional parameter vector $P = \{p_1, p_2, \ldots, p_k, p_{k+1}, p_{k+d}\}$; $D = k + d$, whereby some part of parameters p_i; $i = 1, \ldots, k$ is a priori given, some parameters will be fixed during the loop analysis and $p_i \in [p_{imin}, p_{imax}]$; $i = k_1, \ldots, k + d$ may vary over some (known) intervals. So, the performance portrait will be considered in the space with the total dimension D, whereby the variable parameters forming subspace with the dimension d will take levels $p_{i,j} = p_{imin} + (p_{imax} - p_{imin})j/n_i$; $j = 1, 2, \ldots, n_i > 1$.

By containing information about required loop properties PP may be used both for optimally localizing a nominal operating point by appropriate controller tuning, or for optimally localizing an uncertainty set of all possible operating points corresponding to specified intervals of variable loop parameters. When e.g. working with the plant model (3.1) the loop parameters are K_s, a_0, a_1, T_0, T_d. Many control method are based on inversion of the plant model, whereby model parameters could be denoted as $K_{s0}, a_{00}, a_{10}, T_{00}, T_{d0}$.

Inversion will require at least first order filter with a time constant T_f. Specification of the setpoint response will require determination of at least one time constant T_w. It means that in total there are 12 parameters that determine the resulting dynamics. If e.g. two of them, say $K_s \in [K_{s,min}, K_{s,max}]$, $T_d \in [T_{d,min}, T_{d,max}]$ are variable, the task of the control design will be to choose appropriate model parameters $K_{s0}, a_{00}, a_{10}, T_{00}, T_{d0}$ and free design parameters T_s, T_w in such a way that over all points over the uncertainty set corresponding to $K_s \in [K_{s,min}, K_{s,max}]$, $T_d \in [T_{d,min}, T_{d,max}]$ chosen according to $p_{i,j} = p_{imin} + (p_{imax} - p_{imin})j/n_i$; $j = 1, 2, \ldots, n_i > 1$ required performance measures will be achieved. PP required for such a design may be generated by simulation, or by real time experiments. Although its generation may be connected with numerical problems, especially those related to the nature of grid computations, when one has to balance precision of achieved results (quantization level in considered grid) with the total number of evaluated points and the corresponding computation time, it gives very promising results especially when dealing with dead time systems.

3.5 Dynamical Classes (DC) of Control

By introducing qualitative measures for transient responses, we are now able to categorize all PID controllers that are able to yield MO step responses at the plant output according to the shape of their manipulated (control) variable. If this has properties of nP functions, we will include the corresponding control into the dynamical class of control with index n, shortly DCn.

Today, also people without a background in optimal control understand that if they wish to move with their car monotonically from one point to another they have at least once accelerate (it means to increase kinetic energy of the car) and then to brake (decrease the energy). Or, if they wish to charge a container, they must open the input valve for some interval of time. So, control processes are by its nature related with energy accumulation, or dissipation and the transients are expected to be the fastest one if they are related with maximal values of the manipulated variable. This fact was reflected by the Feldbaum's theorem (Feldbaum, 1965) published firstly in 1949.

Theorem 3.1 (Feldbaum's Theorem). *For the MTC of the n-th order system from one constant reference output value to another one there are required n-intervals of optimal control, when the control signal step-by-step changes from one limit value to the opposite one.*

Despite the fact that later works (by Bushaw (1958); Pontryagin et al (1964), etc.) showed that in controlling oscillatory systems and initial states sufficiently far from the required ones the total number of intervals can also be higher (see e.g. Athans and Falb (1966)), by restricting our treatment to

monotonic output transients from a steady state to another one, Feldbaum's theorem still represents one of the corner stones of optimal control. But when examining majority of existing textbook on PID control, about n-interval of optimal control you are going to find practically nothing - one of few positive exception is the already mentioned book by Glattfelder and Schaufelberger (2003).

It is true that in practice it is just rarely permitted to apply control dealing exclusively with limit control values and with their instantaneous changes. The "rectangular" pulses with sharp rims excite in controlled systems theoretically an infinitely broad frequency spectrum of higher harmonics. Excitation of higher harmonics could cause ineligible reactions. Furthermore, we are mostly not able to perform such control perfectly, and such a control is usually not acceptable with respect to the technological constraints. An admissible rate of the control signal changes or an admissible acceleration use to be constrained by construction, or have to be respected by control. So, if the physical substance of the Feldbaum's theorem has to be respected, then in modified "softer" versions, when the control pulses will not be rectangular, but continuous in time – i.e. somehow "rounded". Besides of the amplitude constraints, one has to respect also rate constraints, or even constraints on higher-order control signal derivatives. Under such constraints it may happen that some interval does not take the limit value, or even some of them fully melt away from the control responses. To cover also such "softer" control responses and to distinguish them from the rectangular train of pulses of the MTC it is then better to speak about *dynamical classes of control* and about corresponding *fundamental* controllers establishing bridges between smooth linear PID control and nonlinear discontinuous MTC.

The aim of this chapter is to explore dynamics of PID control and make it compatible also with the MTC. In relay MTC of simple plants, output responses corresponding to transition from one steady state to a new reference steady state are typically monotonic. The already mentioned exception of systems with complex roots with larger initial deviation from final state is not relevant for such a problem. The rectangular shape of control signal in relay MTC is, however, possible just in a limit case of control

- without constraints on the rate of the control signal and/or on its higher derivatives,
- with negligible nonmodelled dynamics,
- for negligible fluctuations of plant parameters and
- for negligible measurement noise (full information about state).

To respect all these additional factors control signal must become "softer". Thereby some its pulses may not hit the constraints, or even some pulses may fully disappear. So, by stressing importance of the above mentioned limitations, acceptable closed loop dynamics may naturally tend to lower number of control pulses, in the limit case of stable systems to single interval with MO controller output.

Example 3.1 (Smooth control of an n-tuple integrator). When considering stable single integrator $\dot{y} = u_1$ (Fig. 3.14) with output y changing monotonically from an initial value $y_0 = y(0)$ to a final value $y_\infty = y(\infty) > y(0)$, y will be increasing (not decreasing) if its derivative is a positive (non negative) function of time, i.e. $\dot{y}(t) > 0$, or $\dot{y}(t) \geq 0$, $t \in (0, \infty)$. With respect to the plant equation $\dot{y} = u_1$, $t \in (0, \infty)$ 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 satisfy conditions $u_1(0^-) = \dot{y}(0) = 0$ and $u_1(\infty) = \dot{y}(\infty) = 0$ - signal $u_1(t)$ continuous for $t \geq 0$ and satisfying these conditions must take a maximum $u_{m1} - u(t_{m1}) > 0$; $\dot{u}(t_{m1}) = 0$ for some $t_{m1} \in (0, \infty)$. Under constrained control, when the control signal saturates, the maximum value may also be achieved over an interval $t \in [t_{max1}, t_{max2}]$. 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 local extreme points $u_{1e}(t_{ei})$; $i = 1, 2 \ldots$ 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 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 $u(\infty) = 0$.

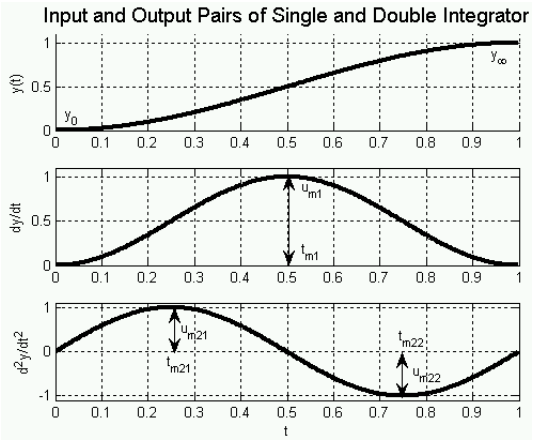


Fig. 3.14 MO output y satisfying (3.18) for $\varepsilon = \varepsilon_y = 0$ (above) with the corresponding 1P input signal of single integrator $u_1(t) = \dot{y}(t)$ (middle), or with the corresponding 2P input signal of the double integrator $u_1(t) = \ddot{y}(t)$ (below)

Similarly, we could treat also a decrease of the setpoint value. 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 given initial and final

conditions and having one extreme point and being monotonic before and after this extreme point. By accepting possible control discontinuity at $t = 0^+$, when the extreme point moves to $t_{m1} = 0^+$, the first MO interval may shrink to zero. However, the MO output increase finishing by reaching steady state cannot be achieved by simpler 0P (e.g. step) control signal.

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. When accepting control discontinuity at $t = 0^+$ the number of intervals may drop to two. But by requiring MO output increase finishing by reaching new steady state control of the double integrator cannot be achieved by simpler input, e.g. by a 0P, or by a 1P signal.

Similar conclusions may also be derived for controlling unstable systems – the number of required pulses cannot be lower than the number of unstable poles. For these systems we get similar conclusions like for the relay MTC.

For stable plant poles poles their influence on the resulting closed loop dynamics may vary. If a plant has only stable poles and its open-loop response is monotonic, then it is always possible to find a controller guaranteeing monotonic closed loop setpoint response at the plant output by a monotonic signal at the plant input. By requiring shorter transients, one extreme point in the setpoint response may become visible, or right two extreme points, etc. A lot will depend on the plant dynamics, on the chosen controller and on required speed of processes. Once you decide to use controller producing at least one extreme point in the control signal, by speeding up the response you may expect problems with the control signal saturation. In order to solve all associated problems, we need to introduce some internal classification of all possible control tasks. This will be achieved by introducing dynamical classes of control.

Definition 3.14 (Dynamical classes (DC) of control). With n being nonnegative integer, under *Dynamical Class n* (shortly DCn) of PID control we understand all control tasks and their solutions with MO plant output and nP plant input.

In characterizing shape of the control signal by nP function n corresponds to the non-negative integer used in denoting number of possible extreme points or intervals with saturated control signal values that also corresponds

to the number of constrained pulses occurring under MTC. Such control signals have to bring the plant output monotonically from one steady state to another one. With respect to Definition 3.1 and the Feldbaum's Theorem it is possible to conclude that all tasks and dynamical processes of the PID control correspond to DC0, DC1 and DC2. What does it mean?

3.5.1 Dynamical Class 0 (DC0)

In this dynamical class, after a step change of reference variable both the manipulated variable (controller output) and the plant output change monotonically, from one steady state to another one.

Definition 3.15 (Step response dynamics of DC0). MO control signal both at the controller and plant output initiated by a setpoint step change characterize step response dynamics of DC0.

Examples of such control signals at controller output are in Fig. 3.15. Limit case of such monotonic transients at the plant input, or output is the step function.

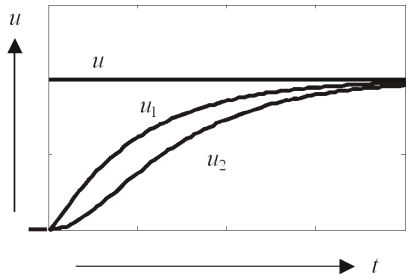


Fig. 3.15 DC0: Control signal reaction to a setpoint step; u – without rate constraints, u_1 – with a rate constraint, u_2 – with constrained 2^{nd} derivative of the control signal

Processes of DC0 can be met in situations, where the dynamics of transients in plant may be neglected, i.e. it is not connected with a reasonable energy, or mass accumulation. Such processes are e.g. typical in controlling flows by valves. After constraining rate of control signal changes, transition to a new control signal value can be exponential one. After constraining also amplitude of the 2^{nd} control signal derivative, the control response takes form of S-function (Fig. 3.15). Since for properly dimensioned actuators and admissible inputs the control constraints will never be active, these control loops are traditionally well treated within the framework of the linear control theory.

It is yet to note that validity of the NO condition (3.12) does not automatically mean that the control transient must be stable. Therefore, condition (3.12) should yet be combined with some measure indicating system stability. Simultaneous fulfillment of monotonicity (3.13) with constrained values at the plant output and input usually fully guarantee also the parent property – BIBO system stability.

3.5.2 Dynamical Class 1 (DC1)

In DC1, for the initial phase of control response initiated by a setpoint step it is typical accumulation of energy in the controlled process. This is associated with a gradual increase (decrease) of the controlled variable that runs most rapidly under impact of the limit control signal value. Control signal may be qualified as one-pulse function.

E.g. by charging a container with liquid, in the first phase of control the input valve should be fully opened, whereas the output value (liquid level) monotonically increases and only in the vicinity of the required level the input flow starts to decrease to a steady state value what will stabilize required output value. Similar transients can frequently be met in speed control in mechatronic systems, in temperature, pressure and concentration control, etc.

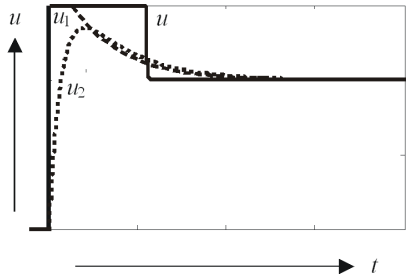


Fig. 3.16 DC1: Control signal reaction to a setpoint step change; u – time optimal (without rate constraints), u_1 – with rate constraints for the transient from the limit to the steady state value, u_2 – as u_1, with an additional limit on the control signal increase.

Definition 3.16 (Step response dynamics of DC1). Dynamics with MO output response and 1P control signal reaction corresponding to a setpoint step change (involving one extreme point, or one control interval with control signal at one of the control signal constraints, Fig. 3.16) will be classified as dynamics of DC1.

From requirement of single extreme point of control signal (one interval at the limit control value) it follows that the transition from initial control signal value to its extreme point and transition from this extreme point to the steady state value u_∞ will be monotonic.

Rectangular pulse of MTC with infinitely short transient from limit control signal value to the steady state value represent limit situation not fully achievable in practice. After limiting rate of changes during the control signal decrease to the steady state (response u_1), the span of the limit control action decreases, but the total length of transient to the new steady state increases. When constraining also the control signal increase (response u_2), the control signal does not catch to reach the limit value, since the necessary control decrease to the steady state has to start yet before it – the length of transient grows further. Whereas the single interval of control is still visible, by constraining rate of the control signal changes the control signal reaction slowly approaches monotonic shape typical for DC0.

With respect to one possible interval with constrained controller output, for dealing with DC1 it is usually not enough to remain within the linear control. Typical solutions for this dynamical class are frequently achieved with different aw - controllers.

3.5.3 Dynamical Class 2 (DC2)

In DC2 output changes are associated with accumulation and recurrence or dissipation of energy required for achieving state and output changes and stabilization at a new steady state.

Definition 3.17 (Step response dynamics of DC2). Dynamics corresponding to MO output response to a setpoint step change with 2P control signal reaction involving two extreme points, or two control intervals (Fig. 3.17) with control value subsequently constrained to the upper and lower limit value (or conversely), will be classified as dynamics of DC2.

Within the DC2 the control signal reaction to a setpoint step bounded to monotonic output response can already involve two extreme points. After these two intervals (two extreme points) control signal is monotonically tending to a new steady state value u_∞ .

According to the Theorem 3.1 the MTC is typical with two (rectangular) control pulses approaching both limit control values (Fig. 3.17). After introducing rate constraints for both the switching from one limit value to the opposite one and for transient to the steady state value u_∞, the 2^{nd} control interval is typically rounded, or even disappears. Such response u_2 is typically converging to the next lower DC.

In the case when the rate constraints allow the control signal to attack both the upper and lower control limit, also the majority of aw approaches

fail (Rönnbäck, 1996) (improved solutions are e.g. given by Hippe (2006)). The windup phenomenon is not only connected with the integral (I) action, but also with the controlled process, when it is denoted as the plant windup Glattfelder and Schaufelberger (2003). In the literature simple and reliable solution for this dynamical class that could enable an arbitrary dynamics shaping ranging from the fully linear one up to the on-off MTC are still missing. For all that the needs on such solutions are very high: Let's mention just the automotive industry. Here, the historically known cascaded linear structures are not able to fulfill sufficiently the existing expectations. Although this task is practically solved (see e.g. Huba (2003, 2006)), the new solutions are not yet widely known.

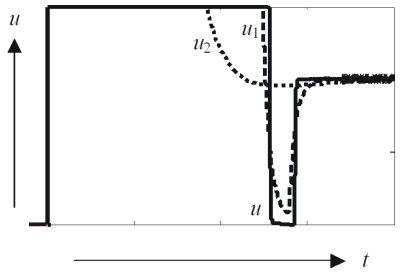


Fig. 3.17 DC2: Control signal reaction to a setpoint step change; u – time optimal (without rate constraints), u_1 – by limiting rate of changes in transient from one control limit to the opposite one and in transient to the steady state, u_2 – as u_1 but with stronger constraints.

3.6 Fundamental and "ad hoc" Solutions

Under fundamental controllers we understand solutions that for the nominal loop dynamics offer continuum of transient responses parameterized by the closed loop poles (or equivalent parameters as time constants of bandwidths) and enable to achieve any speed of control ranging from linear pole assignment control up to the relay MTC. Under "ad hoc" controllers we will understand solutions offering single (not adjustable) closed loop dynamics, or dynamics adjustable just in a limited range.

E.g. the relay MTC may be denoted as a typical "ad hoc" solution, since it offers unique closed loop dynamics that cannot be simply slowed down.

3.6.1 Setpoint Response

Since all solutions of DC0 will always remain linear, the corresponding fundamental solutions may be derived by the linear pole assignment control. How it is possible to characterize their substance?

Let us consider specification of the closed loop dynamics by two n-tuples of poles[1] α_1 and α_2 satisfying

$$-\infty < \alpha_2 < \alpha_1 < 0 \tag{3.25}$$

The setpoint response $\bar{y}(\alpha_i, t) = y(\alpha_i, t)/w(t)$ representing output reaction to the setpoint step $w(t) = w\mathbf{1}(t)$; $w = \text{const}$ is starting for $\alpha_i; i = 1, 2$ in a steady state with zero initial condition $\bar{y}(\alpha_i, 0) = 0$.

Definition 3.18 (Fundamental controller of DC0 – setpoint response). Controllers offering for a setpoint step and poles (3.25) output responses satisfying Ineqs.

$$1 > \bar{y}(\alpha_2, t) > \bar{y}(\alpha_1, t) > 0; \ \forall t > 0 \tag{3.26}$$

and asymptotic properties

$$\lim_{t\to\infty} \bar{y}(\alpha_i, t) = 1; \quad i = 1 \text{ or } 2 \tag{3.27}$$

will be denoted as *fundamental* one.

Fundamental solution simply means that by shifting closed loop poles to the left the corresponding outputs converge to the reference value faster, but monotonically, i.e. without overshooting, or undershooting, or without changing somehow their shape.

Similar effect in increasing speed of control we would like to achieve in constrained systems treated in DC1 and DC2. Here, the step response of the MTC representing the not really achievable limit dynamics will be denoted as $\bar{y}_{topt}(t)$. Let us suppose that the required state may be achieved by monotonic output transient, i.e. the Feldbaum's theorem holds. Expectations on the fundamental controller may then be expressed by following definition.

Definition 3.19 (Fundamental controllers of DC1 and DC2 – setpoint response). Controller yielding for the nominal dynamics $S(s)$ and for the closed loop poles (3.25) MO setpoint step responses of the output variable and fulfilling Ineqs.

$$1 > \bar{y}_{topt}(t) = \bar{y}(-\infty, t) \geq \bar{y}(\alpha_2, t) \geq \bar{y}(\alpha_1, t) > 0; \ \forall t > 0 \tag{3.28}$$

and asymptotic requirement

[1] Poles need not to be n-tuple, but in both vectors there should be kept fixed ratio of corresponding entries to the representative value α_i, $i = 1, 2$

$$\lim_{t\to\infty} \bar{y}(\alpha_i, t) = 1; \quad i = 1 \text{ or } 2 \tag{3.29}$$

will be denoted as *fundamental* one.

Requirement (3.28) means that the closed loop dynamics may be specified by the closed loop poles (or other appropriate parameters, as e.g. closed loop time constants, closed loop bandwidth, etc.), whereas it is ranging arbitrarily from fully linear dynamics of pole assignment control up to the relay MTC one (Fig. 3.17). Both these situations are considered as limit cases of the generalized approach.

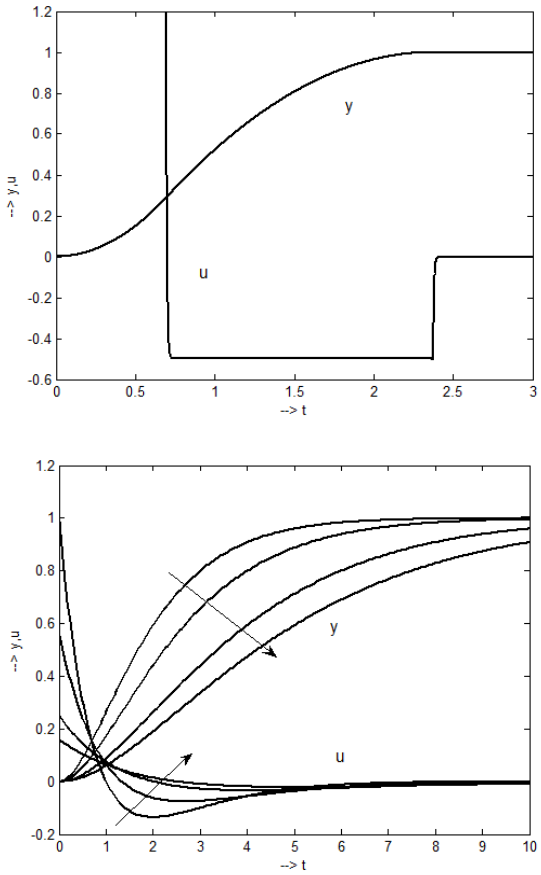


Fig. 3.18 Output and control signal transients of double integrator plant: left - for the relay MTC with control signal constrained by $U_{max} = 1.2$ and $U_{min} = -0.5$; right – for linear pole assignment control with double real pole values $-1, -0.75, -0.5$ and -0.4 arrows indicate pole shifting to zero (i.e. slowing down transients

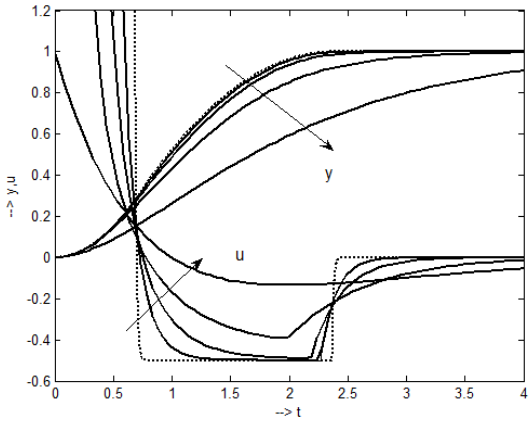


Fig. 3.19 Output and control signal transients of double integrator plant by constrained pole assignment control $U_{max} = 1.2$ and $U_{min} = -0.5$; for double real pole values $-8, -4, -2, -1$ – arrows indicate pole shifting to zero (i.e. slowing down transients); dotted – MTC transients

3.6.2 Disturbance Response

Notion of the dynamical classes can be applied both to the setpoint step responses as well as to the disturbance responses. The main difference is connected with the fact that whereas in the case of a step change of the reference signal the controller is instantaneously able to react, after a disturbance step it take some time up to moment when the disturbance observer sufficiently reconstructs the new disturbance value. Due to this reconstruction time, controller is able to stop control error increase appearing during this phase of disturbance response just with some delay. Just then controller starts to remove the already existing deviation. From this moment, output and control signal behavior can be analyzed in the same way as after a setpoint step. So, when speaking about disturbance response of DC0, instead of MO output considered at the setpoint response we are actually dealing with 1P output that can be characterized as MO just from the turnover point.

In the following we will deal just with controllers offering disturbance output response that after initial deviation monotonically tends to the required reference value.

In evaluating disturbance step responses one can test the same properties as for the setpoint response, but (besides of the already mentioned delay in disturbance compensation) it is to note that:

- By a prefilter in the reference signal it is possible to slow down dynamics of the setpoint step responses and by the disturbance observer filter to modify the disturbance response – in this way it is possible to tune both

responses at least partially separately – we are speaking about two-degree-of-freedom controllers.
- In general, areas of parameters guaranteeing NO, MO, or nP disturbance responses are different from those corresponding to setpoint step responses.

Let the disturbance response $\bar{y}(\alpha_i, t) = y(\alpha_i, t)/d(t)$ represents output reaction to the disturbance step $d(t) = d\mathbf{1}(t)$; $d = \text{const}$ starting for $\alpha_i; i = 1, 2$ in a steady state with zero initial condition $\bar{y}(\alpha_i, 0) = 0$. Again, we would like to deal with controllers that by pushing the closed loop poles to minus infinity enable to increase speed of removal of the control error caused by disturbance step and thereby to decrease its amplitudes up to zero.

Definition 3.20 (Fundamental controller– disturbance response). Controllers offering for a disturbance step and poles (3.25) output responses satisfying to Ineqs.

$$0 \leq |\bar{y}(\alpha_2, t)| \leq |\bar{y}(\alpha_1, t)|; \ \forall t > 0 \tag{3.30}$$

and asymptotic properties

$$\lim_{t\to\infty} \bar{y}(\alpha_i, t) = 0; \quad i = 1 \text{ or } 2 \tag{3.31}$$

$$\lim_{\alpha_i\to\infty} \bar{y}(\alpha_i, t) = 0; \quad \forall t > 0; \quad i = 1 \text{ or } 2 \tag{3.32}$$

will be denoted as *fundamental* ones.

Since the delay in the disturbance reconstruction and compensation may be significantly long and due to this also the control error occurring at the moment of output turnover, it is again possible to consider different DC associated with its removal.

3.6.3 Internal and Zero Dynamics

In systems with relative order of considered output r less than the total system order n there always exist states that are not directly controllable by input. They represent the so called internal dynamics. To achieve some simplicity, in nonlinear control this internal dynamics is usually characterized by simpler expressible zero dynamics.

Definition 3.21 (Relative order of the system output). As the relative degree of the plant output we denote integer r, telling, how many times it is required to differentiate output to get in the resulting formula control signal (plant input).

Definition 3.22 (Zero dynamics). Zero dynamics of a system with relative degree $r < n$ describes dynamics associated with maintaining output and its first r derivatives at zero. For linear system given by its transfer function

with highest power in numerator m and highest power in denominator n the relative degree is given as the pole-zero excess

$$r = n - m \tag{3.33}$$

From Definition 3.22 it follows that the zero dynamics is interesting just for systems with the relative degree less than the system's degree. In designing PID controllers, when dealing with dominant dynamics up to the 2nd order (i.e $n \leq 2$) the only situation with zero dynamics corresponds to systems with the relative degree $r = 1$ and $r = 2$. In a special case of systems with real poles it corresponds to situation when the total plant dynamics may be decomposed into two parallel first order plants

$$F(s) = \frac{K_1}{1 + T_1 s} + \frac{K_2}{1 + T_2 s} = K \frac{1 + T_0 s}{(1 + T_1 s)(1 + T_2 s)} \tag{3.34}$$

$$K = K_1 + K_2, \quad T_0 = \frac{K_1 T_2 + K_2 T_1}{K_1 + K_2} \tag{3.35}$$

In this case, zero dynamics is characterized by the time constant T_0 of the numerator of the transfer function. Number $-1/T_0$ is thereby denoted as the plant *zero*. Fundamental solutions that would enable an arbitrarily close tracking of the reference variable may be found just for systems with stable zero dynamics, when $T_0 > 0$. From the point of view of the MTC this situation corresponds to the so called singular problem (Athans and Falb, 1966), when during the 2nd period of control, the manipulated variable does not go to saturation limit, but takes values denoted as zeroing input (Isidori, 1995) that varies with the time constant T_0. So, Feldbaum's Theorem is valid just for systems with the relative degree equal to the full degree. For $r < n$ the total number of control intervals remains to be given by n, but just r from them may run with the limit control values.

For systems with unstable zero dynamics ($T_0 < 0$) fundamental solutions do not exists and it is possible to design just solutions preserving this unstable zero dynamics leading usually to some output undershooting.

3.7 Dead Time Systems

Another theoretically challenging and due to this being backward segment of the control theory is represented by the time-delayed systems. For this area, two early historical solutions are known: the Disturbance-Response Feedback by Reswick (1956) for dead-time compensation and the Smith Predictor (Smith, 1957). Practical experiences referred by many papers show that both structures have strong limitations: they are highly sensitive to parameters fluctuations and they do not enable an arbitrarily close approximation of optimal solutions. Controllers with dead time compensation are

frequently denoted as the predictive ones (see e.g. predictive PI-controller in Åström and Hägglund (1995)), but sometimes also as the PI - dead-time controllers (Shinskey, 1996, 2000). In the time of analogue pneumatic and electronic controllers the main reason for rare use of the corresponding structures was given by the problems of dead time modeling. So, for many decades' traditional PID controllers without dead time compensation substituted optimal solutions for the dead time compensation. These approaches did not guarantee strictly optimal results and so they have reasonably contributed to the inflation of different "optimal" controller tuning. They further survive due to the conservativeness of practice despite the fact that the new digital controllers enable an easy dead time modeling and compensation.

3.7.1 Delayed Fundamental Controllers

One important question is if it is possible to achieve for the time delayed systems the same dynamics as for the delay-free systems.

For the setpoint response answer to this question depends on the dead time position within the closed loop. If it is situated in the feedback loop, controllers with dead time compensation enable to achieve at the output the same dynamics as in the delay free systems. For known initial conditions the delay-free controller-plant connection enables an immediate action.

Feedback controller compensating dead time present somewhere in the loop, will be, of course, more complicated and the closed loop behavior will be much more sensitive to any model imperfection. But, theoretically, for in advance known input signals it is possible to achieve at the plant output any speed of control transients.

However, when dealing with response to unknown disturbances, or when dealing with step response and dead time is located within the feedforward path, at the output any result of control actions can appear just after the time delay. Therefore, in such situations, the best achievable behavior will be delayed by this dead time and it has to be respected also by the corresponding definitions of fundamental controllers that will be denoted as *delayed fundamental controllers*. The difference will especially be visible in the disturbance response, where it is no more possible to achieve requirement (3.32).

3.7.2 Fundamental Controllers – a New Concept?

Many of the known approaches to the controller design do not fulfill the requirements on the fundamental solutions, since they:

- are linear and so they do not enable to arbitrarily speed up transients to approach in the limit under consideration of constraints the MTC transient responses, or
- do not involve free design parameters at all.

E.g. Klán and Gorez (2000) (as many others) tried to find optimal PI controller tuning for stable 1st order systems with relatively long dead time T_d. The problem, however, is that structure of the PI controller was generically derived for the 1st order plant dynamics without the dead time. Short dead time values can be allowed by limiting choice of the applicable controller parameters (closed loop poles). This, however, violates requirements put on the fundamental solutions. For the 1st order plant with long dead time the fundamental controller will already have more complicated structure than simple PI controller – involving some features of the Smith predictor. So, the above mentioned solution cannot be treated within the group of fundamental solutions for the first order plants with long dead-time, just as a special (ad hoc) suboptimal solution.

The above example represents typical feature of majority of existing solutions. The PI controller represents an easy to use, but not a fundamental solution. In the time of analogue pneumatic and electronic controllers the main reason for rare use of the optimal dead time structures was given by the problems of the dead time implementation. So, for many decades' traditional PID controllers without dead time compensation substituted them. These approaches do not guarantee strictly fundamental properties and so they have reasonably contributed to the inflation of different "optimal" controller tuning. They further survive due to the conservativeness of practice despite the fact that the new digital controllers enable an easy dead time modeling and compensation. Of course, it has no sense to fight against their use, but it should be shown what they are able to offer. In such a way, all the ambiguity of solutions reported e.g. by O'Dwyer (2000, 2006) can be reasonably reduced.

3.8 Table of Fundamental PID Controllers

As it was already mentioned above, Glattfelder and Schaufelberger (2003) tried to design the pole assignment PI controller in such a way that its control signal step reaction would converge to one pulse of the MTC. This point shows other important discrepancy of the PI control theory. When comparing their design criterion with that one introduced by Klán and Gorez who required the optimal PI control signal step reaction to have a stepwise character (see e.g. paper by Klán and Gorez (2000); or the discussion by Strmčnik and Vrančič (2000) we see a clear contradiction. Who is right in this conflict? The response may surprise many people — both requirements are right!

Simply, there exist two dynamical classes of PI control. Whereas the traditional linear PI control having the control signal response required by Klan and Gorez corresponds to DC0, controllers using anti-windup circuitry and trying to approach the MTC response characteristic by one saturated pulse of control represent already solution of DC1. It has no sense to ask, which one is better — each has its unique properties that cover specific group of applications!

Generalizing this way of arguing, it is then possible to define three dynamic classes of the PID control (Tab. 3.1).

Introduction of dynamical classes of control together with introduction of fundamental solution enable transparent practically motivated classification of the existing controller structures and tuning rules.

Up to now, works on PID control usually did not pay attention to the dynamical classes. So it can e.g. happen that whereas Vítečková et al (2000) proposed for the plant

$$S_2(s) = \frac{K_s}{s(T_1 s + 1)} \tag{3.36}$$

PD controller tuning that corresponds to the DC1, for the plant

$$S_2(s) = \frac{K_s}{(T_1 s + 1)(T_2 s + 1)} \tag{3.37}$$

they already gave PID controller tuning that corresponds to the DC0. Because solutions corresponding to a particular DC need not be unique, it is to expect that a rigorous classification of the existing solutions according to the dynamical classes represents a complex and long term problem. Classification of the existing solutions is complicated also by the fact that the practically attractive properties may lie on the border of two dynamical classes.

Rows of the newly introduced table of fundamental controllers correspond to different DCs (Tab. 3.1). These are naturally given by the relative degree of the output defined by the dominant dynamics located in the forward path of the control loop. The type of the controller is then given by the dominant closed loop dynamics including also the feedback dynamics.

The new table of controllers may seem to be much richer than the traditional basis of PID control consisting of the I, P, PI, P-P, P-PI, PD and PID controllers. Besides of the basic dynamics, traditional controllers were classified according to the implementation form (series, parallel, interactive, non-interactive, with setpoint weighting, with different anti-windup structures, etc.).

Similar features are to find also in the new approach. Generic schemes of the constrained PID control are derived by the state space approach and extended by DOB based I action. These are shown to have equivalent schemes that are more or less similar to the traditional structures used in different modifications in practice. Particular controllers are represented by structures – i.e. they are more complex than simple transfer functions. Such a develop-

ment is not surprising – e.g. the generalization to controllers with two-degree-of-freedom controllers started already in 1960s.

All traditional linear structures are involved in the DC0, while the higher DCs are already essentially nonlinear.

Table 3.1 Table of the fundamental PID controllers

		Dominant dynamics							
Dynamic class	I-action	K	$Ke^{-T_d s}$	$\frac{K_s}{s+a}$	$\frac{K_s e^{-T_d s}}{s+a}$	$\frac{K_{s1}}{s+a_1}+\frac{K_{s2}}{s+a_2}$	$\left[\frac{K_{s1}}{s+a_1}+\frac{K_{s2}}{s+a_2}\right]e^{-T_d s}$	$\frac{K_s}{s^2+a_1 s+a_0}$	$\frac{K_s e^{-T_d s}}{s^2+a_1 s+a_0}$
0	N	FF	FF	FF	FF	FF	FF	FF	FF
	Y	I	PrI	PI	PrPI	PID	PrPID	PID	PrPID
1	N	-	-	P	PrP	P-P	PrP-P	PD	PrPD
	Y	-	-	PI	PrPI	P-PI	PrP-PI	PID	PrPID
2	N	-	-	-	-	-	-	PD	PrPD
	Y	-	-	-	-	-	-	PID	PrPID

FF – static feedforward control is involved also in all feedback controllers
Pr – abbreviation for predictive (dead time) controllers with compensation of the dominant dead time

Note that the PI controllers and their predictive versions for dead time compensation are included in two dynamical classes. Each solution is, however, different! The traditional linear PI controller represents optimal solution from DC0. This has either to be combined with a prefilter (usually used in older analogue electronic solutions), or implemented as the I-P controller with error acting on I only $b = 0$. It guarantees dynamics minimizing the actuator wear.

Up to now, the solutions really optimal for the DC1 were approximated by linear controllers equipped by some anti-windup circuitry. However, while the structures used for ages in the series implementation of PI controllers for constrained systems (Åström and Hägglund, 1995; Glattfelder and Schaufelberger, 2003; Kothare et al, 1994) represent substantial part of the newly derived ones – the question is if the missed parts of the new optimal solutions are not simply result of a vague controller description, or of the intentional know-how protection?

The new solutions fully explain needs on setpoint weighting, or prefilters used equivalently in older analogue electronic controllers, needs on structure variations (switching between a linear and a nonlinear PD-controller, etc.

Let us remind that the table brings two structures of the PI and PD controllers and even 3 different structures of the PID ones. Because it is no problem to show that all of them are important for practice, then it is also clear, why each attempt to replace one fundamental solution by an "optimal" retuning of some other structure lead finally to the already mentioned inflation of optimal tunings: for each set of initial states, input signals and process

parameters the optimization gave some results and libraries are crowded by all such results.

While the fundamental PD controller of DC1 is essentially near to the linear PD controller compensating usually the largest loop time constant (it has linear control algorithm extended by saturation), the new PD controllers of DC2 are already fully nonlinear. For each special loop dynamics (the double integrator, single integrator+time constant, two different or equal time constraints, oscillatory dynamics, etc.) it is possible to derive a special controller. Despite the possibility to derive for each plant new controller, a unique importance has the solution derived for the double integrator that can be used universally. Again, it is not something completely new. Feldbaum (1965) cites patent of Russian engineers from 1935 based on improving dynamics of the rolling mills positional control by quadratic velocity feedback that is typical for the time optimal control of the double integrator. Later, similar idea was used in the industrial controller Speedomax produced by Northrup.

Later we will deal with the fundamental solutions that enable by choosing the closed loop poles to modify the closed loop dynamics from the fully linear one up to the on-off dynamics of the MTC. The latter corresponds to pushing poles up to $-\infty$. All controllers are extended by the I-action based on reconstruction and compensation of acting disturbances. Since it would be too demanding to explore in details all possible solutions, this book concentrates on solutions based on 1- and 2-parameter models.

After choosing for the identified dominant loop dynamics a fundamental controller, one has to determine its tuning. When is a controller tuning reliable? Everything depends on the loop properties. If the time constants and gain of the identified dominant dynamic seem to be constant, besides of the possibly fixed nominal dynamics reliable system approximation has to take into account also the nonmodelled (perturbation dynamics). In general it should considers possible plant-model mismatch that determines borders of the closed loop poles choice used for the controller tuning and also other parasitic aspects as e.g. the measurement noise.

3.9 Generic and Intentionally Decreased DC

Introduction of dynamical classes of control into the controller design brings structure for classification of available approaches, methods and solutions. Why such a system may be useful, it may be obvious from the following theorem.

Theorem 3.2. *For a given plant, the generic dynamical class of control is given by the output relative degree. However, for plants with stable subsystems it can be intentionally decreased up to the number of remaining unstable or marginally stable poles.*

Each dynamical class of control is related to some control properties. Since from the above theorem it follows that e.g. for stable 2nd order systems it is possible to design controllers from DC2, DC1 and DC0, without having in mind specific properties of each solution comparing of resulting solutions may be very questionable.

To illustrate related problems, let us start with inspecting possible loop configurations with dynamics of the 1st and 2nd order. It is to remember that whereas identifying some process from the measured input-output behavior (by evaluating step response, relay experiment, measurement at the stability border, etc.) without additional information it is not possible to decide about the actual dynamics distribution within the loop. But, for each particular distribution another controller should be chosen.

For the 1st order loop dynamics (Fig. 3.20) we have to decide upon 2 solutions, for the 2nd order one (Fig. 3.21) upon 3 (or even 4, since we have to distinguish among the configurations b1 and b2). When remembering, how many authors tried to propose universal autotuner based on evaluating the input-output behavior, here you can see one of the reasons, why no of them can be generally accepted. There are too many degrees of freedom: the optimal order of the approximation, distribution of the dynamical term within the loop and choice of the dynamical class of control that can be based on the relative degree of the actual output, or intentionally decreased.

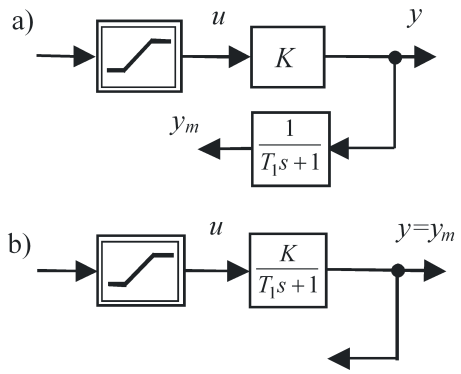


Fig. 3.20 Control loops with the 1st order dominant dynamics and with relative degrees a) 0 and b) 1.

Besides of the natural allocation of the dynamics within the control loop the design has to consider also other aspects as the availability of different signals, the measurement and quantization noise level of particular measurements, nonmodelled dynamics, possible fluctuations of system parameters, etc. It is e.g. to show that for the same measured output the sensitivity to measurement noise is increasing by increasing the dynamical class of used controllers. So, in a noisy environment of industrial control with oftenly put

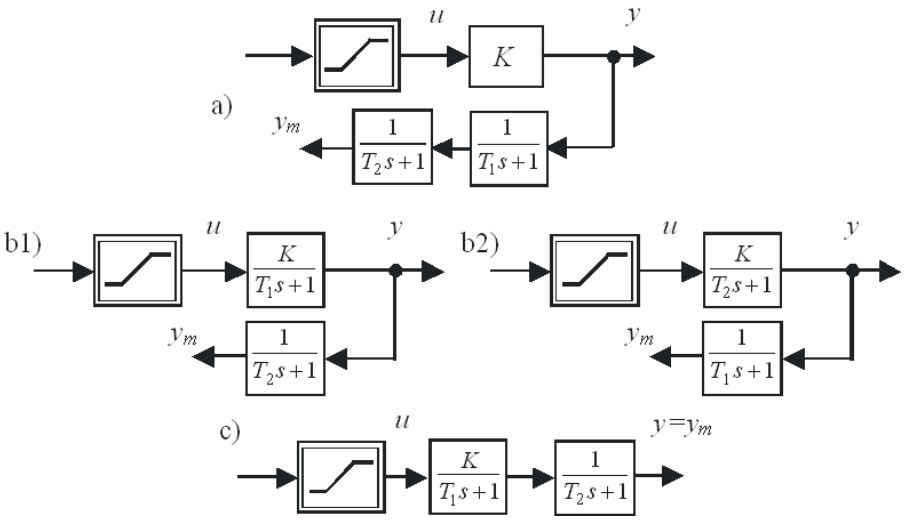


Fig. 3.21 Control loops with the 2nd order dominant dynamics and the relative degrees: a) 0; b) 1 and c) 2.

additional requirements on minimizing wear of actuators one has intentionally to choose solution corresponding to the lowest DC. Similar conclusions can also follow in controlling processes with variable dead time, gain and other parameters. But still there exist many situations, where a decrease of the dynamical class is not permitted – e.g. in controlling unstable systems.

Example 3.2 (Two-tank hydraulic system). For a hydraulic system with two containers and a pump having a dynamics negligible with respect to the dynamics of containers characterize generic dynamical classes of control associated with the output (Fig. 3.20) defined by:

a) $y = q_0$,
b) $y = q_1$ (or $y = y_1$),
c) $y = q_2$ (or $y = y_2$).

Solution: When it is required to control the input flow q_0 of the "memoryless" (noninertial) pump, it is possible to speak about process of DC0. This follows from the assumption of the noninertial pump, when it is possible to react to step setpoint changes by step flow changes. Of course, due to the inertia of the pump and of the liquid contained in pipeline the real processes will be smoother (e.g. exponential as in Fig. 3.15), but these transients can fully be neglected with respect to the time of filling the containers.

This holds without respect to the question, which information about the pump flow is used for establishing feedback. Depending on available sensors the pump can be controlled by measuring the controlled flow by a sensor located immediately at its output, or it is controlled by using flow observer

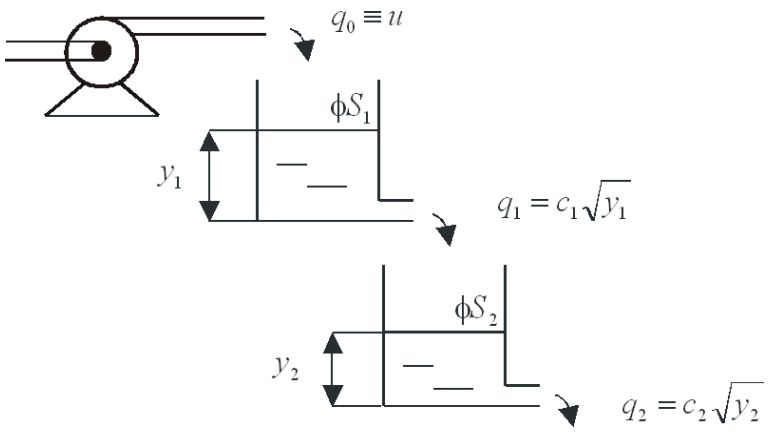


Fig. 3.22 Two tank system with a "noninertial" pump

based on measuring levels y_1 and/or y_2 (or the flows q_1 and q_2). In the 2nd case it may be necessary to respect also the length of the inlet pipeline that brings dead-time into the control loop. Of course, for each situation the final controller (including also the relevant reconstruction) will be different.

When it is required to control the flow q_1, or the level y_1 of the first tank, each larger setpoint step change may already require to let the pump to work for some time in one of the limit regimes: either fully switched on (Fig. 3.16), when the level has to be increase, or fully switched off, when it has to be decreased. A "linear" control has sense just in the vicinity of the reference flow (level). The length of transient from a limit regime to a new steady state can sometimes be fully neglected with respect to the time required for filling the container. By its nature, this process can be classified as from the DC1. As above, the final control algorithms will depend on, which process variables are measured and they will differ from those proposed for the above problem.

When it is required to control the second tank output flow q_2, or its level y_2, after each larger setpoint change upward it will be firstly required to run the pump fully switched on for some time. In such a way the level in the first tank will increase most rapidly what results in the fastest possible filling of the 2nd tank. However, yet before reaching the required level in the 2nd tank, the pump has to be switched off to decrease the first tank level y_1 up to the value corresponding to the required steady state (Fig. 3.17). "Linear"' control process may appear just in the vicinity of the required output flow q_2 (level y_2) and its duration can usually be neglected with respect to the duration of the nonlinear transient. By its nature, the process can be assigned into the DC2. The control algorithms will depend on, which process variables are measured and differ from those proposed for both above problems.

Since the two-tank system can be shown to be stable, according to Theorem 3.2 the output flow q_2 (or level y_2) may be set to the reference level also by control algorithm of DC0 that would simply set the input flow $q_0 = q_2$.

It can be easily shown (e.g. by simulation) that the corresponding transient would be much longer than the transient from DC2. For the same task, the solution of DC1 might be based on bringing the flow q_1 to the value corresponding to the desired output, i.e. by $q_1 = q_2$. Again, the overall transient would take more time than by using controller from DC2, but it would be faster than controller from DC0.

3.10 Summary

1. The existing classification of PID controllers into ISA, series and parallel ones reflects spontaneous development of the technology that shows still to be not completely finished. In early period, many details of developed solutions were considered as proprietary information and the internal structures were frequently kept secret instead of being published in literature. Much useful information was also scattered in the literature and finally forgotten, so that it is not sure that today we know fully to argue all existing forms of controllers and to transfer their essential features to the newer technology solutions. Some effort for a more detailed description was already spent in early period of PID control development motivated by requirements of a reliable analytical tuning.
2. With respect to real needs of practice it is obvious that the traditional modules of PID control do not cover all its requirements: there is lack on solutions for higher dynamical classes (DC) of constrained control (unstable systems), including also time-delayed systems.
3. Problems are also caused by the fact that control community has still not accepted notion of dynamical classes of the PID control and requirements that must fulfill fundamental controllers to be included into the PID basis. Despite to the fact that many authors are already respecting these requirements intuitively, without moving forward in this point, it is not possible to develop consistent theory that would cover all requirements ranging from quasi minimum-time control up to the fully linear "smooth"transients.
4. Solutions of the DC0 are fully compatible with the traditional linear solutions that in transition from a steady state to another one yield monotonic output and control signal responses.
5. Within the DC1 containing already one phase of energy accumulation, the fundamental solutions are up to now typically being replaced by traditional linear controllers extended by anti-windup (aw) circuitry.
6. Within the DC2 containing energy accumulation and dissipation phases the fundamental solutions are usually being replaced by less effective and just locally applicable cascaded structures, by the sliding mode control, or by new development in model predictive control.

7. By index of the dynamical class we denote a non-negative integer denoting number of possible intervals with the limit control signal values (or extreme points) that can occur under MTC with monotonic output.
8. Dynamical classes of control (control processes, controllers) are physically closely related to the energy accumulation/dissipation taking part in controlled plants, when they denote relevant number of energy accumulation phase and, mathematically, to the relative degree of the specified outputs.
9. Definition of dynamical classes of control enables to classify existing design methods and approaches, to increase reliability of the control design, to explain several existing gaps between theory and practice and so finally to improve the acceptance of the control theory by practice.
10. Historically, the notion of dynamical classes of control is closely related to the Feldbaum's theorem about n-intervals of the relay MTC. However, ideal rectangular pulses of such control are considered just as limit case of the constrained pole assignment control (CPAC) with poles shifted to minus infinity. Generating of such rectangular pulses would require an infinitely broad frequency band of the control loop. Putting additional constraints on the rate of control signal changes (or even on its higher derivatives), or on the loop frequency band, respectively, CPAC gives smoother control and some its intervals may shrink to smoother signals with extreme points, or even they fully disappear.
11. Generic DC index related to the specified output relative degree cannot be determined by measuring input-output characteristics, if the measured output is different from the specified one. So, for a rigorous decision in defining optimal controller we usually need also additional information about the distribution of dynamical elements within the control loop (system structure): ideal universal autotuner based fully on input-output measurement is not possible.
12. In the case of relatively high measurement (quantization) noise, or perturbation dynamics it may be useful to intentionally decrease dynamical class of control against the generic one and so to decrease the systems sensitivity, actuators wear, etc.
13. For covering all typical situations in controlling systems with transfer functions up to the 2^{nd} order with dead time we have introduced table of fundamental controllers containing besides of static feedforward control (contained also in all feedback structures) 18 different feedback controllers.
14. The proposed table includes all traditional (linear) PID controllers within the DC0. Besides of this it systematically covers many solutions used in practice that are up to now staying outside of mostly "linear" theory of the PID control. Despite it is much richer than the basis of traditional PID control, it still does not cover all possible situations with the 2^{nd} order plants with dead time – some rarely appearing situations were up to now omitted due to the sake of simplicity.

15. The existing inflation of different PID tuning rules and anti-windup structures results from attempts to replace one fundamental solution by another one by retuning its parameters. By this process based mostly on different optimization procedures one can get a local optimum corresponding to a particular choice of the initial conditions, input signals and system parameters, but never a globally valid solution.
16. There are many arguments supporting acceptance of the proposed table and solutions contained. As at any change of the settled-down theory, against acts the inertia of human thinking. This cannot be eliminated by any arguments or facts: this is also result of the system dynamics.

3.11 Questions and Exercises

1. How could you define PID control?
2. Could you formulate an alternative definition?
3. What does characterize index of a dynamical class?
4. How it is related to the Felbaum's theorem about n intervals of time optimal control?
5. Which criteria must fulfill a controller to be considered as the fundamental one?
6. What was the technological reason for lag of theory of the time delayed systems and by which technology generation it was eliminated?
7. Identify some reasons for lags in development of theory of constrained PID control.
8. Name some factors contributing to the inflations of optimal tuning rules for PID control.
9. Does the well-known method by Ziegler and Nichols optimize the controller tuning for the set point step of for the disturbance step responses?
10. Characterize processes of the dynamical classes 0, 1 and 2!
11. Sketch the table of fundamental PID controllers!

Acknowledgements The author is pleased to acknowledge the financial support by a grant No. NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism.

References

Anderson, B. and Moore, J. B. (1969). Linear system optimization with prescribed degree of stability, Proc. IEEE, 116 (12), 2083-2087

Åström, K. J. and T. Hägglund (1995) PID controllers: Theory, design, and tuning – 2nd ed., Instrument Society of America, Research Triangle Park, NC

Åström, K. J. and Hägglund, T. (2004) Revisiting the Ziegler–Nichols step response method for PID control. *J. Process Control* 14, 635–650

Åström, K. J. and Hägglund T. (2005) *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC

Åström, K. J and Wittenmark, B. (1984) *Computer Controlled Systems. Theory and Design.* Prentice Hall, Englewood Cliffs, N. J.

Athans, M. and P. L. Falb (1966) *Optimal Control*. McGraw-Hill, N. York

Baños, A., Vidal, A. (2007) Definition and Tuning of a PI+CI Reset Controller. *European Control Conference*, July 2-5, 2007, Kos, Greece.

Bennet, S. (1997) A Brief History of Automatic Control. IEEE Control Systems Magazine, 17-25.

Bushaw., D (1958) Optimal discontinuous forcing terms. In Contributions to the theory of nonlinear oscillations, 29-52, Princeton Univ. Press, Princeton, H. J.

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

Fertik, H. A., Ros, C. W. (1967) Direct digital control algorithms with anti-windup feature. ISA Trans., 317-328.

Glattfelder, A. H. und Schaufelberger, W. (2003) Control Systems with Input and Output Constraints. Springer, London

Hägglund, T. (1996) An industrial dead-time compensating PI-controller. *Control Engineering Practice* 4, 749-756

Hippe, P. (2006) *Windup in Control: Its Effects and Their Prevention*. Springer London

Horowitz, I. M. (1963) *Synthesis of Feedback Systems*. Academic Press, N. York

Huba, M., Sovišová, D. and I. Oravec (1999) Invariant Sets Based Concept of the Pole Assignment Control, *Conf. Proc. ECC'99* Karlsruhe

Huba, M. (2003) Syntéza systémov s obmedzeniami I. Základné regulátory. II. Základné štruktúry. (Constrained systems design. I. Basic controllers. II. Basic structures.) Vydavatel'stvo STU Bratislava

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. and Šimunek, M. (2007) Modular Approach to Teaching PID Control. *IEEE Trans. Ind. Electr.*, 54, 6, 3112-3121.

Huba, M., Skogestad, S., Fikar, M., Hovd, M., Johansen, T. A., Rohal'-Ilkiv, B., Editors (2011) Preprints of the Workshop "Selected Topics on Constrained and Nonlinear Control", STU Bratislava – NTNU Trondheim, January 2011.

Huba, M., Skogestad, S., Fikar, M., Hovd, M., Johansen, T. A., Rohal'-Ilkiv, B., Editors (2011) "Selected Topics on Constrained and Nonlinear Control. Workbook", STU Bratislava – NTNU Trondheim, January 2011.

Isidori, A. (1995) *Nonlinear Control Systems*. 3rd edition, Springer Verlag, New York

Johnson, M. A., Moradi, M. H. (Editors) (2005) *PID Control: New Identification and Design Methods*. Springer London

Keel, L. H., Kim, Y. C. and Bhattacharyya, S. P. (2008) *Advances in Three Term Control*. Pre-Congress Tutorials & Workshops. 17th IFAC World Congress Seoul, Korea

Khalil, H. K. (1996) *Nonlinear Systems*, 2nd Ed. Prentice Hall Int. London.

Klán, P. and R. Gorez (2000) Balanced Tuning of PI Controllers. *European Journal of Control*, 6, 541-550

Kothare, M. V., Campo, P. J., Morari, M. and C. V.Nett (1994) A Unified Framework for the Study of Anti-windup Designs. Automatica, Vol 30, 1869-1883.

Kramer, L. C., Jenkins, K. W. (1971) A new technique for preventing direct digital control windup. Proc. Joint Automatic Control Conf., St.&Louis, Missouri, 571-577.

Maxwell, J. C. (1868) On Governors. *Proceedings of the Royal Society of London*, 270-283. Published also in *Mathematical Trends in Control Theory*, R. Bellman and R. Kalaba (Edts.), Dover Publications, N. York, 1964, 601-616.

Morari, M. and E. Zafiriou (1989) *Robust Process Control*. Prentice Hall, Englewood Cliffs, N. Jersey

Newton, Gould, and Kaiser (1957) *Analytical Design of Linear Feedback Controls*. Wiley, New York

O'Dwyer, A. (2000) A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules. Preprints IFAC Workshop on Digital Control: Past, present and future of PID Control PID'2000, Terassa, Spain 175-180.

O'Dwyer, A. (2006) Handbook of PI and PID controller tuning rules. 2nd Ed., Springer London 2006.

Ogata, K. (1997) Modern Control Engineering. 3rd Ed., Prentice Hall, London

Ohishi, K., Nakao, M., Ohnishi, K. and Miyachi, K. (1987) Microprocessor-controlied DC motor for load-insensitive position servo system. IEEE Trans. Ind. Electron., 34, 44-49.

Ohnishi, K. (1987) A new servo method in mechantronics, Trans. Jpn. Soc. Elect. Eng., 107-D, 83-86.

Ohnishi, K., Shibata, M. and Murakami, T. (1996) Motion Control for Advanced Mechatronics. IEEE/ASME Trans. on Mechatronics, Vol. 1, 1, 56-67

Oldenbourg, R. C. and H. Sartorius (1951) Dynamik selbsttätiger Regelungen. 2.Auflage, R. Oldenbourg-Verlag, München

Pontryagin, L. S., Boltyanskij, V. G., Gamkrelidze, R. V., Mischenko, E. F. (1964) *The Mathematical Theory of Optimal Processes*. Pergamon Press Oxford

Reswick, J. B. (1956) Disturbance-Response Feedback - a new control concept. Trans. ASME, No.1, 153-162

Rivera, D. E., M. Morari and S. Skogestad (1986) Internal Model Control. 4. PID Controller Design. *Ind. Eng. Chem. Process Des. Dev.*, 25, 252-265

Rissanen, J. (1960) Control System Synthesis by Analogue Computer Based on 'Generalized Linear Feedback Concept'. *Proceedings of the Symposium on Analog Computation Applied to the Study of Chemical Processes*, Brussels, November 21-23, 1-13

Rönnbäck, S. (1996) Nonlinear Dynamic Windup Detection in Anti-Windup Compensators. Preprints CESA'96, Lille, 1014-1019

Seok, J. K., Kim, K. T., and Lee, D. C. (2007) Automatic Mode Switching of P/PI Speed Control for Industry Servo Drives Using Online Spectrum Analysis of Torque Command. *IEEE Trans. Ind. Electron* 54, 5, 2642-2647

Shigemasa, T.; Yukitomo, M.; Kuwata, R. (2002) A model-driven PID control system and its case studies. *Proc. IEEE Int. Conf. on Control Applications* Glasgow, Scotland, UK, Vol.1, 571 – 576

Shigemasa, T.; Yukitomo, M. (2004) Model-Driven PID Control System, its properties and multivariable application. *APC 2004 Advanced Process Control Applications for Industry Workshop*, April 26, 27 and 28, 2004, Vancouver, Canada

Shinskey, G. (1990) How good are Our Controllers in Absolute Performance and Robustness. *Measurement and Control*, Vol. 23, 1990, 114-121

Shinskey, 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 2000, 14-18

Shinskey, G. (1996) *Process Control Systems*, 4th ed., McGraw-Hill, N. York

Skogestad, S. (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13, 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 (1996) *Multivariable Feedback Control Analysis and Design*, John Wiley, N. York

Smith, O. J.M. (1957) Closser control of loops with dead time. Chemical Engineering Progress, Vol.53, No.5, 217-219.

Strmčnik, S. and Vrančič, D. (2000) Discussion on "Balanced Tuning of PI-Controllers" by P. Klán and R. Gorez. *European J. of Control*, 551-552

Umeno, T. and Hori. Y. (1991) Robust speed control of DC servomotors using modernm two degrees-of-freedom controller design. TIE 38, 363-368.

Van de Vegte, J. (1994) *Feedback Control Systems.* 3rd Ed. Prentice Hall

Vítečková, M., A. Víteček, L. Smutný (2000) Simple PI and PID controllers tuning for monotone self-regulating plants. *IFAC Workhop on Digital Control, Past, present and future of PID Control,* Terrassa, Spain, 283-288

Youla, D. C., Jabr, H. A. and J. J. Bongiorno (1976) Modern Wiener-Hopf design of optimal controllers. Part II: the multivariable case. *IEEE Trans. on Automatic Control*, 21, 319-338

Yukitomo, M.; Baba, Y.; Shigemasa, T. (2002) "A model driven PID control system and its application to chemical processes", SICE, *Proc. 41st SICE Annual Conf.*, Vol. 4, 2656–2660

Yukitomo, M.; Shigemasa, T.; Baba, Y.; Kojima, F. (2004) A two degrees of freedom PID control system, its features and applications. 5th Asian Control Conf., 2004, Vol.1, 456–459

Zhao, S. (2004) Advanced Control of Autonomous. Underwater Vehicles, PhD Thesis in Mechanical. Engineering, University of Hawaii.

Zhong, Q. C. and Mirkin, L. (2002). Control of integral processes with dead-time. Part 2: Quantitative analysis, IEE Proc. Control Theory Appl., 149, (4), 291–296.

Zhong, Q. C. and Normey-Rico, J. E. (2002). Control of integral processes with dead-time. Part 1: Disturbance observer-based 2DOF control scheme. IEE Proc.-Control Theory Appl., Vol. 149, 4, 285–290.

Ziegler, J. G. a Nichols, N. B. (1942) Optimum settings for automatic controllers. Trans. ASME, November, 759-768

Chapter 4
Basic Fundamental Controllers of DC0

Mikuláš Huba

Abstract The Dynamical Class 0 (DC0) contains all known linear PID control structures enabling to achieve monotonic plant output course after a setpoint step by monotonic control signal at the controller output. Ideally, control signal reaction to such a setpoint step may be arbitrarily fast and in a limit case to approache step function. Plants that enable to achieve such transients may be considered as generalization of a memoryless plant with additional stable dynamics. Considered structures will be derived from the static feedforward open-loop control extended by reconsruction and compensation of acting disturbances by measuring the plant and controller output signals. For this purpose, fundamental solutions will be proposed based on parallel plant model (PID-PM, or IMC like PID structures) and inverse models of the (invertible) dominant loop dynamics (PID-IM, or DO based PID structures). In a correctly tuned loop and under effect of admissible input signals the control signal constraint will never be active: neither in steady states nor during monotonic transient responses among them. So, in DC0 the control loop may be fully treated by means of linear control theory. A typical feature and important advantage of transients in DC0 is the lowest wear and energy consumption of the actuators.

4.1 I, $\mathbf{I_0}$ and $\mathbf{FI_0}$ Controllers

Relation of the input and output variable of the memoryless plant (4.1) with constrained control signal u_r, with input disturbance v_i and output disturbance v_o is described as

Mikuláš Huba
Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, e-mail: mikulas.huba@stuba.sk

$$y = K(u_r + v_i) + v_o \tag{4.1}$$

In order to consider also asymptotic behavior, we will deal just with piecewise constant loop inputs.

4.1.1 Output Disturbance Reconstruction

By measuring output y that corresponds to control signal u (Fig. 4.1a) for a plant gain estimate K_0 it is possible to reconstruct the output disturbance value $\hat{v}_o$ by means of $\hat{v}_o = y - K_0 u_r$. This value can then be used for the distrubance compensation by a counteractive signal added to the reference value. In order to avoid algebraic loop, to achieve required noise filtration and robustness, and or to limit rate of control signal changes after a disturbance step, it is required to work with filtered reconstructed signal

$$\hat{v}_{of} = \frac{1}{1 + T_f s}\,[y - K_0 u_r] \tag{4.2}$$

To limit rate of the control & output changes after a reference step, a prefilter with time constant T_p can be used.

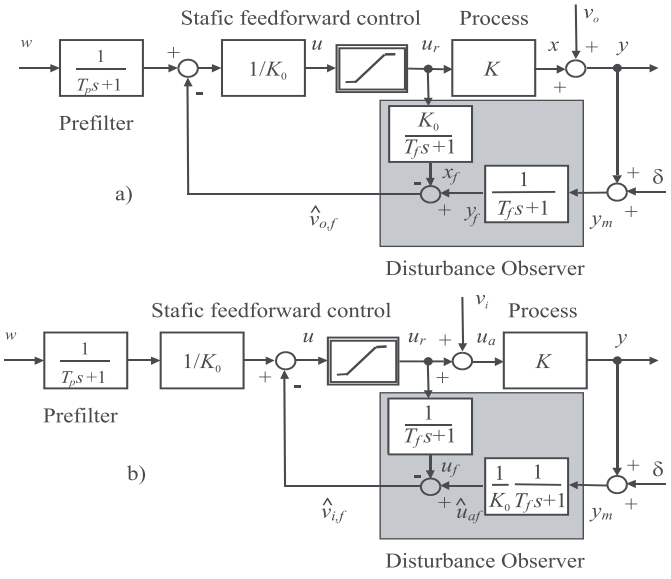


Fig. 4.1 FI_0 controllers: static feedforward control extended 1) by reconstruction and compensation of (a) output and (b) input disturbances of a memoryless plant and 2) by a prefilter; in nominal case $K_0 = K$

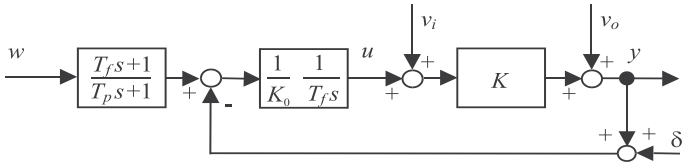
4.1.2 Input Disturbance Reconstruction

An input disturbance v_i may be reconstructed as difference between the reconstructed plant input u_a and the controller output u as $v_i = y/K_0 - u_r$. Again, it will be required to work with filtered reconstructed disturbance

$$\hat{v}_{if} = \frac{1}{1 + T_f s}\,[y/K_0 - u_r] \tag{4.3}$$

To limit rate of the control & output changes after a reference step, also here (Fig. 4.1b) a prefilter with the time constant T_p may be used. For admissible input signals the saturation can be omitted (it will never be active, i.e. $u_r = u$) and the structures in Fig. 4.1 be replaced by more frequently used structure (Fig. 4.2) with integrating controller

$$R\,(s) = \frac{K_I}{s}; \quad K_I = \frac{1}{K_0 T_f} \tag{4.4}$$



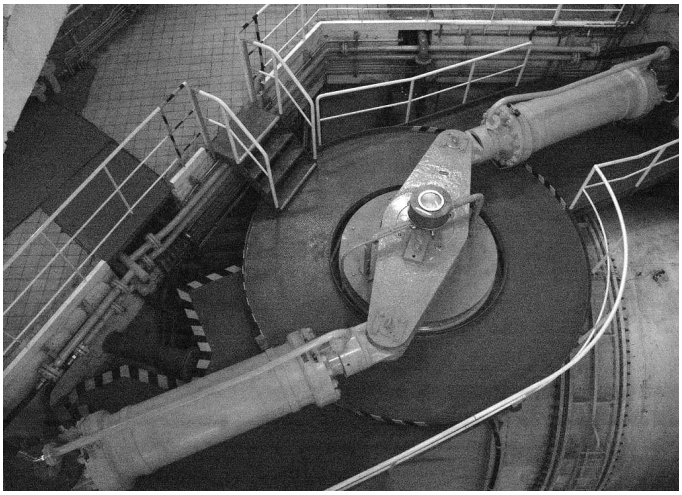


Fig. 4.2 Closed loop with I_0 controller equivalent to the structures from Fig. 4.1 (above) and example of hydraulic actuators with I character (below).

Instead of the filter time constant T_f it may be simpler to work with the reconstruction filter bandwidth

$$\Omega_f = \frac{1}{T_f} = K_0 K_I \tag{4.5}$$

Equivalent loop prefilter is defined as

$$T_e(s) = \frac{1 + T_f s}{1 + T_p s} \tag{4.6}$$

By its omitting, i.e. by setting $T_p = T_f$ in the generic scheme in Fig. 4.1 with

$$T_p(s) = 1/(1 + T_f s) \tag{4.7}$$

one gets a continuous control response after a setpoint step. For keeping the step character of feedforward control after a reference signal step, the equivalent loop should include ideal prefilter with as small as possible value T_p (ideally $T_p \to 0$). This equivalent structure of FI_0 controllers may be usefull not just due to its simplicity, but also in situations when the controller and actuator are physically not separable. Such a situation is typical e.g. in using hydraulic and electrical drives in roles of actuator. These genericly have integral character - for a constant nonzero input their output is linearly increasing.

Since an output disturbance can be fully replaced by equivalent input disturbance and this is more fequent in practice, in the sequel we will mostly limit our treatment to the case of input disturbances whereby the index "i" may be omitted.

Definition 4.1 (I_0 and FI_0 controllers). Under I_0 controller we will understand static feedforward control extended by DO based reconstruction and compensation of input, or output disturbances according to Fig. 4.1. When extended by the prefilter with the time constant T_p to the FI_0 controllers, they may also be represented by the equivalent structure according to Fig. 4.2.

Definition 4.2 (I controller). Under I controller we will understand FI_0 controller with the prefilter time constant $T_p = T_f$ that may also be represented by the equivalent structure according to Fig. 4.2 in which the input filter dissapears.

In order to get acceptable filtration of the measurement noise and also to achieve desired robustenss against non-modelled loop delays and plant-model mismatch the DO time constant T_f cannot be set arbitrarily small. In a closed loop tuned for monotonic transient responses and under effect of admisssible input signals the control saturation will never be active and therefore it can be omitted from Fig. 4.1. That means the loop can be fully treated by linear methods. It is also not to forget that the first order filter used in FI_0-controllers does not represent the only available solution. Usefull

properties as e.g. improved robustness and noise filtering can be achieved by using higher order DO filters.

4.1.3 Fundamental Properties of I_0 and FI_0 Controllers

When the gain K_0 used for the controller tuning is not equal to real plant gain K, the transfer functions corresponding to reference/disturbance signal responses become

$$\begin{aligned} I_0:\ F_{wI0}(s) &= \frac{Y(s)}{W(s)} = \frac{K(T_f s+1)}{K_0 T_f s + K} = \frac{s/\Omega_f + 1}{s\kappa/\Omega_f + 1}\ ;\ \kappa = \frac{K_0}{K} \\ I:\ F_{wI}(s) &= \frac{Y(s)}{W(s)} = \frac{K}{K_0 T_f s + K} = \frac{1}{s\kappa/\Omega_f + 1}\ ;\ \Omega_f = \frac{1}{T_f} \\ F_{v,o}(s) &= \frac{Y(s)}{V_o(s)} = \frac{sK_0T_f}{K_0T_f s + K} = \frac{s\kappa/\Omega_f}{s\kappa/\Omega_f + 1}\ ;\ F_{v,i}(s) = KF_{v,o}(s) \end{aligned} \tag{4.8}$$

So, both responses depend on the reconstruction filter bandwidth Ω_f and on the ratio of the estimated and real plant gains $\kappa = K_0/K$. For $\kappa > 0$, the closed loop in Fig. 4.2 wil be (theoretically) stable for any positive value of Ω_f, (for any negative value of the closed loop pole $\alpha_f = -\Omega_f/\kappa$). By increasing $\Omega_f \to \infty$, $F_w(s) \to 1$, i.e. the exponential closed loop setpoint step responses $1 - \exp(-\Omega_f t/\kappa)$ are approaching unit step, what according to Def. 3.18 means that for the setpoint response this solution represents fundamental controller. Similarly, by increasing $\Omega_f \to \infty$, $F_{vi}(s) \to 0$, i.e. the disturbance closed loop step responses $K\exp(-\Omega_f t/\kappa)$ are converging to zero (more precisely, to the Dirac pulse $K\delta(t)$), what according to requirements of Def. 3.18 means that for the disturbance response this solution again represents fundamental controller.

4.1.4 Nonmodelled Dynamics Approximated by Dead-time – Analytical Treatment

In controlling ideal memoryless plant the controller gain (4.4) may increase to infinitely large values (DO filter time constant T_f may converge to zero). The only condition is that $\kappa > 0$, i.e. the estimated plant gain K_0 has the same sign as the real gain K. However, in controling real plants, no control loop is strictly memoryless one. By increasing reconstruction bandwith $\Omega_f = 1/T_f \to \infty$, the speed of transients increases. Since the concept of memoryless plants covers just situation, when the transients are sufficiently slow and

negligable, at some value of Ω_f this concept will become inappropriate to real loop behavior. As a result, overshooting and oscillations of loop variables occur.

Next, we are going to determine borders for validity of the concept of memoryless plant and to propose measures to enable its reliable use. In doing so, we will start with analyzing influence of the simplest loop dynamics. So, let us consider loop with I controller (Fig. 4.2) for $T_p = T_f$), with a memoryless plant and dead-time T_d, when the relation between the control signal u and the measured output $y_m = y_1$ (Fig. 4.3) is given as

$$F_{yd}(s) = \frac{Y_m(s)}{U(s)} = K\mathrm{e}^{-T_d s} \tag{4.9}$$

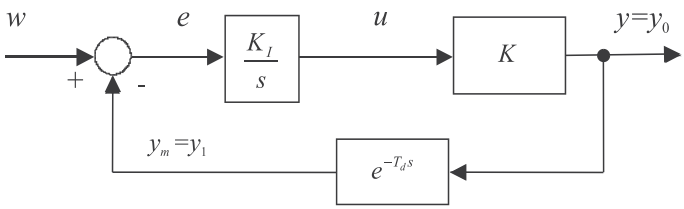


Fig. 4.3 Loop with I controller, memoryless plant and dead time

$$\begin{aligned} F_{w0}(s) &= \frac{Y_0(s)}{W(s)} = \frac{K_I K}{s + \mathrm{e}^{-T_d s} K_I K} = \frac{\mathrm{e}^{T_d s}\Omega_f/\kappa}{s\mathrm{e}^{T_d s} + \Omega_f/\kappa} = \frac{B_0(s)}{A(s)};\ \ \kappa = \frac{K_0}{K} \\ F_{w1}(s) &= \frac{Y_1(s)}{W(s)} = \frac{K_I K\mathrm{e}^{-T_d s}}{s + \mathrm{e}^{-T_d s} K_I K} = \frac{\Omega_f/\kappa}{s\mathrm{e}^{T_d s} + \Omega_f/\kappa} = \frac{B_1(s)}{A(s)};\ \ \Omega_f = \frac{1}{T_f} \end{aligned} \tag{4.10}$$

After introducing new complex variable

$$p = T_d s \tag{4.11}$$

these equation may be fully expressed in a normalized form

$$\begin{aligned} F_{w0}(p) &= \frac{Y_0(p)}{W(p)} = \frac{\mathrm{e}^p \Omega/\kappa}{p\mathrm{e}^p + \Omega/\kappa} = \frac{B_0(p)}{A(p)};\ \ \Omega = \frac{T_d}{T_f} \\ F_{w1}(p) &= \frac{Y_1(p)}{W(p)} = \frac{\Omega/\kappa}{p\mathrm{e}^{T_d p} + \Omega/\kappa} = \frac{B_1(p)}{A(p)};\ \ \kappa = \frac{K_0}{K} \end{aligned} \tag{4.12}$$

It is to see that the setpoint response fully depends on the parameter

$$q = \Omega/\kappa \tag{4.13}$$

For some tasks, it may be more appropriate to introduce instead of the parameter q its reciprocal value

$$\tau = \kappa/\Omega \tag{4.14}$$

In the case of nominal tuning ($\kappa = 1$) it denotes ratio of the filter time constant to the dead time $\tau = T_f/T_d$. One of the first method for analytical controller tuning (Oldenbourg and Sartorius, 1944, 1951) was based on derivation of conditions of the double real dominant pole.

Theorem 4.1 (I controller gain corresponding for T_d to the Double Real Dominant Pole (DRDP)). *Tuning of the I controller in the loop in Fig. 4.3 that should guarantee the fastest possible monotonic transients may be derived by using conditions for the double real dominant pole p_0 of the closed loop characteristic equation $A(p) = 0$ by satisfying conditions*

$$A(p_0) = 0;\ \dot{A}(p_0) = 0 \tag{4.15}$$

as

$$\begin{aligned} q_{opt} = \Omega/\kappa &= \exp(-1) \\ \Omega_f T_d \exp(1) &= \kappa \\ T_f &= T_d \exp(1)/\kappa \end{aligned} \tag{4.16}$$

In the plane of loop parameters (κ, Ω) equation represents a line corresponding to the fastest possible transients without overshooting. For $\Omega \exp(1) > \kappa$ the transients already have overshoots.

It is once more to remind that strictly MO transients can really be achieved for prefilter with $T_p \geq T_f$ in Fig. 4.1. Solution equivalent to $T_p = T_f$ is equivalent to omitting prefilter in the scheme in Fig. 4.2.

Definition 4.3 (Optimal DO bandwith, optimal DO time constant, optimal I controller gain for loop with T_d). As optimal DO time constant T_f, optimal DO bandwith $\Omega_f = 1/T_f$, optimal normalized DO bandwith $\Omega = T_d/T_f$ and optimal integral gain $K_I = 1/(K_0 T_f)$ of the dead-time system in Fig. 4.3 will be denoted those corresponding to the double real dominant pole (DRDP) given as

$$\begin{aligned} \Omega = \kappa/exp(1);\ \ \Omega_f = \kappa/(exp(1)T_d);\ \ T_f = 1/\Omega_f \\ K_I = 1/(K_0 T_d \exp(1)) \end{aligned} \tag{4.17}$$

Theorem 4.2 (Critical I controller gains). *Sustained closed loop oscillation with period $P_u = 2\pi/\omega$ orresponds to the root $s = j\omega$ of the characteristic equation $A(s) = 0$. Critical tuning and the corresponding period of oscillations P_u determined by substituting $s = j\omega$ (Neimark , 1973) into $A(s) = se^{T_d s} + \Omega_f/\kappa$ are*

$$\begin{aligned} \omega = 0 \;\Rightarrow\; P_u \to infty\,;\; \Omega_f/\kappa = 0 \\ \omega = 2\pi/T_d \;\Rightarrow\; P_u = 4T_d\,;\; \Omega_f/\kappa = \pi/\left(2T_d\right) \end{aligned} \tag{4.18}$$

The upper critical DO bandwith and the corresponding critical DO time constant are then given as

$$\Omega_{crit} = \kappa\pi/2\,;\; \Omega_{f,crit} = \kappa\pi/\left(2T_d\right)\;;\; T_{f,crit} = 2T_d/\left(\kappa\pi\right) \tag{4.19}$$

4.1.5 Nonmodelled Dynamics Approximated by Dead-time – Treatment by Performance Portrait

Alhough it might seem at the first glance that the analytically derived border of MO responses based on the DRDP gives reliable results, a detailed computer based analysis based on the Performance Portrait shows that the experimentally determined area of MO responses is slightly larger than the analytically derived one. In this alternative approach the loop behavior is mapped and analyzed over a grid of loop parameters in the plane (κ, Ω). From these data it is then possible to visualize the loop Performance Portrait in Fig. 4.4, or to derive parameters corresponding to a tolerable overshooting shown in Tab. 4.1. It is interesting to note that all values $\tau = \tau\left(\epsilon_y\right)$, including e.g. the simple tuning $\tau = 2$ proposed by Skogestad (2003) that corresponds to $100\epsilon_y = 4.04\%$, which are for $\epsilon_y \to 0$ converging to the value

$$\tau \to 2.703\ldots \tag{4.20}$$

are smaller than

$$\tau_{opt} = exp(1) = 2.718\ldots \tag{4.21}$$

corresponding to (4.14) and (4.17).

Of course, one could deal with the question, if the discrepancy in results is due to the limited precision of numerical computations, or it is expressing influence of infinitely many poles neglected in the double real dominant pole method. Although in this case differences may be observed just on the third decimal position, in general, the experimental qualitative & quantitative computer based analysis of step responses may give much deeper insight into closed loop properties than the analytical analysis of infinitely many closed loop poles of such dead time system. Simultaneously it e.g. shows on numerical issues that may be important not just for simulations, but also for the real time control. However, for vast majority of engineering tasks the identified differences in results do not play a primary role and we could conclude that in the case of the I-controller the analytically derived conclusions a coinciding with the results achieved by using the Performance Portrait. This method gives, however a more detailed information, what will become yet more important in dealing with more complex control tasks.

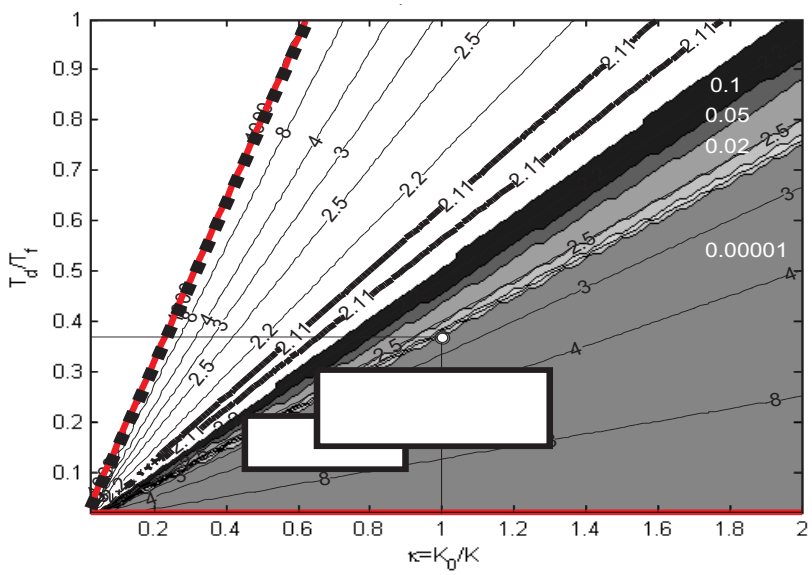


Fig. 4.4 Performance portrait of the loop with I controller and dead time from Fig. 4.3 including level contours corresponding to IAE values of the output y_1; areas of NO&MO output step responses identified for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$; a boundary point of the strictly NO&MO area is given by (4.17); no one of given ϵ_y areas does reach up to the area of optimal IAE1 values outlined by bold curces; the stability border (4.19) is given by bold dotted line; examples of uncertainty boxes are explained in following chapters

Table 4.1 IAE0 and IAE1 values and the corresponding controller tuning corresponding for $\kappa = 1$ to the outputs y_0 and y_1 under ϵ_y NO&MO setpoint step responses of the loop with I controller and nonmodelled dynamics approximated by the dead time T_d

$100\varepsilon_y$ [%]	10	5	4.04	2	1	0.1	0.01	0.001	0	**0**
$\tau = T_f / T_d$	1.724	1.951	2.0	2.162	2.268	2.481	2.571	2.625	2.703	**2.718**
$\Omega = T_d / T_f = 1/\tau$	0.580	0.515	0.5	0.465	0.441	0.403	0.389	0.381	0.37	**0.368**
IAE_0 / T_d	1.105	1.147	1.17	1.240	1.314	1.486	1.571	1.625	1.703	**1.718**
IAE_1 / T_d	2.105	2.147	2.17	2.240	2.314	2.486	2.571	2. 625	2.703	**2.718**

4.1.6 Nonmodelled Dynamics Approximated by Time Constant – Analytical Treatment

Another elementary possibility for approximating the nonmodelled loop dynamics is represented by single time constant (accmulative delay) corresponding to the plant transfer function

$$F_{ya}(s) = \frac{Y_m(s)}{U(s)} = \frac{K}{1 + T_a s} \tag{4.22}$$

For passive compensation of the nonmodelled dynamics (I controller tuning), the location of this delay within the control loop is not important and in order to cover both possible loop configurations we will consider transfer functions corresponding to two possible loop outputs in Fig. 4.2 (i.e. with $T_p = T_f$ in the generic schemes in Fig 4.1). For the new complex variable

$$p = T_a s \tag{4.23}$$

they again depend on the parameters $q = \Omega/\kappa$, or $\tau = \kappa/\Omega$

$$\begin{aligned} F_{w0}(p) &= \frac{Y_0(p)}{W(p)} = \frac{(1+p)\,\Omega/\kappa}{p\,(1+p) + \Omega/\kappa} = \frac{B_0(p)}{A(p)}; \quad \Omega = \frac{T_a}{T_f} \\ F_{w1}(p) &= \frac{Y_1(p)}{W(p)} = \frac{\Omega/\kappa}{p\,(1+p) + \Omega/\kappa} = \frac{B_1(p)}{A(p)}; \quad \kappa = \frac{K_0}{K} \end{aligned} \tag{4.24}$$

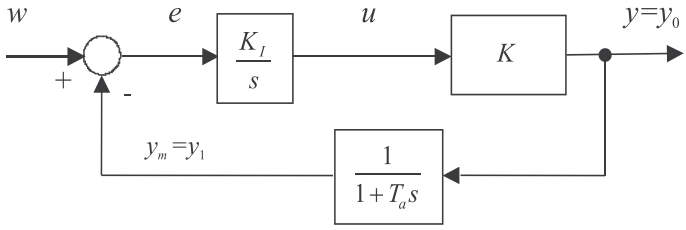


Fig. 4.5 Loop with I controller, memoryless plant and non-modelled dynamics approximated by time constant

Theorem 4.3 (I controller gain corresponding for T_a to the Double Real Dominant Pole (DRDP)). *Tuning of the I controller in the loop in Fig. 4.5 that should guarantee the fastest possible monotonic transients may be derived by using conditions for the double real dominant pole p_0 of the closed loop characteristic equation $A(p) = 0$ by satisfying conditions*

$$A(p_0) = 0; \quad \dot{A}(p_0) = 0 \tag{4.25}$$

as

$$q_{opt} = \Omega_{opt}/\kappa = 1/4; \quad T_f = 4T_a/\kappa \tag{4.26}$$

In the plane of loop parameters (κ, Ω) equation (4.26) represents a line corresponding to the fastest possible transients without overshooting. For $\Omega \exp(1) > \kappa$ the transients should already have overshooting.

Definition 4.4 (Optimal DO bandwith, optimal DO time constant, optimal I controller gain for T_a). As optimal DO time constant T_f, optimal DO bandwith $\Omega_f = 1/T_f$, optimal normalized DO bandwith $\Omega = T_d/T_f$ and optimal integral gain $K_I = 1/(K_0 T_f)$ of the system with time constant in Fig. 4.5 will be denoted those corresponding to the double real dominant pole (DRDP) given as

$$\Omega = \kappa/4\,;\;\; \Omega_f = \kappa/(4T_a)\;;\;\; T_f = 1/\Omega_f\,;\;\; K_I = 1/(4K_0 T_a) \tag{4.27}$$

Theorem 4.4 (Critical I controller gains for T_a). *Sustained closed loop oscillation with period $P_u = 2\pi/\omega$ corresponds to the root $s = j\omega$ of the characteristic equation $A(s) = 0$. In difference to the loop with dead time, for $\kappa > 0$ this loop remains stable for any $T_f > 0$, when the critical tuning and the corresponding period of oscillations P_u determined by substituting $s = j\omega$ into $A(s) = s(T_a s + 1) + \Omega_f/\kappa$ are*

$$\begin{aligned} \omega = 0 \;\Rightarrow\; P_u \to infty\,;\; \Omega_f/\kappa = 0\,;\; K_{I,min} = 0 \\ \omega \to \infty \;\Rightarrow\; P_u \to 0\,;\; \Omega_f/\kappa \to \infty\,;\; K_{I,max} \to \infty \end{aligned} \tag{4.28}$$

4.1.7 Nonmodelled Dynamics Approximated by Time Constant – Treatment by Performance Portrait

Comparison of results achieved for dead time Tab. 4.1 and time constant Tab. 4.2 shows that in the case of the time constant T_a increased values of tolerated overshooting lead to reasonably faster IAE decrease than in the case of the dead time T_d (Fig. 4.6). For 10% tolerated overshooting the IAE values corresponding to T_a and T_d are roughly equal. That has an important consequence on plant identification: by using the step responses based e.g. on measuring the average residence time. For systems with tolerable overshooting it has no sense to distinguish the type of nonmodelled dynamics. And conversely, it may be important in aiming to achieve the fastest possible MO responses with low admissible overshooting and low deviations from monotonicity.

Table 4.2 IAE0 and IAE1 values and the corresponding controller tuning corresponding for $\kappa = 1$ to the outputs y_0 and y_1 under ϵ_y NO&MO setpoint step responses of the loop with I controller and nonmodelled dynamics approximated by the time constant T_a

$100\varepsilon_y$ [%]	10	5	2	1	0.1	0.01	0.001	0	**0**
$y_0 : \tau = T_f / T_a$	1.754	2.169	2.604	2.857	3.367	3.597	3.731	3.968	**4**
$\Omega = T_a / T_f = 1/\tau$	0.570	0.461	0.384	0.350	0.297	0.278	0.268	0.252	**0.25**
IAE_0 / T_a	1.348	1.510	1.760	1.941	2.376	2.598	2.732	2.968	**3**
$y_1 : \tau = T_f / T_a$	1.398	1.908	2.433	2.724	3.311	3.571	3.717	3.968	**4**
$\Omega = T_a / T_f = 1/\tau$	0.715	0.524	0.411	0.367	0.302	0.280	0.269	0.252	**0.25**
IAE_1 / T_a	1.921	2.221	2.581	2.806	3.321	3.573	3.717	3.968	**4**

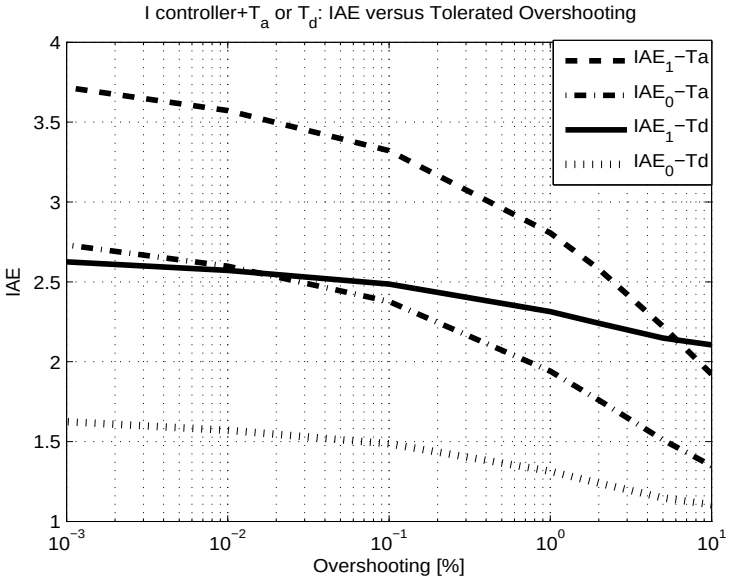


Fig. 4.6 IAE1 and IAE0 values versus tolerated overshooting for the time constantT_a and dead time T_d according to Tab. 4.1 and Tab. 4.2

4.1.8 Tuning Based on Maximal Sensitivity $M_s = 1.4$

Today, controllers are frequently tuned with the aim to guarantee chosen maximal sensitivity to modeling errors. This can be expressed as the maximal value of the sensitivity function defined as $S(s) = 1/(1 + L(s))$, whereby $L(s) = R(s)F(s)$ is the open loop transfer function with $R(s)$ being the transfer function of the controller and $F(s)$ being the plant transfer function. The maximal sensitivity is then given as

$$M_s = \max\{S(j\omega),\ S(s) = 1/(1 + L(s))\}; \quad L(s) = R(s)F(s) \tag{4.29}$$

Thereby M_s represents inverse value of the shortest distance R of the critical point $(-1, j0)$ from the Nyquist curve of the open loop transfer function $L(s)$. This may be determined by making circle with centre in the critical point that tougches Nyquist curve (Fig. 4.7). Typical M_s values (Åström and Hägglund, 1995; Åström et al, 1998) appropriate for control lie in the range $1.2 - 2.0$. Lower M_s values give slower, but less oscillatory transient responses. Why exactly these values? Do they represent universal constants defined by the nature? In order to get some interpretation we may compare the maximal sensitivity method with results achieved by the double real dominant pole.

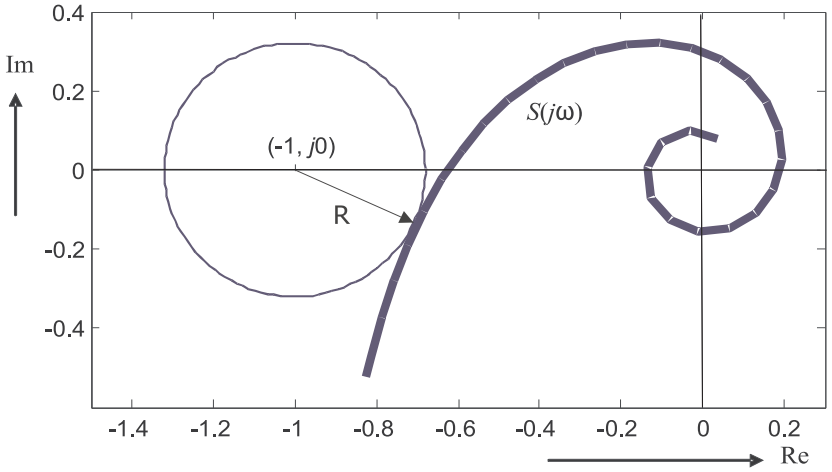


Fig. 4.7 Maximal sensitivity $M_s = 1/R$ is defined as reciprocal value of the maximal radius R of the circle with centre in critical point $(-1, 0j)$ that is touging Nyquist curve $L(j\omega)$

By evaluating maximal sensitivity corresponding in the nominal case to tuning (4.17) one gets $M_s = 1.3936 \approx 1.4$. The result does not depend on the particular dead time value T_d. It shows that the DRDP and the maximal sensitivity approaches are somehow related and explains possible motivation for Aström and coworksers to prefer exactly the value $M_s = 1.4$. However, calculating the maximal sensitivity corresponding to tuning (4.27) gives already different value $M_s = 1.155$. It means that the tuning corresponding to DRDP does not introduced universally valid *optimal* M_s value. Simultaneously, question arrises, which delay is more appropriate for approximations dealing with non-nmodelled loop dynamics: dead time (4.9) or time constant (4.22)? Experimental results show that in dealing with real loops tuning (3.18) gives mostly too conservative controller values (as it is also illustrated by Fig. 4.6)

that gives argument to work with T_d. For approximation (4.22) we might use also other alternative: to derive new controller tuning corresponding to value $M_s = 1.4$ what gives

$$T_f = 1.5T_a\,;\; T_p = 2.1T_f \tag{4.30}$$

Thereby, the prefilter time constant value has to be increased from $T_p = T_f$ more than two times to achieve nearly monotonic step response of y_1. Fig. 4.8 shows that such transients are much faster as for the DRDP tuning. By further increase of T_p it would also be possible to remove slight overshooting in the transients of the output y_0 and so also of the control signal. It is, however, to note that in the disturbance response the overshooting remains. So, approach based on loop shaping and dealing with the maximum sensitivity and the maximum complementary sensitivity (Skogestad and Postlethwaite, 1996; Skogestad, 2003) is indeed possible, but not primarily oriented to respect nonovershooting and monotonicity conditions at the plant and controller outputs. Furthermore, the expected dynamics is guaranteed just around the nominal operating point. Of course, it is expected that the less aggresive tuning with lower M_s values will allow higher degree of the plamt-model mismatch, but there are no easy ways of quantifying these expectations. Similar comments may also be used for other design methods based on the frequency response, as e.g. the Disturbance Rejection Magnitude Optimum method (DRMO, Vrančič et al (2004)) that gives

$$T_f = 2T_a; \quad T_p = 1.35T_a \tag{4.31}$$

These method may be advantageous in working with plant models achieved by identification based on the frequency response. Else, more direct approach of the Performance Portrait method that uses direct technological parameters as tolerated overshooting, or tolerated deviations from monotonicity will be preferred.

4.1.9 Short Summary of Nominal I_0-Controller Tuning

Previous analysis showed that already in designing simple I controller there exist several degrees of freedom: by choice of the prefilter T_p it is to decide about character of the control signal dynamics after setpoint steps – should it have a step character, or a softer, exponential one? That is: are we going to use controller according to Fig. 4.1 with equivalent structure in Fig. 4.2 with two unknown parameters T_p and T_f, or simplified solution according to Fig. 4.1 with prefilter $T_p = T_f$ that is equivalent to Fig. 4.2 without prefilter? This process of deciding about complexity of the solution could be continued by considering higher order DO filters in the generic scheme that might be interesting both from the point of view of noise filtering as well as from the closed loop robustness point of view.

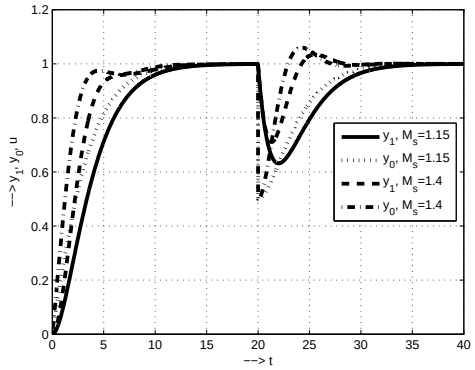


Fig. 4.8 Loop with the FI_0 controller and plant (4.22) with $T_p = T_f$ and T_f corresponding to (4.27) that yields $M_s = 1.15$ and with the tuning (4.30) corresponding to $M_s = 1.4$

It was also shown that the concept of memoryless plant used in deriving controller structure has to be refined by appropriate approximation of the non-modelled dynamics. This plays an important role in controller tuning enabling achieving the fastest possible ϵ_y-MO&NO transient responses. In this step we have analysed basic properties achieved by approximations of the nonmodelled dynamics by the dead time and by single time constant. Of course, in practice also their combination, or higher order approximations may be proposed, as e.g. the approximation by $1/(1 + T_a s)^2$, used by Glattfelder and Schaufelberger (2003). But, when allowing tolerated overshooting around 5% of the setpoint step, influence of both types of approximations of the nonmodelled dynamics is approximately equal and more important question becomes, which approximations are easier to be achieved. A more rigorous approximation of the nonmodelled dynamics has sense just for a high precission control.

In specifying the loop dynamics we have concentrated our effort on conditions of achieving ϵ_y-NO&MO transients that showed to correspond to identical conditions in this case. This may be important both in the technological context, both in decreasing acturator wear and in the constrained control design. At the controller output, MO transient from one admissible steady state to another one will never excite control saturation and so it is possible to omit its effect from the control loop analysis.

Aproximation of the loop dynamics may be based on measuring setpoint step responses, by evaluating system response at the stability border (when it is possible and allowed to bring system to oscillations by appropriate controller tuning), by relay experiment, etc.

For loop with memoryless plant and dead (4.9) gave the I controller tuning based on the DRDP already one of the first control text books by Oldenbourg

and Sartorius (1944, 1951) that indicates importance of the solution for practice. Since the controller is derived for the simplest plant model it may be universally used for controlling broad spectrum of stable plants, what e.g. inspired Datta et al (2000) to speak about "magic of integral control". I controllers are appropriate also for systems with long (and possible variable) dead times. Rugh and Shamma (2000) e.g. presents there use in combustion engines that are typical by long delay between engine fuelling and exhaust emissions. But, above mentioned tuning rules are still fixed just to a nominal point and should futher be extended to more general situation with plant parameters varying over broader intervals.

4.1.10 Robust Controller Tuning and Characteristics

In practical applications, loop parameters are mostly known with some degree of uncertainty. Plant properties may vary in time (time variable plants), due to operating point changes (nonlinear plants), or they may be simply identified with a limited precission. How to tune the controller, when it is required to guarantee some performance, whereas the loop parameters K, T_a or T_d are not known exactly, but they are given just with interval uncertainty as

$$\begin{aligned} K &\in \langle K_{min}, K_{max} \rangle \; ; \; c_K = K_{max}/K_{min} \geq 1 \\ T_d &\in \langle T_{d,min}, T_{d,max} \rangle \; ; \; c_d = T_{d,max}/T_{d,min} \geq 1 \\ T_a &\in \langle T_{a,min}, T_{a,max} \rangle \; ; \; c_a = T_{a,max}/T_{a,min} \geq 1 \end{aligned} \tag{4.32}$$

When interpreting such a situation by means of Fig. 4.4 it is to note that changes of the plant gain K influence possible values of $\kappa = K_0/K$. For a chosen value K_0 it is possible to find limit values

$$\kappa_{min} = K_0/K_{max} \, ; \; \kappa_{max} = K_0/K_{min} \tag{4.33}$$

that within the parameter plane (κ, Ω), determine range of horizontal movement of the working point. For a constant and exactly know value of $\Omega = T_d/T_f$, or $\Omega = T_a/T_f$ the uncertainty set is reduced to a horizontal uncertainty line segment (ULS) with vertices corresponding to (4.33).

Similarly, for a chosen DO badwidth $\Omega_f = 1/T_f$ it is possible to find limit values of $\Omega = T_d/T_f$, or $\Omega = T_a/T_f$ as

$$\begin{aligned} \Omega_{min} &= T_{d,min}/T_f \, ; \; \Omega_{max} = T_{d,max}/T_f \\ \Omega_{min} &= T_{a,min}/T_f \, ; \; \Omega_{max} = T_{a,max}/T_f \end{aligned} \tag{4.34}$$

If the only uncertainty is related to the nonmodelled dynamics, whereby the plant gaing is exactly known, the uncertainty set will be given by vertical ULS with the horizontal position $\kappa = K_0/K$ and vertices (4.34).

By combining extreme values of two independent parameters (4.32) one gets uncertainty box (UB) with vertices corresponding to (4.33) and (4.34) as

$$UB = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} \kappa_{\min}, \Omega_{\max} & \kappa_{\max}, \Omega_{\max} \\ \kappa_{\min}, \Omega_{\min} & \kappa_{\max}, \Omega_{\min} \end{bmatrix} \tag{4.35}$$

To guarantee required property (ϵ_y-NO&MO control with specified tolerance ϵ_y) for all possible situations, it is then required that the whole UB lies in parameter area guaranteeing given property. With respect to the shape of the border of NO & MO control in Fig. 4.4 it is obvious that the critical role will be played by the upper left vertex

$$B_{11} = (\kappa_{min}, \Omega_{max}) \tag{4.36}$$

whereby $\Omega_{max} = T_{d,max}/T_f$, or $\Omega_{max} = T_{a,max}/T_f$. In the analytical design this can be placed at one of the line borders (4.17), or (4.27). That means to fulfill for all possible working points requirements

$$K_0 T_f \geq exp(1) K_{max} T_{d,max}\,;\; K_0 T_f \geq 4 K_{max} T_{a,max} \tag{4.37}$$

Due to the radial shape of the performance portrait, by shifting ULS or UB along chosen lines to any position the closed loop properties do not vary. However, by increasing ratio of the upper and the lower limit value in (4.32) the mean value of IAE index over the uncertainty set will increase. As it is evident from Fig. 4.9, the rate of increase depends on the type of the nonmodelled dynamics and on the tolerated overshooting.

For T_a and strictly MO tuning both IAE values increase due to the ucertainty much more rapidly than for T_d, but for the 10% tolerated overshooting the increase is in both cases practically equivalent and much less intensive than in the MO case. From this point of view we come to a surprising result: in the case of the I controller it is easier to control loops with dead time than loops with equivalent time constant value. Since by increasing the uncertainty coefficients c_a, or c_d the maximal values in (4.32) and so also the required T_f values in (4.37) increase linearly, also the IAE values in 4.9 increase linearly. It is also to note that in the case of a time constant tuning corresponding to certain overshooting of the output y_0 differs from that corresponding to the output y_1. From this point ov view it is to expect that the step disturbances entering to the closed loop at different points will in the case of considering tolerable overshooting require special attention.

Example 4.1. In this illustrative example we will show robust design and the corresponding robust performance achievable by using the simplest possible I controller and then compare these results with the much more complex Filtered Smith Predictor (FSP) according to Normey-Rico and Camacho (2007);

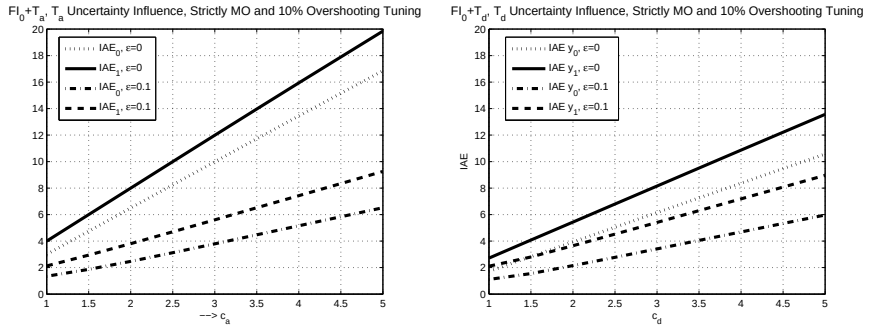


Fig. 4.9 T_a uncertainty influence on average IAE values of outputs y_0 and y_1 for strictly MO tuning ($K_I = 0.25/(KT_{a,max})$) and tuning with 10% tolerable overshooting of y_0 with $K_I = 0.57/(KT_{a,max})$ (left) and equivalent uncertainty influence for T_dăwith strictly MO tuning ($K_I = 0.37/(KT_{d,max})$) and tuning with 10% tolerable overshooting with $K_I = 0.58/(KT_{d,max})$

Example 6.1. The uncertain plant to be controlled is

$$F(s) = \frac{K_p \mathrm{e}^{-Ls}}{(1+s)(1+0.5s)(1+0.25s)(1+0.125s)} \qquad (4.38)$$
$$K_p \in \langle 0.8,\, 1.2\rangle \;;\; L \in \langle 9,\, 12\rangle$$

The FSP controller using primary PI-controller

$$C(s) = K_c \frac{1+T_I s}{T_I s} \qquad (4.39)$$

was tuned using standard robust approach in the frequency domain based on a nominal plant and norm bounded multiplicative uncertainty. As the nominal model an approximation of the original plant by the FOPDT one with

$$F_{apr}(s) = \frac{K_n \mathrm{e}^{-L_n s}}{1+T_n s} \;;\; K_n = 1 \;;\; L_n = 10.5 \qquad (4.40)$$

was used. Robust stability was proven for $K_c = 1\,;\, T_I = T_n$ and $T_f = L_n/2$. But, this method is not able to guarantee higher requirements on MO transients, expressed e.g. by the amplitude related deviations, or TV_0 values as it is evident from the PP in Fig. 4.10 left. So, it does not enable to design controller for more advanced applications. From Fig. 4.10 right it is to see that even the 20 times larger filter time constant does not reasonably improve the considered loop performance for larger plant gains: the controller needs to be fully retuned, possibly by the PP method.

Tuning of the I-controller will be based on the average residence time interpreted as dead time T_d

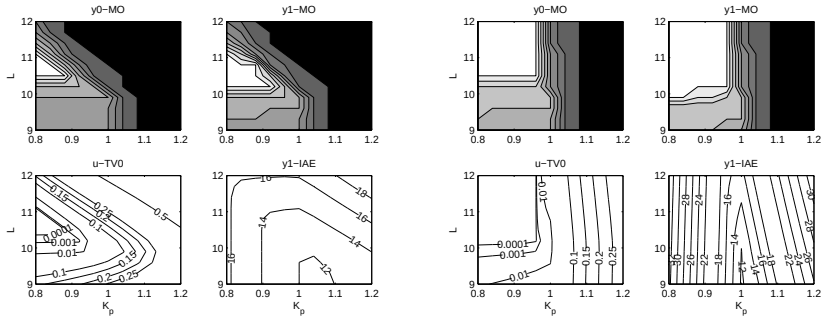


Fig. 4.10 PP of the plant (4.38) with the FSP controller based on (4.39)- (4.40) with $T_f = L_n/2$ (left) and $T_f = 10L_n$ (right); ϵ_y-MO areas identified for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$ with white denoting the best performance

$$A_0 = KT_d\,;\; A_0 = \int_0^\infty \left[y\left(\infty\right) - y\left(t\right)\right] dt \tag{4.41}$$

by the step responses (Åström and Hägglund, 1995), or by a general input signal according to Ingimundarson (2000), when for the maximal dead time L one gets for (4.38)

$$T_{d,max} = L_{max} + 1 + 0.5 + 0.25 + 0.125 = 13.875 \tag{4.42}$$

When choosing $K_0 = 1$, the filter time constant may be determined according to (4.37) as $T_f = \tau(\epsilon_y)T_{d,max}K_{max}$, whereby the values for 2% and 5% were taken from Tab. 4.1. From the PP in Fig. 4.11 it is obvious that for the output y_1 the deviations achieved in the critical corner exactly match the expectations, so that no corrections are necessary. Since the MO conditions are nearly matched also by the output y_0, it means that any output corresponding to some distribution of dynamical terms in (4.38) among the feedback and the feedforward path would match the required specification. Explanation for this (may be surprising result), when the extremely simple model gives precise results, may be taken from the same source as the above example (Normey-Rico and Camacho (2007), pp. 174): "when the dead-time is dominant, the contribution of the open loop poles to the closed loop response will be small thus their elimination will contribute with a small increment in the speed of the transients". Model used for tuning of the I controller fully respects this statement - whereas the model (4.40) used by authors of this statement not. The PPs in Fig. 4.10 and Fig. 4.11 fully confirm also another statement of above authors (pp. 145) "the effect of dead-time error is not symmetric", just the method they have used does not allow dealing effectively with this problem.

Simple I controller yields indeed higher IAE values than the much more complex FSP. However, up to now there exists no method for reliable tuning

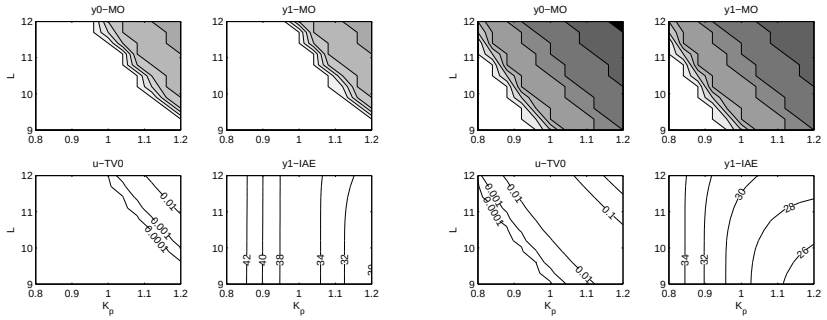


Fig. 4.11 PP of the I controller with T_f tuned to guarantee lower than 2% ($\epsilon_y = 0.02$, left) and lower than 10% deviations (right) from MO. MO areas correspond to $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$ with white showing the best performance

of FSP with respect to higher performance requirements. Having this fact in mind, several authors developed interactive tools to fight with this problem by the 'trial and error' method. Using the PP method the FSP may be redesigned to respect also this problem in a direct way.

4.2 PI_0 Controllers

In order to get MO transients, in the case of increasing time constant (accumulative delay) it is frequently not enough to compensate its influence just by restricting the closed loop bandwidth. Besides of slower transients this way of compensation brings also increased influence of disturbances. In many situations such impact is not acceptable and there is arising demand on active compensation of the time delay that would avoid these negative phenomena. Active compensation of dominant loop time constant leads to new control structures denoted here as PI, PI_0 and FPI_0 controllers. In deriving their structure the time constant T_a aproximating originally nonmodelled dynamics will now be denoted as the dominant loop time constant T_1. We will start by extending structure in Fig. 4.1 (similarly as in Fig. 4.5) by such a time constant denoted now as T_1.

4.2.1 Different Types of PI_0 and FPI_0 Controllers

Definition 4.5 (Active compensation of stable time constant - inversion of dynamics). Within the DC0, under "active compensation" of the loop time constant T_1 ("acumulative delay") it will be understood reconstruction of the estimate $\hat{y}$ of the actual loop output $y = y_0$ from the measured

delayed output $y_m = y_1$ that can be expressed as

$$\begin{aligned} \hat{Y}(s) &= (1 + T_1 s)\, Y_m(s) \\ \hat{y}(t) &= y_m(t) + T_1 \mathrm{d}y_m(t)/\mathrm{d}t \end{aligned} \tag{4.43}$$

Such a reconstruction of the input signal of a dynamical system from measured values of its output is called as inversion of its dynamics.

Active compensation of single time constant, i.e. inversion of its dynamics, is based on using inverse plant transfer function (inverse model). Such inversion may, however, be carried out just for stable systems. Output of an unstable system located in the feedback of Fig. 4.12 would be for a constant output increasing exponentially to infinite magnitudes. Under finite precission, such a signal cannnot be processed by real equipment. Besides of this, with respect to physical feasibility of inversion and to avoid algebraic loops, reconstruction of actual output will require additional filtration.

Active compensation of the time delay incorporated into reconstruction and compensation of disturbances leads to control structure in Fig. 4.12. In a loop with admissible inputs and monotonic transients of control signal the control saturation will never be active and so it can be omitted and the loop may be represented by the well known structure with the linear PI-controller $R(s)$ and the equivalent prefilter $T_e(s)$

$$\begin{aligned} R(s) &= K_c \frac{1 + T_I s}{T_I s}\,;\ K_c = \frac{T_{10}}{K_0 T_f}\,;\ T_I = T_{10} \\ T_e(s) &= \frac{1 + T_{f1} s}{(1 + T_{p1} s)\,(1 + T_{p2} s)}\,;\ T_{f1} = T_f\,;\ T_{p1} = T_{10}\,;\ T_{p2} = T_p \end{aligned} \tag{4.44}$$

After cancelling T_f in the prefilter numerator (4.44) by choosing $T_{p1} = T_{10} = T_f$ and $T_{p2} = T_p > 0$, step changes of control signal produced by a setpoint step will change to smoother exponential changes and the structure corresponds to the Two Degreee of Freedom (2DOF) PI controller that is also equivalent to the PI controller with error acting on I action (integral part) only, while the P-action has as input negative measured output. When furthermore $\mathrm{T}_{p2} = T_p = 0$ the structure reduces to the traditional PI controller without the equivalent prefilter.

$$U(s) = -K_p Y_m(s) + E(s)/\left(K_0 T_f\right)\,;\ E(s) = W(s) - Y_m(s) \tag{4.45}$$

When starting by defining the traditional PI controller parameters K_p and T_I, then according to (4.44) it is possible to calculate the equivalent prefilter parameters of the structure in Fig. 4.12 and to denote them as

$$T_{p1} = T_I\,;\ T_{p2} = 0\,;\ T_{f1} = T_I/\left(K_p K_0\right) \tag{4.46}$$

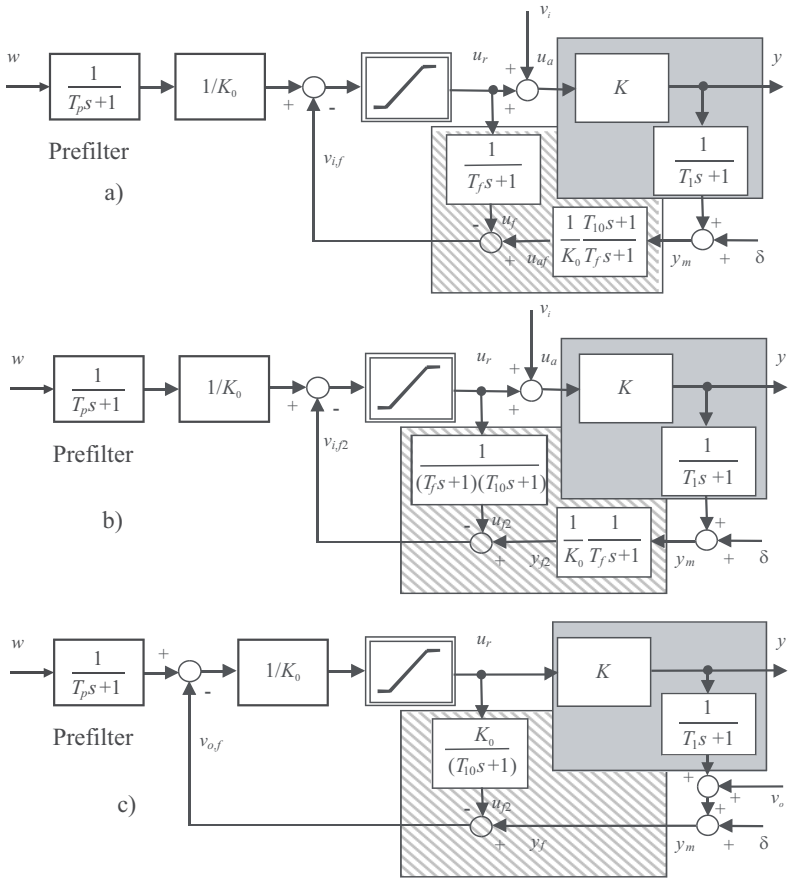


Fig. 4.12 a) Fundamental PI_0-IM controller designed as static feedforward control extended by input disturbance reconstruction and compensation using Inverse plant Model (cancelling time constant T_1); b) Modification of the I_0 controller for input disturbance reconstruction with balanced DO including loop time constant estimate in the channel from the controller output and c) Parallel plant Model of the PI_0-PM controller (PI_0-IMC controller); all structures may be extended by prefilter

Because of pole zero cancellation in the prefilter, controller (4.45) with the error acting on I only that gives smoother (exponential) control error decrease, may be simpler described by the prefilter parameters

$$T_{p1} = T_I\,;\; T_{p2} = 0\,;\; T_{f1} = 0 \tag{4.47}$$

Definition 4.6 (Fundamental PI_0-IM, FPI_0-IM and PI controllers based on Inverse Model of the dominant loop dynamics). The static feedforward control extended by the reconstruction and compensation of in-

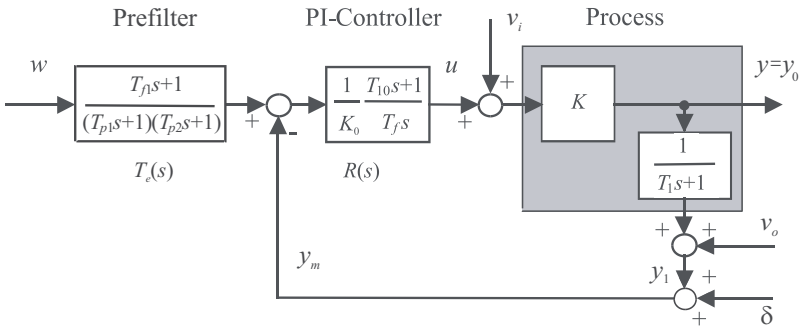


Fig. 4.13 Equivalent scheme of the PI_0-IM controller from Fig. 4.12; tuning $T_{f1} = T_{p1} = T_f$ gives the 2DOF PI controller with the simplest prefilter $T_e = 1/(1 + T_{10}s)$

put disturbances and compensation of single loop time constant using inverse model of the dominant loop dynamics will be denoted as the PI_0-IM controller here. It is a fundamental solution fulfilling conditions of Def. 3.18. When extended by a prefilter with the time constant $T_p > 0$ it will be denoted as the FPI_0-IM controller. For $T_{f1} = T_{p1} = T_f = T_{10}$, when also the prefilter (4.44) of the equivalent loop in Fig. 4.13 reduces to the prefilter with single time constant $T_{p2} = T_p$, one gets the structure of the 2 degree of freedom (2DOF) PI controller. When also $T_p = 0$ the structure reduces to the traditional PI controller. The closed loop transfer functions of the FPI_0-IM and 2DOF PI controller with the plant-model parameter dismatch are given as

$$\begin{aligned}
F_{w0}(s) &= \frac{Y_0(s)}{W(s)} = \frac{K(1+T_f s)(1+T_1 s)}{(1+T_p s)\left[K_0 T_f T_1 s^2 + (K_0 T_f + K T_{10})s + K\right]} \\
F_{w1}(s) &= \frac{Y_1(s)}{W(s)} = \frac{K(1+T_f s)}{(1+T_p s)\left[K_0 T_f T_1 s^2 + (K_0 T_f + K T_{10})s + K\right]} \\
F_{w0p}(s) &= \frac{K(1+T_1 s)}{K_0 T_f T_1 s^2 + (K_0 T_f + K T_{10})s + K};\ T_{f1} = T_{p1} = T_{10} \\
F_{w1p}(s) &= \frac{K}{K_0 T_f T_1 s^2 + (K_0 T_f + K T_{10})s + K};\ T_{f1} = T_{p1} = T_{10}
\end{aligned} \tag{4.48}$$

It is to note that the equivalent scheme according to Fig. 4.13 is now allways realizable, i.e. also for the FPI_0-IM controller with $T_p = 0$. Since for $K_0/K > 0$, $T_f > 0, T_{10} > 0$ and $T_1 > 0$ all denominator coefficients are positive, system remains robustly stable for any such a tuning.

Besides of use of the inverse dynamics, an equal balancing of both reconstruction channels may also be achieved by alternative solutions with the time constant T_1 included in the DO path from the controller output in Fig. 4.12b. Instead of this it is more frequently used solution with reconstruction and compensation of the output disturbance using the parallel plant model in Fig. 4.12c.

Definition 4.7 (PI_0-PM controller with Paralel plant Model typical for the IMC structures). In loops with the time constant T_1 the input disturbance reconstruction used in the I_0 controller can be improved by inserting identified time constant T_10 into the DO reconstruction branch leading from the controller output (Fig. 4.12b) to balance equally both reconstruction channnels. When designed for reconstruction and compensation of the output disturbance (Fig. 4.12c), the resulting PI_0-PM controller will become identical with the IMC control structure that is known by low noise sensitivity and good robustenss. Transfer functions describing setpoint responses may be achieved from (4.48) by susbstituting T_{10} instead of T_f. Similarly, also the responses to input disturbances cannot be arbitrarily speeded up and are determined by the time constant T_{10}. This structure does not fullfill requirements on the fundamental solutions from Def. 3.18 and therefore it will not be further analyzed here.

4.2.2 PI_0-IM: Analytical Versus Numerical Robust Tuning

Analytical approach to controller tuning may be based on the position and character of the closed loop poles. In such a case, character of the transient responses is determined by roots of the characteristic polynomial corresponding to different loop parameters. Closed loop poles corresponding to (4.48) are

$$s_{1,2} = -\frac{K_0 T_f + K T_{10}}{2K_0 T_f T_1} \pm \frac{\sqrt{(K_0 T_f + K T_{10})^2 - 4K K_0 T_f T_1}}{2K_0 T_f T_1} \tag{4.49}$$

Transients are expected to change qualitatively when the discriminant in (4.49) changes its sign, i.e. when

$$(K_0 T_f + K T_{10})^2 - 4K K_0 T_f T_1 = 0 \tag{4.50}$$

By denoting

$$\kappa = K_0/K > 0\,;\ \tau_f = T_f/T_{10} > 0\,;\ \tau_1 = T_1/T_{10} > 0\,;\ \tau_p = T_p/T_{10} > 0 \tag{4.51}$$

the last equation may also be rewritten as

$$\tau_1 = \frac{(\kappa\tau_f + 1)^2}{4\kappa\tau_f} \tag{4.52}$$

The closed loop performance portrait showing dependance of the shape of transient responses on the loop parameters is appropriate to be identified for

dimensionless parameters (4.51). It can be derived from (4.48) by introducing new complex variable

$$p = T_{10}s \tag{4.53}$$

when

$$\begin{aligned} F_{w0}(p) &= \frac{Y_0(p)}{W(p)} = \frac{(1+\tau_f p)(1+\tau_1 p)}{(1+\tau_p p)\left[\kappa\tau_f\tau_1 p^2 + (\kappa\tau_f + 1)p + 1\right]} \\ F_{w0p}(p) &= \frac{(1+\tau_1 p)}{\kappa\tau_f\tau_1 p^2 + (\kappa\tau_f + 1)p + 1};\ \tau_p = \tau_f \end{aligned} \tag{4.54}$$

New complex variable p means also new scale in the time domain, whereby, for the same input, the time transients corresponding to p in (4.54) will be times faster than transients corresponding to s in (4.48). It means that all properties related to time (as e.g. IAE or ISE performance indices) identified in the dimensionless variables have to be multiplied by this factor. Note (Fig. 4.14) that for the output y_0 (input of the time constant) of the PI_0 controller achieved for $T_p = 0$ the NO areas are no more identical with those corresponding to MO output, as in the case of the I controller and that to larger values of $\tau_f = T_f/T_{10}$ correspond enlarged areas of NO and MO control.

From the closed loop transfer functions of the PI_0 controller achieved for $T_p = 0$ from (4.48), or (4.54) it may be deduced that for $K > K_0 \Rightarrow \kappa < 1$ due to the 2nd order polynomials in numerator and denominator giving $F_{w0}(\infty) > 1$ the setpoint step responses of $y_0$incline to overshooting. This will restrict choice of appropriate tuning for achieving NO and MO output to $K_0 \geq K_{max}$. Only here gives the aperiodicity border (4.50), (4.52) some usefull information. Output monotonicity of the setpoint responses of y_0 could be be improved by canceling one of the numerator time constants by prefilter. Since the plant time constant may vary in time, the simplest solution is to use prefilter with tuning $T_p = T_f$, or the equivalent controller without prefilter (i.e. with $T_{p1} = T_{f1}$ in (4.44)). For the output y_1 the areas of NO and MO control coincides. The aperiodicity border (4.50), (4.52) gives for $\kappa < 1$ some usefull information just for small values of τ_f. All these subtle nuances shows that impact of the robust analytical tuning based on the closed loop pole is very restricted by its nature. The u-TV values with $\mathrm{TV}_{min} = 1$ correspond to unit step of the setpoint signal and $K = 1$. In order to eliminate dependance on these parameters, it is more appropriate to work with TV_0 criterion.

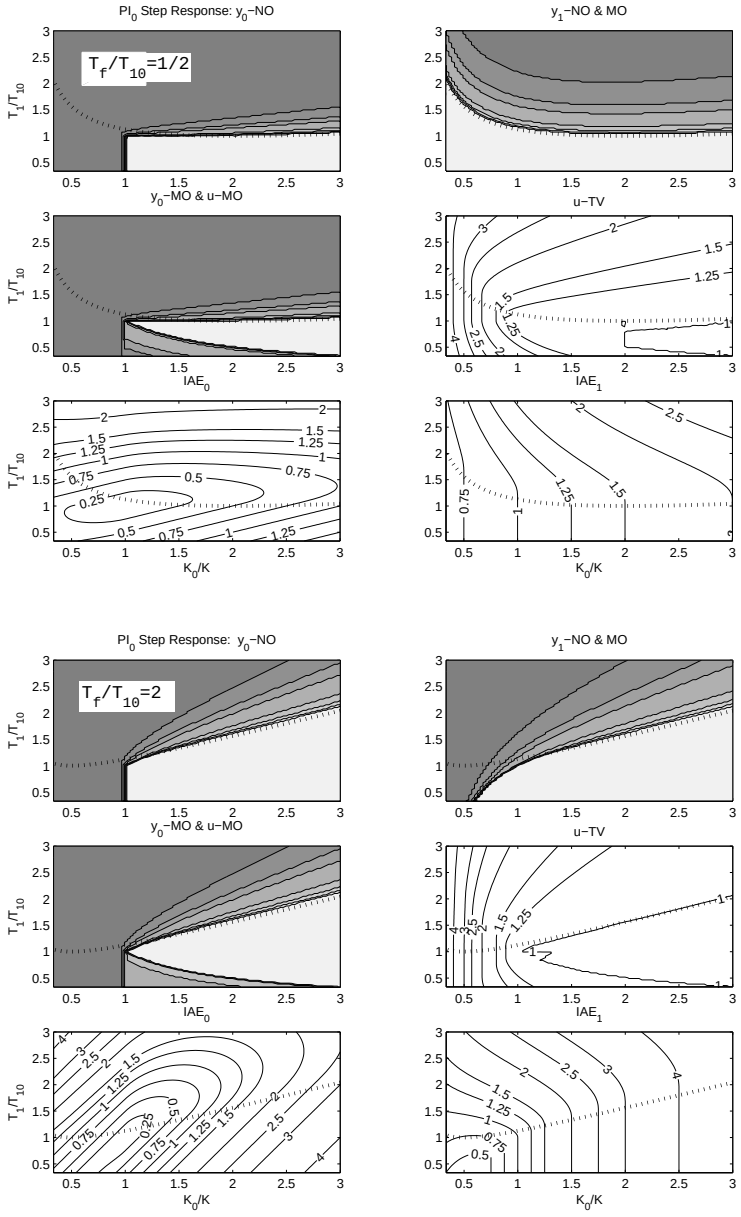


Fig. 4.14 Performance portrait of the PI_0-IM controller for setpoint response with two values of $\tau_f = 1/2$ and $\tau_f = 2$ and different measurement precissions $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$ with white showing the best performance; dotted border of complex poles (4.50), (4.52)

4.2.3 PI_0-IM: Impact of the Parameter Mismatch on Setpoint Steps

In tuning the controller one will usually ask: "which tuning will guarantee chosen qualitative shape of transients and simultaneously give minimal IAE values for the whole possible extent of parameter changes?"

In answering this question, let us formulate the control task in robust design of the PI_0-IM controller more precisely. For the plant uncertainty given as

$$\begin{aligned} K &\in \langle K_{min}, K_{max} \rangle \; ; \; c_K = K_{max}/K_{min} \geq 1 \\ T_1 &\in \langle T_{1,min}, T_{1,max} \rangle \; ; \; c_T = T_{1,max}/T_{1,min} \geq 1 \end{aligned} \tag{4.55}$$

that in the plane of normalized parameters (κ, τ_1) yields uncertainty boxes of all possible operating points

$$UB = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} \kappa_{min}, \tau_{1,max} & \kappa_{max}, \tau_{1,max} \\ \kappa_{min}, \tau_{1,min} & \kappa_{max}, \tau_{1,min} \end{bmatrix} \tag{4.56}$$

or in the case with single uncertain parameter corresponding to the uncertainty line segment (ULS) the task is to find controller tuning guaranteeing fastest possible transients (with minimal average IAE value). In fulfilling this task it is firstly required to identify the loop performance portrait corresponding to dimensionless variables (e.g.by fixing plant values $K = 1, T_1 = 1$ and by mapping system behavior for interesting range of plant values K_0 and T_{10} (for intervals larger than given by (4.55) and for some range of values $\tau_f = T_f/T_{10}$. All interesting results achieved by computer simulation will then be stored within the 3D space of dimensionless parameters (4.51).

Examples of sweeping parameter area corresponding e.g. to ϵ-MO output y_0 and looking for appropriate UB lying completely in it are in Fig. 4.15. By using uncertainty information represented by (4.55) it is necessary to sweep over all possible values of T_f for UB (4.56) or ULS defined by ratios of extreme values of uncertain parameters c_K, or c_T and lying in the required performance area. During this step, from identified values of particular UB (4.55) one has to recalculate the task from fixed controller tuning K_0, T_{10} and variable plant parameters K, T_1 to fixed limit loop values (4.55) and variable controller tuning corresponding to the optimal position of UB according to

$$K_0 = \kappa_{min}^{opt} K_{max} \; ; \; T_{10} = T_{1,min}/\tau_{min}^{opt} \; ; \; IAE_{mean} = T_{10} IAE_{mean}^{opt} \tag{4.57}$$

Finally, the identified optimal tuning has to be verified by simulation to guarantee required degree of output monotonicity and overshooting. Due to truncation errors, results fulfilling given condition may be shifted by one quantization step that may be important especially when working with lower number of points in the parameter grid. In such a case, finer controller tuning

could reasonably improve the resulting control performance. The calculation may be accelerated by generating new performance portrait just for a limited range of unknown parameters. Minimal IAE values usually correspond to UB shifted as much as possible to the values with $\kappa \geq 1$, i.e. to $K_0 \geq K_{max}$.

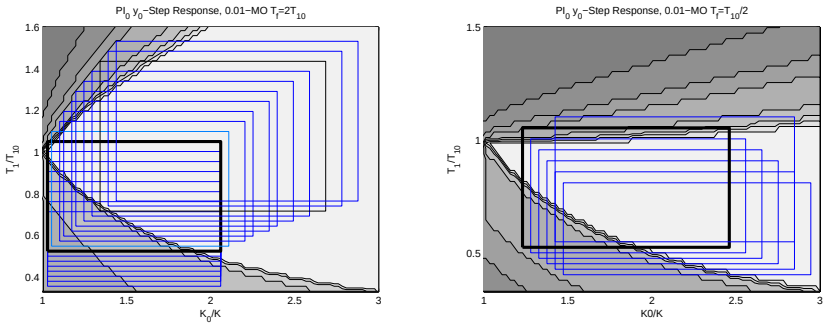


Fig. 4.15 Uncertainty boxes corresponding to interval plant parameters (3.43) for $\tau_f = 2$ (left) and $\tau_f = 1/2$ (right) and tolerated overshooting 1%; optimal UB (bold) $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$ with white showing the best performance;

We may use this fact in simplifying visualization problems in 3D, when by supposing $K_0 = K_{max}$ it is possible to decrease the number of uncertain parameters to one and to work in 2D parameter space. Also here we may expect that for the output y_0 of the PI_0 controller the transients may be monotonic for $\tau_1 \leq 1$ and $\tau_f > 0$. Without considering nonmodelled dynamics, the transient responses may be arbitrarily speeded up by decreasing the DO filter time constant and the IAE value over the ULS may be made to be arbitrarily small. From this point of view, use of more complex PI_0-IM controller seems to be much more advantageous than the use of I_0 controller with uncertainty characteristics in Fig. 4.9. However, even in the situations with negligable uncertainty of the dominant dynamics parameters, in real loops the process of speeding up transient responses by decreasing the DO filter time constant will be limited by the every time present nonmodelled dynamics. The DO filter time constant must remain larger that the largest time constant or dead time approximating the nonmodelled dynamics.

Example 4.2 (Tuning of the PI_0 Controller for Limited TV_0 Values). The task is to tune robustly PI_0 controller to guarantee for setpoint step response limited values of $\mathrm{TV}_0 < \mathrm{TV}_{0,max} = 0.1$ for the plant uncertainty limits

$$T_1 \in \langle 1, 1.5 \rangle \; ; \; K \in \langle 10, 20 \rangle \tag{4.58}$$

Performance Portrait (Fig. 4.16) was generated over 100x100 points for $T_{10} = 1, K = K_0 = 1$, $\tau_1 \in \langle 0.01, 2 \rangle$ and $\tau_f \in \langle 0.01, 5 \rangle$. In order to keep the disturbance response as fast as possible, controller will be tuned by using the

smallest possible T_f value enabling to achieve the required performance. Localization of the corresponding ULS in the PP is shown by bold line segment. The tuing parameters are determined according to (4.57). By sweeping the PP one gets

$$\begin{aligned} \tau_f = 2.1327\,;\ \tau_{1,min}^{opt} = 0.8101\,;\ IAE_{1,mean}^{opt} = 1.0267 \\ T_{10} = T_{1,min}/\tau_{1,min}^{opt} = 1.2344\,;\ T_f = \tau_f T_{10} = 2.6327 \\ IAE_{1,mean} = T_{10} IAE_{1,mean}^{opt} \end{aligned} \tag{4.59}$$

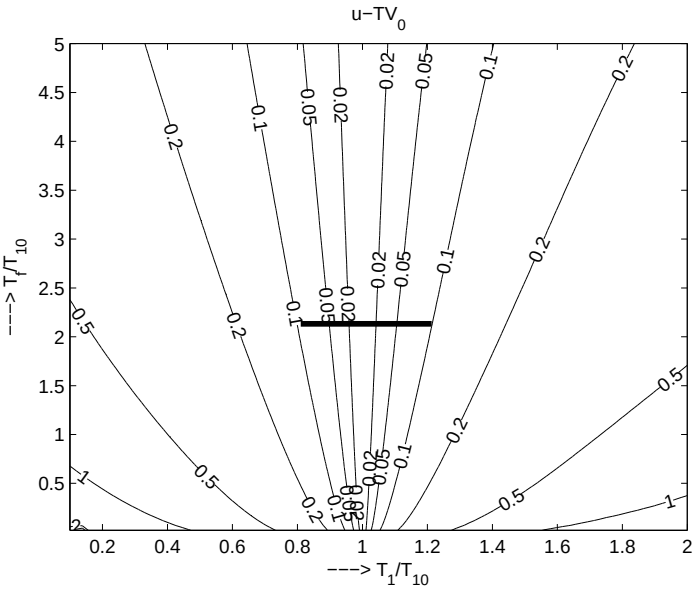


Fig. 4.16 PI_0: Performance portrait for the TV_0 values of the setpoint step responses with the ULS corresponding for the minimal possible value τ_f to the limit T_1 values (4.58) and to $TV_0 < 0.1$

From the setpoint step responses coresonding to limit values (4.58) and satisfying $TV_0 < 0.1$ it is to see that the amplitude deviations from monotonicity and nonovershooting may be approximatelly expressed as

$$\epsilon_y \approx TV_{0,max}/2 \tag{4.60}$$

Such a relation holds, however, just in situations, when the transients show one pulse superimposed on monotonic (in the limit case step) variables.

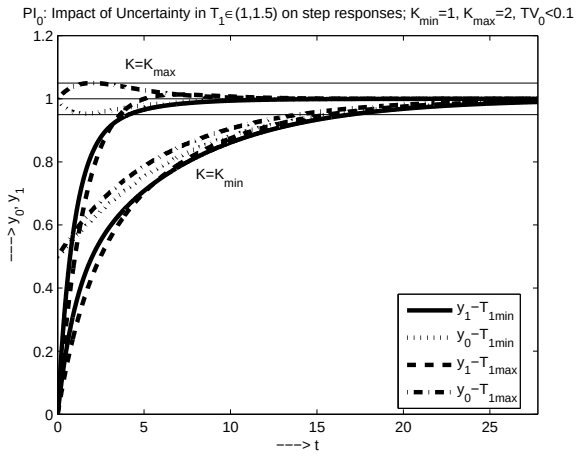


Fig. 4.17 PI_0: Setpoint step responses corresonding to limit values (4.58) and satisfying $\mathrm{TV}_0 < 0.1$ show amplitude deviations from monotonicity $\epsilon_y \approx TV_{0,max}/2$

4.2.4 PI$_0$-IM: Impact of Parameter Mismatch for Disturbance Step

Up to now, all our attention was concentrated on the setpoint response, but already there we tried to find such a controller tuning that would fulfill the performance requirements with the minimal T_f values that are expected to give the fastest possible disturbance reconstruction and compensation. Now we will focuss our attention also to the disturbance response. For an input disturbance v_i the disturbance responses are defined by transfer functions

$$\begin{aligned} F_{vi0}(s) = \frac{Y_0(s)}{V_i(s)} &= \frac{sKK_0T_f\,(1+T_1s)}{[K_0T_fT_1s^2 + (K_0T_f + KT_{10})\,s + K]} \\ F_{vi1}(s) = \frac{Y_1(s)}{V_i(s)} &= \frac{1}{(1+T_1s)}F_{vi0}(s) \end{aligned} \tag{4.61}$$

As it is obvious from these transfer functions that yield $F_{vi0}(0) = 0$ and $F_{vi1}(0) = 0$, a piecewise constant input disturbances will cause no permanent error. From $F_{vij}(0) = 0\,; j = 0, 1$ it is obvious that in steady states influence of admissible piecewise constant input disturbances disturbances is completely eliminated. For generating performance portrait it is again advantageous to introduce normalized loop parameters (4.51) and (4.53) that yield

$$\begin{aligned} F_{vi0}(p) = \frac{Y_0(p)}{V_i(p)} &= \frac{pK_0\tau_f\,(1+\tau_1p)}{[\kappa\tau_f\tau_1p^2 + (\kappa\tau_f+1)\,p+1]} \\ F_{vi1}(p) = \frac{Y_1(p)}{V_i(p)} &= \frac{1}{(1+\tau_1p)}F_{vi0}(p) \end{aligned} \tag{4.62}$$

As it is obvious from these transfer functions, the performance analysis may not be fully realized in the normalized variables and the disturbance response will also depend on the tuning parameter K_0. So, localization of UB in space of normalized parameters will be based on sweeping the parameter portrait for position corresponding to minimal IAE value that has to respect not only the time scale (4.53), but also scaling imposed by K_0.

It is to remember that NO, or MO areas of controller parameters identified by the computer based analysis for a disturbance step are different from equivalent areas corresponding to setpoint step step. When it is required to keep some property for setpoint as well as for disturbance response, the uncertainty box corresponding to possible loop values must lie in intersection of corresponding areas.

4.2.5 Influence of the Nonmodelled Dynamics

Results of the previous analysis show that controller tuning is dominantly influenced by robustness issues. This holds also in situations, when the parameter changes are relatively negligible and plant is supposed to have time invariant dynamics.

The first intuitive expectation migt be that by decreasing plant uncertainty and by canceling the dominant time constants by inverse dynamics, the remaining loop dynamics can be arbitrarily speeded up (as for fundamental solutions) by deceasing $T_f \to 0$. This is, however, not true in practice. In such situations a reliable PI_0 controller tuning would require to determine not only the dominant loop time constant T_1 but also some parameter approximating the nonmodelled dynamics. In the simplest case it is again possible to approximate the nonmodelled dynamics by a time constant T_a (accumulative delay), or by a transport delays T_d. Such loop approximations would be based on models as

$$F_{yd}(s) = \frac{Y_m(s)}{U(s)} = \frac{K\mathrm{e}^{-T_d s}}{1 + T_1 s} \tag{4.63}$$

or

$$F_{yd}(s) = \frac{Y_m(s)}{U(s)} = \frac{K}{(1 + T_1 s)\,(1 + T_a s)} \tag{4.64}$$

In the nominal case with $T_{10} = T_1$, the first estimate of appropriate T_f values for which the nonmodelled dynamics might be important could be based on Tab. 4.1 and Tab. 4.2 derived for the I_0 controller. Such approach was already mentioned by textbooks (Huba, 2003, 2006). Using approximation of the nonmodelled dynamics by dead time and $K_0 = K, T_{10} = T_1$ one gets e.g values recommended by Vítečková et al (2000), or many other results summarized by O'Dwyer (2000).

To get more detailed picture of the resulting loop dynamics for $T_{10} \neq= T_1$, the computer based analysis can be used again. For both models 4.63 and 4.64 introduction of additional parameter for the nonmodell dynamics leads to increase of the dimension of the solved problem. It means that already when using possibility for simplification by choosing $K_0 = K_{max}$ it is necessary to work in a 3D space. Therefore, it is always advantageous to check the possibility to simplify the problem e.g. by choosing $T_{10} = T_{1,max}$ and to solve the problem in 2D space of parameters (τ_d, τ_f), ; $\tau_d = T_d/T_{10}$.

4.2.6 Effect of Measurement and Quantization Noise

For a possible measurement noise δ the responses of both possible outputs are defined by

$$F_{\delta 0}(s) = \frac{Y_0(s)}{\delta(s)} = \frac{K\left(1+T_{10}s\right)\left(1+T_1 s\right)}{\left[K_0 T_f T_1 s^2 + \left(K_0 T_f + K T_{10}\right)s + K\right]}$$
$$F_{\delta 1}(s) = \frac{Y_1(s)}{\delta(s)} = \frac{1}{(1+T_1 s)} F_{\delta 0}(s) \tag{4.65}$$

In the robustness analysis in previous sections we came to conclusion that from the robustness point of view it is better to use the more complex PI_0 controller than the simpler I_0 controller. However, in tuning the fundamental PI_0-IM controller given by Fig. 3.12a it is important to remember that a measurement noise step by $\Delta\delta$ produces in the control signal kick with amplitude

$$\Delta u = \lim_{s\to\infty} s \frac{1+T_{10}s}{K_0 T_f s} \frac{\Delta\delta}{s} = \frac{T_{10}}{K_0 T_f}\Delta\delta \tag{4.66}$$

When for a given value $\Delta\delta$ one chooses the filter time constant T_f too small, noise amplification and due to this the corresponding "kick" of the manipulated variable Δu may increase over acceptable values. So, the filter time constant (or the equivalent gain of the P action (4.44)) should also consider acteptable levels of such control signal kicks defined by the maximal amplitudes of the measurement noise. Whereas for the I_0 controller the integral character of controller is guaranteeing homogenous filtration over all frequencies, for the PI_0 controller filtration properties dominate just for frequencies over the DO bandwidth $\Omega_f = 1/T_f$. So, the measurement noise and required filtration properties represent the key aspects in deciding if to use I_0 or PI_0 control.

4.2.7 Conclusions PI_0

Space available for this contribution has not enabled to go into detailed comparing of all possible structures of the PI_0 controllers, of possible DO filters and prefitlers. But, in interpreting results from the computer based analysis of robust controller tuning it is important to note several points:

1. For the setpoint step responses of output y_0 (input of the time constant) NO areas are different from MO ones. For the disturbance responses of output y_0 and for both responses of output y_1 NO and MO areas are identical.
2. NO and MO areas corresponding to output y_0 are different from those corresponding to the output y_1. It is to remember that just the tasks with output y_0 with the relative degree zero genericly fall into DC0.
3. Transients from DC0 may also be designed for the output y_1 (see Theorem 3.1), but there faster dynamics may be achieved by solutions of DC1, treated e.g in Huba (2011).
4. Setpoint step responses are more sensitive to the plant-model mismatch than the disturbance responses. This sensitivity typical for the setpoint step responses of the output y_0 may be reasonable decreased by using prefilter with $T_p = T_f$.
5. For NO and MO step responses of output y_0 it is important to work with $K_0 \geq K$ and $T_{10} \geq T_1$.
6. By limiting the admissible TV_0 values in tuning the PI_0 controller it is simultaneously possible to limit the amplitude deviations from nonovershooting and monotonicity to approximatelly $\epsilon_y \approx TV_{0,max}$

4.2.8 Performance Portrait of the FPI_0 Controller for $T_p = T_f$

After extending the PI_0 controller by prefilter with the time constant $T_p = T_f$ to the FPI_0 (4.44) it is possible to reasonably enlarge areas of NO and MO step responses without increasing number of tuned parameters. The performance portrait in Fig. 4.18 shows that with the tuning $T_{10} = T_{1,max}$ and $K_0 = K_{max}$it is possible to achieve outputs u, y_0 and y_1 with zero TV_0 values for practically arbitrary T_f values. This, however, holds just for systems with negligable nonmodelled dynamics. Therefore, in real applications this controller could be reliably tuned after approximating the nonmodelled dynamics and by increasing number of normalized parameters by using the PP generated in 3D. A simplified approach could e.g use the analyzes of the I_0 tuning to choose T_f and according to the PP in 2D to set

$$T_{10} = T_{1,max}\,;\; K_0 = K_{max} \tag{4.67}$$

4.3 Predictive $\mathbf{I_0}$ and Filtered Predictive $\mathbf{I_0}$ Controllers ($\mathbf{PrI_0}$ and $\mathbf{FPrI_0}$)

Tuning of the closed loop systems involving dead-time still represents a challenging domain of control research. Thereby, importance of dead-time systems that are being used to describe transport of mass, energy and information and to approximate accumulation of time lags in a chain of low order systems is permanently increasing, to mention just different new applications arising in the field of remote control via computer networks and telecommunication links. As it was shown in many contributions (see e.g. Normey-Rico and Camacho (2007)), an increase of the dead-time values with respect to the dominant plant time constant leads in the loops with PID controllers without active dead time compensation to rapid performance deterioration. Consider a stepwise constant reference signal $w(t)$ and an uncertain plant with dominant dead-time

$$\begin{aligned} F(s) &= K\mathrm{e}^{-T_d s} \\ K \in \langle K_{min}, K_{max}\rangle \ ; \ c_K &= K_{max}/K_{min} \geq 1 \\ T_d \in \langle T_{d,min}, T_{d,max}\rangle \ ; \ c_d &= T_{d,max}/T_{d,min} \geq 1 \end{aligned} \tag{4.68}$$

The task is to design robust controller that would guarantee step responses of the output and control variable with tolerable deviation from monotonicity defined e.g. by specifying the amplitude deviations ϵ_y, or ϵ_u, or by specifying the integral measures for deviations u-TV_0 or y-TV_0.

The plant model (4.68) may be simply identified by evaluating the average residence time (4.41) by the step responses Åström and Hägglund (1995), or by a general input signal according to Ingimundarson (2000).

In the simplest case, based on estimate of the plant gain K_0, to set output of the considered plant to the reference value w the static feedforward control $1/K_0$ maight be used. For the plant with an output disturbance v_o it would be possible to extend this static feedforward control by the Disturbance Observer (DO) inspired by the IMC (Morari and Zafiriou, 1989). For the input disturbance it may similarly be used the DO based on the inverse plant model inspired by Ohnishi et al. (1996). Thereby, in both cases, estimate of the plant dead time T_{d0} was inserted into the DO branch from the controller output.

In controlling plant (4.68) both these alternatives are equivalent, but in order to be clear in controlling more complex plants and with sake of the brevity we will propose following:

Definition 4.8 (Predictive $\mathbf{PrI_0}$ and $\mathbf{FPrI_0}$ Controllers). Under the PrI_0 controller we will understand the static feedforward control with the gain $1/K_0$ extended by the input or output disturbance reconstruction and compensation (Fig. 4.19) with the DO filter time constant T_f and by the prefilter with the time constant T_p (in the simpest case with $T_p = T_f$) giving

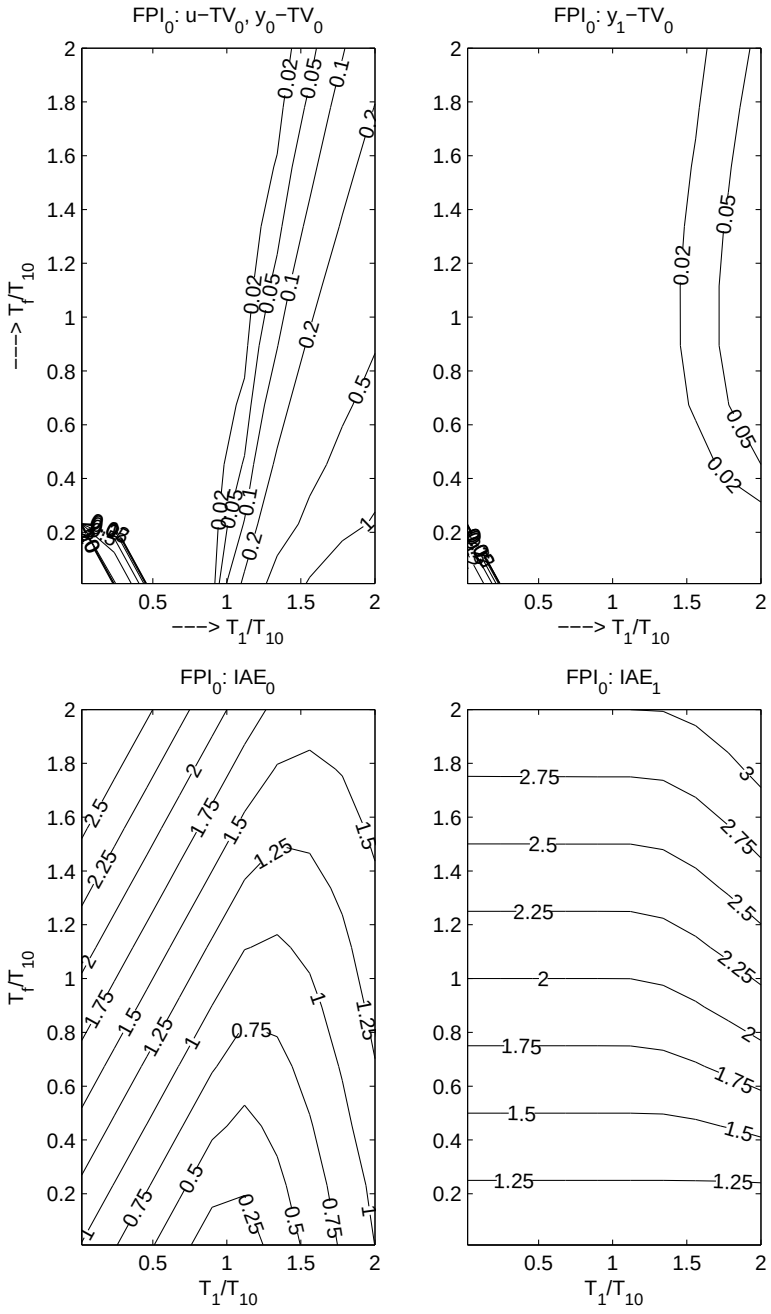


Fig. 4.18 FPI_0 with $T_p = T_f$: Performance portrait for the TV_0 values of the setpoint step responses at the outputs y_0 (controller output) and y_1 and the equivalent IAE values

the resulting control law

$$U(s) = \frac{W(s)}{K_0\,(1+T_f s)} - \left[\frac{Y(s)}{K_0\,(1+T_f s)} - \frac{\mathrm{e}^{-T_{d0}s}U(s)}{K_0\,(1+T_f s)}\right] \tag{4.69}$$

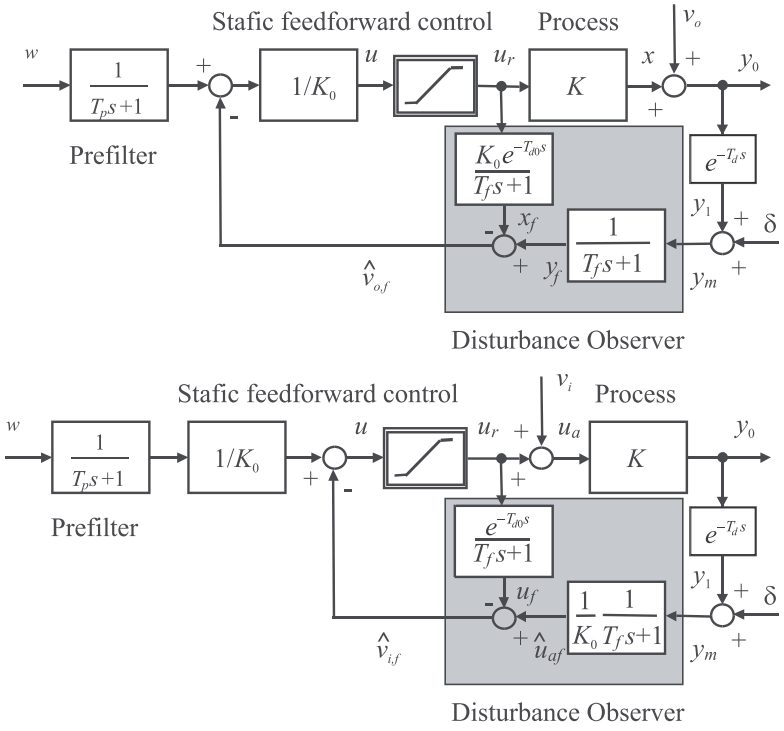


Fig. 4.19 a) Fundamental FPrI_0-PM controller (FPrI_0-IMC controller) controller designed as static feedforward control extended by input disturbance reconstruction and compensation using Parallel plant Model (above) and the FPrI_0-IM controller with Inverse Model of the invertible dynamics reduced to $1/K_0$ (below); in both cases the disturbance reconstruction was balanced by including dead time estimate into the DO channel from the controller output; both structures are extended by a prefilter (in the simpest case with $T_p = T_f$) to FPrI_0 controllers

4.3.1 Performance Portrait of the PrI_0 and $FPrI_0$ Controllers

Intuitively one could expect optimal behavior of this controller for $K_0 = K, T_{d0} = T_d$. However, how to choose K_0 and T_{d0} and what happens in the case of a parameter mismatch?

The setpoint-to-output closed loop transfer functions of the PrI_0 and FPrI_0 controllers corresponding to the output y_1 and a plant-model parameter dismatch are given as

$$\begin{aligned} F_{w1}(s) = \frac{Y_1(s)}{W(s)} &= \frac{K(1+T_f s)\,\mathrm{e}^{-T_d s}}{(1+T_p s)\left[K_0\left(T_f s+1-\mathrm{e}^{-T_{d0}s}\right)+K\mathrm{e}^{-T_d s}\right]} \\ F_{w1p}(s) &= \frac{K\mathrm{e}^{-T_d s}}{K_0\left(T_f s+1-\mathrm{e}^{-T_{d0}s}\right)+K\mathrm{e}^{-T_d s}}\,;\ T_p = T_f \end{aligned} \tag{4.70}$$

Similarly, the input disturbance-to-output y_1 closed loop transfer functions for the plant-model parameter dismatch is given as

$$F_{vi1}(s) = \frac{Y_1(s)}{V_i(s)} = \frac{KK_0\mathrm{e}^{-T_d s}\left(1+T_f s-\mathrm{e}^{-T_{d0}s}\right)}{K_0\left(T_f s+1-\mathrm{e}^{-T_{d0}s}\right)+K\mathrm{e}^{-T_d s}} \tag{4.71}$$

Obviously, $F_{w1}(0) = 1$ and $F_{vi1}(0) = 0$, what guarantees I-behaviour, i.e. rejection of piece-wise constant disturbances also for $K_0 \neq K$ and $T_{d0} \neq T_d$. These properties hold also for the output y_0

$$F_{w0}(s) = F_{w1}(s)\,\mathrm{e}^{T_d s}\,;\ F_{vi0}(s) = F_{vi1}(s)\,\mathrm{e}^{T_d s} \tag{4.72}$$

To be able to use the generated PP for any plant (4.68), the setpoint step responses will be mapped by using 3D coordinate system (κ, τ_f, τ_d) with normalized variables

$$\kappa = K_0/K\,;\ \tau_f = T_f/T_{d0}\,;\ \tau_d = T_d/T_{d0}\,;\ \tau_p = T_p/T_{d0}\,;\ p = T_{d0}s \tag{4.73}$$

that yield

$$\begin{aligned} F_{w1}(p) = \frac{Y_1(p)}{W(p)} &= \frac{(1+\tau_f p)\,\mathrm{e}^{-\tau_d p}}{(1+\tau_p s)\left[\kappa\left(\tau_f p+1-\mathrm{e}^{-p}\right)+\mathrm{e}^{-\tau_d p}\right]} \\ F_{w1p}(p) &= \frac{\mathrm{e}^{-\tau_d p}}{\kappa\left(\tau_f p+1-\mathrm{e}^{-p}\right)+\mathrm{e}^{-\tau_d p}}\,;\ \tau_p = \tau_f \\ F_{vi1}(p) = \frac{Y_1(p)}{V_i(p)} &= \frac{K_0\mathrm{e}^{-\tau_d p}\left(1+\tau_f p-\mathrm{e}^{-p}\right)}{\kappa\left(\tau_f p+1-\mathrm{e}^{-p}\right)+\mathrm{e}^{-\tau_d p}} \end{aligned} \tag{4.74}$$

The transfer functions corresponding to the output y_0 may similarly be derived by means of

$$F_{w0}(p) = F_{w1}(p)\,\mathrm{e}^{\tau_d p};\ F_{vi0}(p) = F_{vi1}(p)\,\mathrm{e}^{\tau_d p} \tag{4.75}$$

Examples of one layer of the 3D PP of the PI_0 and FPI_0 in Fig. 4.20 that by introducing the prefilter the ϵ_y-areas reasonably enlarger and that e.g. the 10^{-5}-MO area is close to the area with $\mathrm{TV}_0 = 10^{-4}$, what again points out possibility to work with the numerically simpler measures for the integral deviations from strictly monotonic control at the plant input and output.

By being based on the 2-parameter plant model the PrI controllers represent alternatives to the PI controllers. For systems with the dominant time delays they enable substantial quality improvement. First version of PrI controller was proposed by Reswick (1956). Yet before the well known Smith Predictor (SP) he proposed active compensation of the whole identified dead time ($T_{d0} = T_d$ and $K_0 = K$) corresponding $T_f = 0$. This caused, however, enormous sensitivity to parameters uncertainty, because for $T_f \to 0$ the monotonicity areas shrink (Fig. 4.21). Because of lacking method for a reliable controller tuning, it was practically forgotten and newer works (Åström & Hägglund, 1995; 2005, Guzman et al., 2008; Normey-Rico et al., 2009) mention just the Smith Predictor. The PP based analysis enables to explain the high sensitivity of the Reswick's solution and importance of the choice of Tf and gives also possibility of robust tuning of these simplest possible predictive controllers.

4.3.2 Robust Tuning of the PrI_0 and FPrI_0 Controllers

In a subplane (κ, τ_d) with a given τ_f the robust design corresponding to plant (4.68) means to locate Uncertainty Box of all possible operating points

$$UB = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} \kappa_{min}, \tau_{d,max} & \kappa_{max}, \tau_{d,max} \\ \kappa_{min}, \tau_{d,min} & \kappa_{max}, \tau_{d,min} \end{bmatrix} \tag{4.76}$$

with vertices corresponding to combinations of the limit values of κ and τ_d by specifying K_0, T_{d0} and $T_f$in such a manner that will guarantee the fastest possible transients (with minimal average IAE value). Examples of sweeping parameter area corresponding e.g. to ϵ_y-MO output y_1, $\epsilon_y = 0.02$ and looking for appropriate UB lying completely in it are in Fig. 4.22.

Due to the relatively rough quantization, the achieved overshooting (Fig. XXX3 below) is not absolutely close to the tolerable value. It is to note that the found "optimal" tuning of this controller is very close to the expected value of the gain $K_0 = K_{max}$) that may reasonably simplify the tuning process by reducing the task to 2D space of parameters (τ_d, τ_f).

In the case with single uncertain parameter the task reduces to finding optimal position of a horizontal (uncertain gain K), or vertical (uncertain T_d) uncertainty line segment (ULS).

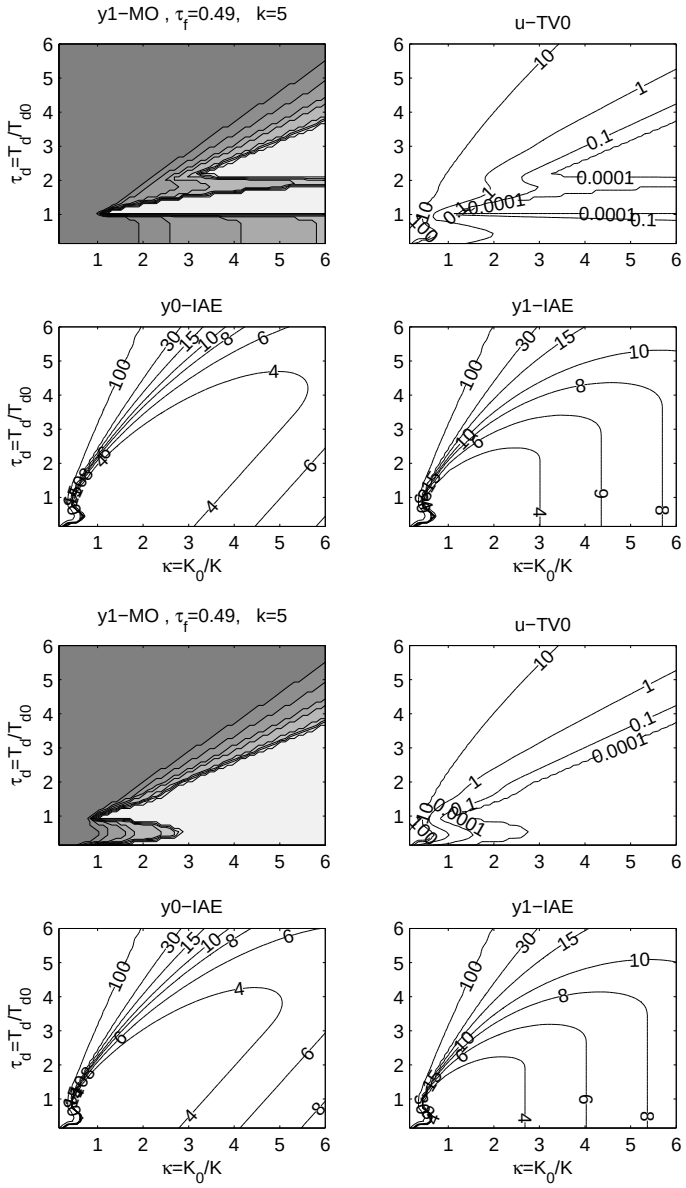


Fig. 4.20 One layer of the PP of the plant (4.68) with the PrI_0 (above) and FPrI_0 controller (below) corresponding to $\tau_f = 0.49$ generated over 61x61x11 points and showing ϵ_y-MO areas (left above) for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$, white denoting the best performance, the u-TV_0 contours (right above) and the IAE0 and IAE1 levels (below) in the plane (κ, τ_d)

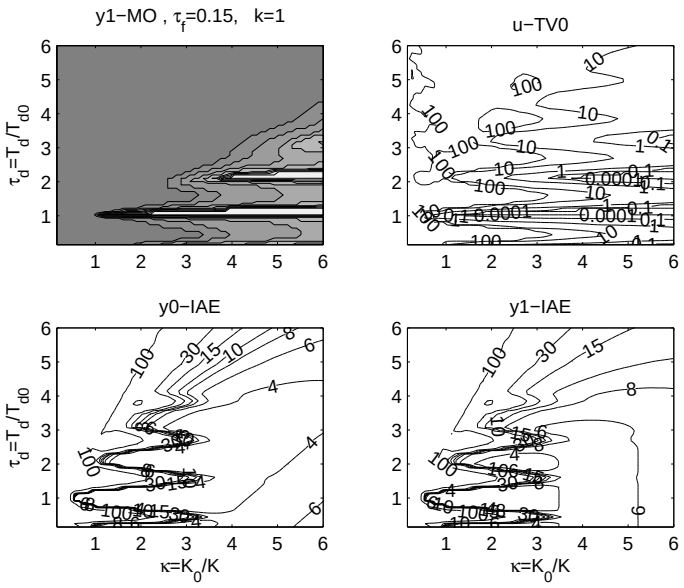


Fig. 4.21 Layer of the PP of the plant (4.68) with the FPrI_0 controller corresponding to $\tau_f = 0.15$ generated over 61x61x11 points and showing ϵ_y-MO areas (left above) for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$, white denoting the best performance, the u-TV_0 contours (right above) and the IAE0 and IAE1 levels (below) in the plane (κ, τ_d)

By using uncertainty information represented by (4.68) it is necessary to sweep over all possible values of T_f for UB (4.76) or ULS defined by ratios of extreme values of uncertain parameters c_K, or c_d and lying in the required performance area. During this step, from identified values of particular UB (4.68) one has to recalculate the task from fixed controller tuning $K_0, T_{d,0}$ and variable plant parameters K, T_d to fixed limit loop values (4.68) and variable controller tuning T_f, K_0 and $T_{d,0}$ corresponding to the optimal position of UB according to

$$K_0 = \kappa_{min}^{opt} K_{max}\,;\; T_{d,0} = T_{d,min}/\tau_{min}^{opt}\,;\; IAE_{mean} = T_{d,0} IAE_{mean}^{opt} \quad (4.77)$$

By increasing the tolerable deviation from monotonicity to $\epsilon_y = 0.05$ (Fig. 4.23), the transient run faster, but simultaneously the additional control effort expressed by increased u-TV_0 value occurs.

In both analyzed cases, due to the relatively rough quantization, the achieved overshooting of the step responses is not absolutely close to the tolerable value. A direct increase of points in one dimension in generating the the performance portrait leads in 3D PP to cubic increase of the total number of points. But, from the shapes of ϵ_y-MO areas of the PP it is evident

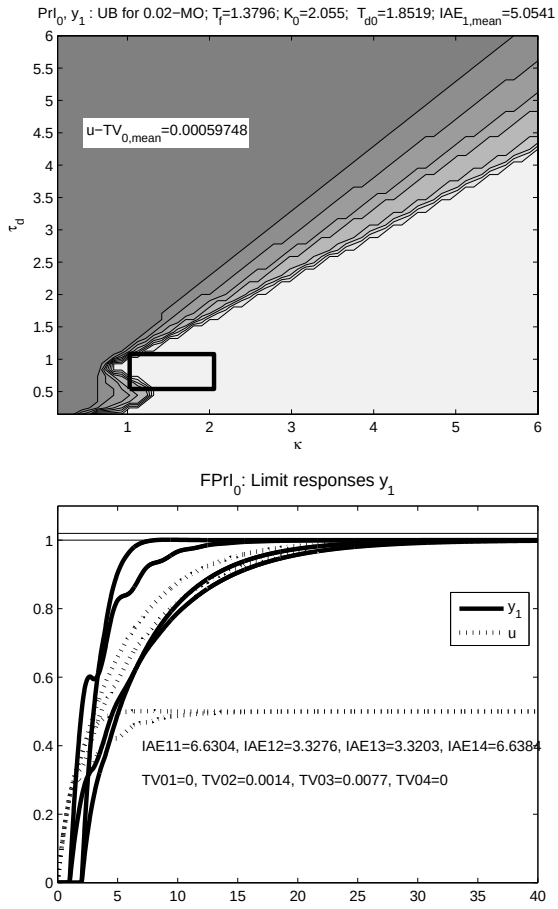


Fig. 4.22 Result for seeping an optimal UB of the plant (4.68) with $K_{min} = 1, K_{max} = 2, T_{d,min} = 1, T_{d,max} = 2$ over PP of 61x61x11 points, areas of ϵ_y-MO output step responses identified for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$, white denoting the best performance, with identified parameters $T_f = 1.3796, K_0 = 2.005$ and $T_{d,0} = 1.8519$ (above) and the corresponding transients (below)

that it is allways possible to set $K_0 = K_{max}$ and reduce the whole design procedure to 2D space (τ_d, τ_f).

Example 4.3 (PrI_0 Controler for Plant from Example 4.1).

This illustrative example compares robust design of the PrI_0 controller with the FSP (Normey-Rico and Camacho (2007); Example 6.1) with the robust design of I controller in Example 4.1. The uncertain plant to be controlled is (4.38). Its Performance Portraits achieved by the FSP and I controller are given by Fig. 4.10 and Fig. 4.11. For tuning the FPrI_0 controller,

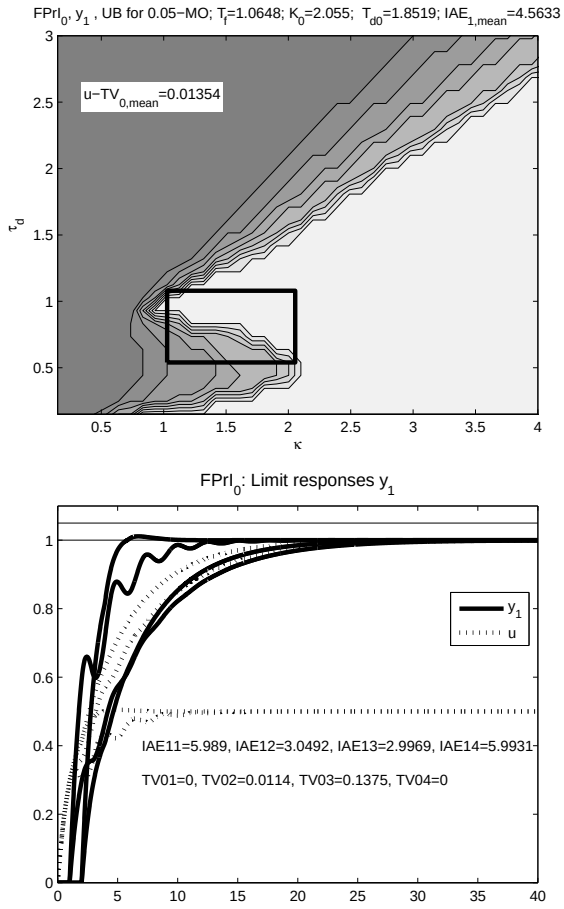


Fig. 4.23 Result for seeping an optimal UB of the plant (4.68) with $K_{min} = 1, K_{max} = 2, T_{d,min} = 1, T_{d,max} = 2$ over PP of 61x61x11 points with identified parameters $T_f = 1.0648, K_0 = 2.005$ and $T_{d,0} = 1.8519$; areas of ϵ_y-MO output step responses identified for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$, white denoting the best performance (above) and the corresponding transients (below)

similarly as in (4.42) the equivalent dead time will be determined by using information about the sum of the plant time constants as

$$T_d = L + S\,;\ S = \sum T_i = 1.875\,;\ T_d \in \langle 10.875, 13.875 \rangle \tag{4.78}$$

According to this, the optimal UB (Fig.) was specified in 3D PP of the plant (4.68) by tuning parameters

$$T_f = 4.7359\,;\ K_0 = 1.584\,;\ T_{d0} = 11.6935 \tag{4.79}$$

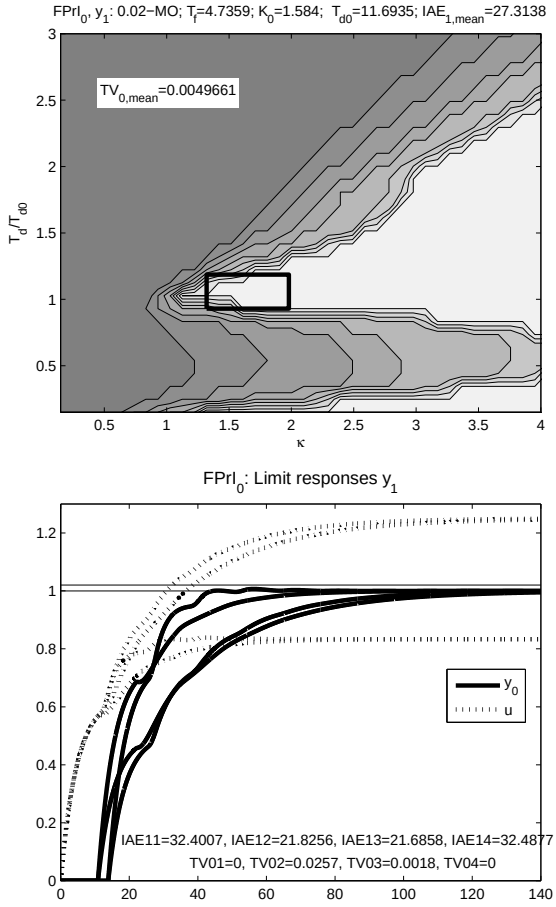


Fig. 4.24 Result for sweeping IAE optimal UB for y_1 and 0.02–MO of the plant (4.38) with $K_{p,min} = 0.8, K_{p,max} = 1.2, L_{min} = 9, L_{max} = 12$ approximated by T_d (4.78) over PP of 61x61x11 points with identified parameters $T_f = 4.7359, K_0 = 1.584$ and $T_{d,0} = 1.8519$; areas of ϵ_y-MO output step responses of y_1 identified for tollerances $\epsilon_y = \{0.1, 0.05, 0.02, 0.01, 0.001, 0.0001, 0.00001\}$, white denoting the best performance (above) and transients corresponding to the limit uncertain parameter values (below)

Due to the relatively rough quantization, the calculated gain K_0 is rather overestimated with respect to $K_{max} = 1.2$ and so the achieved overshooting of the step responses is not absolutely close to the tolerable value. Despite to this the achieved results are comparable with those achieved with retuned Filtered Smith Predictor based on the first order plus dead time model (Huba, 2011). Also now, explanation for this surprising result, when the extremely simple model gives excellent results, may be taken from the same source as this example (Normey-Rico and Camacho (2007), pp. 174): “when the dead-time is dominant, the contribution of the open loop poles to the closed loop

response will be small thus their elimination will contribute with a small increment in the speed of the transients". PrI_0 and $FPrI_0$ fully respect this fact and will surelly find top position in many industrial applications.

4.4 Summary

1. Known input disturbances may be compensated by opposite signal at the controller output. Output disturbances may be compensated by opposite signal correcting the reference setpoint value of the controller.
2. By extending the static feedforward control of a memoryless plant by disturbance observer (DO) for reconstruction of disturbances one gets the generic structure of the I_0 controller. Different stable low-pass filter can be chosen with respect to the measurement noise filtration and loop robustness. Continuity of the setpoint step response may be achieved by using prefilter for the setpoint variable.
3. In loops with strictly memoryless plant represents the I_0 controller a fundamental solution – the DO filter time constant may be arbitrarily small (the gain of the equivalent I_0 controller infinitely large) and the corresponding transient responses infinitely fast.
4. In tuning real loops with memoryless plant it is important to estimate the every time present nonmodelled loop dynamics. This can be approximated by dead time, by time constant, or by more complex dynamics. Controller parameters corresponding to the fastest non-overshooting and monotonic control may be well approximated by analyzing conditions of double real dominant close doop pole (DRDP). Approximations by dead time usually lead to faster monotonic transients than approximations by time constant.
5. Tuning of the I_0 controller gain is equivalent to simultaneous tuning of the DO filter time constant T_f used in disturbance reconstruction and tuning of the reciprocal gain of the feedforward control. For achieving setpoint step responses with defined overshooting, maximal dead time values and minimal plant gains have to be identified.
6. Tuning of the I_0 controller brings several degrees of freedom. One can decide about dynamics of the control signal corresponding to a setpoint step that may either have stepwise character (achieved by using controller according to Fig. 4.1) or softer exponential one (given by controller in Fig. 4.12 with prefilter time constant $T_p = T_f$, or by controller in Fig. 4.13 without prefilter). The nonmodelled loop dynamics may be approximated by a dead time, by a time constant, or by more complex transfer function. The loop dynamics may be approximated by providing a step response experiment, by measurement on stability border, by relay experiment, etc.

7. In control loops with a memoryless plant and one (stable) dominant time constant it is possible to cancel its effect in DO by filtered inverse of this dominant loop dynamics that gives structure of the PI_0 controller. For a neglected nonmodelled dynamics it represents a fundamental solution – ideally, the DO filter time constant may be arbitrarily small (the equivalent I_0 controller gain may be infinitely large) and the corresponding transient responses may be infinitely fast. Different stable low-pass filters can be chosen with respect to the measurement noise filtration and loop robustness.
8. In nominal case, a reliable controller tuning has to respect the nonmodelled loop dynamics that remains after cancelling the dominant time constant. The dominant time plant constant effect on the reconstruction dynamics may also be balanced by adding its estimate into the branch leading from the controller output. In this way one gets IMC like structure of the PI_0 controller that has no more properties of fundamental solutions: its dynamics cannot be arbitrarily speeded up, just to a limit value given by the dominant loop time constant. This solution may, however, be interesting by low noise sensitivity and robustness against parameters uncertaintny.
9. Active compensation of the loop (plant) time constants by the inverse terms in the DO based PI_0 controllers may lead to increased sensitivity to the measurement noise. But, the loop sensitivity to parameters uncertainty may be decreased. Higher order models usually also give lower effect of the nonmodelled dynamics. Ideal controllers (corresponding to models with neglected nonmodelld dynamics) represent fundamental solutions enabaling to shift closed loop pole (observer pole) theoretically to minus infinity and so to speed up transients to stepwise changes of control signal and output variable. However, in all real loops it is necessary to limit admissible closed loop poles (filter poles) to values giving acceptable noise amplification, robustness to model uncertainty and nonmodelled dynamics.
10. For the closed loop with monotonic nonoverhooting control signal transients and for admissible inputs (reference signals and acting disturbances) the control saturation will never be activated and so it can be omitted from considered control structures. This enables to describe all problems considered within the dynamical class 0 (DC0) by linear control theory. Therefore, in dealing with linear PID control structures we will consider their use within the DC0, even in situations when for the sake of simplicity the index "0" was omitted.
11. Active compensation of dead time by inversion is not possible. In this case, the dead time introducing time shift of the measured output may be compensated by including estimate of dead time into the observer branch leading from the controller output. The disturbance will be reconstructed by the time delay, but its values will be not distorted by different time shifts of both DO branches. In this way it is possible to construct pre-

dictive I_0 controller (PrI_0) and its filtered version $FPrI_0$ equipped by a prefilter.
12. Introduction of DO based I action designed as reconstruction and compensation of input, or output disturbances plays a key role in designing constrained integrating controllers for higher dynamical classes of control that do not exhibit integrator windup.

4.5 Questions and Exercises

1. Which controller is more sensitive to the measurement noise: the PI_0, or the PrI_0 one?
2. How could you define PID_0 controller for active compensation of two time constants?
3. Could you formulate alternative solutions to this problem?
4. Which criteria must fulfill proposed controllers to be considered as the fundamental ones? Do all solution proposed by you fulfill these requirements?
5. What does characterize index "0" of the dynamical class DC0?
6. How could you define $PrPI_0$ controller for active compensation of one time constant and of long dead time?
7. Could you formulate alternative solutions to this problem?
8. Which criteria must fulfill propoes controllers to be considered as the fundamental ones? Do all solution proposed by you fulfill these requirements?

Acknowledgements The author is pleased to acknowledge the financial support 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 (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.

Glattfelder, A. H. und Schaufelberger, W. (2003) Control Systems with Input and Output Constraints. Springer, London.

Huba, M. (2003) Syntéza systémov s obmedzeniami I. Základné regulátory. II. Základné štruktúry. (Constrained systems design. I. Basic controllers. II. Basic structures.) Vydavatel'stvo STU Bratislava.

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. (2011) Filtered Smith Predictor Tuning by the Performance Portrait Method. *Preprints of the Workshop "Selected topics on constrained and nonlinear control"*, Huba, M., Skogestad, S., Fikar, M., Hovd, M., Johansen, T. A., Rohal'-Ilkiv, B., Editors, STU Bratislava-NTNU Trondheim, January 2011.

Ingimundarson, A. (2000). Robust tuning procedures of deadtime compensating controllers. Department of Automatic Control. Lund Institute of Technology.

Morari, M. and E. Zafiriou (1989) *Robust Process Control*. Prentice Hall, Englewood Cliffs, N. Jersey.

Neimark, Ju.,I. (1973) 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., 95-131.

Normey-Rico, J. E. and Camacho, E. F. (2007) Control of dead-time processes. Springer.

O'Dwyer, A. (2000) A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules. Preprints IFAC Workshop on Digital Control: Past, present and future of PID Control PID'2000, Terassa, Spain 175-180.

Ohnishi, K., Shibata, M. and Murakami, T. (1996) Motion Control for Advanced Mechatronics. IEEE/ASME Trans. on Mechatronics, Vol. 1, 1, 56-67.

Oldenbourg, R. C. and H. Sartorius (1944, 1951) Dynamik selbsttätiger Regelungen. 2.Auflage, R. Oldenbourg-Verlag, München.

Rugh, W. J and Shamma, J. S. (2000) Research on gain scheduling. Automatica 36, 1401-1425.

Skogestad, S. (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13, 291–309.

Skogestad, S. and I. Postlethwaite (1996) *Multivariable Feedback Control Analysis and Design*, John Wiley, N. York.

Vítečková, M., A. Víteček, L. Smutný (2000) Simple PI and PID controllers tuning for monotone self-regulating plants. *IFAC Workhop on Digital Control, Past, present and future of PID Control,* Terrassa, Spain, 283-288.

Vrančič, D., Strmčnik, S., Kocijan, J. (2004). Improving disturbance rejection of PI controllers by means of the magnitude optimum method. ISA Trans. 43, 73-84.

Chapter 5
Introduction to Nonlinear Model Predictive Control and Moving Horizon Estimation

Tor A. Johansen

Abstract Nonlinear model predictive control and moving horizon estimation are related methods since both are based on the concept of solving an optimization problem that involves a finite time horizon and a dynamic mathematical model. This chapter provides an introduction to these methods, with emphasis on how to formulate the optimization problem. Both theoretical and practical aspects are treated, ranging from theoretical concepts such as stability, existence and uniqueness of the solution, to practical challenges related to numerical optimization methods and computational complexity.

5.1 Introduction

The purpose of this chapter is to give an introduction to two of the most powerful tools that can be used to address nonlinear control and estimation problems - nonlinear model predictive control (NMPC) and nonlinear moving horizon estimation (NMHE). They are treated together since they are almost identical in approach and implementation - even though they solve two different and complementary problems.

The text is intended for advanced master and doctoral level students that have a solid background in linear and nonlinear control theory, and with a background in linear MPC, numerical methods for optimization and simulation, and state estimation using observers and the Extended Kalman Filter. Other excellent surveys to the topic and introductory texts can be found in Allgöwer et al (1999); Findeisen et al (2003b); Mayne et al (2000); Morari and Lee (1999); Rawlings (2000).

Tor A. Johansen
Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: tor.arne.johansen@itk.ntnu.no

5.1.1 Motivation and Main Ideas

5.1.1.1 Nonlinear Control

Consider the problem of controlling a multi-variable nonlinear system, subject to physical and operational constraints on the input and state. Well known systematic nonlinear control methods such as feedback linearization (Isidori (1989); Marino and Tomei (1995); Nijmeijer and van der Schaft (1990)) and constructive Lyapunov-based methods (Krstic et al (1995); Sepulchre et al (1997)) lead to very elegant solutions, but they depend on complicated design procedures that does not scale well to large systems and they are not developed in order to handle constraints in a systematic manner. The concept of optimal control, and in particular its practical implementation in terms of Nonlinear Model Predictive Control (NMPC) is an attractive alternative since the complexity of the control design and specification increases moderately with the size and complexity of the system. In particular for systems that can be adequately modeled with linear models, MPC has become the de-facto standard advanced control method in the process industries (Qin and Badgwell (1996)). This is due to its ability to handle large scale multivariable processes with tens or hundreds of inputs and states that must fulfill physical and operational constraints.

MPC involves the formulation and solution of a numerical optimization problem corresponding to a finite-horizon optimal control problem at each sampling instant. Since the state of the system is updated during each sampling period, a new optimization problem must be solved at each sampling interval. This is know as the receding horizon approach. With linear models the MPC problem is typically a quadratic or linear program, which is known to be convex and for which there exists a variety of numerical methods and software. While the numerical complexity of linear MPC may be a reasonable challenge with powerful computers being available, there is no doubt that NMPC is limited in its industrial impact due to the challenges of guaranteeing a global (or at least sufficiently good) solution to the resulting nonlinear optimization problem within the real-time requirements (Qin and Badgwell (2000)). Other limiting factors are the challenges of developing nonlinear dynamic models and state estimators. The nonlinear programming problem may have multiple local minima and will demand a much larger number of computations at each sample, even without providing any hard guarantees on the solution. Hence, NMPC is currently not a panacea that can be plugged in to solved any control problem. However, it is a powerful approach of great promise that has proven itself in several applications, Qin and Badgwell (2000); Foss and Schei (2007), and with further research in the direction of numerical implementation technology and modeling and state estimation methods, it may strengthen its position as the most powerful method available for certain classes of systems.

5.1.1.2 Nonlinear Estimation

Consider the state estimation problem of nonlinear systems. 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 nonlinear Moving Horizon State Estimator (NMHE) 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 known constraints on the state and uncertainty, and a state estimate and error covariance matrix estimate to estimate the arrival-cost at the beginning of the data window, see Rao et al (2003); Moraal and Grizzle (1995b); Alessandri et al (1999, 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 NMHE 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)) and are competitive alternatives to NMHE. Finally, we remark that nonlinear observers based on constructive Lyapunov design methods Krstic et al (1995); Sepulchre et al (1997) and nonlinear system theory (Marino and Tomei (1995); Isidori (1989)) are developed for certain classes of nonlinear systems and leads to very elegant and computationally efficient solutions, but are not easy to develop for large classes of high order multi-variable systems.

5.1.2 Historical Literature Review

Originally, the MPC and MHE methods were developed fairly independently. More recently, with the development of algorithms for constrained NMPC and

NMHE their developments have converged and the methods are more often presented as duals of each other and with similar notation and terminology. One reason is the fundamental duality between estimation and control, Goodwin et al (2005), but equally important may be their dependence on nonlinear numerical optimization and similarities in the formulation of the optimization problems that leads to synergies when implementing practical solutions.

5.1.2.1 Nonlinear Model Predictive Control

The nonlinear optimal control theory was developed in the 1950's and 1960's, resulting in powerful characterizations such as the maximum principle, Athans and Falb (1966) and dynamic programming, Bellman (1957). In the direct numerical optimal control literature, Hicks and Ray (1971); Deuflhard (1974); Biegler (1984); Bock and Plitt (1984); Betts (2001); Gill et al (1997); Bock et al (1999); von Stryk (1993), numerical methods to compute open loop control trajectories were central research topics. Problem formulations that included constraints on control and state variables were treated using numerical optimization.

NMPC involves the repetitive solution of an optimal control problem at each sampling instant in a receding horizon fashion. Unfortunately, there is no guarantee that the receding horizon implementation of a sequences of open loop optimal control solutions will perform well, or even be stable, when considering the closed loop system. This challenge, in combination with the tremendous success of *linear* MPC in the process industries, Qin and Badgwell (1996), lead to an increasing academic interest in NMPC research with focus on stability analysis and design modifications that guarantee stability and robustness. The early results Chen and Shaw (1982); Keerthi and Gilbert (1988); Mayne and Michalska (1990) boosted a large series of research, including Michalska and Mayne (1993); Alamir and Bornard (1995); Chen and Allgöwer (1998); Nicolao et al (2000); Scokaert et al (1999); Magni et al (2001a,b); Jadbabaie et al (2001); Mayne et al (2000). Industrial applications of NMPC have been reported, and are surveyed in Qin and Badgwell (2000); Foss and Schei (2007).

One of the early contributions of NMPC are given in Li and Biegler (1989), that uses linearization procedures and Gauss-Newton methods to provide a numerical procedure for NMPC based on SQP that makes only one Newton-iteration at each sampling instant. Theoretical results are also given in Li and Biegler (1990). The continuation/GMRES method of Ohtsuka (2004) is based on a similar philosophy of only one Newton-iteration per sample, while it is based on interior point methods. Recent NMPC research along similar ideas has benefited considerably from progress in numerical optimization, being able to take advantage of structural properties on the NMPC problem and general efficiency improvements, e.g. Biegler (2000); Diehl et al (2009); Tenny

et al (2004); Zavala and Biegler (2009), in addition to important issues such as robustness Magni et al (2003); Magni and Scattolini (2007); Limon et al (2006).

In parallel with the development of NMPC, researchers have developed so-called Real-Time Optimization (RTO) approaches, Sequeira et al (2002); Xiong and Jutan (2003). They are conceptually similar to NMPC, as they are generally based on nonlinear models (usually first principles models) and nonlinear programming. Their conceptual difference is that RTO uses static nonlinear models, while NMPC uses dynamic nonlinear models.

5.1.2.2 Nonlinear Moving Horizon Estimation

Generalizing ideas from linear filtering, Jazwinski (1968), early formulations of NMHE were developed in Jang et al (1986); Ramamurthi et al (1993); Kim et al (1991); Tjoa and Biegler (1991); Glad (1983); Zimmer (1994); Michalska and Mayne (1995). 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, Moraal and Grizzle (1995b).

Such discrete-time observers are formulated in the context of numerical nonlinear optimization and analyzed with respect to convergence in Rao et al (2003); Alessandri et al (1999, 2008); Raff et al (2005); Alamir (1999). In recent contributions, Biyik and Arcak (2006) provides results on how to use a continuous time model in the discrete time design, while issues related to parameterization are highlighted in Alamir (2007) computational efficiency are central targets of Zavala et al (2008); Alamir (2007); Alessandri et al (2008).

Uniform observability is a key assumption in most formulations and analysis of NMHE. For many practical problems, like combined state and parameter estimation problems, uniform observability is often not fulfilled and modifications are needed to achieve robustness, Moraal and Grizzle (1995a); Sui and Johansen (2010).

5.1.3 Notation

Norms: For a vector $x \in \mathbb{R}^n$, let $||x|| = ||x||_2 = \sqrt{x^T x}$ denote the Euclidean norm, and $||x||_1 = |x_1| + ... + |x_N|$ and $||x||_\infty = \max_i |x_i|$. The weighted norms are for a given symmetric matrix $Q \succ 0$ given as $||x||_Q = \sqrt{x^T Q x}$ and we use the same notation also when $Q \succeq 0$. Vectors $x_1, x_2, ..., x_N$ are stacked into one large vector x by the notation $x = \text{col}(x_1, x_2, ..., x_N)$.

For a continuous signal $x(t)$, where t denotes continuous time, we let $x[t_0, t_1]$ denote the trajectory between $t_0 \leq t \leq t_1$.

5.1.4 Organization

This chapter is organized in three main sections. In section 5.2 the formulation of NMPC optimization problems is described, focusing on the consequences of the various choices and challenges an engineer will face when designing and tuning an NMPC. Likewise, section 5.3 considers the formulation of NMHE optimization problems. The more detailed aspects of implementation in terms of numerical computations and solving the optimization problem, are treated on a general level common for both NMPC and NMHE, in section 5.4.

5.2 NMPC Optimization Problem Formulation

This section will focus on the *formulation* of the NMPC problem, while the detailed issues related to its *numerical solution* are postponed until section 5.4. It is, however, important to have in mind that these two issues are closely linked. While the NMPC problem formulation is driven by the specification of the control objective, constraints and dynamic model formulations, one should also consider potential numerical challenges at this point. In particular, important characteristics of the tradeoff between numerical accuracy and computational complexity are determined already at the point when the NMPC optimization problem is formulation through discretization, choice of parameterizations, and choice of decision variables and constraint formulations in the optimization problem. Some of these relationships are treated also in this section, together with fundamental properties of the optimization problem, including stability, convexity and the link between controllability and well-posedness of the optimization problem.

5.2.1 Continuous-time Model, Discretization and Finite Parameterization

This section will introduce a basic nonlinear optimal control formulation starting from a continuous time model and a finite horizon where the objective is to minimize a cost function

$$J(u[0,T],x[0,T]) \triangleq \int_0^T \ell(x(t),u(t),t)dt + S(x(T),T) \tag{5.1}$$

$$\tag{5.2}$$

subject to the inequality constraints for all $t \in [0,T]$

$$u_{min} \leq u(t) \leq u_{max} \tag{5.3}$$

$$g(x(t),u(t),t) \leq 0 \tag{5.4}$$

and the evolution of the ordinary differential equation (ODE) given by

$$\frac{d}{dt}x(t) = f(x(t),u(t),t) \tag{5.5}$$

with given initial condition $x(0) \in \mathbb{R}^n$. The function ℓ is know as the stage cost, S is the terminal cost, $T > 0$ is the horizon, and together these define the cost function J. The evolution of the state $x(t)$ is given by the function f according to (5.5) and depends on the input signal $u(t) \in \mathbb{R}^m$ and time t, and forms an infinite-dimensional equality constraint to the optimal solution in the formulation above. In addition there is saturation on the input with minimum and maximum thresholds u_{min} and u_{max}, respectively, and general inequality constraints jointly on states and inputs, point-wise in time $t \in [0,T]$, defined by the function g. These constraints may result from both physical and operational constraints of the control system and stability-preserving terminal sets that will be discussed later in section 5.2.3, see also Mayne et al (2000).The properties of ℓ and S have consequences for the control performance, including stability, and must be carefully understood and tuned, Mayne et al (2000). We will return to this important issue in section 5.2.3. The explicit time-dependence in f, g, ℓ allows for time-varying reference trajectories, known disturbances and exogenous input signals to be accounted for in the optimal control problem formulation. Throughout this chapter we implicitly assume all the functions involved satisfy the necessary regularity assumptions, such as continuity and smoothness.

The above formulation basically defines an infinite-dimensional optimal control problem whose solution can be characterized using classical tools like calculus of variations, Pontryagin's maximum principle (Athans and Falb (1966)) and dynamic programming, Bellman (1957). In these *indirect methods* such characterizations of the solution can help us only in a very limited number of special cases to find an analytic exact representation of the solution. The most interesting and well known is the unconstrained linear quadratic regulator (LQR) where the feedback solution is a linear state feedback $u = Kx$ under additional assumptions on T and S that makes the cost function equivalent to an infinite horizon cost Athans and Falb (1966). More recently, explicit piecewise linear state feedback representation of the solution can be made for the linearly constrained LQR problem (Bemporad et al (2002)) and more generally for linearly constrained discrete-time piecewise

linear systems, Bemporad et al (2000), although the complexity of the exact representation may be prohibitive for anything but small scale systems.

Although numerical solutions can be found based on the characterizations of the indirect methods, In the context of NMPC we choose to restrict our attention to so-called *direct methods* that seems most promising and popular. They are characterized by discretization and finite parameterization being introduced in the optimal control problem formulation which is then directly solved with numerical methods. The principle of NMPC is to repeatedly solve finite-horizon optimal control problems of the above kind at each sampling instant. This means that the initial state $x(0)$ to (5.5) is viewed as the current state based on the most recent measurements, and the optimal control trajectory $u[0,T]$ solving the above problem is implemented for a short period of time (usually one sampling interval, typically much smaller than T) until the procedure is repeated and an updated optimal control trajectory is available. However, the solution of the above optimal control problem, requires reformulations for the following reasons

- The solution to the ordinary differential equation (5.5) with given initial conditions must generally be based on discretized to be handled by numerical integration since exact closed-form solutions of the ODE are usually not possible to formulate in the general nonlinear case. Viewed in a different way, the infinite number of equality constraints (5.5) must be represented by a finite approximation.
- The infinite-dimensional unknown solution $u[0,T]$ should be replaced by a finite number of decision variables to be able to define a finite-dimensional optimization problem that can be solved using numerical optimization.
- Measurements are typically sampled data available only at the sampling instants, such that an updated initial state $x(0)$ will normally be available only at defined sampling instants.
- Arbitrary control trajectories cannot be implemented since typically the control command can only be changed at defined sampling instants and is typically assumed to be constant (or some other simple sample-and-hold function such as linear) between the sampling instants.

In order to reformulate the problem into a finite-dimensional and practical setting, we will make the following assumptions that will allow the integral and differentiation operators to be approximated by numerical integration methods.

- The horizon T is finite and given.
- The input signal $u[0,T]$ is assumed to be piecewise constant with a regular sampling interval t_s such that T is an integer multiple of t_s, and parameterized by a vector $U \in \mathbb{R}^p$ such that $u(t) = \mu(t,U) \in \mathbb{R}^r$ is piecewise continuous.
- An (approximate) solution to (5.5) is assumed to be defined in the form $x(t) = \phi(t,U,x(0))$ at N discrete time instants $T_d = \{t_1, t_2, ..., t_N\} \subset [0,T]$ for some ODE solution function $\phi(\cdot)$. The discrete set of time instants T_d

results from discretization of the ODEs and its time instants may not be equidistant. A simulation of the ODEs embedded in the function $\phi(\cdot)$ may incorporate additional intermediate time-steps not included in T_d, since the purpose of T_d is primarily to discretize the inequality constraints (5.3)-(5.4) at a finite number of representative points in time and to approximate the integral in (5.1) with a finite sum. In general, the time instants T_d need not coincide with sampling instants.

The assumption of given horizon T is typical for many NMPC problems, but there are important exceptions such as minimum-time formulations in e.g. robotics, Shin and McKay (1985), batch process control (Foss et al (1995); Nagy and Braatz (2003); Nagy et al (2007)), and other problems such as diving compression (Feng et al (2009)), where the horizon T may be considered a free variable. The resulting modifications to the problem formulations may lead to additional challenges related to the time discretization and may make the optimization problem more challenging.

The basis for the NMPC is the nominal model (5.5), and we remark that model uncertainty, unknown disturbances and measurement errors are not accounted for in this formulation of the NMPC problem. Various extensions and variations that can relax many of the assumptions above can be made relatively easy as straightforward modifications to the basic problem formulation. Obviously, the ODEs (5.5) can result from the spatial discretization of a partial differential equation (PDE), and the problem formulation can be augmented with nonlinear algebraic constraints in a straightforward way to account for a differential-algebraic model (DAE) model formulation (Cervantes and Biegler (1998); Diehl et al (2002)). For simplicity of presentation, we stick to the formulation above and return to some alternatives and opportunities that will be discussed in later sections.

The parameterization of the input signal $\mu(t, U)$ on the horizon $t \in [0, T]$ is important and will influence both the control performance and computational performance. In general, it should satisfy the following objectives

- Be sufficiently flexible in order to allow for a solution of the reformulated optimal control problem close to the solution original problem (5.1)-(5.5).
- Be parsimonos in the sense that it does not contain unnecessary parameters that will lead to unnecessary computational complexity and numerical sensitivity.
- Be implementable within the capabilities of the control system hardware and software, meaning that particular consideration may be needed for any parameterization beyond a piecewise constant input trajectory that is restricted to change its value only at the sampling instants.

Based on the last very practical point, a general choice is the piecewise constant control input $\mu(t, U) = U_k$ for $t_k \leq t < t_{k+1}$ parameterized by the vector $U = col(U_0, ..., U_{N-1}) \in \mathbb{R}^{mN}$. Practical experience shows that the receding horizon implementation offers considerable flexibility for a NMPC to recover performance due to sub-optimality at each step. Consequently, it is

common practice to implement move-blocking strategies such that a smaller number of parameters is required by restricted the input from change at every sampling instant on the horizon, in particular towards the end of the horizon. For example, MPC has been successfully implemented for stable plants based on linear models by optimizing a constant input on the whole horizon, Qin and Badgwell (1996).

5.2.2 Numerical Optimal Control

In this section the basic optimal control formulation in section 5.2.1 is reformulated into a form suitable for numeric solution by a nonlinear optimization solver.

As classified in Diehl et al (2009) there are two main avenues to direct numerical optimal control

- **The sequential approach.** The ODE constraint (5.5) is solved via numeric simulation when evaluating the cost and constraint functions. This means that the intermediate states $x(t_1), ..., x(t_N)$ disappear from the problem formulation by substitution into the cost and constraint functions, while the control trajectory parameters U are treated as unknowns. This leads to a sequence of simulate-optimize iterations, often known as *Direct Single Shooting*, Hicks and Ray (1971); Sargent and Sullivan (1977); Kraft (1985).
- **The simultaneous approach.** The ODE constraints (5.5) are discretized in time and the resulting finite set of nonlinear algebraic equations are treated as nonlinear equality constraints. The intermediate states $x(t_1), ..., x(t_N)$ are treated as unknown variables together with the control trajectory parameters U, and the cost function is evaluated simply by replacing the integral (5.1) by a finite sum. This leads to simultaneous solution of the ODEs and the optimization problem with a larger number of constraints and variables. The most well known methods of this type are *Direct Multiple Shooting* (Deuflhard (1974); Bock and Plitt (1984); Bock et al (1999); Leineweber et al (2003)) and *Collocation methods*, (Tsang et al (1975); Biegler (1984); von Stryk (1993)).

It is fair to say that all the above mentioned approaches have advantages that could make them the method of choice when considering a specific problem. Already now we are in position to understand some of the differences

- The simultaneous approach involves a larger number of constraints and variables and therefore leads to "bigger problems". On the other hand, the cost and constraint function evaluation is much simpler and there are structural properties of the equations and numerical advantages that can be exploited in some cases. This will be discussed in section 5.4.

- Neglecting errors due to discretization and numerical approximations, all methods results in the same optimal control trajectory. Hence, one may expect the main difference between these alternatives to be related to numerical properties and computational complexity. Numerical accuracy of the solution is a consequence of discretization, round-off errors, sensitivity to initial conditions and input, differences in linear algebraic methods, etc. and must be balanced against computational cost. These aspects will be treated in more detail in section 5.4.
- Nonlinear optimization problems are generally non-convex, and the convergence and success of a given optimization algorithm depend largely on the initial guess provided for the solution. The sequential and simultaneous approach are in this sense fundamentally different, since the simultaneous approach not only requires an initial control trajectory guess, but also one for the state trajectory. The availability of a good initial guess for the state trajectory is an advantage that can be exploited by the simultaneous approach. On the other hand, the presence of nonlinear equality constraints (which by definition are non-convex) in the simultaneous approach, one cannot expect feasible initial guesses, which has consequences for the choice of numerical methods, and will be further discussed in section 5.4.
- The sequential approach may use more or less arbitrary and separate ODE and optimization solvers, which may in some cases be simple and convenient when compared to the simultaneous approach that tend to require more specialized and integrated numeric software combining these tasks. This may be a particularly important issue for industrial users that must use software tools based on an extensive set of requirements and constraints.

5.2.2.1 Direct Single Shooting

In direct single shooting (Hicks and Ray (1971); Sargent and Sullivan (1977); Kraft (1985)), the ODE constraint (5.5) is eliminated by substituting its discretized numerical solution $x(t_k) = \phi(t_k, U, x(0))$ into the cost and constraints; minimize with respect to U the cost

$$V^*(x(0)) = \min_{U \in \mathbb{R}^p} V(U; x(0)) \triangleq \sum_{k=1}^{N} \ell(\phi(t_k, U, x(0)), \mu(t_k, U), t_k)(t_k - t_{k-1}) + S(\phi(T, U, x(0)), T) \tag{5.6}$$

subject to

$$u_{min} \leq \mu(t_k, U) \leq u_{max}, \quad t_k \in T_d \tag{5.7}$$

$$g(\phi(t_k, U, x(0)), \mu(t_k, U), t_k) \leq 0, \quad t_k \in T_d \tag{5.8}$$

and the ODE solution function $\phi(\cdot)$ is the result of a numerical integration scheme. In its simplest form, an explicit integration scheme may be used

$$x(t_{k+1}) = F(x(t_k), \mu(t_k, U), t_k), \;\; x(t_0) = x(0) \text{ given}, \tag{5.9}$$

for $k = 0, ..., N-1$, leading to

$$\phi(t_k, U, x(0)) = F(\ldots F(F(x(0), \mu(t_0, U), t_0), \mu(t_1, U), t_1), ..., \mu(t_{k-1}, U), t_{k-1}) \tag{5.10}$$

However, $\phi(t_k, U, x(0))$ may also be computed using any other (implicit) discretization scheme in the simulation.

The problem (5.6) - (5.8) is a nonlinear program in U parameterized by the initial state vector $x(0)$ and time. Dependence on time-varying external signals such as references and known disturbances are left implicit in order to keep the notation simple. The receding horizon MPC strategy will therefore re-optimize U when new state or external input information appears, typically periodically at each sample. We assume the solution exists, and let it be denoted U^*.

We note that the introduction of common modifications such as terminal constraints and infeasibility relaxations still gives a nonlinear program, but with additional decision variables and constraints.

5.2.2.2 Direct Collocation

In direct collocation (Tsang et al (1975); Biegler (1984); von Stryk (1993)) the numerical solution for $x(t_k)$ is not substituted into the cost and constraint functions, but the associated nonlinear algebraic equations resulting of an ODE discretization scheme are kept. Hence, the variables $x(t_k), \;\; k = 1, ...N$ are treated as unknown decision variables:

$$\begin{aligned} V^*(x(0)) &= \min_{U \in \mathbb{R}^p, x(t_1) \in \mathbb{R}^n, \ldots, x(t_N) \in \mathbb{R}^n} V(U, x(t_1), ..., x(t_N); x(0)) \\ &\triangleq \sum_{k=1}^{N} \ell(x(t_k), \mu(t_k, U), t_k)(t_k - t_{k-1}) + S(x(t_N), T) \end{aligned} \tag{5.11}$$

subject to

$$u_{min} \;\; \leq \mu(t_k, U) \leq u_{max}, \quad t_k \in T_d \tag{5.12}$$

$$g(x(t_k), \mu(t_k, U), t_k) \leq 0, \quad t_k \in T_d \tag{5.13}$$

$$F(x(t_{k+1}), x(t_k), \mu(t_k, U), t_k) = 0, \quad k = 0, ..., N-1 \tag{5.14}$$

$$x(t_0) = x(0) \text{ given} \tag{5.15}$$

where F is a function defined by the discretization scheme of the ODE (5.5). We observe from (5.14) that it directly allows for implicit numerical integration methods to be used, and that the algebraic equations resulting from the implicit integration scheme will be solved simultaneously with the optimization.

The problem (5.11) - (5.13) is a nonlinear program in the variables $U, x(t_1), ..., x(t_N)$ parameterized by the initial state vector $x(0)$. In addition, dependence on time-varying external signals such as references and known disturbances are left implicit in order to keep the notation simple. The receding horizon MPC strategy will therefore re-optimize U when new state or external input information appears, typically periodically at each sample. We assume the solution exists, and let it be denoted $U^*, x^*(t_1), ..., x^*(t_N)$.

5.2.2.3 Direct Multiple Shooting

Direct multiple shooting (Deuflhard (1974); Bock and Plitt (1984); Bock et al (1999); Leineweber et al (2003)) combines elements of both direct single shooting and direct collocation. It is a simultaneous approach in the sense it reformulates the ODE (5.5) to a set of nonlinear algebraic equality constraints that are solved simultaneously with the optimization. It differs from the direct collocation method since an ODE solver is used to simulate the ODE (5.5) in each time interval $t_k \leq t \leq t_{k+1}$ for $k = 0, ..., N-1$:

$$\begin{aligned} V^*(x(0)) &= \min_{U \in \mathbb{R}^p, (x(t_1),...,x(t_N))^T \in \mathbb{R}^{nN}} V(U, x(t_1), ..., x(t_N); x(0)) \\ &\triangleq \sum_{k=1}^{N} \ell(x(t_k), \mu(t_k, U), t_k)(t_k - t_{k-1}) + S(x(t_N), T) \end{aligned} \tag{5.16}$$

subject to

$$u_{min} \leq \mu(t_k, U) \leq u_{max}, \quad t_k \in T_d \tag{5.17}$$

$$g(x(t_k), \mu(t_k, U), t_k) \leq 0, \quad t_k \in T_d \tag{5.18}$$

$$x(t_{k+1}) = \phi(x(t_k), \mu(t_k, U), t_k), \quad k = 0, ..., N-1 \tag{5.19}$$

$$x(t_0) = x(0) \text{ given}, \tag{5.20}$$

where ϕ is a function defined by the simulation of the ODE (5.5). The main difference between direct multiple shooting and direct collocation is due to the use of an arbitrary ODE solver between the time-instants in T_d. Direct multiple shooting may have advantages when adaptive discretization schemes are needed (due to stiff dynamics, for example) since they might require a varying number of grid points for each iteration of the solver. With multiple shooting this can in principle be "hidden" within the direct single shooting solver used between each time-instant in T_d, while it directly leads to a change

in the dimensions of the optimization problem at each iteration with a direct collocation method. Direct multiple shooting decouples the grids required for the point-wise discretization of the constraints (5.18) and the discretization grid required to integrated the ODE. In a sense, direct multiple shooting provides additional flexibility compared to both direct single shooting and direct collocation. On the other hand, direct collocation leads to a more sparse structure that can be exploited by the numerical optimization solver.

5.2.2.4 The Nonlinear Program – Feasibility and Continuity

This section summarizes some features of the numeric optimization problem resulting from the direct approach to numerical optimal control in NMPC. Important issues related to the well-posedness of the problem are reviewed. They are related to existence and uniqueness of the solution and continuous dependence of the solution on data such as the initial state $x(0)$. These are again related to regularity properties and fundamental properties such as controllability.

In summary, all formulations reviewed in this section lead to a nonlinear optimization problem of the form

$$V^*(\theta) = \min_z V(z, \theta) \tag{5.21}$$

subject to

$$G(z, \theta) \leq 0 \tag{5.22}$$

$$H(z, \theta) = 0 \tag{5.23}$$

where z is a vector of decision variables (control trajectory parameters, intermediate states, slack variables, etc.) while θ is a vector of parameters to the problem (initial states, parameters of reference trajectories, exogenous inputs, etc.).

Existence of a solution corresponds to feasibility of the optimization problem. We define the feasible set of parameters Θ_F as the set that contains all θ for which the optimization problem (5.21)-(5.23) has a solution $z^*(\theta)$

$$\Theta_F = \{z \mid \text{there exists a } z \text{ such that } G(z, \theta) \leq 0,\ H(z, \theta) = 0\} \tag{5.24}$$

The feasible set is a result of the dynamics of the systems and basically all design parameters of the NMPC problem. Generally speaking, it is desired to make this set as large as possible while fulfilling the physical and operational constraints of the control system. We will return to this design issue in section 5.2.3.

For simplicity, let us for the time being neglect the equality constraints (5.23). Using direct single shooting they can be eliminated and are thus not

important for the understanding of the fundamental issues in this section. For a given parameter $\theta_0 \in \Theta_F$, consider the Karush-Kuhn-Tucker (KKT) first-order necessary conditions for local optimality of (5.21)-(5.22); Nocedal and Wright (1999)

$$\nabla_z L(z_0;\theta_0) = 0 \tag{5.25}$$
$$G(z_0;\theta_0) \leq 0 \tag{5.26}$$
$$\mu_0 \geq 0 \tag{5.27}$$
$$\text{diag}(\mu_0)G(z_0;\theta_0) = 0 \tag{5.28}$$

are necessary for a local minimum z_0, with associated Lagrange multiplier μ_0 and the Lagrangian defined as

$$L(z,\mu;\theta) \triangleq V(z;\theta) + \mu^T G(z;\theta) \tag{5.29}$$

Consider the optimal active set $\mathcal{A}_0$ at θ_0, i.e. a set of indices to active constraints in (5.26). The above conditions are sufficient for local optimality of z_0 provided the following second order condition holds:

$$y^T \nabla_z^2 L(z_0,\mu_0;\theta_0) y > 0, \quad \text{for all } y \in \mathcal{F} - \{0\} \tag{5.30}$$

with $\mathcal{F}$ being the set of all directions where it is not clear from first order conditions if the cost will increase or decrease:

$$\mathcal{F} = \{y \mid \nabla_z G_{\mathcal{A}_0}(z_0;\theta_0)y \geq 0, \nabla_z G_i(z_0;\theta_0)y = 0, \text{ for all } i \text{ with } (\mu_0)_i > 0\} \tag{5.31}$$

The notation $G_{\mathcal{A}_0}$ means the rows of G with indices in $\mathcal{A}_0$. The following result gives local regularity conditions for the optimal solution, Lagrange multipliers and optimal cost as functions of θ.

Assumption A1. V and G are twice continuously differentiable in a neighborhood of (z_0,θ_0).

Assumption A2. The sufficient conditions (5.25)-(5.28) and (5.30) for a local minimum at z_0 hold.

Assumption A3. Linear independence constraint qualification (LICQ) holds, i.e. the active constraint gradients $\nabla_U G_{\mathcal{A}_0}(z_0;\theta_0)$ are linearly independent.

Assumption A4. Strict complementary slackness holds, i.e. $(\mu_0)_{\mathcal{A}_0} > 0$.

Theorem 5.1. *For a given z_0 and θ_0 then under assumptions A1-A3, z_0 is a local isolated minimum, and for θ in a neighborhood of θ_0, there exists a unique continuous function $z^*(\theta)$ satisfying $z^*(\theta_0) = z_0$ and the sufficient conditions for a local minimum.*

If in addition A4 holds, then for θ in a neighborhood of θ_0 the function $z^*(\theta)$ is differentiable and the associated Lagrange multipliers $\mu^*(\theta)$ exists, and are unique and continuously differentiable. Finally, the set of active con-

straints is unchanged, and the active constraint gradients are linearly independent at $z^*(\theta)$.

The first part is proven in Kojima (1980), and the 2nd part follows from Theorem 3.2.2 in Fiacco (1983).

The concept of controllability of nonlinear systems can be defined in several ways. Here we have taken a pragmatic point of view, and focus on conditions that leads to feasibility of the solution, and continuity of the value function or solution as a function of the time-varying data θ that includes the initial conditions. In the context of numerical optimal control, issues related to lack of controllability or inappropriate design choices will typically manifest themselves in terms of infeasibility (no solution exists), indefiniteness of the Hessian (a global solution is not found), or singularity or poor conditioning of the Hessian (the solution is not unique and continuously dependent on the input data, or is highly sensitive to changes in decision variables). The latter case means that small changes in the state may require very large control actions to compensate. Since the above sufficient optimality conditions are practically impossible to verify a priori, these are important issues to be monitored by the practical NMPC algorithm based on output from the numerical solver in order to asses the quality of the NMPC design and identify problems related to lack of controllability or inappropriate design or tuning of the NMPC criterion and constraints.

The simplest special case for which strong properties can be guarantees a priori is the case of joint convexity:

A5. V and G are jointly convex for all (z, θ).

The optimal cost function can now be shown to have some regularity properties, Mangasarian and Rosen (1964):

Theorem 5.2. *Suppose A1-A5 holds. Then* X_F *is a closed convex set, and* $V^* : \Theta_F \to \mathbb{R}$ *is convex and continuous.*

Convexity of Θ_F and V^* is a direct consequence of A5, while continuity of V^* can be established under weaker conditions; Fiacco (1983). We remark that V^* is in general not differentiable, but properties such as local differentiability and directional differentiability can be investigated as shown in e.g. Fiacco (1983). Regularity properties of the solution function z^* is a slightly more delicate issue, and essentially relies on stronger assumptions such as strict joint convexity that ensure uniqueness of the solution.

5.2.3 Tuning and Stability

The specification of the NMPC control functionality and dynamic performance is essentially provided through the cost function and the constraints. We will not go into details on the practical tuning tradeoffs and the types of

physical and operational constraints, but note that one may typically choose l_2 or l_1 type cost function

$$\ell(x,u,t) = ||x - r_x(t))||_Q^2 + ||u - r_u(t)||_R^2 \tag{5.32}$$
$$\ell(x,u,t) = ||Q(x - r_x(t))||_1 + ||R(u - r_u(t)||_1 \tag{5.33}$$

where the properties of the weight matrices $Q \succeq 0$ and $R \succeq 0$ are essential for performance, and in some cases also stability. In the simplest case when there exists an $\varepsilon > 0$ such that

$$\ell(x,u,t) \geq \varepsilon(||x||^2 + ||u||^2) \tag{5.34}$$

it is clear that all states and control actions are directly observable through the cost function, an it follows intuitively that minimization of the cost function will influence all states that are controllable. Based on the similar arguments, it is in fact sufficient for stabilization that only the unstable modes of the system are observable through the cost function, such that $Q \succeq 0$ may be sufficient if weights are given on the unstable modes, Mayne et al (2000). In order to ensure uniqueness of the control trajectory it is generally recommended that $R \succ 0$. In summary, conventional LQR tuning guidelines (e.g. Athans and Falb (1966)) are very helpful as a starting point also for NMPC.

Although the effect of modeling errors and disturbances will be discussed in section 5.2.4.2, we remark that incorrect choice of the reference $r_u(t)$ for the control input may lead to a steady-state error that will be important to compensate for in many applications.

NMPC is based on the receding horizon control principle, where a finite horizon open loop optimal control problem solved at each sampling instant and the optimized control trajectory is implemented until a new optimized control trajectory is available at the next sampling instant. This leads to closed-loop control since each new optimized control trajectory is based on the most recent state information. However, the numerical optimal control problem solved at each sampling instant provides essentially an open-loop control trajectory. The finite-horizon cost function imposes in principle no stability requirement by itself, and with an unfortunate choice of design parameters (horizon T, weight matrices Q and R, terminal cost S, and certain constraints) the closed loop NMPC may be unstable. In particular for open loop unstable systems, it is important to understand how these design parameters should be chosen to avoid an unstable NMPC.

5.2.3.1 Stability Preserving Constraints And Cost-to-go

This section discusses stability of the NMPC in more depth, and how this property is related to design parameters in the cost function and constraints. The description will be fairly informal, and we avoid the technical details in

order to focus on the most important concepts. For simplicity we assumme that the objective is regulation to a constant set-point r. Further details and a more rigorous treatment of the topic are found in Chen and Allgöwer (1998); Mayne et al (2000); Michalska and Mayne (1993); Keerthi and Gilbert (1988); Mayne and Michalska (1990), and we remark that the concepts relevant for NMPC are essentially the same as for linear MPC.

The following principles are generally useful to ensure stability of an NMPC Mayne et al (2000):

- The control trajectory parameterization $\mu(t, U)$ must be "sufficiently rich" - most theoretical work assume piecewise constant control input trajectory that is allowed to move at each sampling instant.
- From the optimality principle of dynamic programming, Bellman (1957), an infinite horizon cost may be expected to have a stabilizing property. Theoretically, this leads to an infinite dimensional problem (except in simple special cases), so more practical approaches are
  - Sufficiently large horizon T. However, it is not obvious to know what is large enough, in particular for an open loop unstable system and when the constrained outputs are non-minimum phase (see Saberi et al (2002) for results on the importance of the zero-dynamics of the constrained outputs for the linear case).
  - A terminal cost chosen to approximate the cost-to-go, i.e. $S(x(T), T) \approx \int_{t=T}^{\infty} \ell(x(t), u(t), t)dt$ such that the total cost function approximates an infinite horizon cost. Unfortunately, the cost-to-go is generally hard to compute and simple approximations are usually chosen.
- Terminal set constraints of the type $x(t_N) \in \Omega$ that ensures that the state is regulated "close enough" to the set-point such that after T it is a priori known that there exists a feasible and stabilizing controller that will ensure that $x(t), t \geq T$ never leaves Ω and eventually goes asymptotically to the set-point. There are many algorithms based on this philosophy, some of them are defined as dual mode NMPC (Michalska and Mayne (1993)) since they switch to a stabilizing simpler (non-NMPC) control law once Ω is reached, while others continue to use NMPC also in Ω with the confidence that there exist an (explicit or implicit) stabilizing control law that the NMPC may improve upon.
- Terminal equality constraints of the type $x(t_N) = r$, Keerthi and Gilbert (1988), that ensures convergence in finite time. This basically implies that the cost after time T is zero, and is therefore related to both an infinite-cost strategy and a stability-preserving-constraint strategy.
- Finally, the idea of choosing the cost-to-go to approximate an infinite-horizon cost and the use of a terminal set may be combined. With the use of a terminal set it will be sufficient to approximate the cost-to-go for states that are within the terminal set, and simple tools like local linearization can be applied to make this a fairly practical approach; Chen and Allgöwer (1998).

A formal treatment of these issues are found in the references, see Mayne et al (2000) for additional references. The main tools are the use of either the value function $V^*(x)$ as a Lyapunov function, or investigating monotony of a sequences of value function values. Instead, we provide an example that is essentially similar to the method in Chen and Allgöwer (1998).

Example. Consider the discrete-time non-linear system

$$x(t_{k+1}) = F(x(t_k), u(t_k)) \tag{5.35}$$

where $x \in \mathbb{R}^n$ is the state, and $u \in \mathbb{R}^m$ is the input. We assume the control objective is regulation to the origin. For the current $x(t_k)$, we formulate the optimization problem

$$V^*(x(t_k)) = \min_U J(U, x(t_k)) \tag{5.36}$$

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

$$\begin{aligned} & y_{\min} \leq y_{k+i|k} \leq y_{\max}, \quad i = 1, ..., N \\ & u_{\min} \leq u_{k+i} \leq u_{\max}, \quad i = 0, 1, ..., N-1, \\ & x_{k+N|k} \in \Omega \\ & x_{k+i+1|k} = F(x_{k+i|k}, u_{k+i}), \quad i = 0, 1, ..., N-1 \\ & y_{k+i|k} = C x_{k+i|k}, \quad i = 1, 2, ..., N \end{aligned} \tag{5.37}$$

with $U = \{u_k, u_{k+1}, ..., u_{k+N-1}\}$ and the cost function given by

$$J(U, x(t_k)) = \sum_{i=0}^{N-1} \left(||x_{k+i|k}||_Q^2 + ||u_{k+i}||_R^2 \right) + ||x_{k+N|k}||_P^2 \tag{5.38}$$

The compact and convex terminal set Ω is defined by

$$\Omega = \{x \in \mathbb{R}^n \mid x^T P x \leq \alpha\} \tag{5.39}$$

where $P = P^T \succ 0$ and $\alpha > 0$ will be specified shortly. An optimal solution to the problem (5.36)-(5.37) is denoted $U^* = \{u_t^*, u_{t+1}^*, ..., u_{t+N-1}^*\}$, and the control input is chosen according to the receding horizon policy $u(t_k) = u_t^*$. This and similar optimization problems can be formulated in a concise form

$$V^*(x) = \min_U J(U, x) \quad \text{subject to} \quad G(U, x) \leq 0 \tag{5.40}$$

Define the set of N-step feasible initial states as follows

$$X_F = \{x \in \mathbb{R}^n \mid G(U, x) \leq 0 \text{ for some } U \in \mathbb{R}^{Nm}\} \tag{5.41}$$

Suppose Ω is a control invariant set, such that X_F is a subset of the N-step stabilizable set, Kerrigan and Maciejowski (2000). Notice that the origin is an

equilibrium and interior point in X_F. It remains to specify $P \succ 0$ and $\alpha > 0$ such that Ω is a control invariant set. For this purpose, we use the ideas of Chen and Allgöwer (1998), where one simultaneously determine a linear feedback such that Ω is positively invariant under this feedback. Define the local linearization at the origin

$$A = \frac{\partial f}{\partial x}(0,0), \quad B = \frac{\partial F}{\partial u}(0,0) \tag{5.42}$$

Now, the following assumptions are made:

- (A, B) is stabilizable.
- $P, Q, R \succ 0$.
- $y_{min} < 0 < y_{max}$ and $u_{min} < 0 < u_{max}$.
- The function f is twice continuously differentiable, with $f(0,0) = 0$.

Since (A, B) is stabilizable, let K denote the associated LQ optimal gain matrix, such that $A_0 = A - BK$ is strictly Hurwitz. A discrete-time reformulation of Lemma 1 in Chen and Allgöwer (1998) can be made, Johansen (2004):

Lemma 5.1. *If $P \succ 0$ satisfies the Lyapunov-equation*

$$A_0^T P A_0 - P = -\kappa P - Q - K^T R K \tag{5.43}$$

for some $\kappa > 0$, there exists a constant $\alpha > 0$ such that Ω defined in (5.39) satisfies

1. *$\Omega \subset \mathcal{C} = \{x \in \mathbb{R}^n \mid u_{min} \leq -Kx \leq u_{max}, y_{min} \leq Cx \leq y_{max}\}$.*
2. *The autonomous nonlinear system*

$$x(t_{k+1}) = F(x(t_k), -Kx(t_k)) \tag{5.44}$$

   is asymptotically stable for all $x(0) \in \Omega$, i.e. Ω is positively invariant.
3. *The infinite-horizon cost for the system (5.44)*

$$J_\infty(x(t_k)) = \sum_{i=0}^{\infty} \left(||x_{k+i|k}||_Q^2 + ||Kx_{k+i|k}||_R^2 \right) \tag{5.45}$$

   satisfies $J_\infty(x) \leq x^T P x$ for all $x \in \Omega$.

In order to prove this result we first remark that the Lyapunov-equation (5.43) is generally satisfied for sufficiently small $\kappa > 0$ because A_0 is strictly Hurwitz and the right-hand side is negative definite. One may define a set of the form

$$\Omega_{\alpha_1} = \{x \in \mathbb{R}^n \mid x^T P x \leq \alpha_1\} \tag{5.46}$$

with $\alpha_1 > 0$, such that $\Omega_{\alpha_1} \subseteq \mathcal{C}$, i.e. an ellipsoidal inner approximation Ω_{α_1} to the polyhedron $\mathcal{C}$ where the input and state constraints are satisfied. Hence, the first claim holds for all $\alpha \in (0, \alpha_1]$.

Define the positive definite function $W(x) = x^T P x$. Along trajectories of the autonomous system (5.44) we have

$$\begin{aligned} W(x(t_{k+1})) - W(x(t_k)) &= \left(A_0 x(t_k) + \phi(x(t_k))\right)^T P \left(A_0 x(t_k) + \phi(x(t_k))\right) \\ &\quad - x^T(t_k) P x(t_k) \\ &= x^T(t_k) \left(A_0^T P A_0 - P\right) x(t_k) + \phi^T(x(t_k)) P \phi(x(t_k)) \\ &\quad + x^T(t_k)(A_0^T P + P A_0)\phi(x(t_k)) \end{aligned}$$

where $\phi(x) = F(x, -Kx) - A_0 x$ satisfies $\phi(0) = 0$. From (5.43)

$$\begin{aligned} W(x(t_{k+1})) - W(x(t_k)) &= -x^T(t_k) \left(Q + K^T R K + \kappa P\right) x(t_k) \\ &\quad + x^T(t_k)(A_0^T P + P A_0)\phi(x(t_k)) + \phi^T(x(t_k)) P \phi(x(t_k)) \end{aligned}$$

Let L_ϕ be a Lipschitz constant for ϕ in Ω_α (which must exist because f is differentiable). Since $\partial\phi/\partial x(0) = 0$ and ϕ is twice differentiable we can choose $L_\phi > 0$ as close to zero as desired by selecting $\alpha > 0$ sufficiently small. Hence, there exist $\alpha \in (0, \alpha_1]$ such that

$$W(x(t_{k+1})) - W(x(t_k)) \leq -x^T(t_k) \left(\frac{\kappa}{2} P + Q + K^T R K\right) x(t_k) \quad (5.47)$$

for all $x(t_k) \in \Omega$ and positive invariance of Ω follows since Ω is a level set of W.

Notice that from (5.47) we have

$$W(x(\infty)) - W(x(0)) \leq -J_\infty(x(0)) - \frac{\kappa}{2} \sum_{k=0}^{\infty} ||x(t_k)||_P^2 \quad (5.48)$$

and the third claim holds because $W(x(\infty)) = 0$ for all $x(0) \in \Omega$.

Hence, the result is proven, and it follows from Mayne et al (2000); Chen and Allgöwer (1998) that the RHC makes the origin asymptotically stable with region of attraction equal to the feasible set X_F. A procedure for selecting P, κ and α can be adapted from Chen and Allgöwer (1998).

5.2.3.2 Sub-optimal NMPC

It may be difficult to establish a non-conservative hard bound on the number of iterations required for convergence of the nonlinear programming problem that NMPC must solve numerically at each sampling instant. Furthermore, there may not be computational resources available to guarantee that a sufficient number of iterations can be computed, and only a local minimum may

be found. As an example, some NMPC methods will assume that only one iteration is performed per sample, Li and Biegler (1989, 1990). Hence, it is of interest to understand the consequences of not converting in terms of control performance loss. A fundamental result is given in Scokaert et al (1999), where it is shown that feasibility and descent (reduction in cost function compared to the control trajectory computed at the previous sample) is sufficient for asymptotic stability of NMPC provided that terminal constraints are included in the formulation. Hence, optimality is not required. In the same spirit, a computationally efficient and robust implementation of these ideas are pursued in Lazar et al (2008), and also exploited in the context of approximate NMPC Bemporad and Filippi (2003); Johansen (2004).

5.2.3.3 Example: Compressor Surge Control

Consider the following 2nd-order compressor model Greitzer (1976); Gravdahl and Egeland (1997) with x_1 being normalized mass flow, x_2 normalized pressure and u normalized mass flow through a close coupled valve in series with the compressor

$$\dot{x}_1 = B\left(\Psi_e(x_1) - x_2 - u\right) \tag{5.49}$$

$$\dot{x}_2 = \frac{1}{B}\left(x_1 - \Phi(x_2)\right) \tag{5.50}$$

The following compressor and valve characteristics are used

$$\Psi_e(x_1) = \psi_{c0} + H\left(1 + 1.5\left(\frac{x_1}{W} - 1\right) - 0.5\left(\frac{x_1}{W} - 1\right)^3\right)$$

$$\Phi(x_2) = \gamma \text{sign}(x_2)\sqrt{|x_2|}$$

with $\gamma = 0.5$, $B = 1$, $H = 0.18$, $\psi_{c0} = 0.3$ and $W = 0.25$. The control objective is to avoid surge, i.e. stabilize the system. This may be formulated as

$$\ell(x,u) = \alpha(x - x^*)^T(x - x^*) + \kappa u^2$$

$$S(x) = Rv^2 + \beta(x - x^*)^T(x - x^*)$$

with $\alpha, \beta, \kappa, \rho \geq 0$ and the set-point $x_1^* = 0.40$, $x_2^* = 0.60$ corresponds to an unstable equilibrium point. We have chosen $\alpha = 1$, $\beta = 0$, and $\kappa = 0.08$. The horizon is chosen as $T = 12$, which is split into $N = p = 15$ equal-sized intervals, using piecewise constant control input parameterization. Valve capacity requires the constraint

$$0 \leq u(t) \leq 0.3 \tag{5.51}$$

to hold, and the pressure constraint

$$x_2 \geq 0.4 - v \tag{5.52}$$

avoids operation too far left of the operating point. The variable $v \geq 0$ is a slack variable introduced in order to avoid infeasibility and $R = 8$ is its weight in the cost function.

A nonlinear optimization problem is formulated using direct single shooting where explicit Euler integration with step size 0.02 is applied to solve the ODE. Due to the unstable dynamics, this may not be the best choice, but it is sufficient for this simple example.

The NLP solution is shown in Figure 5.1 as a function $u^*(x)$. The corresponding optimal cost $V^*(x)$ is shown in Figure 5.2, and simulation results are shown in Figure 5.3, where the controller is switched on after $t = 20$. We note that it quickly stabilizes the deep surge oscillations.

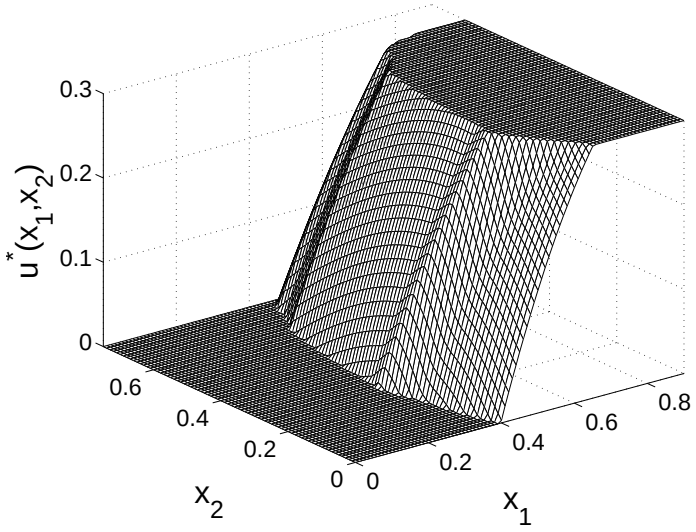


Fig. 5.1 Feedback control law.

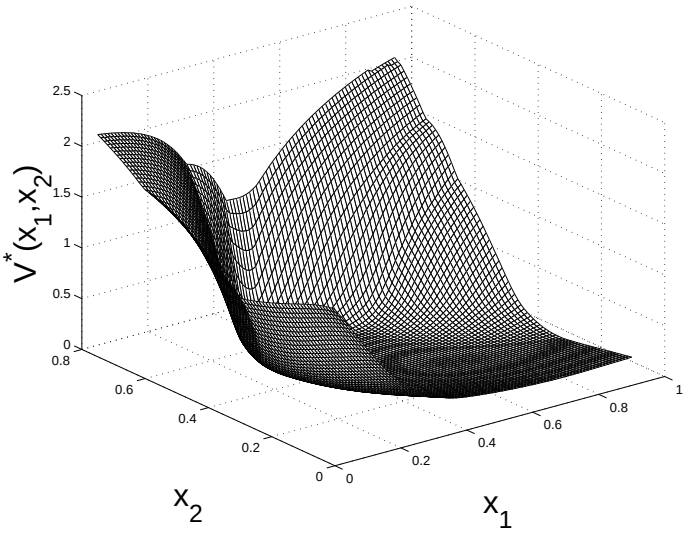


Fig. 5.2 Optimal costs of the feedback control law.

5.2.4 Extensions and Variations of the Problem Formulation

5.2.4.1 Infeasibility Handling and Slack Variables

Feasibility of the NMPC optimization problem is an essential requirement for any meaningful state and reference command, and it is importance in practice that the NMPC optimization problem is formulated such that feasibility is ensured as far as possible by relaxing the constraints when needed and when possible. Obviously, physical constraints like input saturation can never be related, but operational constraints can generally be relaxed according to certain priorities under the additional requirement that safety constraints are fulfilled by a separate system (like an emergency shutdown system, pressure relief valves, or by functions in a decentralized control system). Stability-enforcing terminal constraints may also be relaxed in practice, or even skipped completely, since they tend to be conservative and often not needed when the NMPC is otherwise carefully designed, in particular for open loop stable systems.

A general way to reformulate an optimization problem to guarantee feasibility is to use slack variables (e.g. Vada et al (1999)). Taking the fairly general NLP formulation (5.21)-(5.23) as the starting point, we reformulate it in the following way

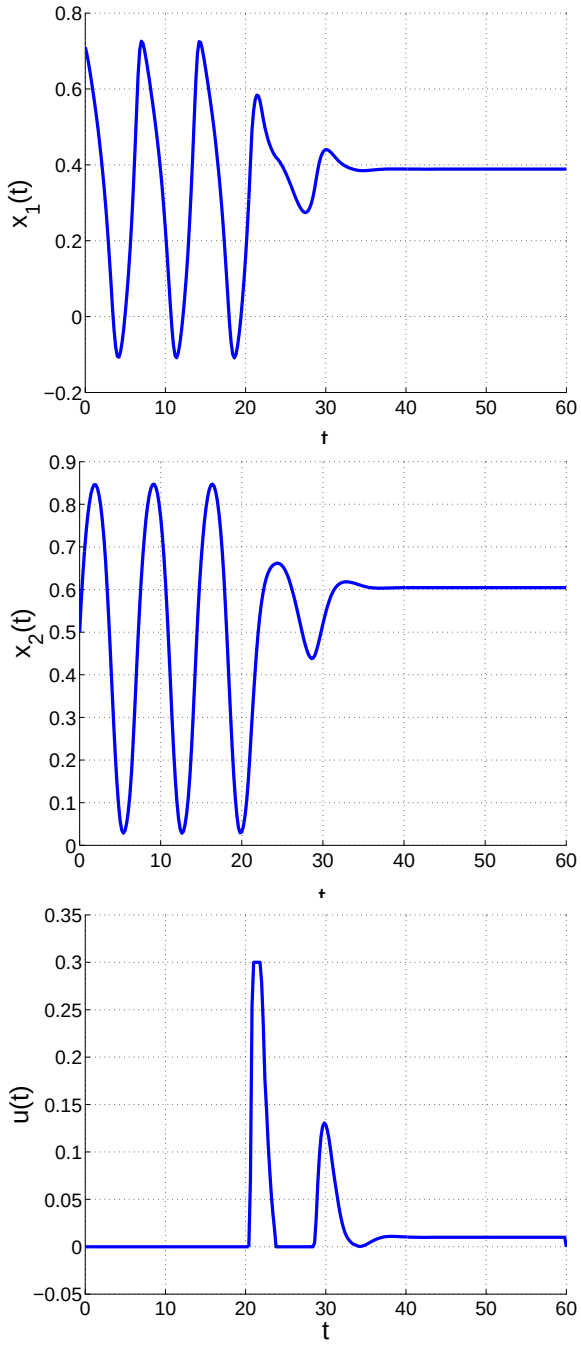


Fig. 5.3 Simulation of compressor with nonlinear MPC.

$$V_s^*(\theta) = \min_{z,s,q} V(z,\theta) + ||W_s s||_1 + ||W_q q||_1 \tag{5.53}$$

subject to

$$G(z,\theta) \leq s \tag{5.54}$$
$$H(z,\theta) = q \tag{5.55}$$
$$s \geq 0 \tag{5.56}$$

where $W_s \succeq 0$ and $W_q \succeq 0$ are weight matrices of appropriate dimension. They are usually chosen such that the two latter penalty terms of (5.53) dominates the first term in order to ensure that the feasibility constraints are not relaxed when not needed.

5.2.4.2 Robustness

Practical industrial experience shows that MPC tend to be inherently robust, Qin and Badgwell (1996, 2000), even without any particular consideration in the design phase beyond ensuring the accuracy of dynamic models and formulating realistic specifications in terms of operational constraints and cost function weights. In addition, mechanisms to handle steady state model errors (integral action like mechanisms) are usually implemented.

As a contrast to this practical experience, it is shown by examples, Grimm et al (2004), that when the NMPC problem involves state constraints, or terminal constraints in combination with short prediction horizons, the asymptotic stability of the closed-loop may have not be robust. A necessary condition for lack of robustness is that the value function and state feedback law are discontinuous, Grimm et al (2004), while at the same time lack of continuity does not necessarily lead to lack of robustness, Lazar et al (2007).

There exist a wide range of NMPC formulation that include robustness into the formulation of the optimization problem. One can mainly distinguish between three types of approaches; stochastic NMPC, min-max NMPC, and mechanisms to avoid steady-state errors.

There are two formulations of min-max NMPC: the open-loop and the closed-loop formulation (see Magni and Scattolini (2007) for review of the min-max NMPC approaches). The open-loop min-max NMPC (Michalska and Mayne (1993); Limon et al (2002); Magni and Scattolini (2007)) guarantees the robust stability and the robust feasibility of the system, but it may be very conservative since the control sequence has to ensure constraints fulfillment for all possible uncertainty scenarios without considering the fact that future measurements of the state contain information about past uncertainty values. As a result, the open-loop min-max NMPC controllers may have a small feasible set and a poor performance because they do not include the effect of feedback provided by the receding horizon strategy of MPC.

Most min-max MPC robustness approaches assume a fairly simple additive uncertainty model of the form

$$x_{k+1} = F(x_k, u_k) + w_k \tag{5.57}$$

where some bound on the unknown uncertainty w_k is assumed. The conservativeness of the open-loop approaches is overcome by the closed-loop min-max NMPC (Magni et al (2003); Magni and Scattolini (2007); Limon et al (2006)), where the optimization is performed over a sequence of feedback control policies. With the closed-loop approach, the min-max NMPC problem represents a differential game where the controller is the minimizing player and the disturbance is the output of the maximizing player. The controller chooses the control input as a function of the current state so as to ensure that the effect of the disturbance on the system output is sufficiently small for any choice made by the maximizing player. In this way, the closed-loop min-max NMPC would guarantee a larger feasible set and a higher level of performance compared to the open-loop min-max NMPC (Magni et al (2003)).

Stochastic NMPC formulations are based on a probabilistic description of uncertainty, and can also be characterized as open-loop Cannon et al (2009); Kantas et al (2009) and closed-loop Goodwin et al (2009); Arellano-Garcia et al (2007) similarly to min-max robust NMPC as described above. They also share similar challenges due to significantly increased computational complexity when compared to nominal NMPC formulations.

The reformulation of nonlinear models as Linear Parameter Varying (LPV) models allows for the use of linear and bi-linear matrix inequality formulations of robust NMPC, Angeli et al (2000); Casavola et al (2003); Wan and Kothare (2004). The embedding of nonlinear systems into the class of LPV models

$$x_{k+1} = A(p_k)x_k + B(p_k)u_k + w(p_k) \tag{5.58}$$

leads to loss of information in the model that leads to more conservative robust control. However, using tools of semi-definite and convex programming, Boyd et al (1994), the LPV re-formulation allows for the computational complexity to be significantly reduced in many cases. In (5.58), p_k is a parameter whose value is known to belong to some bounded set, and some approaches also assume that its time-derivative has a known bound, and the LPV re-formulation clearly allows a richer class of uncertainty to be modeled, compared to (5.57).

Steady-state control errors may result if there are steady-state model errors. While linear control design offers several tools to deal with this problem (including integral action, integrating models in linear MPC, and others), not all of them are directly transferable to nonlinear systems. The commonly used cure for steady-state errors in MPC, which can be directly transferred to NMPC, appears to be the use of a state estimator or observer that estimates an input or output disturbance for direct compensation in the NMPC

cost function, Muske and Badgwell (2002); Pannocchia and Rawlings (2003); Pannocchia and Bemporad (2007); Borrelli and Morari (2007).

5.2.4.3 Observers and Output Feedback

Most formulations of nonlinear MPC assume state feedback. They are usually based on state space models, e.g. Balchen et al (1992); Foss and Schei (2007), although certain black-box using discrete-time nonlinear input/output models have also been proposed Nørgaard et al (2000); Åksesson and Toivonen (2006). Since all states are usually not measured, any implementation of NMPC based on a state space model will require a state estimator, which is often a critical component of an NMPC Kolås et al (2008). State space models have the advantage that they are most conveniently based on first principles.

Although practical rules of thumb for observer design such as separation of time-scales (typically one order of magnitude faster state estimator relative to the control loop response time) tend to be applicable in practical implementations also for NMPC, there also exist a number of rigorous theoretical results on the stability of the combination of observers with NMPC, see Findeisen et al (2003b) for an overview. Although a general separation principles does not exists for NMPC, there are some results in this direction, Findeisen et al (2003a); Adetola and Guay (2003); Messina et al (2005); Roset et al (2006).

5.2.4.4 Mixed-integer MPC

General NMPC formulations based on nonlinear models suffer from the fact that it is hard to verify whether the underlying optimization problem is convex or not, such that in general it must be assumed to be non-convex. At the same time, all practical optimization solvers will assume some form of local convexity and guarantee convergence only to good initial guesses for the solution. This challenge will be further discussed in section 5.4. On the other hand, NMPC based on piecewise linear (PWL) models and cost functions will in general lead to mixed-integer linear programs (MI-LP) for which there exists solvers that guarantee global convergence, Tyler and Morari (1999); Bemporad and Morari (1999). The equivalence between a wide class of hybrid systems models, mixed logic models and PWL models, Heemels et al (2001), makes this approach attractive in many practical applications. Despite its applicability and importance, we only remark that the MI-LP theory and software are well developed, and refer to the references above and the large literature on MI-LP, Williams (1999).

5.2.4.5 Decentralized and Distributed NMPC

Recently, several approaches for decentralized and distributed implementation of NMPC algorithms have been developed. A review of architectures for distributed and hierarchical MPC can be found in Scattolini (2009). The possibility to use MPC in a decentralized fashion has the advantage to reduce the original, large size, optimization problem into a number of smaller and more tractable ones.

In Magni and Scattolini (2006), a stabilizing decentralized MPC algorithm for nonlinear systems consisting of several interconnected local subsystems is developed. It is derived under the main assumptions that no information can be exchanged between local control laws, i.e. the coupling between the subsystems is ignored, and only input constraints are imposed on the system. In Dunbar and Murray (2006), it is supposed that the dynamics and constraints of the nonlinear subsystems are decoupled, but their state vectors are coupled in a single cost function of a finite horizon optimal control problem. In Keviczky et al (2006), an optimal control problem for a set of dynamically decoupled nonlinear systems, where the cost function and constraints couple the dynamical behavior of the systems, is solved.

5.3 NMHE Optimization Problem Formulation

In this section we consider the formulation of the NMHE optimization problem, and we follow as similar organization as section 5.2, with focus on the formulation of the optimization problem and the link between fundamental properties such as observability, detectability and existence and uniqueness of the solution.

5.3.1 Basic Problem Formulation

The state estimation problem is to determine the current state based on a sequence of past and current measurements at discrete time instants, and the use of a dynamic model. For simplicity, we will assume data are available via synchronous sampling. Extension to be more general situation when data from the different sensors and data channels are asynchronous are conceptually straightforward and does not lead to any fundamental complications, but the mathematical notation requires many more indices and becomes unnecessarily tedious for an introduction to the topic. The problem can be treated by careful discretization of the continuous-time system to take asynchronous data into account, or a more pragmatic approach would be to rely on digital signal processing technique of interpolation and extrapolation for

pre-processing the data before used in the NMHE in order to artificially provide synchronized data as required at each sampling instant, Proakis and Manolakis (1996).

At the time t_k corresponding to the discrete time index k we consider a set of $N+1$ sampling instants $T_s = \{t_{k-N}, t_{k-N+1}, ..., t_k\}$, where the following synchronized window of output and input data are available

$$Y_k = \text{col}(y(t_{k-N}), y(t_{k-N+1}), ..., y(t_k))$$
$$U_k = \text{col}(u(t_{k-N}), u(t_{k-N+1}), ..., u(t_k))$$

where $y(t) \in \mathbb{R}^r$ and $u(t) \in \mathbb{R}^m$. We assume without loss of generality that sampling is periodic, i.e. the horizon $T = t_k - t_{k-N}$ and the sampling interval $t_s = t_i - t_{i-1}$ are constant. The inputs and outputs may be related by an ODE model

$$\frac{d}{dt}x(t) = f(x(t), u(t), w(t)) \tag{5.59a}$$
$$y(t) = h(x(t), u(t)) + v(t) \tag{5.59b}$$

with unknown initial condition $x(t_{k-N}) \subset \mathbb{R}^n$. The variable w includes unknown model errors and disturbances, and v includes unknown additive measurement errors. In addition, one may have available a priori information about $x(t)$ in the form of constraints on states and uncertainty

$$\text{col}(x(t), w(t), v(t)) \in X \times W \times V, \quad t \in [t_{k-N}, t_k] \tag{5.60}$$

for some compact sets X, W and V. The constraints may result from operational knowledge of the system, or physical properties of the states (such as chemical composition never being negative at any point in time). More generally, such a priori knowledge may incorporate more complex statements that motivates a more general constraint formulation

$$C(x(t), w(t), v(t), t) \leq 0, \quad t \in [t_{k-N}, t_k] \tag{5.61}$$

The above constraint could incorporate time-varying information and statements that involves the interaction between two ore more variables - for example that a gas pressure is always below a certain threshold, expressed through the product of gas mass and temperature through the ideal gas law. One may also have a priori information that is not linked to a particular time instant, like that the average value of a certain variable is known to stay between certain upper and lower bounds or that the measurement noise has zero mean, which can be expressed as

$$\int_{t_{k-N}}^{t_k} c(x(t), w(t), v(t))dt \leq 0 \tag{5.62}$$

The state estimation problem is essentially to estimate $x(t_k)$ based on the $N+1$ data samples, the model, and the a priori information given in the form of constraints.

5.3.1.1 Observability

The concept of observability is essential in order to formulate and understand the NMHE problem. In this section we will for convenience assume that the dynamic model system (5.59) is discretized in the form of a state space formulation

$$x_{k+1} = F(x_k, u_k, w_k) \tag{5.63a}$$

$$y_k = h(x_k, u_k) + v_k \tag{5.63b}$$

with the convenient notation $u_k = u(t_k), y_k = y(t_k), v_k = v(t_k), w_k = w(t_k)$. In this section, we will neglect the constraints (5.60)-(5.62) since they are not important for the observability concept. Furthermore, the process noise v_k and measurement noise w_k will also be set to zero and neglected in this section when defining the concept of observability. Note that by using the discrete-time equation (5.63a) recursively with initial condition $x(t_{k-N})$ and $v_k = 0$ and $w_k = 0$, one will uniquely determine $x(t), t \geq t_{k-N}$, including the current state $x(t_k)$ that we want to estimate.

To express Y_k as a function of x_{k-N} and U_k under these conditions, denote $F_k(x_k) = F(x_k, u_k, 0)$ and $h_k(x_k) = h(x_k, u_k)$, and note from (5.63b) that the following algebraic map can be formulated, Moraal and Grizzle (1995b):

$$Y_k = H(x_{k-N}, U_k) = \begin{bmatrix} h^{u_{k-N}}(x_{k-N}) \\ h^{u_{k-N+1}} \circ F_{k-N}(x_{k-N}) \\ \vdots \\ h^{u_k} \circ F_{k-1} \circ \cdots \circ F_{k-N}(x_{k-N}) \end{bmatrix} \tag{5.64}$$

Hence, without the presence of any uncertainty and constraints, the state estimation problem is equivalent to the inversion of this set of nonlinear algebraic equations, like in the case of a linear system when full rank of the observability matrix is equivalent to observability. In order to better understand the similarities between the linear and nonlinear case, consider the linear system $x_{k+1} = Ax_k + Bu_k$ with output $y_k = Cx_k$. The (5.64) corresponds to

$$Y_k = \mathbb{C}_N x_{k-N} + \mathbb{B}_N U_k \tag{5.65}$$

where the matrix $\mathbb{C}_N$ is defined by

$$\mathbb{C}_N = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^N \end{pmatrix} \tag{5.66}$$

and $\mathbb{B}_N$ is a matrix that contains blocks of the form CA^iB. Clearly, the state can be uniquely determined from the window of past inputs and outputs by inverting the linear algebraic equations (5.65) if and only if $\mathbb{C}_N$ has full rank. It is well known from linear systems theory that $\text{rank}(\mathbb{C}_N) = \text{rank}(\mathbb{C}_n)$ for $N \geq n$, where $\mathbb{C}_n$ is known as the observability matrix. Similarly, in the nonlinear case, conditions that ensure that the inverse problem is well-posed (Tikhonov and Arsenin (1977)) in the sense that the inverse of (5.64) exists, is unique, and depends continuously on the data U_k and Y_k are of fundamental importance and essentially amounts to the concept of observability.

Definition 5.1 (Moraal and Grizzle (1995b)). The system (5.63) 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 $U_k \in \mathbb{U}^{N+1}$ such that

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

Definition 5.2 (Sui and Johansen (2010)). The input $U_k \in \mathbb{U}^{N+1}$ is said to be N-*exciting* for the N-observable system (5.63) at time index k if there exists a K-function φ_k that for all $x_1, x_2 \in \mathbb{X}$ satisfies

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

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

$$H(x_1, U_k) - H(x_2, U_k) = \Phi_k(x_1, x_2)(x_1 - x_2), \tag{5.67}$$

where

$$\Phi_k(x_1, x_2) = \int_0^1 \frac{\partial}{\partial x} H((1-s)x_2 + sx_1, U_k) ds. \tag{5.68}$$

An observability rank condition can be formulated similar to the linear case outlined above (see also Moraal and Grizzle (1995b); Alessandri et al (2008); Fiacco (1983) and others for similar results):

Lemma 5.2. *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 $\frac{\partial H}{\partial x}(x, U_k)$ has full rank (equal to n) for all $x \in \mathbb{X}$ and some $U_k \in \mathbb{U}^{N+1}$, then the system is N-observable and the input U_k is N-exciting for the system (5.63) at time index k.*

Proof (Sui and Johansen (2010)). Due to the observability rank condition being satisfied, $\Phi_k^T(\cdot)\Phi_k(\cdot) > 0$ and the system of nonlinear algebraic equations (5.67) can be inverted as follows:

$$x_1 - x_2 = \Phi_k^+(x_1, x_2)\big(H(x_1, U_k) - H(x_2, U_k)\big),$$
$$\Rightarrow \frac{1}{\pi_k^2(x_1, x_2)}\|x_1 - x_2\|^2 \leq \|H(x_1, U_k) - H(x_2, U_k)\|^2,$$

where $\pi_k(x_1, x_2) = \|\Phi_k^+(x_1, x_2)\|$. This proves that the conditions in Definitions 5.1 and 5.2 hold with $\varphi(s) = s/\overline{p}^2$ where

$$\overline{p} = \sup_{x_1, x_2 \in \mathbb{X}, U_k \in \mathbb{U}^{N+1}} ||\Phi_k^+(x_1, x_2)|| \tag{5.69}$$

is bounded due to F and h are twice differentiable on the compact set $\mathbb{X} \times \mathbb{U}$. □

This condition is a natural generalization of the linear observability matrix rank condition since

$$\frac{\partial H}{\partial x}(x, U_k) = \mathbb{C}_N \tag{5.70}$$

for a linear system, and the full rank condition of $\mathbb{C}_n$ is completely equivalent to observability for $N \geq n$. A fundamental difference is that in the nonlinear case the rank of the matrix $\frac{\partial H}{\partial x}(x, U_k)$ depends on both the current state x and the current and past inputs U_k. This means that in the nonlinear case, successful asymptotic state estimation may depend on state and input trajectories, in strong contrast to the linear case where only the initial state influences the transient behavior of the observer (neglecting the influence of noise and disturbances in this discussion).

The role of the horizon parameter N can also be understood from the above discussion. While $N = n$ is generally sufficient for an estimate to be computable for observable linear systems, the benefits of choosing N larger is two-fold: The input data U_k may be N-exciting for a nonlinear system for sufficiently large N, but not for $N = n$, and second, a larger N will improve robustness to noise and uncertainty via a filtering effect. The possible disadvantages of choosing N very large are increased computational complexity and too much filtering leading to slow convergence of the estimates.

Define the N-information vector at time index k as

$$I_k = \text{col}(y_{k-N}, \ldots, y_k, u_{k-N}, \ldots, u_k).$$

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 $\mathbb{T} : \mathbb{X} \to \mathbb{D} \subseteq R^n$, where $\mathbb{D}$ is the convex hull of $\mathbb{T}(\mathbb{X})$:

$$d = \text{col}(\xi, z) = \mathbb{T}(x) \tag{5.71}$$

such that the following dynamics are equivalent to (5.63) for any initial condition in $\mathbb{X}$ and inputs in $\mathbb{U}$,

$$\xi_{k+1} = F_1(\xi_k, z_k, u_k) \tag{5.72a}$$
$$z_{k+1} = F_2(z_k, u_k) \tag{5.72b}$$
$$y_k = g(z_k, u_k). \tag{5.72c}$$

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 5.3. The system (5.63) 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 (5.72);
(2) the sub-system (5.72b)-(5.72c) is N-observable;
(3) the sub-system (5.72a) 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

$$||F_1(\xi_1, z, u) - F_1(\xi_2, z, u)||' \leq L_1 ||\xi_1 - \xi_2||'. \tag{5.73}$$

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 contraction assumption in part 3 of the definition is not very restrictive. For linear systems, it is equivalent to the conventional detectability definition.

Definition 5.4. The input U_k is said to be *N-exciting* for a strongly N-detectable system (5.63) at time index k if it is N-exciting for the sub-system (5.72b)-(5.72c) at time index k.

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 unacceptably 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.

5.3.1.2 Objective Function and Constraints

The topic of this section is to formulate the NMHE problem in terms of a nonlinear optimization problem that is convenient to solve using numerical opti-

mization. Defining $W_k = \text{col}(w_{k-N}..., w_{k-1}, w_k)$, and $V_k = \text{col}(v_{k-N}, ..., v_k)$ we introduce the following cost function similar to Rao et al (2003)

$$J'(x_{k-N}, ..., x_k, W_k, V_k) = \sum_{i=k-N}^{k} L(w_i, v_i) + Z_{k-N}(x_{k-N}) \tag{5.74}$$

where $L(w, v)$ is a stage cost typically of the least-squares type $L(w, v) = ||w||^2_M + ||v||^2_\Xi$ for some $M = M^T \succeq 0$ and $\Xi = \Xi^T \succeq 0$, there is a second term Z that we will discuss shortly, and the minimization must be performed subject to the model constraints

$$x_{i+1} = F(x_i, u_i, w_i) \tag{5.75}$$
$$y_i = h(x_i, u_i) + v_i \tag{5.76}$$

and the additional constraints resulting from (5.60)-(5.62)

$$\text{col}(x_i, w_i, v_i) \in X \times W \times V, \quad i = k - N, ..., k \tag{5.77}$$
$$C(x_i, w_i, v_i, t_i) \leq 0, \quad i = k - N, ..., k \tag{5.78}$$
$$\sum_{i=k-N}^{k} c(x_i, w_i, v_i) \leq 0 \tag{5.79}$$

It is straightforward to eliminate the variables v_i from this optimization problem, leading to

$$\begin{aligned} \Phi^*_{k-N} &= \min_{x_{k-N}, ...x_k, W_k} J(x_{k-N}, ..., x_k, W_k) \\ &= \sum_{i=k-N}^{k} L(w_i, y_i - h(x_i, u_i)) + Z_{k-N}(x_{k-N}) \end{aligned} \tag{5.80}$$

subject to

$$x_{i+1} = F(x_i, u_i, w_i), \quad i = k - N, ..., k \tag{5.81}$$
$$\text{col}(x_i, w_i, v_i) \in X \times W \times V, \quad i = k - N, ..., k \tag{5.82}$$
$$C(x_i, w_i, y_i - h(x_i, u_i), t_i) \leq 0, \quad i = k - N, ..., k \tag{5.83}$$
$$\sum_{i=k-N}^{k} c(x_i, w_i, y_i - h(x_i, u_i)) \leq 0 \tag{5.84}$$

By defining the solution function $\phi(i, U_k, x_{k-N})$ for $i \geq k - N$ using (5.81) recursively we can make further elimination of the nonlinear equality constraints (5.81) similar to the direct single shooting approach and re-define the cost function and constraints as follows:

$$\min_{x_{k-N},W_k} J(x_{k-N},W_k) = \sum_{i=k-N}^{k} L(w_i, y_i - h(\phi(i,U_k,x_{k-N}),u_i)) + Z_{k-N}(x_{k-N}) \tag{5.85}$$

subject to

$$\begin{aligned} \mathrm{col}(x_i,w_i,v_i) &\in X \times W \times V, \quad i = k-N,...,k \\ C(\phi(i,U_k,x_{k-N}),w_i,y_i - h(\phi(i,U_k,x_{k-N}),u_i),t_i) &\leq 0, \quad i = k-N,...,k \\ \sum_{i=k-N}^{k} c(\phi(i,U_k,x_{k-N}),w_i,y_i - h(\phi(i,U_k,x_{k-N}),u_i)) &\leq 0 \end{aligned} \tag{5.86}$$

The simple choice $Z(\cdot) = 0$ means that the state estimate is defined as the best least squares match with the data on the horizon. This means that no information from the data before the start of the horizon is used in the estimation, which is a clear weakness especially when the information content in the data is low due to lack of excitation, noise and other uncertainty. In other words, the estimation formulation contains no other mechanisms to introduce filtering of noise or regularization than to increase the horizon N, which also increases the computational complexity of the optimization problem and may still be insufficient.

In order to improve our ability to tune the NMHE and systematically introduce filtering of the state estimates, the term $Z(\cdot)$ in the formulation may be used as an arrival-cost estimate as discussed in e.g. Rao et al (2003) or in an ad hoc way to penalize deviation from an a priori estimate as in e.g Alessandri et al (2008); Sui and Johansen (2010); Alessandri et al (2003). Arrival cost estimation is discussed further in section 5.3.1.3, and a link between arrival cost estimation and the approach of Alessandri et al (2008) is illustrated in Poloni et al (2010).

We remark that the formulations make no particular assumptions on the uncertainty, and minimizes the impact of uncertainty on the estimates in a least-squares sense. Introduction of stochastic models can be envisioned and lead to better estimates in some cases, Lima and Rawlings (2010).

5.3.1.3 Arrival-cost Estimates

The term $Z(\cdot)$ in the cost function J defined in (5.80) may be used to make the finite (moving) window cost function J approximate the full (still finite) window cost (Rao et al (2003))

$$J''(x_{k-N},...,x_k,W_k) = \sum_{i=0}^{k} L(w_i, y_i - h(x_i,u_i)) + \Gamma(x_0) \tag{5.87}$$

such that

$$Z_{k-N}(x_{k-N}) \approx \sum_{i=0}^{k-N-1} L(w_i, y_i - h(x_i, u_i)) + \Gamma(x_0) \tag{5.88}$$

where $\Gamma(x_0)$ is such that $\Gamma(x_0) = 0$ for the a priori most likely estimate of x_0, and $\Gamma(x) \succ 0$ for other values. The motivation for more closely approximating the full window cost (as opposed to a moving window cost) is to capture as much information as possible from time index $i = 0, 1, ..., k - N - 1$. Using arguments of dynamic programming, Rao et al (2003), an exact arrival cost completely captures the information up to time index $k - N - 1$. This would lead to more accurate estimates through improved filtering.

The effect of the arrival cost can be understood by comparing the moving horizon approach to Extended Kalman Filtering (EKF); Gelb (2002). In an EKF the information in past data is summarized in the covariance matrix estimate. Under assumptions that include linearity of the system and the noise and disturbances being Gaussian white noise with known covariances that are reflected in a quadratic cost function, it is known that the Kalman filter is an optimal filter, Gelb (2002), that provides states estimates with minimum variance. An EKF is an approximate sub-optimal filter that allows for nonlinearities and makes certain simplifying computations such neglecting higher order statistics and higher order (nonlinear) terms. In a similar manner, the NMHE with an arrival cost estimate captures the information of data until the start of the window in the arrival cost. Unfortunately, it is hard to find an explicit representation of the arrival cost for nonlinear systems, and practical methods attempts to approximate the arrival cost. The use of covariance matrix estimates from EKF and similar ideas is a useful way to define the arrival cost, Rao et al (2003):

$$Z_k(x) = (x - \hat{x}_k)^T \Pi_k^{-1} (x - \hat{x}_k) + \Phi_k^* \tag{5.89}$$

The matrix Π_k is assumed to be non-singular such that its inverse is well defined, and obtained by solving the recursive Riccati-equation

$$\Pi_{k+1} = G_k Q_k G_k^T + A_k \Pi_k A_k^T - A_k \Pi_k C_k^T \left(R_k + C_k \Pi_k C_k^T\right)^{-1} C_k \Pi_k A_k^T$$

with some given positive definite matrix as initial condition Π_0. The matrices A_k, G_k, C_k are defined as linearizations about the NMHE estimated trajectory:

$$A_k = \frac{\partial F(\hat{x}_k, u_k, \hat{w}_k)}{\partial x} \tag{5.90}$$

$$G_k = \frac{\partial F(\hat{x}_k, u_k, \hat{w}_k)}{\partial w} \tag{5.91}$$

$$C_k = \frac{\partial h(\hat{x}_k, u_k)}{\partial x} \tag{5.92}$$

and for simplicity we assume Q_k and R_k are defined through a quadratic cost function $L(w,v) = w^T Q_k^{-1} w + v^T R_k^{-1} v$. More generally, Q_k and R_k may be defined as Hessians of L as in Rao et al (2003).

It is well known that alternative nonlinear Kalman Filters may perform better than the EKF in many situations. In the context of NMHE arrival cost estimation some useful methods are sample based filters (Ungarala (2009)), particle filtes (Lopez-Negrete et al (2009)), and Unscented Kalman Filtering (UKF) (Qu and Hahn (2009)).

5.3.1.4 Combined State and Parameter Estimation

Many practical estimation problems are characterized by both states and parameters being unknown or uncertain. In Kalman filtering (Gelb (2002)) and observer design, a common approach to the joint state and parameter estimation problem is to augment the state space with constant parameters. Assuming a vector of constant parameters θ^* appears in the model equations:

$$\xi_{i+1} = F_m(\xi_i, u_i, \omega_i, \theta^*) \tag{5.93}$$

$$y_i = h_m(\xi_i, u_i, \theta^*) + v_i \tag{5.94}$$

with the new notation where ξ_i is the state and ω_i is the disturbance. An *augmented* state space model assumes that the parameters are constant or slowly time-varying by the following model of the unknown parameter vector

$$\theta_{i+1} = \theta_i + \varrho_i \tag{5.95}$$

Combining (5.93)-(5.94) with (5.95) leads to

$$\begin{pmatrix} \xi_{i+1} \\ \theta_{i+1} \end{pmatrix} = \begin{pmatrix} F_m(\xi_i, u_i, \omega_i, \theta_i) \\ \theta_i + \varrho_i \end{pmatrix} \tag{5.96}$$

$$y_i = h_m(z_i, u_i, \theta_i) + v_i \tag{5.97}$$

With the augmented state $x = \mathrm{col}(\xi, \theta)$ and augmented disturbance vector $w = \mathrm{col}(\omega, \varrho)$ we observe that these equations are in the assumed form (5.75)-(5.76) such that the NMHE algorithm formulation can be applied without any modifications.

It is common to encounter combined state and parameter estimation problems where convergence conditions of uniform observability or persistence of excitation are not fulfilled, Moraal and Grizzle (1995a); Sui and Johansen (2010). In such cases various mechanisms of regularization should be implemented to get graceful degradation of the estimation when insufficient information is available to determine the estimates. The use of a term in the cost function that preserves the history and makes the observer degrade to an open-loop observer is one such mechanism, that can be combined with more

advanced monitoring of the Hessian matrix of the cost function to detect and resolve lack of excitation, Sui and Johansen (2010).

5.4 Numerical Optimization

For simplicity of notation, we assume in this section that the NMHE or NMPC problem is formulated as a general nonlinear programming problem at each time instant

$$\min_z V(z) \text{ subject to } G(z) \leq 0,\ H(z) = 0 \tag{5.98}$$

where z is a vector with the unknown decision variables. In practice, as implemented in most numerical solver software, it will be important to exploit structural properties of the constraints and objective functions such that further separation of the functions G and H into simple bounds ($z_{min} \leq z \leq z_{max}$), linear constraints and "truly" nonlinear constraints is usually made for efficient implementation. For simplicity of presentation, we does not make such separation here.

5.4.1 Problem Structure

The choice of numerical optimization solver strategy will have significant impact on both the need for computational resources and the quality of the solution in NMPC and NMHE. In this context, computational resources usually means the CPU time required for the solution to converge to meet the tolerance requirements, while quality of solution is related to lack of convergence or high sensitivity to initial guesses.

There are several features of NMPC and NMHE problems that are relevant to consider

- Formulation of the numerical optimal control or estimation problem, e.g. sequential or simultaneous approaches. The sequential approach leads to a smaller, denser problem with a computationally complex cost function usually without nonlinear equality constraints, while the simultaneous approach leads to a larger, more structured, sparse problem with nonlinear equality constrains and relatively simple cost and constraint functions to evaluate.
- NMPC and NMHE solves a sequence of numerical optimal control or estimation problems, where the parameters of the problem are usually subject to fairly small changes from one run to the next. There is usually benefits of warm starting the next optimization run using the solution and other internal data from the previous run as initial guesses, data or conditions.

- Since the optimization will be repeated at the next sample, and the optimization problem is formulated using uncertain data, it may not always be essential that the solver has converged (or equivalently that the tolerances may not need to be very strict) due to the forgiving effect of feedback. However, a feasible solution is generally required at each run in order to operate the control and monitoring systems. This means that problems tend to be re-formulated using slack variables with some prioritization of constraints that can be relaxed, and that is it generally desirable to start the next optimization run with a feasible initial guess generated from the previous run such that even with a limited number of iterations one can guarantee feasibility.
- Safety and reliability are essential features of most control and monitoring systems, which means that post-optimal analysis and checks on the quality of the solution must usually be implemented. Issues such as non-convexity and non-smoothness of models and constraints are essential to understand and take into account.

Although all nonlinear MPC and MHE problems have certain features in common, they may also differ considerably with respect to size, models, cost functions and constraints. This means that there will not be a single numerical method that will be the best, in general. Below, we briefly outline some commonly used numerical methods with emphasis on sequential quadratic programming and interior point methods. We point out that there exist a wide range of alternative methods that may perform better in certain types of problems, like derivative-free methods (e.g. Conn et al (2009)) that may be better suited if the computation of gradients is expensive or not possible to achieve accurately.

5.4.2 Nonlinear Programming

Newton's method for iterative solution of nonlinear algebraic equations is the backbone of most numerical optimization methods. For a nonlinear vector equation $f(z) = 0$, Newton's method starts with an initial guess vector z^0 and generates a sequence of guesses z^k indexed by the integer $k = 1, 2, 3, ...$ according to the following formula that results from linearization using Taylor's theorem and truncation:

$$f(z^k) + \nabla_z^T f(z^k)(z^{k+1} - z^k) = 0 \tag{5.99}$$

Eq. (5.99) defines a set of linear algebraic equations that can be solved for z^{k+1} using numerical linear algebra, which is the workhorse at the core of nonlinear programming and is the main contribution to computational complexity in addition to the computation of the function f and its gradient (Jacobian matrix) $\nabla_z f$. As Newton's method is based on linearization, it has

only local convergence, but with a quadratic convergence rate, Nocedal and Wright (1999).

Newton's method is used in nonlinear programming to solve nonlinear algebraic equations closely related to the first order optimality conditions of (5.98), known as the Karush-Kuhn-Tucker (KKT) conditions Nocedal and Wright (1999)

$$\nabla_z L(z^*, \lambda^*, \mu^*) = 0 \tag{5.100}$$
$$H(z^*) = 0 \tag{5.101}$$
$$G(z^*) \leq 0 \tag{5.102}$$
$$\mu^* \geq 0 \tag{5.103}$$
$$G_i(z^*)\mu_i^* = 0, \ \ i = 1, ..., n_G \tag{5.104}$$

where n_G is the number of inequality constraints and the Lagangian function is defined as

$$L(z, \lambda, \mu) = V(z) + \lambda^T H(z) + \mu^T G(z) \tag{5.105}$$

Obviously, the KKT conditions also involves inequalities which means that Newton's method cannot be applied directly. The different nonlinear programming methods differ conceptually in the way the KKT conditions, being mixed equations and inequalities, are used to formulate a sequence of nonlinear equations. The different nonlinear programming methods also differ with respect to approximations used for the gradient $\nabla_z f$ of the resulting set of equations. Since the evaluation of (5.100) already requires gradient computations (for the Jacobian matrix of the Lagrangian $\nabla_z L$) in the formulation of the equations to be solved, the computation of $\nabla_z f$ generally requires the expensive computation or approximation of the matrix $\nabla_z^2 L$, known as the Hessian matrix of the Lagrangian.

5.4.2.1 Sequential Quadratic Programming (SQP)

SQP methods linearize the KKT conditions (5.100)-(5.104) at the current iterate z^k, leading to a set of linear conditions that can be interpreted as the KKT conditions of the following quadratic program (QP), Nocedal and Wright (1999):

$$\min_z V_{QP}^k(z) \tag{5.106}$$

subject to

$$H(z^k) + \nabla_z^T H(z^k)(z - z^k) = 0 \tag{5.107}$$
$$G(z^k) + \nabla_z^T G(z^k)(z - z^k) \leq 0 \tag{5.108}$$

with the cost function

$$V_{QP}^k(z) = \nabla_z^T V(z^k)(z - z^k) + \frac{1}{2}(z - z^k)^T \nabla_z^2 L(z^k, \lambda^k, \mu^k)(z - z^k) \tag{5.109}$$

This QP interpretation is highly useful since it provides a practical way to deal with the fact that the KKT conditions include inequalities, which are not straightforward to solve using Newton's method directly. The vast knowledge and numerical methods of solving QP problems, typically using active set methods, Nocedal and Wright (1999); Gill et al (1981), is exploited at this point. Active set methods replace inequality constraints with equality constraints based on an active set assumption that is improved iteratively as the method converges towards an optimal solution.

However, there are three major challenges remaining:

- The first key challenge is related to the Hessian matrix $\nabla_z^2 L(\cdot)$. Problems arise if this matrix is not positive definite such that the QP is not convex and a global optimum may not exist or is not unique. In the context of NMPC or NMHE, problems will also arise if the computational complexity of computing the Hessian is beyond the CPU resources available. Approximations such as quasi-Newton and Gauss-Newton methods are commonly used to approximate the Hessian from the Jacobian, see below, in a positive definite form.
- The second key challenge is related related to the accuracy of the underlying linearizations (or equivalently, the local quadratic approximations of the QP to the NLP). In order to have control over this issue, it is common to solve the QP to generate a search direction only, and then generate the next iterate z^{k+1} not as the minimum of the QP defined above, but through a search procedure along this direction. Common search procedures are line search and trust region methods, as outlined below.
- The third key challenge is related to feasibility. To ensure convergence it is common to use a merit function to control the step size length in both line search and trust region methods. The merit function adds a penalty on constraint violations to the original cost function to ensure that the next iterate moves towards a combined objective of reducing the cost function and being feasible.

Quasi-Newton methods approximate the Hessian of the Lagrangian by an update formula that only requires computation of the Jacobian. Common methods, such as the BFGS update, Nocedal and Wright (1999), leads to significant computational reduction and ensures that the Hessian approximation is positive definite. The price to pay is that the convergence rate may no longer be quadratic, but typically only super-linear, Nocedal and Wright (1999).

Gauss-Newton methods are particularly useful for least-squares type of problems, like NMHE and certain NMPC formulations, where the cost function is the squared norm of some nonlinear functions since a reliable

estimate of the Hessian can be computed directly from the Jacobian as the product of the Jacobian and its transpose, Nocedal and Wright (1999).

Line search methods are designed to account for the fact that the QP is only a locally valid approximation. As the name indicates, one performs a one-dimensional search in the descent direction computed by the QP (solution) to ensure that sufficient descent of the actual merit function is achieved; Nocedal and Wright (1999).

Trust region methods define a maximum step length for the next iterate based on a trust region, where the linearization is sufficiently accurate. This aims to ensure that the next iterate is well defined and accurate, and the size of the trust region is adapted to ensure that the merit function reduction predicted by the QP is sufficiently close to the actual merit function reduction, Conn et al (2000); Wright and Tenny (2004).

5.4.2.2 Interior Point Methods (IP)

Interior point methods deal with the inequality constraints of the KKT conditions in a fundamentally different way than SQP methods. The KKT conditions concerning the inequality constraints, in particular (5.104), is replaced by a smooth approximation (Wright (1997); Diehl et al (2009)):

$$G_i(z^*)\mu_i^* = \tau, \ i = 1, ..., n_G \tag{5.110}$$

Solving the resulting set of algebraic nonlinear equations with Newton's methods is equivalent to a solution of the following appoximate problem, where the inequality constraints are handled by a $\log(\cdot)$ barrier function:

$$\min_z \left(V(z) - \tau \sum_{i=1}^{n_G} \log\left(-G_i(z)\right) \right) \text{ subject to } H(z) = 0 \tag{5.111}$$

The parameter $\tau > 0$ parameterizes a central path in the interior of the feasible region towards the optimum as $\tau \to 0$, which motivates the name of IP methods. Once the solution for a given $\tau > 0$ is found, the parameter τ can be reduced by some factor in the next Newton iteration. The practical implementation of an IP method will typically use Newton's method to compute a search direction. Challenges related to the computation of the Hessian matrix and limited validity of the linearization of the Newton method, remain similar to SQP, and the ideas of quasi-Newton methods, merit functions, line search and trust regions are relevant and useful also for IP methods.

5.4.2.3 Linear Algebra

At heart of both the QP sub-problems of SQP and the Newton-step of IP methods are the solution of a set of linear algebraic equations. Efficiency of the numerical optimization solver heavily depends on the efficiency of solving this problem, since it will be repeated many times towards the solution of the NLP at each sampling instant of an NMPC or NMHE. Exploiting structural properties is essential.

Depending on the solution strategy and properties of the problem, such structural properties are often related to positive definiteness of the Hessian (approximation), sparseness and block-diagonal structure of the linear systems of equations, and what information from the previous optimization run can be used to initialize the next run. Using factorization methods one may eliminate algebraic variables and operate in reduced spaces to save computations. Being able to efficiently maintain and update factorized matrices between the various iterations is usually essential to implement this. Although this is essential in any practical implementation of NMHE and NMPC, it is a fairly complex bag of tricks and tools that we consider outside the scope of this introduction. Instead, we refer to excellent and comprehensive treatments in Nocedal and Wright (1999); Diehl et al (2009); Gill et al (1997, 1981) and the references therein.

5.4.3 Warm Start

The NLP problem at one sampling instant is usually closely related to the NLP problem at the previous sampling instant in NMPC and NMHE problem, since the sampling interval is usually short compared to the dynamics of the plant and the controller. Assuming the reference signals and other input to the controller changes slowly, this means that the solution in terms of past state trajectories (for MHE problems) or future input and state trajectories (for MPC problems) can be time shifted one sampling period and still provide a reasonably accurate solution to the next NLP. Assuming no uncertainty in MPC problems, this is a perfectly valid assumption and is commonly used to guarantee feasibility at the next step in stability arguments, e.g Scokaert et al (1999); Mayne et al (2000). Even without time-shifting, the previous solution itself also provides a good initialization for warm start purposes in NMPC, Boch et al (1999); Diehl et al (2004).

Unlike SQP methods, IP methods can usually not make effective use of initial guesses of the solution due to the reformulation of the KKT conditions that follows the parameterized center path controlled by the parameter $\tau > 0$ that is sequentially reduced towards zero. This does not necessarily imply that IP methods are less suited for NMPC and NMHE problems, in particular for large scale problems where IP methods have advantages that may compensate

for this shortcoming. Modified IP methods that can efficiently incorporate warm start is a current research topic, Gondzio and Grothey (2008); Shahzad et al (2010).

Warm start is potentially most efficient when including data beyond just the solution point, but also consider the internal data of the optimization algorithm such as initial estimates of the Hessian approximation (in case exact Hessians are not computed), or initial estimates of factorizations of the Hessian (approximation), initial estimates of optimal active sets, and other data. This is in particular a challenge when the dimensions and structure of these internal data will change from one sample to the next. This may for example be the case in the simultaneous formulations (in particular direct collocation) of numerical optimal control (see section 5.2.2), since the discretization may be changed from one sample to the next, in general. One must also have in mind that simultaneous formulations require that both state and control trajectories are initialized, while sequential formulations only require the control trajectory initialization. What is most beneficial will depend on the accuracy of the available information for initialization, amongst other things. We refer to Diehl et al (2009); Houska et al (2010) and the references therein for a deeper treatment of this topic.

5.4.4 Computation of Jacobians and Hessians

The computation of the Jacobians of the cost and constraint functions is often the main computational cost of numerical optimization methods, and even fairly small inaccuracies in the calculation of the Jacobians due to may lead to severe convergence problems.

Simultaneous approaches offer advantages over sequential approaches with respect to Jacobian computations:

- The prediction horizon is broken up into several intervals where ODE solutions are computed from given initial conditions. Since these intervals will be shorter than the single interval of a single shooting approach, numerical errors due to the ODE solver tend to accumulate less.
- Implicit ODE solvers, which generally have more stable numerical properties than explicit solvers, can in general be used in simultaneous approach.
- Simultaneous approaches are characterized by simpler cost and constraint functions, where automatic differentiation is more easily exploited to avoid numerical Jacobian computation errors, see section 5.4.4.2.

The numerical challenges are in particular important to consider for plants that are unstable or marginally stable. Like in linear MPC, there may be advantages of pre-stabilizing an open-loop unstable plant model with a feedback compensator before used in NMPC or NMHE, Cannon and Kouvaritakis (2005); Sui et al (2010).

5.4.4.1 Finite Difference

The finite difference method approximates the (i, j)-th element of the Jacobian of a vector function $f(z)$ as

$$(\nabla_z f(z))_{i,j} \approx \frac{f_i(z_j + \delta) - f_i(z_j)}{\delta} \tag{5.112}$$

for some small $\delta > 0$. If δ is too large there will be inaccuracies due to the nonlinearity of f_i, since the method computes the average slope between two points. If the two points are not infinitely close and the function is not linear, there will be a "nonlinearity error". If δ is too small, any finite numerical error ε_1 in the computation of $f_i(z_j+\delta)$ and ε_2 in the computation of $f_i(z_j)$ will lead to an error $\epsilon = (\varepsilon_1 - \varepsilon_2)/\delta$ in the computation of the derivative. Obviously, this error goes to infinity when $\delta \to 0$, so a tradeoff between these errors must be made. It should be noticed that the finite difference approximation error ϵ depends on the difference between the errors in the two point-wise evaluations of f_i. This means that systematic errors (i.e. the same error in both ε_1 and ε_2) will have a much smaller effect than a random error of the same magnitude. Practical experience shows that the use of variable-step (adaptive) ODE solvers tend to give a small random numerical error, while the use of fixed-step ODE solvers tend to give a larger systematic error, but even smaller random error. For the reasons described above, one may find that a fixed-step ODE solver leads to considerably smaller error in finite difference Jacobian computations and performs better with less convergence problems in many numerical methods for NMPC and NMHE.

It is also worthwhile to remind the reader that scaling of all variables involved in the optimization problem to the same order of magnitude is in many cases a pre-requisite for numerical nonlinear optimization methods to work satisfactorily. This is evident in the context of finite difference Jacobian computations, but also relevant for other numeric computations.

As a final remark, it is possible to exploit square-root factorizations (like Cholesky factorization) for improved numerical accuracy and computational complexity in finite difference computations, Schei (1997).

5.4.4.2 Symbolic and Automatic Differentiation

The most accurate result and computationally most efficient approach is to calculate gradients by symbolically differentiating the cost and constraint functions. Doing this by hand, or even using symbolic computations in Matlab, Maple or Mathematica, may easily become intractable for NMPC and NMHE problems that may contain a large number of variables, equations and inequalities. A more convenient solution is to rely on so-called *automatic differentiation* software (Griewank and Walther (2008)) that achieved this objective either by overlaying operators in object oriented languages such

as C++ (Griewank et al (1996)), or automatically generates source code for gradient functions based on source code of the original function, Bischof et al (1996).

Acknowledgements The work 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.

References

Abraham R, Marsden JE, Ratiu T (1983) Manifolds, Tensor Analysis, and Applications. Springer–Verlag New York

Adetola V, Guay M (2003) Nonlinear output feedback receding horizon control. In: Proc. ACC, Denver

Åksesson BM, Toivonen HT (2006) A neural network model predictive controller. Journal of Process Control 16:937–946

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

Alamir M (2007) Nonlinear moving horizon observers: Theory and real-time implementation. In: Besancon G (ed) Nonlinear Observers and Applications, LNCIS 363, Springer, pp 139–179

Alamir M, Bornard G (1995) Stability of a truncated infinite constrained receding horizon scheme: The general discrete nonlinear case. Automatica 31:1353–1356

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

Alessandri A, Baglietto M, Battistelli G (2003) Receding–horizon estimation for discrete–time linear systems. IEEE Transactions Automatic Control 48:473–478

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

Allgöwer F, Badgwell TA, Qin JS, Rawlings JB, Wright SJ (1999) Nonlinear predictive control and moving horizon estimation – An introductory overview. In: Frank PM (ed) Advances in Control, Highlights of ECC-99, Springer, pp 391–449

Angeli D, Casavola A, Mosca E (2000) Constrained predictive control of nonlinear plants via polytopic linear system embedding. Int J Robust Nonlinear Control 10:1091 – 1103

Arellano-Garcia H, Wendt M, Barz T, Wozny G (2007) Closed-loop stochastic dynamic optimization under probabilistic output constraints. In: Findeisen R, Allgöwer F, Biegler LT (eds) Assessment and future directions of nonlinear model predictive control, LNCIS, vol. 358, Springer-Verlag, pp 305–315

Athans M, Falb PL (1966) Optimal Control. An Introduction to the Theory and Its Applications. McGraw–Hill Ltd.

Balchen JG, Ljungquist D, Strand S (1992) State–space predictive control. Chemical Engineering Science 47:787–807

Bellman R (1957) Dynamic Programming. Princeton University Press, New Jersey

Bemporad A, Filippi C (2003) Suboptimal explicit RHC via approximate multiparametric quadratic programming. J Optimization Theory and Applications 117:9–38

Bemporad A, Morari M (1999) Control of systems integrating logic, dynamics, and constraints. Automatica 35:407–427

Bemporad A, Borrelli F, Morari M (2000) Optimal controllers for hybrid systems: Stability and piecewise linear explicit form. In: Proc. Conference on Decision and Control, Sydney

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

Betts JT (2001) Practical methods for optimal control using nonlinear programming. SIAM, Philadelphia

Biegler L (2000) Efficient solution of dynamic optimization and NMPC problems. In: Allgöwer F, Zheng A (eds) Nonlinear Predictive Control, Birkhäuser, pp 219 – 244

Biegler LT (1984) Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Computers and Chemical Engineering 8 pp 243–248

Bischof C, Carle A, Kadhemi P, Mauer A (1996) ADIFOR2.0: Automatic differentiation of Fortran 77 programs. IEEE Computational Science and Engineering 3:18–32

Biyik E, 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 PJ, Christophersen N, Størdal JM (2001) Monte Carlo filters for non–linear state estimation. Automatica 37:177–183

Boch HG, Diehl M, Leineweber DB, Schlöder JP (1999) Efficient direct multiple shooting in nonlinear model predictive control. In: Keil F, Mackens W, Voss H, Werther J (eds) Scientific computing in chemical engineering, vol 2, Springer, Berlin

Bock HG, Plitt KJ (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings 9th IFAC World Congress, Budapest, Pergamon Press, Oxford, pp 243–247

Bock HG, Diehl M, Leineweber DB, Schlöder JP (1999) Efficient direct multiple shooting in nonlinear model predictive control. In: Keil F, Mackens W, Voß H, Werther J (eds) Scientific Computing in Chemical Engineering II, vol 2, Springer, Berlin, pp 218–227

Borrelli F, Morari M (2007) Offset free model predictive control. In: Proc. IEEE Conference on Decision and Control, pp 1245–1250

Boyd S, Ghaoui LE, Feron E, Balahrishnan V (1994) Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia

Cannon M, Kouvaritakis B (2005) Optimizing prediction dynamics for robust mpc. IEEE Trans Automatic Control 50:1892–1897

Cannon M, Ng D, Kouvaritakis B (2009) Successive linearization NMPC for a class of stochastic nonlinear systems. In: Magni L, Raimondo DM, Allgöwer F (eds) Nonlinear Model Predictive Control: Towards New Challenging Applications, LNCIS, vol. 384, Berlin/Heidelberg: Springer-Verlag, pp 249–262

Casavola A, Famularo D, Franze G (2003) Predictive control of constrained nonlinear systems via LPV linear embeddings. Int J Robust Nonlinear Control 13:281 – 294

Cervantes A, Biegler L (1998) Large–scale DAE optimization using a simultaneous nlp formulation. AIChE Journal 44:1038–1050

Chen CC, Shaw L (1982) On receding horizon feedback control. Automatica 18:349–352

Chen H, Allgöwer F (1998) A quasi–infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34:1205–1217

Conn A, Scheinberg K, Vicente LN (2009) Introduction to Derivative-Free Optimization. SIAM

Conn AR, Gould NIM, Toint PL (2000) Trust region methods. SIAM

Deuflhard P (1974) A modified Newton method for the solution of ill–conditioned systems of nonlinear equations with application to multiple shooting. Numer Math 22:289–315

Diehl M, Bock HG, Schlöder JP, Findeisen R, Nagy Z, Allgöwer F (2002) Real–time optimization and nonlinear model predictive control of processes governed by differential–algebraic equations. J Process Control 12:577–585

Diehl M, Magni L, de Nicolao G (2004) Online NMPC of unstable periodic systems using approximate infinite horizon closed loop costing. Annual Reviews in Control 28:37–45

Diehl M, Ferreau HJ, Haverbeke N (2009) Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: et al LM (ed) Nonlinear Model Predictive Control, LNCIS 384, Springer, pp 391–417

Dunbar WB, Murray RM (2006) Distributed receding horizon control for multi-vehicle formation stabilization. Automatica 42:549–558

Feng L, Gutvik CR, Johansen TA, Sui D (2009) Barrier function nonlinear optimization for optimal decompression of divers. In: IEEE Conf. Decision and Control, Shanghai

Fiacco AV (1983) Introduction to sensitivity and stability analysis in nonlinear programming. Orlando, Fl: Academic Press

Findeisen R, Imsland L, Allgöwer F, Foss BA (2003a) Output feedback stabilization for constrained systems with nonlinear model predictive control. Int J Robust and Nonlinear Control 13:211–227

Findeisen R, Imsland L, Allgöwer F, Foss BA (2003b) State and output feedback nonlinear model predictive control: An overview. European J Control 9:179–195

Foss BA, Schei TS (2007) Putting nonlinear model predictive control into use. In: Assessment and Future Directions Nonlinear Model Predictive Control, LNCIS 358, Springer Verlag, pp 407 – 417

Foss BA, Johansen TA, Sørensen AV (1995) Nonlinear predictive control using local models – applied to a batch fermentation process. Control Engineering Practice 3:389–396

Gelb A (2002) Applied Optimal Estimation, 17th edn. MIT Press

Gill P, Murray W, Wright M (1981) Practical optimization. Academic Press, Inc.

Gill P, Barclay A, Rosen JB (1997) SQP methods and their application to numerical optimal control. Tech. Rep. NA 97–3, Department of Mathematics, University of California, San Diego

Glad ST (1983) Observability and nonlinear dead beat observers. In: IEEE Conf. Decision and Control, San Antonio, pp 800–802

Gondzio J, Grothey A (2008) A new unblocking technique to warmstart interior point methods based on sensitivity analysis. SIAM J Optimization 19:1184–1210

Goodwin GC, De Dona JA, Seron MM, Zhuo XW (2005) Lagrangian duality between constrained estimation and control. Automatica 41:935–944

Goodwin GC, Østergaard J, Quevedo DE, Feuer A (2009) A vector quantization approach to scenario generation for stochastic NMPC. In: Magni L, Raimondo DM, Allgöwer F (eds) Nonlinear Model Predictive Control: Towards New Challenging Applications, LNCIS, vol. 384, Berlin/Heidelberg: Springer-Verlag, pp 235–248

Gravdahl JT, Egeland O (1997) Compressor surge control using a close–coupled valve and backstepping. In: Proc. American Control Conference, Albuquerque, NM., vol 2, pp 982 –986

Greitzer EM (1976) Surge and rotating stall in axial flow compressors, part i: Theoretical compression system model. J Engineering for Power 98:190–198

Griewank A, Walther A (2008) Evaluating Derivatives, second edition. SIAM

Griewank A, Juedes D, Utke J (1996) ADOL–C, A package for the automatic differentiation of algorithms written in C/C++. ACM Trans Mathematical Software 22:131–167

Grimm G, Messina MJ, Tuna SE, Teel AR (2004) Examples when nonlinear model predictive control is nonrobust. Automatica 40:1729–1738

Haseltine EL, Rawlings JB (2005) Critical evaluation of extended Kalman filtering and moving–horizon estimation. Ind Eng Chem Res 44:2451–2460
Heemels WPMH, Schutter BD, Bemporad A (2001) Equivalence of hybrid dynamical models. Automatica 37:1085 – 1091
Hicks GA, Ray WH (1971) Approximation methods for optimal control systems. Can J Chem Engng 49:522–528
Houska B, Ferreau HJ, Diehl M (2010) ACADO toolkit – an open–source framework for automatic control and dynamic optimization. Optimal Control Applications and Methods
Isidori A (1989) Nonlinear Control Systems, 2nd Ed. Springer Verlag, Berlin
Jadbabaie A, Yu J, Hauser J (2001) Unconstrained receding–horizon control of nonlinear systems. IEEE Trans Automatic Control 46:776–783
Jang SS, Joseph B, Mukai H (1986) Comparison of two appraoches to on-linear parameter and state estimation of nonlinear systems. Ind Chem Proc Des Dev 25:809–814
Jazwinski AH (1968) Limited memory optimal filtering. IEEE Trans Automatic Control 13:558–563
Johansen TA (2004) Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40:293–300
Kandepu R, Foss B, Imsland L (2008) Applying the unscented Kalman filter for nonlinear state estimation. J Process Control 18:753–768
Kantas N, Maciejowski JM, Lecchini-Visintini A (2009) Sequential Monte Carlo for model predictive control. In: Magni L, Raimondo DM, Allgöwer F (eds) Nonlinear Model Predictive Control: Towards New Challenging Applications, LNCIS, vol. 384, Berlin/Heidelberg: Springer-Verlag, pp 263–274
Keerthi SS, Gilbert EG (1988) Optimal infinite horizon feedback laws for a general class of constrained discrete–time systems: Stability and moving horizon approximations. J Optimization Theory and Applications 57:265–293
Kerrigan E, Maciejowski JM (2000) Invariant sets for constrained nonlinear discrete–time systems with application to feasibility in model predictive control. In: Proc. IEEE Conf. Decision and Control, Sydney
Keviczky T, Borrelli F, Balas GJ (2006) Decentralized receding horizon control for large scale dynamically decoupled systems. Automatica 42:2105 – 2115
Kim I, Liebman M, Edgar T (1991) A sequential error-in-variables method for nonlinear dynamic systems. Comp Chem Engr 15:663–670
Kojima M (1980) Strongly stable stationary solutions in nonlinear programs. In: Robinson SM (ed) Analysis and Computation of Fixed Points, Academic Press, New York, pp 93–138
Kolås S, Foss B, Schei TS (2008) State estimation IS the real challenge in NMPC. In: Int. Workshop on Assessment and Future Directions of NMPC, Pavia, Italy
Kraft D (1985) On converting optimal control problems into nonlinear programming problems. In: Schittkowski K (ed) Computational Mathematical Programming, vol F15, NATO ASI Series, Springer–Verlag, pp 261–280
Krstic M, Kanellakopoulos I, Kokotovic P (1995) Nonlinear and Adaptive Control Design. Wiley and Sons
Lazar M, Heemels W, Bemporad A, Weiland S (2007) Discrete-time non-smooth nonlinear mpc: Stability and robustness. In: Findeisen R, Allgöwer F, Biegler LT (eds) Assessment and Future Directions of Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences, Vol. 358, Springer Berlin / Heidelberg, pp 93 – 103
Lazar M, Heemels WPMH, Roset BJP, Nijmeijer H, van den Bosch PPJ (2008) Input-to-state stabilizing sub-optimal NMPC with an application to DC-DC converters. International Journal of Robust and Nonlinear Control 18:890 – 904

Leineweber DB, Bauer I, Boch HG, Schlöder JP (2003) An efficient multiple shooting based reduced SQP strategy for large–scale dynamic process optimization. part i: Theoretical aspects. Comp Chem Eng 27:157–166

Li WC, Biegler LT (1989) Multistep Newton-type control strategies for constrained nonlinear processes. Chem Eng Res Des 67:562–577

Li WC, Biegler LT (1990) Newton-type controllers for constrained nonlinear processes with uncertainty. Ind Engr Chem Res 29:1647–1657

Lima FV, Rawlings JB (2010) Nonlinear stochastic modeling to improve state estimation in process monitoring and control. AIChE Journal

Limon D, Alamo T, Camacho EF (2002) Input–to–state stable mpc for constrained discrete–time nonlinear systems with bounded additive uncertainties. In: Proc. IEEE Conference on Decision and Control, Las Vegas, NV, pp 4619–4624

Limon D, Alamo T, Salas F, Camacho EF (2006) Input–to–state stability of min–max MPC controllers for nonlinear systems with bounded uncertainties. Automatica 42:797–803

Lopez-Negrete R, Patwardhan SC, Biegler LT (2009) Approximation of arrival cost in moving horizon estimation using a constrained particle filter. In: 10th Int. Symposium Process Systems Engineering, pp 1299–1304

Magni L, Scattolini R (2006) Stabilizing decentralized model predictive control of nonlinear systems. Automatica 42:1231–1236

Magni L, Scattolini R (2007) Robustness and robust design of MPC for nonlinear discrete–time systems. In: Findeisen R, Allgöwer F, Biegler L (eds) Assessment and Future Directions of Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences, vol 358, Springer–Verlag, pp 239–254

Magni L, Nicolao GD, Scattolini R (2001a) Output feedback and tracking of nonlinear systems with model predictive control. Automatica 37:1601–1607

Magni L, Nicolao GD, Scattolini R (2001b) A stabilizing model-based predicitve control algorithm for nonlinear systems. Automatica 37:1351–1362

Magni L, De Nicolao G, Scattolini R, Allgöwer F (2003) Robust model predictive control for nonlinear discrete–time systems. International Journal of Robust and Nonlinear Control 13:229–246

Mangasarian OL, Rosen JB (1964) Inequalities for stochastic nonlinear programming problems. Operations Research 12:143–154

Marino R, Tomei P (1995) Nonlinear Control Design: Geometric, Adaptive and Robust. Prentice Hall, UK

Mayne DQ, Michalska H (1990) Receding horizon control of nonlinear systems. IEEE Trans Automatic Control 35:814–824

Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: Stability and optimality. Automatica 36:789–814

Messina MJ, Tuna SE, Teel AR (2005) Discrete-time certainty equivalence output feedback: allowing discontinuous control laws including those from model predictive control. Automatica 41:617–628

Michalska H, Mayne DQ (1993) Robust receding horizon control of constrained nonlinear systems. IEEE Trans Automatic Control 38:1623–1633

Michalska H, Mayne DQ (1995) Moving horizon observers and observer–based control. IEEE Transactions Automatic Control 40:995–1006

Moraal PE, Grizzle JW (1995a) Asymptotic observers for detectable and poorly observable systems. In: IEEE Conf. Decision and Control, New Orleans, pp 109–114

Moraal PE, Grizzle JW (1995b) Observer design for nonlinear systems with discrete–time measurement. IEEE Transactions Automatic Control 40:395–404

Morari M, Lee J (1999) Model predicitve control: Past, present and future. Comp and Chem Eng 23:667–682

Muske KR, Badgwell TA (2002) Disturbance modeling for offset-free linear model predictive control. J Process Control 12:617 – 632

Nagy Z, Braatz RD (2003) Robust nonlinear model predictive control of batch processes. AIChE Journal 49:1776–1786

Nagy Z, Mahn B, Franke R, Allgöwer F (2007) Real-time implementation of nonlinear model predictive control of batch processes in an industrial framework. In: Findeisen R, Allgöwer F, Biegler LT (eds) Assessment and Future Directions of Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences Vol. 358, Springer Berlin / Heidelberg, pp 465 – 472

Nicolao GD, Magni L, Scattolini R (2000) Stability and robustness of nonlinear receding horizon control. In: Allgöwer F, Zheng A (eds) Nonlinear Predictive Control, Birkhäuser, pp 3–23

Nijmeijer H, van der Schaft AJ (1990) Nonlinear Dynamical Control Systems. Springer–Verlag, New York

Nørgaard M, Ravn O, Poulsen NK, Hansen LK (2000) Neural Networks for Modelling and Control of Dynamic Systems. Springer, London

Nocedal J, Wright SJ (1999) Numerical Optimization. Springer–Verlag, New York

Ohtsuka T (2004) A continuation/GMRES method for fast computation of nonlinear receding horizon control. Automatica 40:563–574

Pannocchia G, Bemporad A (2007) Combined design of disturbance model and observer for offset-free model predictive control. IEEE Trans Automatic Control 52:1048–1053

Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model-predictive control. AIChE J 49:426 – 437

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

Proakis JG, Manolakis DG (1996) Digital Signal Processing. Prentice Hall

Qin SJ, Badgwell TA (1996) An overview of industrial model predictive control technology, preprint CPC–V, Lake Tahoe

Qin SJ, Badgwell TA (2000) An overview of nonlinear model predictive control applications. In: Nonlinear Predictive Control, Springer–Verlag, pp 369—-392

Qu CC, Hahn J (2009) Computation of arrival cost for moving horizon estimation via unscented Kalman filtering. J Process Control 19:358–363

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

Ramamurthi Y, Sistu P, Bequette BW (1993) Control-relevant data reconciliation and parameter estimation. Comp Chem Engr 17:41–59

Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete–time systems: Stability and moving horizon approximation. IEEE Transactions Automatic Control 48:246–258

Rawlings JB (2000) Tutorial overview of model predictive control. IEEE Contr Syst Magazine 20(3):38–52

Rawlings JB, Bakshi BR (2006) Particle filtering and moving horizon estimation. Computers and Chemical Engineering 30:1529–1541

Roset BJP, Lazar M, Heemels WPMH, Nijmeijer H (2006) A stabilizing output based nonlinear model predictive control scheme. In: Proceedings of 45th IEEE Conference on Decision and Control, San Diego, USA, pp 4627–4632

Saberi A, Han J, Stoorvogel AA (2002) Constrained stabilization problems for linear plants. Automatica 38:639–654

Sargent RWH, Sullivan GR (1977) The development of an efficient optimal control package. In: Proceedings of the 8th IFIP Conference on Optimization Techniques, Springer, Heidelberg

Scattolini R (2009) Architectures for distributed and hierarchical model predictive control – a review. Journal of Process Control 19:723–731
Schei TS (1997) A finite-difference method for linearization in nonlinear estimation algorithms. Automatica 33:2053–2058
Scokaert POM, Mayne DQ, Rawlings JB (1999) Suboptimal model predictive control (feasibility implies stability). IEEE Trans Automatic Control 44:648–654
Sepulchre R, Jankovic M, Kokotovic P (1997) Constructive nonlinear control. Springer–Verlag, London
Sequeira SE, Graells M, Luis P (2002) Real-time evolution for on-line optimization of continuous processes. Industrial and Engineering Chemistry Research pp 1815–1825
Shahzad A, Kerrigan EC, Constantinides GA (2010) A warm-start interior-point method for predictive control. In: Proc. UKACC
Shin KG, McKay ND (1985) Minimum-time control of robotic manipulators with geometric path constraints. IEEE Transactions on Automatic Control 30:531–541
von Stryk O (1993) Numerical solution of optimal control problems by direct collocation. In: in Optimal Control, (International Series in Numerical Mathematics 111), pp 129–143
Sui D, Johansen TA (2010) Regularized nonlinear moving horizon observer for detectable systems. In: IFAC Nonlinear Control Symposium, Bologna, Italy
Sui D, Johansen TA, Feng L (2010) Linear moving horizon estimation with pre-estimating observer. IEEE Trans Automatic Control 55:2363 – 2368
Tenny MJ, Rawlings JB, Wright SJ (2004) Closed-loop behavior of nonlinear model-predictive control. AIChE Journal 50:2142–2154
Tikhonov AN, Arsenin VY (1977) Solutions of Ill–posed Problems. Wiley
Tjoa IB, Biegler LT (1991) Simultaneous strategies for data reconciliation and gross error detection for nonlinear systems. Comp Chem Engr 15:679–690
Tsang TH, Himmelblau DM, Edgar TF (1975) Optimal control via collocation and non–linear programming. Int J Control 21:763–768
Tyler ML, Morari M (1999) Propositional logic in control and monitoring problems. Automatica 35:565–582
Ungarala S (2009) Computing arrival cost parameters in moving horizon estimation using sampling based filters. J Process Control 19:1576–1588
Vada J, Slupphaug O, Foss BA (1999) Infeasibility handling in linear MPC subject to prioritized constraints. In: IFAC World Congress, Beijing, vol D, pp 163–168
Wan ZY, Kothare MV (2004) Efficient scheduled stabilizing output feedback model predictive control for constrained nonlinear systems. IEEE Trans Automatic Control 49:1172 – 1177
Williams HP (1999) Model Building in Mathematical Programming
Wright SJ (1997) Primal-dual interior-point methods. SIAM, Philadelphia
Wright SJ, Tenny MJ (2004) A feasible trust-region sequential quadratic programming algorithm. SIAM Journal on Optimization 14:1074–1105
Xiong Q, Jutan A (2003) Continuous optimization using a dynamic simplex method. Chemical Engineering Science pp 3817–2828
Zavala VM, Biegler LT (2009) The advanced step NMPC controller: Optimality, stability and robustness. Automatica 45:86–93
Zavala VM, Laird CD, Biegler LT (2008) A fast moving horizon estimation algorithm based on nonlinear programming sensitivity. Journal of Process Control 18:876–884
Zimmer G (1994) State observation by on–line minimization. Int J Control 60:595–606

Chapter 6
Complexity Reduction in Explicit Model Predictive Control

Michal Kvasnica and Miroslav Fikar and Ľuboš Čirka and Martin Herceg

Abstract This chapter discusses recent advances in model predictive control (MPC) and treats issues and challenges in real-time implementation. We investigate the explicit approach to MPC. The idea of explicit MPC is to find the optimal control input as an explicit function of the initial conditions. Such a function is known to take the Piecewise Affine (PWA) form, and allows MPC to be applied to systems with fast dynamics. For most practical cases, however, the function is often too complex to be processed by a typical control hardware setup in real time. Therefore, two novel methods are proposed which aim at deriving a simpler representation of the optimal MPC feedback law. Both methods provide guarantees of closed-loop stability and constraint satisfaction and are able to reduce the real-time complexity by several orders of magnitude.

6.1 Introduction

Real-time implementation of MPC in the Receding Horizon fashion (RHMPC) is a challenging task since it requires that the optimal solution of an optimi-

Michal Kvasnica
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, e-mail: michal.kvasnica@stuba.sk

Miroslav Fikar
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, e-mail: miroslav.fikar@stuba.sk

Ľuboš Čirka
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, e-mail: lubos.cirka@stuba.sk

Martin Herceg
Swiss Federal Institute of Technology, Zurich, e-mail: herceg@control.ee.ethz.ch

sation problem for a given initial condition is obtained within of one sampling instance. Recently, the concept of *parametric programming* has been adopted to pre-compute the optimal solution for all possible initial conditions $\boldsymbol{x}$ as a Piecewise Affine (PWA) function $\boldsymbol{u}^*(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ (Bemporad et al, 2002; Borrelli, 2003; Kvasnica, 2009). Such a function consists of a set of polyhedral regions with the optimal solution being affine on each region. This allows one to apply RHMPC to systems with fast dynamics. Since the optimal solution is explicitly obtained in a functional form, such an approach is often referred to as *explicit RHMPC*.

Complexity of the real-time implementation of such solutions is determined by the amount of memory needed to describe the function and by the amount of CPU time needed to evaluate it for a particular value of $\boldsymbol{x}$. In the simplest form, the explicit solution can viewed as a table with rows representing individual regions. The table size depends exponentially on the number of constraints and on the number of binary variables. As both memory and CPU time grow proportionally with the table size, for a successful real-time implementation it is of imminent importance to keep the complexity of $\boldsymbol{\kappa}(\boldsymbol{x})$ (expressed in terms of number of its regions) as low as possible.

In the existing literature, the issue of complexity of $\boldsymbol{\kappa}(\boldsymbol{x})$ is usually attacked from various perspectives. The most simple approaches reduce the prediction horizons or consider move blocking of inputs.

One option is to approximate the optimal feedback $\boldsymbol{\kappa}(\boldsymbol{x})$ by a simpler function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ either by solving a sub-optimal MPC problem. In Bemporad and Filippi (2003), relaxed Karush-Kuhn-Tucker (KKT) conditions are assumed. Partition of state-space to orthogonal hypercubes was considered in Johansen and Grancharova (2003) and recursive procedure is employed to obtain desired accuracy. Jones and Morari (2009)) use bilevel optimisation to generate a low complexity PWA function directly from the MPC formulation. Optimal control is interpolated over a small number of states. In Rossiter and Grieder (2005), two different control laws from the feasible region boundary are interpolated and achieve a large decrease in the number of regions with possible performance degradation. Laguerre functions are used in Valencia-Palomo and Rossiter (2010) for reparametrisation of degrees of freedom and give significant reduction of complexity with a little loss in performance.

Other possibility is to augment the underlying regions of $\boldsymbol{\kappa}(\boldsymbol{x})$ (see e.g. Johansen and Grancharova (2003); Cychowski and O'Mahony (2005); Grieder et al (2004); Scibilia et al (2009)). In all cases a sub-optimal replacement $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is obtained. However, a direct guarantee of actual complexity reduction, or impact on closed-loop stability and performance are usually not provided.

Recursive formulations of simple MPC formulations with one step prediction and varying terminal set obtained in previous iteration are considered in Grieder et al (2005). Simplicity of MPC formulation here usually leads to a significant reduction of the number of the regions.

Some of the solutions avoid to store the full table and keep only some selected regions. Pannocchia et al (2007) propose partial enumeration of active

constraints at optimality. Several off-line simulations are needed to identify the most important combinations of active constraints.

Another option is to post-process $\boldsymbol{\kappa}(\boldsymbol{x})$ in order to obtain a simpler representation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ by merging together regions which share the same expression for the control law (Geyer et al, 2008). Such an approach, however, is computationally very demanding and thus limited to small-scale problems only. The same paper discusses also suboptimal strategy based on divide&conquer approach that can handle larger problems.

The main aim of this chapter is to provide methods and implementations that reduce execution time needed for evaluate the explicit feedback law $\boldsymbol{\kappa}(\boldsymbol{x})$ on-line for a particular value of $\boldsymbol{x}$. We will concentrate on two approaches that post-process the optimal solution: (i) significant reduction of number of regions by clipping, and (ii) approximation of optimal piecewise affine control law by a polynomial. Another possible solution based on the concept of separation function is proposed in the workshop preprints (Kvasnica et al, 2011)

The first approach constructs a replacement function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ for the optimal control law $\boldsymbol{\kappa}(\boldsymbol{x})$ which is guaranteed to contain less regions than the original one for a vast majority of MPC setups. In addition, as will be illustrated in Section 6.4, such a replacement maintains all closed-loop properties of $\boldsymbol{\kappa}(\boldsymbol{x})$ and therefore does not induce any loss of optimality or stability. The approach is based on the premise that MPC controller operates at the limits of the admissible control freedom for some states. The idea therefore is to extend the unsaturated regions such that they cover the saturated ones. In the next step we propose to pass the calculated function value through a so-called clipping function such that the equivalence to the original function is established for all feasible initial conditions.

The second method approximates the optimal control law defined within multiple state space regions by a higher degree polynomial valid over the entire available state-space boundaries. This polynomial, when applied as a state-feedback, guarantees closed-loop stability, constraint satisfaction, and a bounded performance decay. The advantage of the proposed scheme lies in faster controller evaluation and lower storage demand compared to currently available techniques. As it will be shown, such a polynomial can be constructed by solving a linear programming problem.

6.2 Notation

We denote by $\mathbb{R}^n$ the n-dimensional real vectors and by $\mathbb{R}^{n\times m}$ the $n \times m$-dimensional real matrices. For a matrix or a vector $\boldsymbol{A}$, $[\boldsymbol{A}]_{\setminus\mathcal{I}}$ represents all rows of $\boldsymbol{A}$ except of those belonging to some index set $\mathcal{I}$. The interior of a set $\mathcal{S}$ is denoted by $\text{int}\,(\mathcal{S})$. Given a function $\boldsymbol{\kappa}(\boldsymbol{x})$, $\text{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$ denotes its domain.

Definition 6.1 (Polyhedron). A polyhedron is the convex intersection of c closed affine half-spaces, i.e. $\mathcal{R} := \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{R}^x \boldsymbol{x} \leq \boldsymbol{R}^0\}$ with $\boldsymbol{R}^x \in \mathbb{R}^{c \times n_x}$ and $\boldsymbol{R}^0 \in \mathbb{R}^c$.

Definition 6.2 ($\mathcal{P}$-collection). The set $\mathcal{R} \subseteq \mathbb{R}^{n_x}$ is called the $\mathcal{P}$-collection if it is a collection of a finite number of polyhedra, i.e. $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^R$.

Definition 6.3 (Set difference (Baotić and Torrisi, 2003)). The set difference between a polyhedron $\mathcal{Q} \subseteq \mathbb{R}^{n_x}$ and a $\mathcal{P}$-collection $\mathcal{P} \subseteq \mathbb{R}^{n_x}$ is the $\mathcal{P}$-collection $\mathcal{R} = \mathcal{Q} \setminus \mathcal{P} := \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{x} \in \mathcal{Q},\ \boldsymbol{x} \notin \mathcal{P}\}$.

Definition 6.4 (Partition). We call the collection of polyhedra $\{\mathcal{R}_i\}_{i=1}^R$ the *partition* of polyhedron $\mathcal{R}$ if $\mathcal{R} = \bigcup_{i=1}^R \mathcal{R}_i$, and $\text{int}\,(\mathcal{R}_i) \cap \text{int}\,(\mathcal{R}_j) = \emptyset$ for all $i \neq j$. Each polyhedron $\mathcal{R}_i$ will be referred to as the *region* of the partition.

Definition 6.5 (Adjacent regions). Regions $\mathcal{R}_i$ and $\mathcal{R}_j$ of the partition $\mathcal{R}$ are called adjacent if $\mathcal{R}_i \cap \mathcal{R}_j$ is an $(n_x - 1)$-dimensional facet of both $\mathcal{R}_i$ and $\mathcal{R}_j$, $i \neq j$.

Definition 6.6 (Adjacency list). For each facet j of region $\mathcal{R}_i$ of the partition $\mathcal{R}$ we denote by $\mathcal{A}_{i,j}(\mathcal{R})$ the index set of regions adjacent to $\mathcal{R}_i$ along the j-th facet.

Definition 6.7 (PWA function over polyhedra). Function $\boldsymbol{\kappa}(\boldsymbol{x}) : \mathcal{R} \to \mathbb{R}^{n_z}$ with $\boldsymbol{x} \in \mathcal{R} \subseteq \mathbb{R}^{n_x}$, $\mathcal{R}$ being a polyhedron, is called Piecewise Affine (PWA) over polyhedra, if $\{\mathcal{R}_i\}_{i=1}^R$ is the partition of $\mathcal{R}$ and

$$\boldsymbol{\kappa}(\boldsymbol{x}) := \boldsymbol{K}_i \boldsymbol{x} + \boldsymbol{L}_i \quad \forall \boldsymbol{x} \in \mathcal{R}_i, \tag{6.1}$$

with $\boldsymbol{K}_i \in \mathbb{R}^{n_z \times n_x}$, $\boldsymbol{L}_i \in \mathbb{R}^{n_z}$, and $i = [1, \ldots, R]$.

Definition 6.8 (Continuous PWA function). PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ is continuous if $\boldsymbol{K}_i \boldsymbol{x} + \boldsymbol{L}_i = \boldsymbol{K}_j \boldsymbol{x} + \boldsymbol{L}_j$ holds $\forall \boldsymbol{x} \in \mathcal{R}_i \cap \mathcal{R}_j$, $i \neq j$.

6.3 Explicit Model Predictive Control

In MPC, optimal control actions are calculated by solving a suitable optimisation problem, which usually takes the following form:

$$J(\boldsymbol{x}_0) = \min_{\boldsymbol{U}_N} \ \ell_N(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{6.2a}$$

$$\text{s.t.} \ \ \boldsymbol{x}_0 = \boldsymbol{x}(t), \tag{6.2b}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{6.2c}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \tag{6.2d}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \tag{6.2e}$$

$$\boldsymbol{x}_N \in \mathcal{X}^f, \tag{6.2f}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ and $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ denote, respectively, the state and input predictions at time instance $t+k$, initialised by the measurements of the current state $\boldsymbol{x}(t)$. These quantities are constrained to reside within of chosen sets $\mathcal{X}$ and $\mathcal{U}$. Evolution of the predictions is driven by the *prediction model*, represented by the function $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$. The prediction is carried out over a finite prediction horizon N. The terminal predicted state $\boldsymbol{x}_N$ is constrained to reside in a suitable terminal set $\mathcal{X}^f \subseteq \mathcal{X}$. The aim is to find the vector $\boldsymbol{U}_N := [\boldsymbol{u}_0^T, \boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_{N-1}^T]^T$ of optimal control inputs which minimises the objective function (6.2a) composed of the *terminal penalty* $\ell_N(\cdot)$ and the *stage costs* $\ell(\cdot, \cdot)$:

$$\ell_N(\boldsymbol{x}_N) = \|\boldsymbol{Q}_N \boldsymbol{x}_N\|_p, \tag{6.3a}$$

$$\ell(\boldsymbol{x}_k, \boldsymbol{u}_k) = \|\boldsymbol{Q}_x \boldsymbol{x}_k\|_p + \|\boldsymbol{Q}_u \boldsymbol{u}_k\|_p. \tag{6.3b}$$

Here, $\boldsymbol{Q}_N$, $\boldsymbol{Q}_x$ and $\boldsymbol{Q}_u$ are penalty matrices of suitable dimensions and $p = \{1, 2, \infty\}$ denotes a standard polyhedral vector norm. Two types of prediction models are usually considered in practise:

1. Discrete-time linear time-invariant (LTI) models of the form

$$\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k}_{\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}, \tag{6.4}$$

2. Discrete-time piecewise affine (PWA) models represented by a set of n_D distinct local linear models

$$\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{A}_j \boldsymbol{x}_k + \boldsymbol{B}_j \boldsymbol{u}_k \text{ if } \left[\begin{smallmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k\end{smallmatrix}\right] \in \mathcal{D}_j}_{\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}. \tag{6.5}$$

   These systems originate naturally when nonlinear process dynamics is approximated by multiple local linear models. Here, the index $j = 1, \ldots, n_D$ represents the j-th local linear dynamics out of the total number of dynamics n_D. Each local expression is only valid within of the polyhedron $\mathcal{D}_j = \{\left[\begin{smallmatrix}\boldsymbol{x} \\ \boldsymbol{u}\end{smallmatrix}\right] \mid \boldsymbol{D}_j^x \boldsymbol{x} + \boldsymbol{D}_j^u \boldsymbol{u} \leq \boldsymbol{D}_j^0\}$, where $\boldsymbol{D}_j^x$, $\boldsymbol{D}_j^u$, and $\boldsymbol{D}_j^0$ are matrices of suitable dimensions.

In RHMPC, the optimal sequence $\boldsymbol{U}_N^*$ is calculated by solving (6.2) for a given value of $\boldsymbol{x}(t)$. Subsequently, only $\boldsymbol{u}_0^*$ is extracted from $\boldsymbol{U}_N^*$ and it is applied to the plant. At the next time instance the procedure is repeated again for a fresh measurements $\boldsymbol{x}(t)$, hence introducing feedback into the MPC scheme. Since only $\boldsymbol{u}_0^*$ is required at each time step, the RHMPC feedback is given by

$$\boldsymbol{u}_0^*(\boldsymbol{x}(t)) = [\boldsymbol{I}_{n_u}\ \boldsymbol{0}_{n_u}\ \cdots\ \boldsymbol{0}_{n_u}]\boldsymbol{U}_N^*. \tag{6.6}$$

In explicit MPC approach the optimal solution to problem (6.2) is "pre-calculated" *for all possible* values of the initial condition $\boldsymbol{x}(t)$ using *parametric programming* (Bemporad et al, 2002; Kvasnica, 2009). An introduction to this

technique for a linear time-invariant model and a quadratic cost function is given below.

6.3.1 Quadratic Programming Definition

Consider optimal control problems for a discrete-time linear, time-invariant system

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{6.7}$$

with $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$.

Now consider the constrained finite-time optimal control problem

$$J(\boldsymbol{x}_0) = \min \boldsymbol{x}_N^T \boldsymbol{Q}_N \boldsymbol{x}_k + \sum_{k=0}^{N-1} \boldsymbol{x}_k^T \boldsymbol{Q}_x \boldsymbol{x}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_u \boldsymbol{u}_k \tag{6.8a}$$

$$\text{subj. to } \boldsymbol{x}_k \in \mathcal{X},\ \boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \{0, \ldots, N-1\}, \tag{6.8b}$$

$$\boldsymbol{x}_N \in \mathcal{X}^f, \tag{6.8c}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{6.8d}$$

As future state predictions are constrained by (6.7), we can recursively substitute for them yielding

$$\boldsymbol{x}_k = \boldsymbol{A}^i \boldsymbol{x}_0 + \sum_{j=0}^{k-1} \boldsymbol{A}^j \boldsymbol{B} \boldsymbol{u}_{k-1-j} \tag{6.9}$$

Thus, optimal solution of problem (6.8) can be reformulated as

$$J^*(\boldsymbol{x}_0) = \boldsymbol{x}_0^T \boldsymbol{Y} \boldsymbol{x}_0 + \min_{\boldsymbol{U}_N} \frac{1}{2} \left\{ \boldsymbol{U}_N^T \boldsymbol{H} \boldsymbol{U}_N + \boldsymbol{x}_0^T \boldsymbol{F} \boldsymbol{U}_N \right\} \tag{6.10a}$$

$$\text{subj. to } \boldsymbol{G}\boldsymbol{U}_N \leq \boldsymbol{W} + \boldsymbol{E}\boldsymbol{x}_0 \tag{6.10b}$$

where the column vector $\boldsymbol{U}_N$ is the optimisation vector and $\boldsymbol{H}$, $\boldsymbol{F}$, $\boldsymbol{Y}$, $\boldsymbol{G}$, $\boldsymbol{W}$, $\boldsymbol{E}$ can easily be obtained from the original formulation.

The reformulated problem (6.10) is a standard quadratic programming formulation. Taking any initial value $\boldsymbol{x}_0$, optimal future control trajectory $\boldsymbol{U}_N(\boldsymbol{x})$ can be found from which only the first element is used to close the loop.

6.3.2 Explicit Solution

In explicit solution of (6.10) we use the so-called multi-parametric programming approach to optimisation. In multi-parametric programming, the objective is to obtain the optimiser $\boldsymbol{U}_N$ for a whole range of parameters $\boldsymbol{x}_0$, i.e. to obtain $\boldsymbol{U}_N(\boldsymbol{x})$ as an explicit function of the parameter $\boldsymbol{x}$. The term *multi* is used to emphasise that the parameter $\boldsymbol{x}$ (in our case the actual state vector $\boldsymbol{x}_0$) is a vector and not a scalar. If the objective function is quadratic in the optimisation variable $\boldsymbol{U}_N$, the terminology *multi-parametric Quadratic Program* (mp-QP) is used.

In this formulation, it is useful to define

$$\boldsymbol{z} = \boldsymbol{U}_N + \boldsymbol{H}^{-1}\boldsymbol{F}^T\boldsymbol{x}_0 \tag{6.11}$$

and to transform the formulation to problem, where the state vector $\boldsymbol{x}_0$ appears only in constraints

$$J^*(\boldsymbol{x}_0) = \min_{\boldsymbol{z}} \frac{1}{2}\left\{\boldsymbol{z}^T\boldsymbol{H}\boldsymbol{z}\right\} \tag{6.12a}$$

$$\text{subj. to } \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{W} + \boldsymbol{S}\boldsymbol{x}_0 \tag{6.12b}$$

where $\boldsymbol{S} = \boldsymbol{E} + \boldsymbol{G}\boldsymbol{H}^{-1}\boldsymbol{F}^T$.

An mp-QP computation scheme consists of the following three steps:

1. Active Constraint Identification: A feasible parameter $\hat{\boldsymbol{x}}$ is determined and the associated QP (6.12) is solved. This will yield the optimiser $\boldsymbol{z}$ and active constraints $\mathcal{A}(\hat{\boldsymbol{x}})$ defined as inequalities that are active at solution, i.e.

$$\mathcal{A}(\hat{\boldsymbol{x}}) = \{i \in \mathcal{J} \mid \boldsymbol{G}_{(i)}\boldsymbol{z} = \boldsymbol{W}_{(i)} + \boldsymbol{S}_{(i)}\hat{\boldsymbol{x}}\},\ \mathcal{J} = \{1, 2, \ldots, q\}, \tag{6.13}$$

   where $\boldsymbol{G}_{(i)}$, $\boldsymbol{W}_{(i)}$, and $\boldsymbol{S}_{(i)}$ denote the i-th row of the matrices $\boldsymbol{G}$, $\boldsymbol{W}$, and $\boldsymbol{S}$, respectively, and q denotes the number of constraints. The rows indexed by the active constraints $\mathcal{A}(\hat{\boldsymbol{x}})$ are extracted from the constraint matrices $\boldsymbol{G}, \boldsymbol{W}$ and $\boldsymbol{S}$ in (6.12) to form the matrices $\boldsymbol{G}_{\mathcal{A}}, \boldsymbol{W}_{\mathcal{A}}$ and $\boldsymbol{S}_{\mathcal{A}}$.
2. Region Computation: Next, it is possible to use KKT conditions to obtain an explicit representation of the optimiser $\boldsymbol{U}_N(\boldsymbol{x})$ which is valid in some neighbourhood of $\hat{\boldsymbol{x}}$. These are for our problem defined as

$$\boldsymbol{H}\boldsymbol{z} + \boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{0} \tag{6.14a}$$

$$\boldsymbol{\lambda}^T(\boldsymbol{G}\boldsymbol{z} - \boldsymbol{W} - \boldsymbol{S}\hat{\boldsymbol{x}}) = \boldsymbol{0} \tag{6.14b}$$

$$\boldsymbol{\lambda} \geq \boldsymbol{0} \tag{6.14c}$$

$$\boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{W} + \boldsymbol{S}\hat{\boldsymbol{x}} \tag{6.14d}$$

   Optimised variable $\boldsymbol{z}$ can be solved from (6.14a)

$$\boldsymbol{z} = -\boldsymbol{H}^{-1}\boldsymbol{G}^T\boldsymbol{\lambda} \tag{6.15}$$

Condition (6.14b) can be separated into active and inactive constraints. For inactive constraints holds $\boldsymbol{\lambda}_{\mathcal{I}} = 0$. For active constraints are the corresponding Lagrange multipliers $\boldsymbol{\lambda}_{\mathcal{A}}$ positive and inequality constraints are changed to equalities. Substituting for $\boldsymbol{z}$ from (6.15) into equality constraints gives

$$-\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T\boldsymbol{\lambda}_{\mathcal{A}} + \boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\hat{\boldsymbol{x}} = \boldsymbol{0} \tag{6.16}$$

and yields expressions for active Lagrange multipliers

$$\boldsymbol{\lambda}_{\mathcal{A}} = -(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\hat{\boldsymbol{x}}) \tag{6.17}$$

The optimal value of optimiser $\boldsymbol{z}$ and optimal control trajectory $\boldsymbol{U}_N$ are thus given as affine functions of $\hat{\boldsymbol{x}}$

$$\boldsymbol{z} = -\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\hat{\boldsymbol{x}}) \tag{6.18}$$

$$\begin{aligned}\boldsymbol{U}_N &= \boldsymbol{z} - \boldsymbol{H}^{-1}\boldsymbol{F}^T\hat{\boldsymbol{x}} \\ &= -\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\hat{\boldsymbol{x}}) - \boldsymbol{H}^{-1}\boldsymbol{F}^T\hat{\boldsymbol{x}} \\ &= \boldsymbol{F}_r\hat{\boldsymbol{x}} + \boldsymbol{G}_r\end{aligned} \tag{6.19}$$

where

$$\boldsymbol{F}_r = \boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}\boldsymbol{S}_{\mathcal{A}} - \boldsymbol{H}^{-1}\boldsymbol{F}^T \tag{6.20}$$

$$\boldsymbol{G}_r = \boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}\boldsymbol{W}_{\mathcal{A}} \tag{6.21}$$

In the next step, the set of states is determined where the optimiser $\boldsymbol{U}_N(\boldsymbol{x})$ satisfies the same active constraints and is optimal. Such a region is characterised by two inequalities (6.14c), (6.14d) and is written compactly as $\boldsymbol{H}_r\boldsymbol{x} \leq \boldsymbol{K}_r$ where

$$\boldsymbol{H}_r = \begin{bmatrix}\boldsymbol{G}(\boldsymbol{F}_r + \boldsymbol{H}^{-1}\boldsymbol{F}^T) - \boldsymbol{S} \\ (\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}\boldsymbol{S}_{\mathcal{A}}\end{bmatrix} \tag{6.22}$$

$$\boldsymbol{K}_r = \begin{bmatrix}\boldsymbol{W} - \boldsymbol{G}\boldsymbol{G}_r \\ -(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T)^{-1}\boldsymbol{W}_{\mathcal{A}}\end{bmatrix} \tag{6.23}$$

3. State Space Exploration: Once the controller region is computed, the algorithm proceeds iteratively in neighbouring regions until the entire feasible state space is covered with controller regions.

After completing the algorithm, the explicit model predictive controller consists of definitions of multiple state regions with different affine control laws. Its actual implementation reduces to search for an active region of states and calculation of the corresponding control.

6.3.3 Summary

Let us now summarise and generalise the obtained results. If we assume that the sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{X}^f$ in (6.2d)–(6.2f) are polyhedra containing the origin in their respective interiors, the closed-form solution to (6.2) is characterised by the following Theorem.

Theorem 6.1 (Borrelli (2003)). *The RHMPC feedback $\boldsymbol{u}_0^*(\boldsymbol{x}(t))$ for problem (6.2) with the prediction model (6.2c) represented by (6.4) or (6.5) is given by*

$$\boldsymbol{u}_0^*(\boldsymbol{x}(t)) = \boldsymbol{\kappa}(\boldsymbol{x}(t)) \tag{6.24}$$

where:

1. *$\boldsymbol{\kappa}(\boldsymbol{x}(t))$ is a PWA function of the form (6.1),*
2. *$\boldsymbol{\kappa}(\boldsymbol{x}(t))$ is defined over R polyhedral regions $\mathcal{R}_i$,*
3. *$\boldsymbol{\kappa}(\boldsymbol{x}(t))$ has the domain $\Omega = \bigcup_i \mathcal{R}_i$,*
4. *the optimal cost $J^*(\boldsymbol{x}(t))$ is a PWA function $J^*(\boldsymbol{x}(t)) = \boldsymbol{M}_i\boldsymbol{x}(t) + \boldsymbol{L}_i$ defined over the same regions $\mathcal{R}_i$.*

Theorem 6.1 states that RHMPC feedback $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ can be constructed offline as PWA function. Henceforth, $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ will be called the *explicit RHMPC feedback law*. The advantage of such an approach is that value of $\boldsymbol{u}_0^*$ for a particular value of $\boldsymbol{x}(t)$ can be obtained by simply evaluating $\boldsymbol{\kappa}(\boldsymbol{x}(t))$. For the type of problems investigated in this work, such an evaluation is usually faster compared to solving problem (6.2) as an optimisation problem with a fixed initial condition using off-the-shelf software.

There are two main factors which decide whether it will be possible to employ $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ as an RHMPC feedback in real time:

- whether the memory footprint of PWA function $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ fits into the storage limits of the control device,
- whether it possible to evaluate such a function, for a particular value of $\boldsymbol{x}(t)$, within of one sampling instance.

Clearly, as the number of regions of $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ grows, memory consumption increases and evaluation gets slower. Therefore, in the next sections we provide two different strategies which aim at replacing $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ by a different feedback $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$ of lower complexity. As will be illustrated on concrete examples, the presented procedures allow to significantly decrease the memory and runtime requirements needed to implement MPC in real time.

6.3.4 Numerical Example

We consider a double integrator whose transfer function representation is given by

$$P(s) = \frac{1}{s^2}$$

With a sampling time $T_s = 1\,\mathrm{s}$, the corresponding state-space form can be written as

$$\boldsymbol{x}_{k+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \boldsymbol{x}_k + \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} u_k$$
$$y_k = x_{2,k}$$

We want to design an explicit optimal state-feedback law which minimises quadratic performance index (6.8a) with $N = 5$, $\boldsymbol{Q}_N = \boldsymbol{0}$, $\boldsymbol{Q}_x = \boldsymbol{I}$, and $Q_u = 1$. System states and the control signals are subject to constraints $\boldsymbol{x}_k \in [-1, 1] \times [-1, 1]$ and $u_k \in [-1, 1]$, respectively.

We will implement MPC using the Multi-Parametric Toolbox (MPT). To do so, we describe the dynamical model of the plant:

```
model.A = [1 1; 0 1];
model.B = [1; 0.5];
model.C = [0 1];
model.D = 0;
```

along with the system constraints:

```
model.umin = -1;
model.umax = 1;
model.xmin = [-1; -1];
model.xmax = [1; 1];
```

Next, parameters of the performance index have to be specified:

```
cost.N = 5;
cost.Q = [1 0; 0 1];
cost.R = 1;
cost.P_N = 0;
cost.norm = 2;
```

Finally, the explicit optimal state-feedback control law can be calculated by executing

```
controller = mpt_control(model, cost)
```

The obtained explicit control law is defined by

$$u = \begin{cases} \begin{pmatrix} -0.52 & -0.94 \end{pmatrix} \boldsymbol{x}, \text{ if } \begin{pmatrix} 0.48 & 0.88 \\ -0.48 & -0.88 \\ -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{x} \leq \begin{pmatrix} 0.93 \\ 0.93 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \\ \quad \text{(Region \#1)} \\ -1, \quad \text{if } \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -0.48 & -0.88 \end{pmatrix} \boldsymbol{x} \leq \begin{pmatrix} 1 \\ 1 \\ -0.93 \end{pmatrix} \\ \quad \text{(Region \#2)} \\ 1, \quad \text{if } \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ 0.48 & 0.88 \end{pmatrix} \boldsymbol{x} \leq \begin{pmatrix} 1 \\ 1 \\ -0.93 \end{pmatrix} \\ \quad \text{(Region \#3)} \end{cases}$$

and it can be plotted using the command

```
plot(controller)
```

which will show the regions of the state-space over which the optimal control law is defined, as illustrated in Figure 6.1.

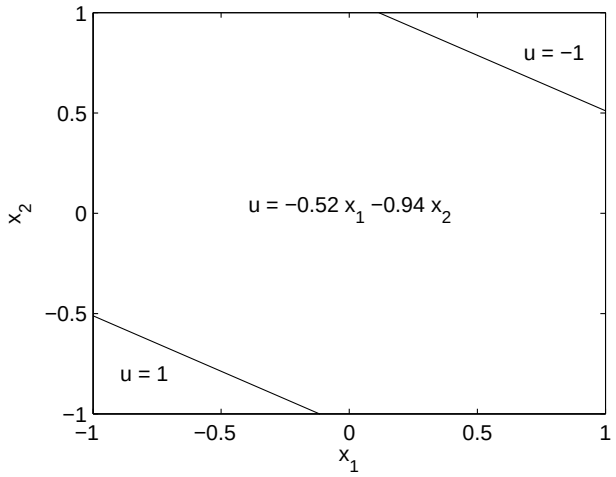


Fig. 6.1 Controller regions and the optimal control for the double integrator example

6.4 Performance-Lossless Complexity Reduction of Explicit MPC

6.4.1 Introduction

By solving the optimisation problem (6.2) using parametric programming, one obtains an explicit representation of the RHMPC feedback law as a PWA function $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ defined over R regions. In this section we show that it is possible to replace this function by a significantly simpler expression $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$ if the following standing assumption holds.

Assumption 6.2 *The explicit RHMPC feedback law $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ is a continuous PWA function.*

Theorem 6.3 (Borrelli (2003)). *If the optimisation problem (6.2) is formulated using a linear prediction model in (6.2c) and solved using parametric programming, then $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ satisfies Assumption 6.2.*

Remark 6.1. For PWA-based prediction models (cf. (6.5)) in (6.2c), continuity of $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ is not guaranteed a-priori. In such a case the methods presented in this section can still be applied if an a-posteriori continuity check is performed.

For linear prediction models in (6.2c), Theorem 6.3 guarantees that the explicit RHMPC feedback (6.24) is a continuous PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ defined over R polyhedral regions. Our primary objective is to replace $\boldsymbol{\kappa}(\boldsymbol{x})$ by a simpler function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$, which requires less memory for its description, is faster to evaluate, and preserves the equivalence $\boldsymbol{\kappa}(\boldsymbol{x}) = \phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$ with $\phi(\cdot)$ being a *clipping function*. It will be shown that $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is a PWA function defined over $\tilde{R}$ polyhedral regions such that $R_{\mathrm{unsat}} \leq \tilde{R} \leq R$, where R_{unsat} is the number of *unsaturated* regions of the original function $\boldsymbol{\kappa}(\boldsymbol{x})$ (cf. Definition 6.9). In addition, it will be illustrated that, typically, $\tilde{R} = R_{\mathrm{unsat}}$ and $\tilde{R} \ll R$ for the case of problems investigated in Section 6.3. Therefore, replacing the explicit RHMPC feedback $\boldsymbol{\kappa}(\boldsymbol{x})$ by $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$ does not sacrifice any performance, and usually leads to a significant reduction of the memory consumption and to an increased on-line evaluation speed.

6.4.2 Theoretical Background

Definition 6.9 (Saturated region). Let $\overline{\boldsymbol{\kappa}}$ and $\underline{\boldsymbol{\kappa}}$ denote, respectively, the element-wise maximum and minimum which the PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ attains over $\mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$. Denote by $\mathcal{I}_{\max}$ the index set of regions saturated at the maximum of $\boldsymbol{\kappa}(\boldsymbol{x})$ (i.e. $\boldsymbol{K}_i = \boldsymbol{0}$ and $\boldsymbol{L}_i = \overline{\boldsymbol{\kappa}}$ for all $i \in \mathcal{I}_{\max}$), by $\mathcal{I}_{\min}$ the

index set of regions saturated at the minimum (i.e. $\boldsymbol{K}_i = \boldsymbol{0}$ and $\boldsymbol{L}_i = \underline{\boldsymbol{\kappa}}$ for all $i \in \mathcal{I}_{\min}$), and $\mathcal{I}_{\text{sat}} = \mathcal{I}_{\max} \cup \mathcal{I}_{\min}$. We call the region $\mathcal{R}_i$ the *saturated region* if it is either saturated at the minimum or at the maximum, i.e. if $i \in \mathcal{I}_{\text{sat}}$. Otherwise the region is called *unsaturated*. The index set of unsaturated regions is denoted by $\mathcal{I}_{\text{unsat}}$.

Definition 6.10 (Saturated PWA function). We call the PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ *saturated* if its partition contains at least one saturated region, i.e. $\mathcal{I}_{\text{sat}} \neq \emptyset$.

Remark 6.2. Not every explicit RHMPC feedback $\boldsymbol{u}^*(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ is necessarily a saturated PWA function. If it is not, then no simplification can be achieved using the procedure discussed here.

Remark 6.3. If $\boldsymbol{\kappa}(\boldsymbol{x})$ is a vector-valued function (i.e. when $n_u > 1$), its regions are considered saturated in the sense of Definition 6.9 if all its elements $\kappa_1(\boldsymbol{x}), \ldots, \kappa_{n_u}(\boldsymbol{x})$ are saturated *jointly* at maximum, or at minimum. E.g. a region with $\kappa_1(\boldsymbol{x}) = \underline{\kappa}_1$ and $\kappa_2(\boldsymbol{x}) = \underline{\kappa}_2$ is considered saturated, but the case with $\kappa_1(\boldsymbol{x}) = \overline{\kappa}_1$ and $\kappa_2(\boldsymbol{x}) = \underline{\kappa}_2$ is not.

Remark 6.4. Although the joint saturation outlined in Remark 6.3 may sound too conservative, in Section 6.4.4 we demonstrate that it is fulfilled in practise often enough for the presented procedure to achieve considerable reduction of complexity. Reducing the conservatism w.r.t. the requirement of joint saturation is subject of ongoing research.

Definition 6.11 (Suitable augmentation). Given is a saturated continuous PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ as in (6.1), defined over the partition $\{\mathcal{R}_i\}_{i=1}^{R}$. We call the function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ a *suitable augmentation* of $\boldsymbol{\kappa}(\boldsymbol{x})$ if following properties hold:

P1: $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is defined over the $\mathcal{P}$-collection $\{\tilde{\mathcal{R}}_j\}_{j=1}^{\tilde{R}}$ such that $\bigcup_i \mathcal{R}_i = \bigcup_j \tilde{\mathcal{R}}_j$, i.e. $\operatorname{dom}(\boldsymbol{\kappa}(\boldsymbol{x})) = \operatorname{dom}(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$,
P2: $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$,
P3: $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \geq \overline{\boldsymbol{\kappa}}$ for all $\boldsymbol{x} \in \mathcal{R}_{\mathcal{I}_{\max}}$,
P4: $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \leq \underline{\boldsymbol{\kappa}}$ for all $\boldsymbol{x} \in \mathcal{R}_{\mathcal{I}_{\min}}$,

where $\overline{\boldsymbol{\kappa}}$, $\underline{\boldsymbol{\kappa}}$, $\mathcal{I}_{\text{unsat}}$, $\mathcal{I}_{\max}$, and $\mathcal{I}_{\min}$ are as in Definition 6.9, and $\mathcal{R}_{\mathcal{I}}$ denotes the subset of regions $\{\mathcal{R}_i\}_{i \in \mathcal{I}}$ for some index set $\mathcal{I} \subseteq 1, \ldots, R$.

Figure 2(a) shows an illustrative 1-D PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$, while Fig. 2(c) depicts its suitable augmentation.

6.4.3 Main Results

Notice that a suitable augmentation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is not, by Definition 6.11, required to be continuous, nor does it require that $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \operatorname{dom} \boldsymbol{\kappa}(\boldsymbol{x})$.

It merely suggests that one can replace the affine expression $\boldsymbol{\kappa}(\boldsymbol{x}) = \boldsymbol{K}_i\boldsymbol{x}+\boldsymbol{L}_i$ in the saturated regions by an arbitrary $\tilde{\boldsymbol{K}}_i\boldsymbol{x} + \tilde{\boldsymbol{L}}_i$ which satisfies P3–P4. As will be shown in the sequel, this freedom allows to construct a simpler function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ by enlarging the unsaturated regions such that they completely cover the saturated ones. Once such a function is obtained, we recover $\boldsymbol{\kappa}(\boldsymbol{x})$ by applying a simple clipping function $\phi(\bullet)$ such that $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \boldsymbol{\kappa}(\boldsymbol{x})\ \forall \boldsymbol{x} \in \mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$. A procedure for computing $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is reported as Algorithm 1, which is the first main result of the chapter.

Algorithm 1 Construction of a suitable augmentation

INPUT: Saturated continuous PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ defined over the polyhedral partition $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^{R}$ with $\mathcal{R}_i = \{\boldsymbol{x} \mid \boldsymbol{R}_i^x\boldsymbol{x} \le \boldsymbol{R}_i^0\}$ and $\Omega = \bigcup_i \mathcal{R}_i$ being a convex polyhedron.

OUTPUT: Suitable augmentation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \tilde{\boldsymbol{K}}_j\boldsymbol{x} + \tilde{\boldsymbol{L}}_j$ if $\boldsymbol{x} \in \tilde{\mathcal{R}}_j$, $j = 1, \ldots, \tilde{R}$.

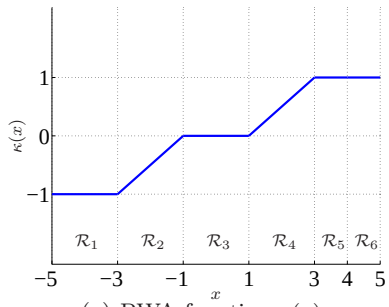
1: Obtain the adjacency list $\mathcal{A}_{i,j}(\mathcal{R})$ and the index set $\mathcal{I}_{\mathrm{unsat}}$ representing indices of unsaturated regions.
2: **for each** unsaturated region $\mathcal{R}_i \in \mathcal{R}_{\mathcal{I}_{\mathrm{unsat}}}$ **do**
3: Using the adjacency list $\mathcal{A}_{i,j}$ identify the subset of half-space indices $\mathcal{J}$ over which the neighbour of $\mathcal{R}_i$ is a saturated region.
4: **Phase A:** Form a new polyhedron $\tilde{\mathcal{R}} = \{\boldsymbol{x} \mid \tilde{\boldsymbol{R}}^x\boldsymbol{x} \le \tilde{\boldsymbol{R}}^0\}$ by removing from $\mathcal{R}_i$ the half-spaces indexed by $\mathcal{J}$, i.e. $\tilde{\boldsymbol{R}}^x = [\boldsymbol{R}_i^x]_{\setminus\mathcal{J}}$ and $\tilde{\boldsymbol{R}}^0 = [\boldsymbol{R}_i^0]_{\setminus\mathcal{J}}$.
5: **Phase B:** Let $\tilde{\mathcal{R}} = \tilde{\mathcal{R}} \cap \Omega$.
6: Determine which unsaturated regions other than $\mathcal{R}_i$ intersect with $\tilde{\mathcal{R}}$. Denote the index set of intersecting regions by $\mathcal{I}$.
7: **if** $\mathcal{I} \neq \emptyset$ **then**
8: **Phase C:** Let $\tilde{\mathcal{R}} = \tilde{\mathcal{R}} \setminus \mathcal{R}_{\mathcal{I}}$, cf. Definition 6.3.
9: **end if**
10: Store region(s) $\tilde{\mathcal{R}}$ and set $\tilde{\boldsymbol{K}}_r = \boldsymbol{K}_i$, $\tilde{\boldsymbol{L}}_r = \boldsymbol{L}_i$ for each $\tilde{\mathcal{R}}_r \in \tilde{\mathcal{R}}$.
11: **end for**
12: If $\tilde{R} > R$, take $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$.
13: **return**

We will explain the algorithm on the following example. Consider a 1-D PWA function $\kappa(x)$ as shown in Figure 2(a), where $R = 6$ and domain of $\kappa(x)$ is $\Omega = \{x \mid -5 \le x \le 5\}$:

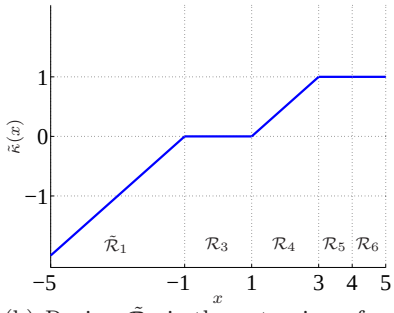
1. $\mathcal{R}_1 = \{-5 \le x \le -3\}$, $\kappa(x) = -1$
2. $\mathcal{R}_2 = \{-3 \le x \le -1\}$, $\kappa(x) = 0.5x + 0.5$
3. $\mathcal{R}_3 = \{-1 \le x \le 1\}$, $\kappa(x) = 0$
4. $\mathcal{R}_4 = \{1 \le x \le 3\}$, $\kappa(x) = 0.5x - 0.5$
5. $\mathcal{R}_5 = \{3 \le x \le 4\}$, $\kappa(x) = 1$
6. $\mathcal{R}_6 = \{4 \le x \le 5\}$, $\kappa(x) = 1$

Since $\overline{\kappa} = 1$ and $\underline{\kappa} = -1$, $\mathcal{I}_{\mathrm{sat}} = \{1, 5, 6\}$ and $\mathcal{I}_{\mathrm{unsat}} = \{2, 3, 4\}$.

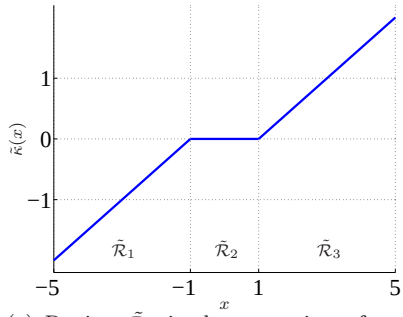
The algorithm iterates through all unsaturated regions in an arbitrary order. Take $i = 2$. Then in Step 3 the region $\mathcal{R}_2$ has $\mathcal{R}_1$ as a saturated neighbour over the half-space $x \ge -3$. Therefore, in Phase A on Step 4 a new region $\tilde{\mathcal{R}}$ is formed by removing this half-space from $\mathcal{R}_2$. This gives



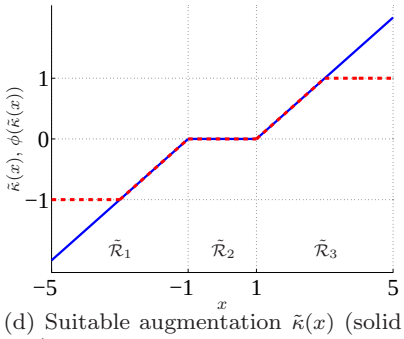
(a) PWA function $\kappa(x)$.



(b) Region $\tilde{\mathcal{R}}_1$ is the extension of region $\mathcal{R}_2$ over its saturated neighbour $\mathcal{R}_1$.



(c) Region $\tilde{\mathcal{R}}_3$ is the extension of region $\mathcal{R}_4$ over its saturated neighbours $\mathcal{R}_5$ and $\mathcal{R}_6$.



(d) Suitable augmentation $\tilde{\kappa}(x)$ (solid line) and the result of clipping $\phi(\tilde{\kappa}(x))$ (dashed line). Note that $\phi(\tilde{\kappa}(x)) = \kappa(x)$ for all x.

Fig. 6.2 Illustration of a suitable augmentation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ of a PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$.

$\tilde{\mathcal{R}} = \{x \leq -1\}$. As this region is unbounded, it is then intersected with Ω in Step 5, which gives $\tilde{\mathcal{R}} = \{-5 \leq x \leq -1\}$ in Phase B. As this new region does not intersect with any other unsaturated regions (i.e. with $\mathcal{R}_3$ or $\mathcal{R}_4$), Step 8 is skipped and region $\tilde{\mathcal{R}}$ is stored, cf. Figure 2(b).

Then the algorithm continues with the second unsaturated region, i.e $i = 3$. As region $\mathcal{R}_3$ has no saturated neighbours, $\mathcal{J} = \emptyset$ in Step 4, and therefore $\tilde{\mathcal{R}} = \mathcal{R}_3$ on Step 10, i.e. the region remains unchanged.

Finally, for $i = 4$ the region $\mathcal{R}_4$ has region $\mathcal{R}_5$ as a saturated neighbour over the half-space $x \leq 3$. Therefore $\tilde{\mathcal{R}} = \{x \geq 1\}$ in Phase A, $\tilde{\mathcal{R}} = \{1 \leq x \leq 5\}$ in Phase B, and Phase C is not needed, as $\tilde{\mathcal{R}}$ doesn't intersect with other unsaturated regions $\mathcal{R}_2$ and $\mathcal{R}_4$, cf. Figure 2(c). The algorithm terminates after exploring all unsaturated regions.

Remark 6.5. The adjacency list is automatically generated as a by-product of most pQP solvers, see e.g. Bemporad et al (2002); Tøndel et al (2003a); Kvasnica et al (2004); Baotić (2005); Spjøtvold et al (2005). Should it not be available at hand, it can be computed by the MPT Toolbox (Kvasnica et al, 2004). The toolbox can also be used to detect polyhedral intersections in Step 6 and to compute the set difference between two $\mathcal{P}$-collections in Step 8.

Remark 6.6. As noted by Baotić and Torrisi (2003), the set difference between a polyhedron $\tilde{\mathcal{R}}$ and a $\mathcal{P}$-collection $\mathcal{R}_{\mathcal{I}}$ in Step 8 is, in general, the $\mathcal{P}$-collection $\{\tilde{\mathcal{R}}_r\}_{r=1}^{\tilde{R}}$. If, however, the union $\cup\tilde{\mathcal{R}}_r$ is convex, $\tilde{\mathcal{R}}$ can be replaced by a single region. If it is not, a minimal representation $\tilde{\mathcal{R}}$ can be obtained e.g. by the method of Geyer et al (2008).

Remark 6.7. In theory, the set difference operation in Step 8 can produce exponentially many regions. Therefore Step 12 is formally needed to ensure that $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is no more complex than the original function $\boldsymbol{\kappa}(\boldsymbol{x})$. We remark that we have never observed such a case, though.

Next, we provide a formal proof of correctness of Algorithm 1.

Theorem 6.4. *For an arbitrary saturated continuous PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$, Algorithm 1 constructs its suitable augmentation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ which fulfils all prerequisites of Definition 6.11.*

Proof. First, we show that $\bigcup_j \tilde{\mathcal{R}}_j = \bigcup_i \mathcal{R}_i = \Omega$. As $\boldsymbol{\kappa}(\boldsymbol{x})$ is assumed to be saturated, at least one half-space must be have been removed in Step 4, hence $\bigcup_j \tilde{\mathcal{R}}_j \supseteq \Omega$. However, due to Step 5 we have $\bigcup_j \tilde{\mathcal{R}}_j = \bigcup_j (\tilde{\mathcal{R}}_j \cap \Omega) = \Omega \cap (\bigcup_j \tilde{\mathcal{R}}_j) = \Omega$. Hence $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ fulfil-ls P1 of Def. 6.11. To prove that $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ for all $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ it is enough to show that $\tilde{\mathcal{R}} \cap \mathcal{R}_{\mathcal{I}_{\text{unsat}}} = \emptyset$, i.e. that the extended regions $\tilde{\mathcal{R}}$ do not overlap with unsaturated regions. Due to Step 8, we have $(\tilde{\mathcal{R}} \setminus \mathcal{R}_{\mathcal{I}_{\text{unsat}}}) \cap \mathcal{R}_{\mathcal{I}_{\text{unsat}}} = \tilde{\mathcal{R}} \cap (\mathcal{R}_{\mathcal{I}_{\text{unsat}}} \setminus \mathcal{R}_{\mathcal{I}_{\text{unsat}}}) = \tilde{\mathcal{R}} \cap \emptyset = \emptyset$. Therefore $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ meets P2 of Def. 6.11. Finally, P3 and P4 follow directly since $\boldsymbol{\kappa}(\boldsymbol{x})$ is assumed to be continuous.

Theorem 6.5. *The number of regions $\tilde{R}$ of the augmented function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ generated by Algorithm 1 is bounded by $R_{unsat} \leq \tilde{R} \leq R$.*

Proof. The lower bound comes from two facts: (i) Algorithm 1 does not modify the number of unsaturated regions; and (ii) the saturated regions are replaced by "expansion" of unsaturated regions, therefore $\tilde{R} = R_{\text{unsat}}$ in the best case. However, Phase C in Step 8 can give rise to additional regions due to Remark 6.6, and therefore $\tilde{R} \geq R_{\text{unsat}}$, in general. The upper bound follows directly from Step 12, cf. Remark 6.7.

Corollary 6.1. *If Step 8 is never invoked during the run of Algorithm 1, then $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is defined over $\tilde{R} = R_{unsat}$ regions.*

Theorems 6.4 and 6.5 say that $\boldsymbol{\kappa}(\boldsymbol{x})$ can be replaced by its suitable augmentation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ of (possibly) lower complexity in terms of number of regions. As will be documented in Section 6.4.4, usually $\tilde{R} \ll R$ for the case of problems considered in this chapter. The augmentation, however, cannot be readily applied as an RHMPC feedback since, in general, $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \neq \boldsymbol{\kappa}(\boldsymbol{x})$ for some $\boldsymbol{x} \in \mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$. The equivalence can be achieved by passing $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ through a very simple clipping function, as noted by Theorem 6.6, which is the second main result.

Theorem 6.6. *Consider a saturated continuous PWA function $\boldsymbol{\kappa}(\boldsymbol{x})$ and its suitable augmentation $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$. Let*

$$\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) := \max(\min(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}), \overline{\boldsymbol{\kappa}}), \underline{\boldsymbol{\kappa}}). \tag{6.25}$$

Then the equivalence $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \boldsymbol{\kappa}(\boldsymbol{x})$ is established for all $\boldsymbol{x} \in \mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$, and therefore $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$ is a performance-lossless replacement of $\boldsymbol{\kappa}(\boldsymbol{x})$.

Proof. Notice that (6.25) is a compact encoding of three IF-THEN rules:

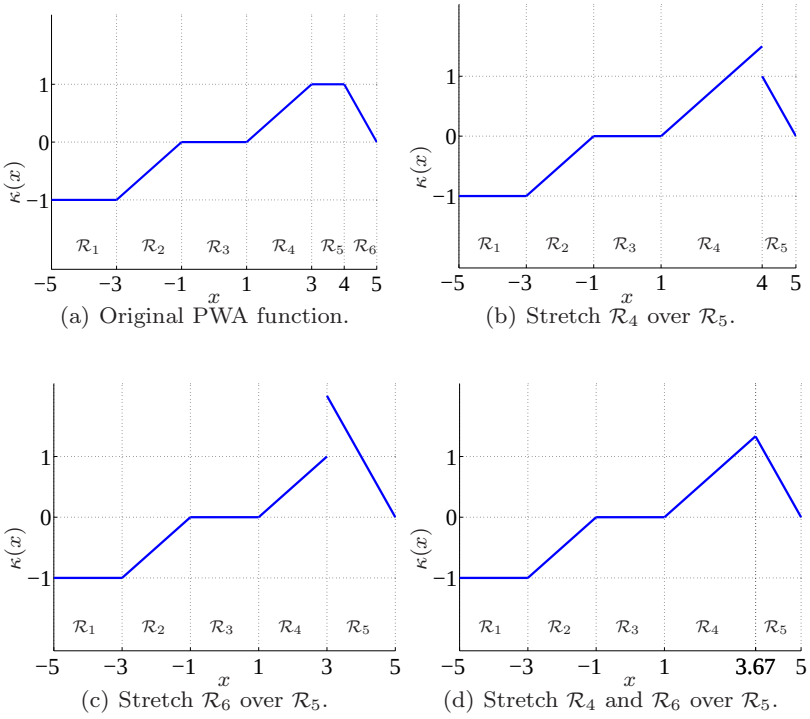
$$\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \begin{cases} \overline{\boldsymbol{\kappa}} & \text{if } \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \geq \overline{\boldsymbol{\kappa}}, \\ \underline{\boldsymbol{\kappa}} & \text{if } \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \leq \underline{\boldsymbol{\kappa}}, \\ \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) & \text{otherwise.} \end{cases} \tag{6.26}$$

Then we get $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \overline{\boldsymbol{\kappa}}$ for all $x \in \mathcal{R}_{\mathcal{I}_{\max}}$ by P3 of Definition 6.11, $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \underline{\boldsymbol{\kappa}}$ for all $x \in \mathcal{R}_{\mathcal{I}_{\min}}$ by P4, and $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ for all $x \in \mathcal{R}_{\mathcal{I}_{\mathrm{unsat}}}$ by P2. Finally, we have that $\mathcal{R}_{\mathcal{I}_{\max}} \cup \mathcal{R}_{\mathcal{I}_{\min}} \cup \mathcal{R}_{\mathcal{I}_{\mathrm{unsat}}} = \mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$ by Def. 6.4 and Theorem 6.1, and hence $\mathrm{dom}(\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))) \supseteq \mathrm{dom}(\boldsymbol{\kappa}(\boldsymbol{x}))$.

Evaluation of $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$ for a given value of $\boldsymbol{x}$ is a three stage process. First, the index a of the region which contains $\boldsymbol{x}$ is identified. This can be achieved e.g. by searching through the regions $\tilde{\mathcal{R}}_i$ sequentially, and stopping once $\boldsymbol{x} \in \tilde{\mathcal{R}}_a$. Alternatively, one can construct the binary search tree (Tøndel et al, 2003b) to perform the region traversal in time logarithmic in $\tilde{R}$. Once the index a is found, the a-th elements $\tilde{\boldsymbol{K}}_a$ and $\tilde{\boldsymbol{L}}_a$ are extracted from memory and $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \tilde{\boldsymbol{K}}_a \boldsymbol{x} + \tilde{\boldsymbol{L}}_a$ is computed. Finally, the function value is clipped by passing it through (6.25), which always performs only $2n_u$ comparisons, insignificant compared to the complexity of region traversal. The extra memory required to store $\overline{\boldsymbol{\kappa}}$ and $\underline{\boldsymbol{\kappa}}$ for $\phi(\cdot)$ is $2n_u$ floating point numbers, negligible compared to the memory footprint of regions $\tilde{\mathcal{R}}_i$.

By Theorem 6.6 we have that $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$ is a performance-lossless replacement of the explicit RHMPC feedback $\boldsymbol{\kappa}(\boldsymbol{x})$ since $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})) = \boldsymbol{\kappa}(\boldsymbol{x}) = \boldsymbol{u}_0^*(\boldsymbol{x})$ for all initial conditions $\boldsymbol{x}$ for which problem (6.2) is feasible. Therefore if $\boldsymbol{\kappa}(\boldsymbol{x})$ guarantees certain closed-loop properties (e.g. stability, optimality, feasibility), so will $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}))$. In addition, it will be demonstrated in the next section that $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is, in majority of practical cases, significantly simpler than $\boldsymbol{\kappa}(\boldsymbol{x})$. To show this, we perform an extensive case study aimed at illustrating

that, typically, $\tilde{R} = R_{\text{unsat}} \ll R$, hence showing that the procedure described in this chapter leads to RHMPC controllers of significantly lower complexity.



(a) Original PWA function. (b) Stretch $\mathcal{R}_4$ over $\mathcal{R}_5$.

(c) Stretch $\mathcal{R}_6$ over $\mathcal{R}_5$. (d) Stretch $\mathcal{R}_4$ and $\mathcal{R}_6$ over $\mathcal{R}_5$.

Fig. 6.3 Order of region traversal and set difference operation

The next example shows a situation when phase C is necessary. Let us assume that there are some unsaturated regions at the border of the feasible state region enclosed by saturated regions. This represents the situation in Figure 3(a) from the modified introductory example where the region $\mathcal{R}_6$ would be unsaturated and the control law decreasing, for example $\kappa(x) = -x + 5$. The presented algorithm based on set difference can now produce two types of solutions.

1. We start with region $\mathcal{R}_4$ that is stretched to the right border of Ω covering regions $\mathcal{R}_5$ and $\mathcal{R}_6$. Afterwards, set difference procedure removes the region $\mathcal{R}_6$ leading to a discontinuity at point $x = 4$ (Figure 3(b)). Of course, this discontinuity will be removed by clipping.
2. We start with region $\mathcal{R}_6$ that is stretched to the left border of Ω covering all other regions. Afterwards, set difference procedure removes regions $\mathcal{R}_1$

to $\mathcal{R}_4$ leading to a discontinuity at point $x = 3$ (Figure 3(c)). Again, this discontinuity will be removed by clipping.

This ambiguity in order in which regions are processed can lead to different region shapes and sometimes even to different number of resulting regions (see Section 6.4.5 where this behaviour was observed).

Another possible algorithm could be to stretch regions $\mathcal{R}_4$ and $\mathcal{R}_6$ over $\mathcal{R}_5$ until the corresponding feedback laws intersect (Figure 3(d)). An advantage of this approach lies in the fact that it does not depend on the order in which regions are processed. On the other hand, it might lead to combinatorial problems in higher state dimensions and to a need to decompose the resulting intersection of unsaturated neighbours into convex regions.

6.4.4 Numerical Examples

6.4.4.1 Double Integrator I

Consider again the double integrator problem from Section 6.3.4. As it consists only of three regions (one unconstrained LQR and two saturated ones), the resulting controller is simply clipped LQR controller

$$v(t) = -0.52x_1(t) - 0.94x_2(t), \quad u(t) = \max(\min(v(t), 1), -1). \tag{6.27}$$

This controller is valid over the whole state constraints $\boldsymbol{x} \in [-1, 1] \times [-1, 1]$.

6.4.4.2 Double Integrator II

Consider again a double integrator sampled at 1 second given by the state-space representation

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t), \tag{6.28}$$

which is subject to constraints $\mathcal{X} = \{-30 \leq \boldsymbol{x}(t) \leq 30\}$ and $\mathcal{U} = \{-1 \leq u(t) \leq 1\}$. Compared to the previous case, feasible state box is much larger. The MPC problem (6.2) was formulated with prediction horizon $N = 25$, $\boldsymbol{Q}_x = \boldsymbol{Q}_N = \boldsymbol{I}$, $Q_u = 1$, and $\mathcal{X}^f = \mathcal{X}$. Problem (6.2) was then solved as a parametric QP according to Theorem 6.1. Using the MPT Toolbox, the explicit RHMPC feedback $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ was obtained in 63 seconds (on a 2.4 GHz CPU with 2GB of RAM using MATLAB 7.4) as a PWA function defined over 847 regions shown in Fig. 6.4.

A simple analysis showed that the function contains 37 unsaturated regions (the coloured stripe in Fig. 6.4). We have subsequently applied Algorithm 1

to construct a replacement function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$. The algorithm terminated after just 0.5 seconds, giving $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$ defined over 37 regions, cf. Corollary 6.1. The regions of $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$ are depicted in Fig. 6.5. It follows that complexity of RHMPC implementation can be reduced by a factor of 22 by using $\phi(\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t)))$ instead of $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ as a feedback in this scenario.

The optimal region merging (ORM), as proposed in Geyer et al (2008), was not directly applicable in this case because the problem size was too large to be handled. However, the same reference also introduces a sub-optimal way of merging based on the divide&conquer strategy. Applying such an approach resulted in 71 regions after 278 seconds.

As it was mentioned above, different order in which regions are processed gives different region shapes. Another possible region shapes with the exactly the same number of regions (37) are shown in Fig. 6.6.

6.4.5 Random Systems

Next we analyse random stable and unstable LTI systems with 2 and 3 states, 1 and 2 inputs, subject to constraints $\mathcal{X} = \{\boldsymbol{x}(t) \mid -30 \leq \boldsymbol{x}(t) \leq 30\}$ and $\mathcal{U} = \{u(t) \mid -1 \leq u(t) \leq 1\}$. For each system the MPC problem (6.2) is constructed with $\boldsymbol{Q}_x = \boldsymbol{I}$, $\boldsymbol{Q}_u = \boldsymbol{I}$, $N = 15$, and $\boldsymbol{Q}_N$ and $\mathcal{X}^f$ designed such that closed-loop stability is attained, i.e. setting $\boldsymbol{Q}_N$ to the solution of DARE and using a positively control invariant terminal set $\mathcal{X}^f$. For each random system we have then solved the MPC problem (6.2) parametrically using the MPT Toolbox Kvasnica et al (2004). Each resulting PWA solution $\boldsymbol{u}_0^*(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ was subsequently post-processed independently by Algorithm 1 and by the ORM method of Geyer et al (2008).

Results obtained for $n_x = 2$ and $n_u = 1$ are shown in Table 6.1. Columns of the table represent, respectively, the index of the random system, R – the number of regions of $\boldsymbol{\kappa}(\boldsymbol{x})$, $\tilde{R}$ – the number of regions of $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ calculated by Algorithm 1, $\tilde{T}$ – the runtime Algorithm 1, R_{ORM} – the number of regions obtained by the ORM method of Geyer et al (2008), and T_{ORM} – its runtime. Entries with † denote cases where the ORM procedure did not finish within of 12 hours. Entries marked with $\star$ denote cases where optimal region merging crashed due to a prohibitive size of the problem. In such case, results of ORM sub-optimal divide&conquer strategy are presented instead.

For convenience, systems in tables are sorted in the ascending order of number of original regions.

On average, the number of regions of $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ is decreased by a factor of 8.4. In 90% of cases Algorithm 1 generated $\tilde{R}$ equal to the number of unsaturated regions of $\boldsymbol{\kappa}(\boldsymbol{x})$, cf. Corollary 6.1. The only exception was the random system reported in row no. 10, where $R_{\text{unsat}} = 29$.

Table 6.2 shows the results obtained for random systems with $n_x = 2$ states and $n_u = 2$ inputs. Again, the reported numbers show that Algorithm 1 is

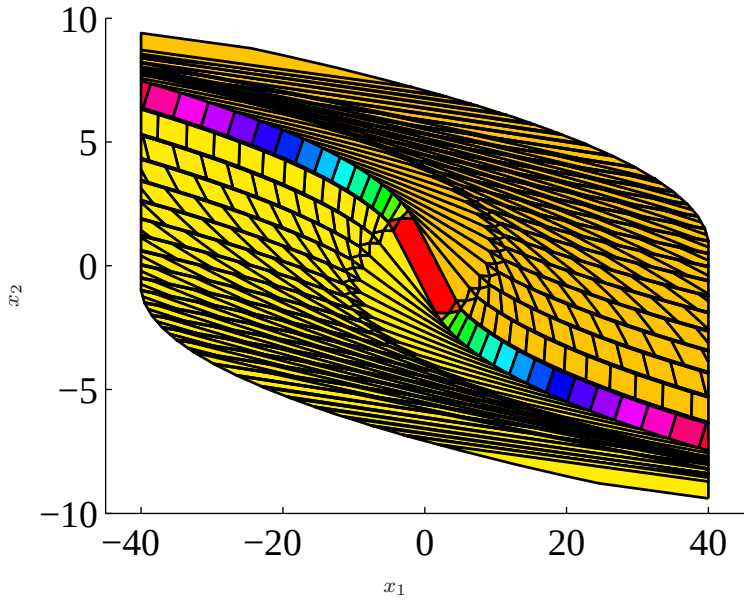


Fig. 6.4 Double integrator: regions of $\boldsymbol{\kappa}(\boldsymbol{x}(t))$. The coloured strip in the middle are unsaturated regions. The two large yellow areas represent regions where $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ is saturated at $\overline{\boldsymbol{\kappa}}$ or $\underline{\boldsymbol{\kappa}}$.

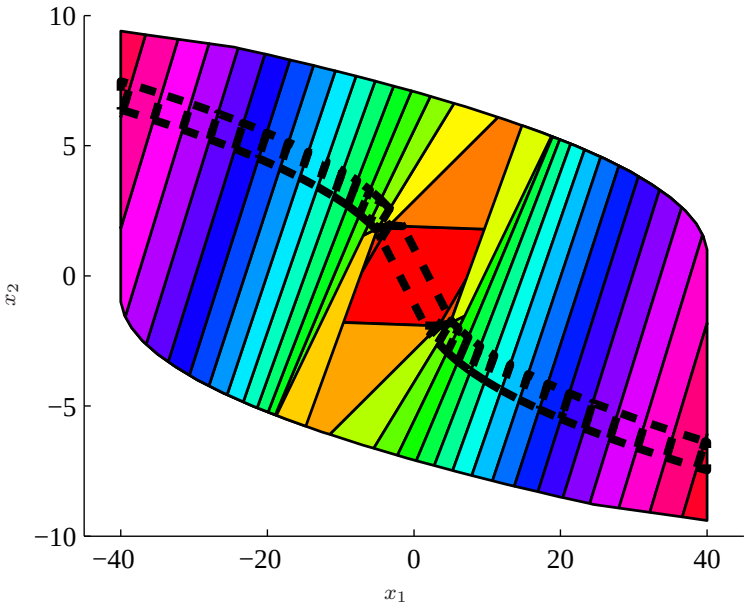


Fig. 6.5 Double integrator: regions of the replacement function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$ (solid shapes) formed as the extensions of unsaturated regions of $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ (dashed regions).

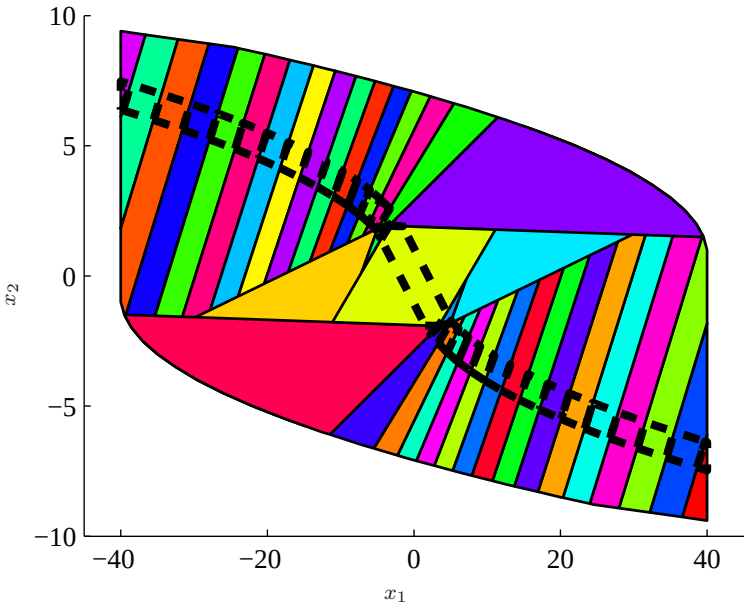


Fig. 6.6 Double integrator: different regions of the replacement function $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$ resulting from a different region order.

able to considerably decrease the complexity of the explicit RHMPC feedback law with relatively low computational effort. On average, complexity in terms of number of regions was decreased by a factor of 4.7. This factor is lower compared to the previous test case. This is due to the fact that Algorithm 1 only removes those regions in which *all* control inputs are saturated, cf. Remark 6.3. The likelihood that this would happen naturally decreases with increasing dimension of the input vector. It is worth noting, though, that even under the conservative assumption of joint saturation, the procedure presented here still achieved better results than the ORM approach for all cases except of two (rows no. 6 and 10 in Table 6.2). In 90% of cases the generated replacement RHMPC feedback law $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ was defined over the corresponding number of unsaturated regions, once again confirming that the conclusions of Corollary 6.1 often hold in practise.

The only exception was case no. 10, where the theoretically achievable minimum number of regions was $R_{\text{unsat}} = 495$ and the obtained number of regions was 769. When subsequent runs of the algorithm with randomly changed processing order were applied, the following numbers resulted: $734, 693, 735, 712, 760$. Thus, the algorithm based on set-difference can indeed produce suboptimal results.

Table 6.1 Results for random systems with $n_x = 2$ and $n_u = 1$.

Case	R	$\tilde{R}$	R_{ORM}	$\tilde{T}$ [s]	T_{ORM} [s]
1	35	5	11	0.1	0.6
2	59	7	15	0.2	1.5
3	75	9	19	0.2	2.5
4	83	9	19	0.2	2.5
5	91	9	19	0.2	3.1
6	127	15	27*	0.2	13.3
7	153	23	39*	0.4	28.4
8	173	17	31*	0.3	18.2
9	221	23	43*	0.4	42.9
10	225	39	49*	0.8	40.5

Table 6.2 Results for random systems with $n_x = 2$ and $n_u = 2$.

Case	R	$\tilde{R}$	R_{ORM}	$\tilde{T}$ [s]	T_{ORM} [s]
1	37	5	7	0.1	0.5
2	61	7	9	0.2	0.7
3	63	5	9	0.2	0.8
4	71	29	41	0.5	7.8
5	73	33	39	0.7	11.4
6	83	53	45	1.2	15.1
7	165	63	83*	1.5	98.3
8	271	119	155*	4.6	283.9
9	943	193	210*	11.3	606.9
10	2029	769	463*	95.2	860.7

Results for random systems with 3 states and 1 input are reported in Table 6.3. They show that complexity of $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ was reduced by factors ranging from 5 for case no. 2 to 15 for case no. 3. On average, the number of regions $\tilde{R}$ of $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ was 8 times less than the number of regions of $\boldsymbol{\kappa}(\boldsymbol{x})$. It is also worth noting that $\tilde{R}$ was equal to R_{unsat} for all investigated cases except of system no. 5 (where $R_{\mathrm{unsat}} = 55$) and system no. 10 (where $R_{\mathrm{unsat}} = 353$).

Overall, obtained results indicate that the significant reduction of RHMPC complexity was achieved. Using the proposed method it was always possible to reduce the number of regions considerably in a relatively short time. In addition, Algorithm 1, compared to the ORM approach, scales significantly better with increasing size of the problem. On the other hand, the ORM method is more general, as it also allows post-processing of discontinuous PWA functions.

Table 6.3 Results for random systems with $n_x = 3$ and $n_u = 1$.

Case	R	$\tilde{R}$	R_{ORM}	$\tilde{T}$ [s]	T_{ORM} [s]
1	43	7	16	1	19
2	223	49	57⋆	1	324
3	481	33	80⋆	1	261
4	503	41	71⋆	1	2100
5	523	69	101⋆	2	575
6	527	51	75⋆	2	1808
7	547	73	188⋆	3	3420
8	837	139	†	7	†
9	1628	274	†	23	†
10	1795	358	†	38	†

6.4.6 Combination of Clipping and ORM

In principle, clipping and ORM can be combined and applied in sequence. The preferred way can be to apply clipping first and ORM afterwards. There are several supporting points for this:

- ORM cannot handle problems with a number of regions having the same control law larger than about 50. Thus, for a majority of realistic problems, the optimal method does not converge. For a class of slightly larger problems, modified ORM based on divide&conquer strategy may be used with a relative success. This can also be confirmed by examining Tables 6.1–6.3.
- Experience shows that the most of regions with the same control law are the saturated ones. ORM has to join these into larger convex polytopes whereas clipping needs not to take care of convexity.
- Clipping keeps resulting regions convex and removes a large fraction of situations that would produce nonconvex union of polytopes with ORM.
- Clipping cannot handle regions with unconstrained control laws.

On the other hand, the set difference step in the clipping algorithm can produce some new regions that were not in the original formulation. In that case, it can theoretically happen that ORM followed by clipping could produce smaller number of regions as for the reverse sequence.

For illustration of advantages when the combined approach is used, consider again the double integrator process with the horizon $N = 3$ and absolute value cost function using MPT:

```
Double_Integrator
probStruct.norm = 1; probStruct.N = 3;
ctrl = mpt_control(sysStruct, probStruct);
```

The resulting controller consists of 50 regions and is shown in Fig. 6.7. More specifically, there are 16 regions on the upper control constraint, 16 regions on the lower control constraint and there are also some regions with the same control law that is not saturated.

We can now apply both ORM and clipping

```
orm1 = mpt_simplify(ctrl, 'optimal')
[clip1, sat, unsat] = remove_saturated_regions(ctrl);
```

This produces results shown in Fig. 6.8. We can see that ORM can reduce the original controller to 16 regions whereas clipping to 18. Thus, in this case ORM is more successful than clipping. This can be attributed to the fact that the number of saturated regions is not very high and that ORM can converge.

Finally, we apply again ORM and clipping for the already simplified controllers

```
orm2 = mpt_simplify(clip1, 'optimal')
[clip2, sat, unsat] = remove_saturated_regions(orm1);
```

In this case, both sequences of methods converge to the same controller with 10 regions shown in Fig. 6.9.

6.5 Polynomial Approximation of RHMPC

6.5.1 Introduction

In this section we shown how to approximate the given explicit RHMPC feedback law $\boldsymbol{\kappa}(\boldsymbol{x})$ by a single multivariate polynomial

$$\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = \sum_{i=0}^{d} \sum_{j=1}^{n_x} [\boldsymbol{\alpha}_i]_j x_j^i \tag{6.29}$$

of pre-specified degree d in such a way that closed-loop stability, feasibility, and bounded performance decay are guaranteed. Here, $\boldsymbol{\alpha}_i \in \mathbb{R}^{n_u \times n_x}$ are the coefficients to be determined, $[\boldsymbol{\alpha}_i]_j$ denotes the j-th column of $\boldsymbol{\alpha}_i$, x_j^i is the i-th power of the j-th element of vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$, and n_x and n_u denote, respectively, the number of states and control inputs. Once calculated, the polynomial can replace the explicit MPC solution as a feedback controller, without negative impact on stability or constraint satisfaction. The added benefit is that evaluation of the polynomial feedback (6.29) for a given state measurements $\boldsymbol{x}$ can be done much faster compared to traversing the look-up table of the optimal MPC controller (Kvasnica et al, 2008). Memory footprint of the approximate controller is also significantly smaller compared to that of the optimal MPC feedback law.

The approximation is performed in two steps. First, given an explicit representation of the RHMPC feedback law $\boldsymbol{\kappa}(\boldsymbol{x})$ and a corresponding PWA Lyapunov function $V(\boldsymbol{x})$, the set of feedback laws which render $V(\boldsymbol{x})$ a Control

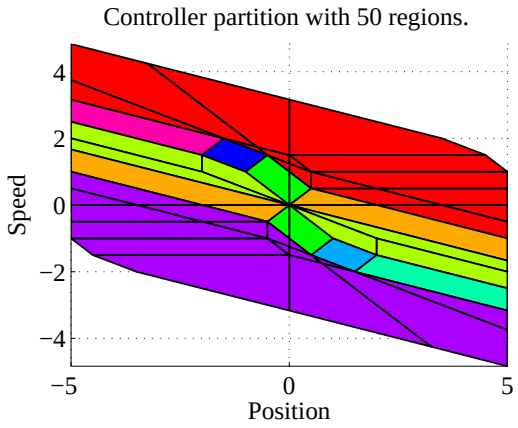


Fig. 6.7 Double integrator: original solution without any region reduction

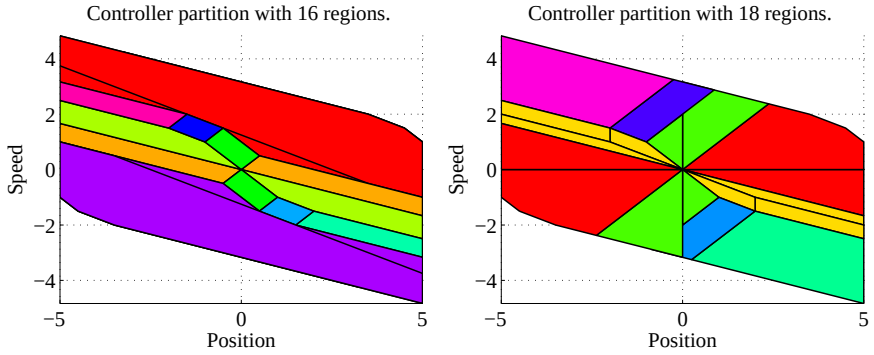


Fig. 6.8 Double integrator: region reduction with ORM (left) and clipping (right)

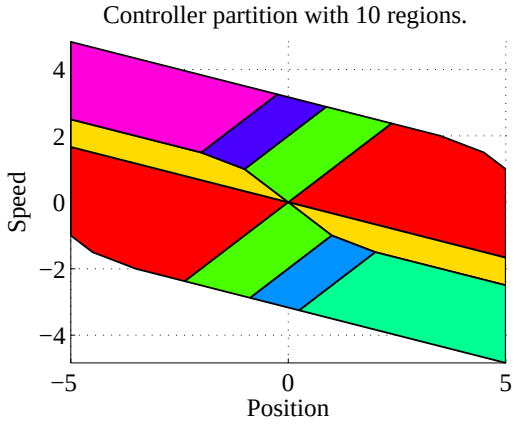


Fig. 6.9 Double integrator: final solution with both ORM and clipping applied (in any order)

Lyapunov Function is calculated using basic computational geometry tools. It is shown that any control law from this set asymptotically stabilises the given system while also providing constraint satisfaction for all time. Then, in the second step, we show how to search for the coefficients of the approximation polynomial such that it is always contained in the set of stabilising feedback laws by solving a single linear program (LP).

6.5.2 Theoretical Background

Assumption 6.7 (Stability, feasibility) *Note that in the following it is assumed that the parameters* $N, \boldsymbol{Q}_x, \boldsymbol{Q}_u, \boldsymbol{Q}_N$, *and* $\mathcal{X}^f$ *in* (6.2) *are chosen in such a way that the explicit RHMPC feedback* $\boldsymbol{\kappa}(\boldsymbol{x}(t))$ *as in* (6.24) *is closed-loop stabilising, feasible for all time (Christophersen, 2007) and that a polyhedral piecewise affine Lyapunov function of the form*

$$V(\boldsymbol{x}(t)) = \boldsymbol{V}_i^x \boldsymbol{x}(t) + V_i^0, \quad \textit{if} \quad \boldsymbol{x}(t) \in \mathcal{R}_i, \tag{6.30}$$

for the closed-loop system

$$\boldsymbol{f}^{CL}(\boldsymbol{x}(t)) := \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{\kappa}(\boldsymbol{x}(t))), \tag{6.31}$$

$\boldsymbol{x}(t) \in \Omega$, *exists and is given.*

This is not a restricting requirement but rather the aim of most (if not all) control strategies. Furthermore, we remark that if the parameters are chosen according to e.g. Mayne et al (2000) one can simply take $V(\boldsymbol{x}(t))$ equal to the optimal cost $J^*(\boldsymbol{x}(t))$.

In order to present the complete result for the new controller approximation approach, the two underlying core ideas need to be explained. The first idea is based on the inherent freedom of the Lyapunov function (6.30):

Theorem 6.8 (Asymptotic/exponential stability (Lazar et al, 2008)). *Let* Ω *be a bounded positively invariant set in* $\mathbb{R}^{n_x}$ *for the autonomous (closed-loop) system* $\boldsymbol{x}(t+1) = \boldsymbol{f}^{CL}(\boldsymbol{x}(t))$ *with* $\boldsymbol{x}(t) \in \Omega$ *and let* $\underline{\alpha}(\cdot)$, $\overline{\alpha}(\cdot)$, *and* $\beta(\cdot)$ *be* K*-class functions (Vidyasagar, 1993). If there exists a non-negative function* $V : \Omega \to \mathbb{R}_{\geq 0}$ *with* $V(\mathbf{0}_{n_x}) = 0$ *such that*

$$\underline{\alpha}(\|\boldsymbol{x}\|) \leq V(\boldsymbol{x}) \leq \overline{\alpha}(\|\boldsymbol{x}\|), \tag{6.32a}$$

$$\Delta V(\boldsymbol{x}) := V(\boldsymbol{f}^{CL}(\boldsymbol{x})) - V(\boldsymbol{x}) \leq -\beta(\|\boldsymbol{x}\|), \tag{6.32b}$$

for all $\boldsymbol{x} \in \Omega$, *then the following results hold:*

(a)The equilibrium point $\mathbf{0}_{n_x}$ *is* asymptotically stable *(Vidyasagar, 1993) in the Lyapunov sense in* Ω.

(b)If $\underline{\alpha}(\|\boldsymbol{x}\|) := \underline{a}\|\boldsymbol{x}\|^{\gamma}$, $\overline{\alpha}(\|\boldsymbol{x}\|) := \overline{a}\|\boldsymbol{x}\|^{\gamma}$, *and* $\beta(\|\boldsymbol{x}\|) := b\|\boldsymbol{x}\|^{\gamma}$ *for some positive constants* $\underline{a}, \overline{a}, b, \gamma > 0$ *then the equilibrium point* $\mathbf{0}_{n_x}$ *is* stable *(Vidyasagar, 1993) in the Lyapunov sense in* Ω.

Simply speaking, if all the prerequisites of Theorem 6.8 are fulfilled with a given controller $\boldsymbol{\kappa}(\cdot)$, the resulting behaviour of the closed-loop system is stabilising. If, for the given function $V(\cdot)$, $\beta(\cdot)$ is now relaxed, one can (possibly) find a set of controllers that will render the closed-loop system stabilising and feasible. These sets of controllers are denoted in the following as *stability tubes*. The concept and results of stability tubes – along with their computation – are elaborated in further detail in (Christophersen, 2007, Ch. 10).

Definition 6.12 (Stability tube). Let $V(\cdot)$ be a Lyapunov function for the general nonlinear, closed-loop system $\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, with $\boldsymbol{x}(t) \in \Omega$, under the stabilising control $\boldsymbol{u}(t) = \boldsymbol{\kappa}(\boldsymbol{x}(t))$ and constraints $\left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{smallmatrix}\right] \in \mathcal{D}$ and let the prerequisites of Theorem 6.8 be fulfilled. Furthermore, let $\beta(\cdot)$ be a K-class function. Then the set

$$\mathcal{S}(V,\beta) := \left\{ \left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{smallmatrix}\right] \in \mathbb{R}^{n_x \times n_u} \;\middle|\; \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \in \Omega, \; \left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{smallmatrix}\right] \in \mathcal{D}, \; V(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})) - V(\boldsymbol{x}) \leq -\beta(\|\boldsymbol{x}\|) \right\}$$

is called *stability tube*.

Theorem 6.9 (Christophersen (2007)). *Let the assumptions of Definition 6.12 be fulfilled. Then every control law* $\boldsymbol{u}(t) = \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}(t))$, $\boldsymbol{x}(t) \in \Omega$, *(also* any *sequence of control samples* $\boldsymbol{u}(t)$*) fulfilling*

$$\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{u}(t)\end{bmatrix} \in \mathcal{S}(V,\beta) \tag{6.33}$$

asymptotically stabilises the system $\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, *where* $\boldsymbol{x}(t) \in \Omega$, *to the origin.*

Naturally, for general nonlinear systems, the stability tube $\mathcal{S}(V,\beta)$ can basically take any form. Note, however, that for the considered class of PWA systems (6.5), PWA control laws $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\kappa}(\boldsymbol{x})$ of the form (6.24), and PWA Lyapunov functions of the form (6.30) with $\beta(\cdot)$ consisting of a sum of weighted vector 1-/∞-norms, the stability tube can be described by a collection of polytopic sets in the state-input space and can be computed with basic polytopic operations:

$$\mathcal{S}(V,\beta) := \left\{ \left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{smallmatrix}\right] \;\middle|\; \boldsymbol{f}_{\mathrm{PWA}}(\boldsymbol{x},\boldsymbol{\kappa}(\boldsymbol{x})) \in \Omega, \; V(\boldsymbol{f}_{\mathrm{PWA}}(\boldsymbol{x},\boldsymbol{\kappa}(\boldsymbol{x}))) - V(\boldsymbol{x}) \leq -\beta(\|\boldsymbol{x}\|) \right\}. \tag{6.34}$$

In the case considered here, the stability tube can be represented and 'easily' be obtained as a collection (or union) of polytopes of the form $\mathcal{S}(V,\beta) := \cup_{j=1}^{N_\mathcal{S}} \mathcal{S}_j$, where the closure of $\mathcal{S}_j$ is $\bar{\mathcal{S}}_j := \left\{ \left[\begin{smallmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{smallmatrix}\right] \in \mathbb{R}^{n_x+n_u} \mid \boldsymbol{S}_j^{xu} \left[\begin{smallmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{smallmatrix}\right] \leq \boldsymbol{S}_j^0 \right\}$.

Without going into details, by construction, we have the following properties: *(a)* for some index set $\mathcal{I}_i \subseteq \{1,\ldots,N_\mathcal{S}\}$, the union $\cup_{j\in\mathcal{I}_i}\mathcal{S}_j$ is defined over the controller region $\mathcal{R}_i$, and *(b)*, $\sum_{i=1}^{N_\mathcal{P}} |\mathcal{I}_i| = N_\mathcal{S}$. This means that each $\mathcal{S}_j$ is defined over a single region $\mathcal{R}_i$, i.e. if for some i_1 and j we have $\text{proj}_x(\mathcal{S}_j) \subseteq \mathcal{P}_{i_1}$ then there does not exist a $i_2 \neq i_1$ with $\text{proj}_x(\mathcal{S}_j) \subseteq \mathcal{P}_{i_2}$. We remark that simulations seem to indicate that most often $\mathcal{I}_i = 1$ for all i, i.e. only one $\mathcal{S}_j$ is defined over $\mathcal{R}_i$.

6.5.3 Main Results

We aim at approximating the optimal RHMPC feedback law $\boldsymbol{\kappa}(\boldsymbol{x})$ by a single polynomial

Problem 6.1. Given $\boldsymbol{\kappa}(\boldsymbol{x})$ as in (6.24) and $V(\boldsymbol{x})$ of the form (6.30) as an optimal closed-form solution to the CFTOC problem (6.2) for a PWA system (6.5) with $p = 1$ or $p = \infty$ in (6.3), find coefficients $\boldsymbol{\alpha}_0,\ldots,\boldsymbol{\alpha}_d$ of the polynomial state-feedback law (6.29) of fixed degree d which approximates $\boldsymbol{\kappa}(\boldsymbol{x})$ in such a way that closed-loop stability, constraint satisfaction, and a bounded performance decay are guaranteed.

Assumption 6.10 *For the pair $\boldsymbol{\kappa}(\boldsymbol{x})$, $V(\boldsymbol{x})$ there exists a stability tube $\mathcal{S}(V,\beta) = \bigcup \mathcal{S}_i(V,\beta)$ of the form (6.34) with $\mathcal{S}_i$ defined over the i-th regions $\mathcal{R}_i$ being convex (i.e. $|\mathcal{I}_i| = 1$), and the union $\bigcup \mathcal{S}_i(V,\beta)$ being connected.*

Existence of $\mathcal{S}(V,\beta)$ hints at existence of control laws other than $\boldsymbol{\kappa}(\boldsymbol{x})$ which would yield the same closed-loop properties (stability and constraint satisfaction). Connectivity is implied by the assumption that a single polynomial covers the whole space of interest and convexity is assumed in order to obtain a unique solution.

Theorem 6.9 provides a sufficient condition for existence of $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ which solves Problem 6.1:

Lemma 6.1. *Let a stability tube $\mathcal{S}(V,\beta)$ satisfying Assumption 6.10 be given and denote by $p_i(\boldsymbol{\alpha},\boldsymbol{x})$ a set of polynomials*

$$p_i(\boldsymbol{\alpha},\boldsymbol{x}) := \boldsymbol{S}_i^0 - \boldsymbol{S}_i^{xu} \left[\begin{smallmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \end{smallmatrix}\right]. \tag{6.35}$$

Then $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ as in (6.29) is a solution to Problem 6.1 if

$$p_i(\boldsymbol{\alpha},\boldsymbol{x}) \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{R}_i,\ \forall i \in [1,\ldots,R]. \tag{6.36}$$

Proof. By assuming convexity of $\mathcal{S}_i(\cdot)$ we have

$$\mathcal{S}_i(\cdot) = \left\{ \begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \end{bmatrix} \mid \boldsymbol{S}_i^{xu} \begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) \end{bmatrix} \leq \boldsymbol{S}_i^0 \right\} \tag{6.37}$$

Hence (6.36) is equivalent to (6.33) with $p_i(\boldsymbol{\alpha}, \boldsymbol{x})$ as in (6.35). From Theorem 6.9 it follows that any control law, i.e. also $\boldsymbol{u} = \tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$, satisfying $\left[\begin{smallmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{smallmatrix}\right] \in \bigcup \mathcal{S}_i(V, \beta)$ will provide closed-loop stability, constraint satisfaction, and a guaranteed worst-case performance decay of $\beta(\|\boldsymbol{x}\|)$.

Lemma 6.1 suggests that finding $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ of the form (6.29) as a solution to Problem 6.1 can be cast as finding coefficients $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_d$ such that polynomials $p_i(\boldsymbol{\alpha}, \boldsymbol{x})$ are non-negative for all points $\boldsymbol{x} \in \mathcal{R}_i$, $\forall i \in [1, \ldots, R]$. The proposed approach is based on the following theorem, due to Pólya (Hardy et al, 1952):

Theorem 6.11 (Pólya's theorem). *If a homogeneous polynomial $p_i(\boldsymbol{\alpha}, \boldsymbol{x})$ is positive $\forall \boldsymbol{x} \in \mathcal{R}_i$ with $\mathcal{R}_i$ being a simplex, all the coefficients of $p_i^M(\boldsymbol{\alpha}, \boldsymbol{x}) = p_i(\boldsymbol{\alpha}, \boldsymbol{x}) \cdot (\sum_{j=1}^{n_x} x_j)^M$ are positive for a sufficiently large Pólya degree M.*

Remark 6.8. Search for $\boldsymbol{\alpha}$ such that $p_i^M(\boldsymbol{\alpha}, \boldsymbol{x}) \geq 0, \forall \boldsymbol{x} \in \mathcal{R}_i$ can be performed by using the more obvious reverse of Pólya's theorem, i.e. that positive coefficients of the extended polynomial imply positivity over the whole simplex.

Remark 6.9. The advantage of Theorem 6.11 over other conservative techniques for ensuring positivity of polynomials (such as the SOS formulation of Kvasnica et al (2008)) stems from the fact that given a symbolic representation of $p_i^M(\boldsymbol{\alpha}, \boldsymbol{x})$, the coefficients $\boldsymbol{\alpha}$ can be found by solving a linear program (LP). To see this, observe that $\boldsymbol{\alpha}$ enters (6.35) in a linear fashion and that all constraints (6.36) are linear in $\boldsymbol{\alpha}$.

Notice, however, that Theorem 6.11 is not directly applicable to find $\boldsymbol{\alpha}$ from (6.36) as $\mathcal{R}_i$ are not simplices, in general. To overcome this limitation, we observe that, by Theorem 6.1, we have $\mathcal{R}_i = \{\boldsymbol{x} \mid \boldsymbol{R}_i^x \boldsymbol{x} \leq \boldsymbol{R}_i^0\}$, which is a polytope described by an intersection of finitely many half-spaces. Given $\mathcal{V}_i = \text{vertices}(\mathcal{R}_i)$ being a set of extremal vertices of $\mathcal{R}_i$, the i-th region can be equivalently expressed as a convex combination of $\mathcal{V}_i$:

$$\mathcal{R}_i = \left\{ \boldsymbol{x} \;\middle|\; \boldsymbol{x} = \sum_{j=1}^{|\mathcal{V}_i|} \lambda_j [\mathcal{V}_i]_j, \ \forall \boldsymbol{\lambda} \in \Lambda_i \right\}, \tag{6.38}$$

$$\Lambda_i = \left\{ \boldsymbol{\lambda} \;\middle|\; 0 \leq \lambda_j \leq 1, \ \sum_{j=1}^{|\mathcal{V}_i|} \lambda_j = 1 \right\}, \tag{6.39}$$

where $|\mathcal{V}_i|$ stands for the number of extremal points of the i-th region, $[\mathcal{V}_i]_j$ denotes the j-th vertex of $\mathcal{R}_i$, and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{|\mathcal{V}_i|}]$. By substituting for $\boldsymbol{x} = \sum_j \lambda_j [\mathcal{V}_i]_j$ into (6.36) we get

$$p_i(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \geq 0, \quad \forall \boldsymbol{\lambda} \in \Lambda_i, \ \forall i \in [1, \ldots, R]. \tag{6.40}$$

Notice that Λ_i in (6.40) are now simplices and Theorem 6.11 can therefore be applied to find $\boldsymbol{\alpha}$ such that $p_i(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ is non-negative $\forall \boldsymbol{\lambda} \in \Lambda_i$.

We can now state the second main result, which is Theorem 6.12 and Algorithm 2 for calculating values of the coefficients $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_d$ of the polynomial feedback law $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ which is an admissible solution to Problem 6.1.

Algorithm 2 Polynomial approximation

INPUT: PWA system (6.5), parameters N, $\boldsymbol{Q}_x$, $\boldsymbol{Q}_u$, $\boldsymbol{Q}_N$, $\mathcal{X}^f$ of the CFTOC problem (6.2), desired maximal performance decay $\beta(\|\boldsymbol{x}\|)$, degree of the approximation polynomial d, Pólya degree M.

OUTPUT: Coefficients $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_d$ of the polynomial feedback (6.29) which, when applied as a state-feedback, asymptotically stabilises the given PWA system.

1: Obtain a closed-form solution $\boldsymbol{\kappa}(\boldsymbol{x})$, $\mathcal{R}_i$, $V(\boldsymbol{x})$ to the CFTOC problem (6.2) according to Theorem 6.1.
2: Calculate the stability tube $\mathcal{S}(V, \beta)$ per (6.34).
3: Calculate extremal vertices $\mathcal{V}_i$ of all regions $\mathcal{R}_i$.
4: Formulate polynomials $p_i(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ per (6.40).
5: Homogenise $p_i(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ by multiplying single monomials by $(\sum_{j=1}^{|\mathcal{V}_i|} \lambda_j)$ until all monomials have the same degree.
6: Compute, symbolically, coefficients c_i^M of the Pólya's polynomial $p_i^M(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = p_i(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \cdot (\sum_{j=1}^{|\mathcal{V}_i|} \lambda_j)^M$.
7: Search for $\boldsymbol{\alpha}$ by solving the following linear program:

$$\text{find} \quad \boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_d, \tag{6.41}$$

$$\text{s.t.} \quad c_i^M \geq 0, \quad \forall i \in [1, \ldots, R]. \tag{6.42}$$

8: **return** $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_d$.

Theorem 6.12. *Let the input arguments of Algorithm 2 satisfy the conditions of Assumptions 6.7 and 6.10. Then the polynomial feedback $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ of the form (6.29) calculated by Algorithm 2 is a solution to Problem 6.1.*

Proof. Directly by Lemma 6.1 and Theorem 6.11.

Remark 6.10. All conditions of Assumption 6.7 will be satisfied (and hence $V(\boldsymbol{x}) = J^*(\boldsymbol{x})$) for $N = \infty$ and $\boldsymbol{Q}_N$, $\mathcal{X}^f$ calculated as shown in Baotić et al (2006).

Remark 6.11. Computation in Steps 1 and 3 of Algorithm 2 can be carried out using Multi-Parametric Toolbox. The code for calculating $\mathcal{S}(V, \beta)$ can be obtained upon mail request from the authors. Steps 4–7 can be solved using YALMIP (Löfberg, 2004), which takes care of all symbolic and non-symbolic calculations.

Remark 6.12. Algorithm 2 is a non-iterative procedure and therefore it always terminates.

Remark 6.13. Instead of a pure feasibility objective in (6.41), an alternative is to look for $\boldsymbol{\alpha}$ which minimise the point-wise distance $\|\kappa(x_j) - \tilde{\kappa}(x_j)\|_1$ with $x_j = [\mathcal{V}_i]_j$, $\forall j = [1, \ldots, |\mathcal{V}_i|]$, $\forall i = [1, \ldots, R]$. Another approach is to try to aim for low-order polynomials by minimising coefficients for higher-order terms. Alternatively, one can even aim for low-complexity controller by minimising the number of non-zero coefficients, which would lead to a mixed-integer LP problem.

6.5.4 Numerical Example

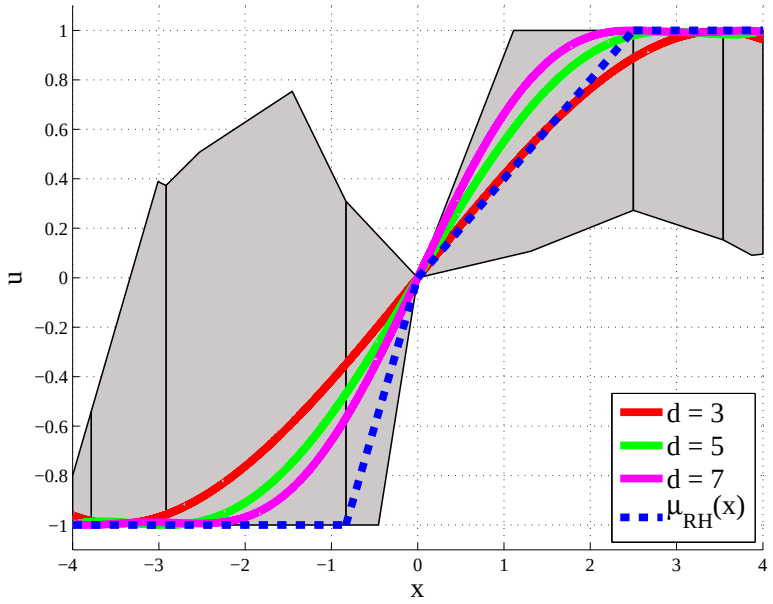


Fig. 6.10 Stability tubes $\mathcal{S}(\cdot)$ (gray sets), optimal control law $\boldsymbol{\kappa}(\boldsymbol{x})$ (blue dashed line), and stabilising polynomial approximations of different degrees.

To illustrate the results of Theorem 6.12, consider the following 1D PWA system (Kvasnica et al, 2008):

$$x_{k+1} = \begin{cases} 4/5\, x_k + 2u_k \text{ if } x_k > 0, \\ -6/5\, x_k + u_k \quad \text{if } x_k \leq 0, \end{cases} \tag{6.43}$$

with $u_k \in [-1,\ 1]$ and $x_k \in [-4,\ 4]$. The CFTOC problem (6.2) was solved with $p = 1, Q_x = 1, Q_u = 1, Q_N = 0, N = \infty$ and the corresponding stability tube $\mathcal{S}(\cdot)$ was calculated for $\beta(\|x\|) = b\|x\|^\gamma$ with $b = 1 \cdot 10^{-6}$ and $\gamma = 1$. The closed-form solution consisted of 7 regions and the stability tube satisfied Assumption 6.10. The sets $\mathcal{S}(\cdot)$ are depicted in gray in Figure 6.10 along with the optimal feedback law $\kappa(x)$. Coefficients of three approximation polynomials with $d = 3, 5, 7$ have been subsequently calculated using Theorem 6.11 with $M = 1$ and are also depicted in Fig. 6.10. The distance-minimisation criterion suggested in Remark 6.13 was used when solving the LP in Step 7 of Algorithm 2.

Note that clipping method introduced in the previous section could be used to enhance quality of the polynomial controller as well. This means that the lower bound (-1) and upper bound (1) on control can be removed (or set to $\pm\infty$) to find the polynomial coefficients. This increases the feasible region of stability tubes. In our example, a clipped linear control law (degree of polynomial $d = 1$)

$$u_k = \max(\min(2x_k, 1), -1), \quad x_k \in [-4,\ 4] \tag{6.44}$$

will satisfy all control and performance specifications. The clipped signal will remain in stability tubes and approximates the optimal control law fairly well. Similarly, clipped higher order polynomials will approximate the optimal control law even closer.

6.5.5 Real-time Control of a Thermo-Optical Device

The uDAQ28/LT thermal-optical system is an experimental device aimed primarily for education purposes Huba et al (2006). The device allows for real time measurement and control of temperature and light intensity. It can be connected to a personal computer via an universal serial bus without requiring an input-output card (Fig. 6.11). Data acquisition and real-time control of the uDAQ28/LT device is carried out in the Matlab/Simulink environment which allows very easy manipulation with the device.

The plant represents a dynamical system which combines slow and fast dynamics. The slow process is characterised by a heat transfer and the fast process corresponds to light emission. Both processes are caused by an embedded light bulb which is controlled by an input voltage signal. In general, the plant is characterised by five inputs and eight outputs whereas only three controlled inputs and three measured outputs are of interest. A precise description of these signals is given in Tab. 6.4.

The construction of the device suggests offers two main control loops. The primal loop regulates the light bulb intensity by manipulating the input voltage. The second loop maintains the inner temperature in safety limits by

manipulating the revolutions of a cooling fan. Presence of physical constraints on manipulated and controlled variables makes the control task challenging and the device has often been used for benchmark of constrained PID control approaches (Huba and Vrančič, 2007).

Table 6.4 Description of measured and controlled signals.

Signal Name	Range
Input voltage to light bulb	0–5 V
Input voltage to cooling fan	0–5 V
Input voltage to LED	0–5 V
Inner temperature	0–100 deg C
Light intensity	not given
Revolutions of the cooling fan	0–6000 rpm

Fig. 6.11 Front view on a thermo-optical device uDAQ28/LT.

6.5.5.1 Identification and PWA Model

In the sequel, only the optical channel of the light-bulb is considered. This decision is motivated by the fact that this channel is represented by a fast dynamics, which makes real-time implementation of a control system a challenging task. Due to very fast responses of the light channel, the sampling rate was selected the lowest admissible by Windows, i.e. $T_s = 0.05\,\mathrm{s}$. As the optical channel is sampled, it immediately suggests identification of input-output relations in discrete time.

Input-output relations of the optical channel have been identified with the help of IDTOOL Toolbox Čirka. et al (2006) as a second order discrete transfer function

$$G(z^{-1}) = \frac{bz^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{6.45}$$

where b, a_1, a_2 are constant parameters and z^{-1} is a discrete time delay operator Mikleš and Fikar (2007). IDTOOL toolbox contains the recursive least squares method LDDIF (Kulhavý and Kárný, 1984) which provides very good estimates of the unknown parameters. However, as the transfer function is valid only locally, the identification was performed over four operating points and the results are summarised in Tab. 6.5.

For the use in explicit MPC scheme, the input-output representation (6.45) is transformed to a discrete state-space model. It is achieved by introducing state variables with discrete time instant k, i.e. $v_{1,k} = y_{k-1}$, $v_{2,k} = y_{k-2}$ and the state space model reads

$$v_{1,k+1} = -a_1 v_{1,k} - a_2 v_{2,k} + b w_k \tag{6.46a}$$

$$v_{2,k+1} = v_{1,k} \tag{6.46b}$$

$$y_k = v_{2,k}. \tag{6.46c}$$

In (6.46) w_k represents the input voltage applied directly to the plant and y_k is the measured output. Voltage input is constrained

$$w_k \in [0, 5]\,\mathrm{V} \tag{6.47}$$

and the measured output lies inside the interval

$$y_k \in [0, 55] \tag{6.48}$$

of light intensity units (not given in the reference manual). The overall input-output behaviour of the optical channel can be recovered by aggregation of the local linear models (6.46) which forms PWA model. Here, the operating area is first split into regions and local linear models are assigned to each such region. The overall behaviour of PWA model is then driven by switching between the locally valid models using logical IF-THEN rules. To perform partitioning of the operating area according to linearisation points in Tab. 6.5, a Voronoi diagram (Aurenhammer, 1991) is constructed, which

Table 6.5 Identification data over four operation points.

	input	output	b	a_1	a_2
(1)	1.3	6.84	2.03	-1.07	0.46
(2)	2.5	19.46	3.56	-0.97	0.43
(3)	3.5	32.09	4.51	-0.91	0.41
(4)	4.5	45.86	5.39	-0.87	0.40

directly returns partitions of the state space as a sequence of convex polytopes. This operation was executed using one of the routines included in MPT toolbox (Kvasnica et al, 2004) and it returned following regions:

$$\mathcal{D}_1 = \{\boldsymbol{v}_k \in \mathbb{R}^2 \mid 0 \leq v_{2,k} \leq 13.15\} \tag{6.49a}$$
$$\mathcal{D}_2 = \{\boldsymbol{v}_k \in \mathbb{R}^2 \mid 13.15 \leq v_{2,k} \leq 25.77\} \tag{6.49b}$$
$$\mathcal{D}_3 = \{\boldsymbol{v}_k \in \mathbb{R}^2 \mid 25.77 \leq v_{2,k} \leq 38.97\} \tag{6.49c}$$
$$\mathcal{D}_4 = \{\boldsymbol{v}_k \in \mathbb{R}^2 \mid 38.97 \leq v_{2,k} \leq 55\} \tag{6.49d}$$

To each of the regions (6.49), a corresponding local linear dynamics (6.46) is assigned, and it forms overall PWA model.

The output from PWA model has been compared to the real measured output from the plant and the result is depicted in Fig. 6.12. For the given scenario PWA model follows correctly the plant's output, thus the accuracy of the model is verified. It can be noticed that at the beginning there is larger mismatch between the plant and the model. It is caused by physical properties of a filament in the bulb which requires certain time to incandesce from a cold startup. As this phase is over, PWA model correctly captures the optical channel of the plant and it can be employed for MPC design.

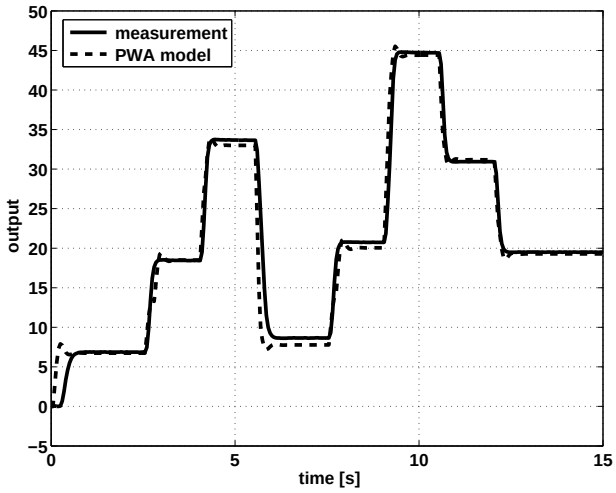


Fig. 6.12 Verification of PWA model.

Table 6.6 Matrices of the normalised model (6.52).

$\boldsymbol{A}_1$	$\boldsymbol{B}_1$	$\boldsymbol{f}_1$
$\boldsymbol{A}_2$	$\boldsymbol{B}_2$	$\boldsymbol{f}_2$
$\boldsymbol{A}_3$	$\boldsymbol{B}_3$	$\boldsymbol{f}_3$
$\boldsymbol{A}_4$	$\boldsymbol{B}_4$	$\boldsymbol{f}_4$

$=$

1.072	-0.464	0.277	-1.492
1	0	0	0
0.969	-0.431	0.485	-0.642
1	0	0	0
0.913	-0.410	0.616	0
1	0	0	0
0.868	-0.402	0.735	0.471
1	0	0	0

6.5.5.2 Control Design

Prediction Model

In order to prevent numerical issues when employing the PWA model for MPC synthesis, it is advised to perform coordinate transformation and normalisation. This can be achieved by introducing normalised variables x_1, x_2 and u as follows:

$$x_1 = \frac{v_{1,k} - v_{1,\text{ref}}}{\bar{v}_1}, \tag{6.50a}$$

$$x_2 = \frac{v_{2,k} - v_{2,\text{ref}}}{\bar{v}_2}, \tag{6.50b}$$

$$u = \frac{w_k - w_{\text{ref}}}{\bar{w}}. \tag{6.50c}$$

The suffix "ref" represents the desired steady state value, i.e.

$$v_{1,\text{ref}} = 32.09, \quad v_{2,\text{ref}} = 32.09, \quad w_{\text{ref}} = 3.5 \tag{6.51}$$

which is basically the linearisation point of the third dynamics (see Tab. 6.5) and $\bar{v}_1 = 3.67$, $\bar{v}_2 = 3.67$, $\bar{w} = 0.5$ are constants. Applying the normalisation, the transformed PWA model yields

$$\boldsymbol{f}_{\text{PWA}}(\boldsymbol{x}_k, u_k) = \boldsymbol{A}_i \boldsymbol{x}_k + \boldsymbol{B}_i u_k + \boldsymbol{f}_i \tag{6.52}$$

where $i = 1, 2, 3, 4$ and state update matrices are given in Tab. 6.6. The state space model (6.52) is associated with the following regions

$$\mathcal{D}_1 = \{\boldsymbol{x} \in \mathbb{R}^2 \mid -8.75 \leq x_2 \leq -5.16\} \tag{6.53a}$$

$$\mathcal{D}_2 = \{\boldsymbol{x} \in \mathbb{R}^2 \mid -5.16 \leq x_2 \leq -1.72\} \tag{6.53b}$$

$$\mathcal{D}_3 = \{\boldsymbol{x} \in \mathbb{R}^2 \mid -1.72 \leq x_2 \leq 1.88\} \tag{6.53c}$$

$$\mathcal{D}_4 = \{\boldsymbol{x} \in \mathbb{R}^2 \mid 1.88 \leq x_2 \leq 6.25\} \tag{6.53d}$$

Besides the dynamics as in (6.52), the following constraints are assumed to be imposed on the behaviour of the prediction model:

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^2 \mid -8.75 \leq x_1 \leq 6.25,\ -8.75 \leq x_2 \leq 6.25\} \tag{6.54a}$$
$$\mathcal{U} = \{u \in \mathbb{R} \mid -7 \leq u \leq 3\}. \tag{6.54b}$$

State constraints $\mathcal{X}$ are derived from the operating range of light intensity (6.48) and input constraints $\mathcal{U}$ represent the saturation limits (6.47).

Control Problem

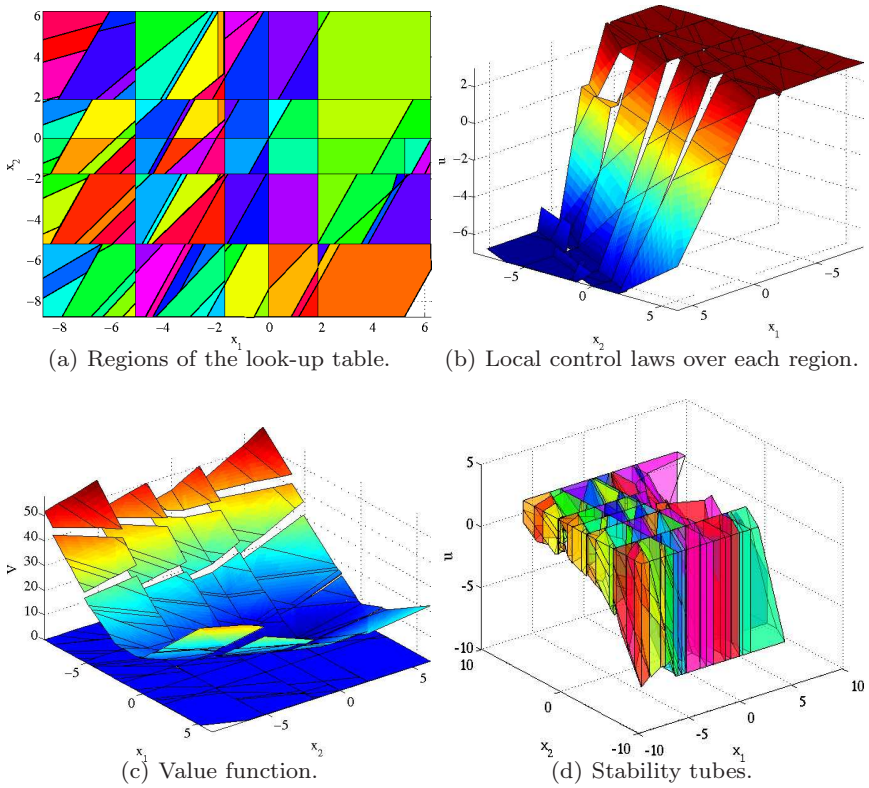
The aim of the control strategy is to find an optimal sequence of control inputs such that all system states are driven to a desired equilibrium. The equilibrium is given by the linearisation point for the third PWA dynamics (6.52) and in the transformed coordinates (6.50) it is exactly the origin, i.e. $x_1 = 0$, $x_2 = 0$, $u = 0$. Mathematically, the problem can be formulated as to find a sequence of future control moves $\boldsymbol{U}_N$ up to horizon N which steer the system states/input to the origin while satisfying constraints (6.54). More precisely,

$$\min_{U_N} \sum_{k=0}^{\infty} \|\boldsymbol{Q}_x \boldsymbol{x}_k\|_1 + \|Q_u u_k\|_1 \tag{6.55a}$$
$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}_{\text{PWA}}(\boldsymbol{x}_k, u_k) \tag{6.55b}$$
$$\boldsymbol{x}_k \in \mathcal{X} \tag{6.55c}$$
$$u_k \in \mathcal{U} \tag{6.55d}$$

where $\boldsymbol{x}_k = [x_1, x_2]^T$ represents the state vector, the function $\boldsymbol{f}_{\text{PWA}}(\cdot)$ describes the PWA model defined in (6.52) and the sets $\mathcal{X}, \mathcal{U}$ are the constraints on input and state variables given by (6.54). Due to the presence of switching rules in PWA model (6.52), the overall optimisation problem (6.55) is cast using additional binary variables as mpMILP. The problem is consequently solved using MPT toolbox (Kvasnica et al, 2004).

Explicit Solution

The problem (6.55) has been solved with parameters $\boldsymbol{Q}_x = \boldsymbol{I}$, $Q_u = 0.5$. The infinite choice of prediction horizon guarantees that the obtained MPC feedback law will provide closed-loop stability (Baotić et al, 2006). The resulting PWA control law builds a look-up table divided into 118 regions, defined in variables x_1, x_2, and these regions are plotted in Fig. 13(a). Over each one of these regions a local feedback law is defined as illustrated in Fig. 13(b). Similarly, the cost function is shown in Fig. 13(c). Note that in the case



(a) Regions of the look-up table. (b) Local control laws over each region.

(c) Value function. (d) Stability tubes.

Fig. 6.13 Explicit solution to Problem (6.55) consists of PWA map defined over 118 regions.

of multiparametric MILP solutions, the resulting PWA control law can be discontinuous (Fig. 13(b)) and defined over a nonconvex set. This is a consequence of using binary variables to encode the IF-THEN rules which describe behaviour of the PWA prediction model.

To implement the resulting look-up table in the on-line experiment, one has to store and evaluate the data. While storing part is limited by the available memory, the evaluation task is limited by the sampling time. The complexity of both tasks depends on the number of regions R. Assuming that we have enough memory to store the look-up table, one have to still evaluate the PWA law. In fact, this task comprises of two steps

1. region identification
2. evaluation of PWA law

from which the first part consumes the most time. Even with the use of binary search tree algorithm, where the evaluation time is logarithmic in R (Tøndel et al, 2003b), the scheme can still be prohibitive for implementation. Moti-

vated by this fact, the goal is to apply the polynomial approximation scheme presented where the whole look-up table is replaced by one polynomial, which is very cheap to implement. To do so, we have to find the set of all perturbations of the control law under which the closed loop renders stability. This will be implemented in the next section.

Polynomial Approximation

Using the polynomial approximation scheme the goal is to find a polynomial control law of the form

$$\tilde{\kappa}(\boldsymbol{x}) = (a_{11},\, a_{12}) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (a_{21},\, a_{22}) \begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix} + (a_{31},\, a_{32}) \begin{pmatrix} x_1^3 \\ x_2^3 \end{pmatrix} \tag{6.56}$$

which, when applied as a state feedback, guarantees closed-loop stability and constraint satisfaction. Theorem 6.12 provides a sufficient condition for existence of such a polynomial feedback law in the sense that if $(\boldsymbol{x}, \tilde{\kappa}(\boldsymbol{x})) \in \mathcal{S}(V, \beta),\ \forall \boldsymbol{x} \in \bigcup_i \mathcal{R}_i$, then $\tilde{\kappa}(\boldsymbol{x})$ will provide closed-loop stability and constraint satisfaction. Therefore the search for suitable polynomial coefficients of (6.56) can be cast as the following optimisation problem:

$$\min_{a_{11},\ldots,a_{32}} \sum_j \|(u(\boldsymbol{x}) - \tilde{\kappa}(\boldsymbol{x}))\|_2 \tag{6.57a}$$

$$\text{s.t.} \quad (\boldsymbol{x}, \tilde{\kappa}(\boldsymbol{x})) \in \mathcal{S}(V, \beta). \tag{6.57b}$$

From all possible choices of $\tilde{\kappa}(\boldsymbol{x})$ which satisfy (6.57b), cost function (6.57a) is used to select the coefficients which provide best approximation of the optimal feedback law $u(\boldsymbol{x})$.

The main advantage of the polynomial feedback law (6.56), compared to MPC controller based on evaluating PWA feedback law, is reduction of the total implementation and storage cost. On the storage side, only the coefficients a_{ij} need to be recorded in the memory, compared to storing the regions $\mathcal{P}_i$ and the feedback laws $\boldsymbol{F}_i$ and $\boldsymbol{g}_i$ for PWA feedback law. The on-line implementation cost is also greatly reduced, as only polynomial evaluation for a given $\boldsymbol{x}$ need to be performed to obtain a stabilising control action.

Table 6.7 Coefficients of the approximated polynomial (6.56).

$a_{11},\, a_{12}$	-0.8718,	-0.0007
$a_{21},\, a_{22}$	-0.0519,	0.0004
$a_{31},\, a_{32}$	0.0019,	0.0001

The approximation scheme has been applied to obtain polynomial control law of type (6.56) with help of YALMIP (Löfberg, 2004). Computed coefficients are given in Tab. 6.7.

Graphical representation of the computed polynomial of order 3 is shown in Fig. 14(b). To visibly see the differences comparing to optimal controller (shown in Fig. 14(a)), a cross-section through $x_2 = 0$ is provided in Fig. 6.15. Illustration of the approximation scheme is shown in Fig. 6.15 which represents a cross-section in stability tubes along the coordinate $x_2 = 0$. The polyhedral sets in Fig. 6.15 demonstrate the space of the stability tubes where there exist a stabilising control law according to Theorem 6.12. Inside this space the approximated polynomial (6.56) has been fitted and it is shown in Fig. 6.15 with a dashed line while the optimal control law is depicted with solid line.

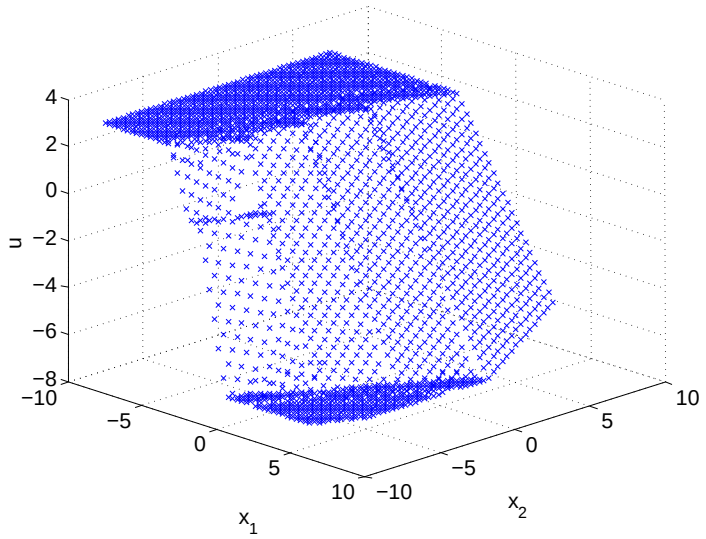
6.5.5.3 Real-Time Implementation

In this section computational requirements are evaluated for the optimal and approximated controller. Both controllers are applied in the real-time experiment and measured performance is discussed.

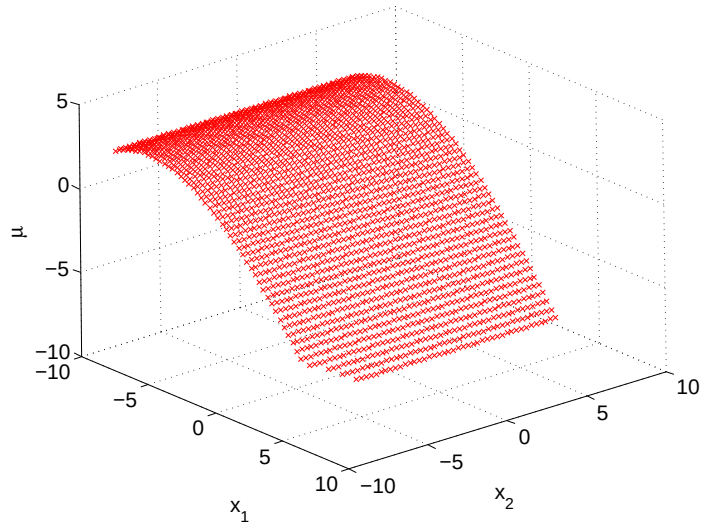
Computational Demands

Implementation of the optimal controller in the on-line experiment is limited by the sampling time $T_s = 0.05\,\text{s}$. If the look-up table, obtained previously and consisting of 118 regions, is stored and evaluated using the binary search tree algorithm Tøndel et al (2003a), the number of FLOPS which are required to evaluate such a controller for a given initial condition is at most 41. The memory requirements are 2832 bytes for the control law and 1536 bytes for the search tree which gives a total of 4368 bytes.

In the polynomial approximation scheme, the number of FLOPS depend on the degree of approximated polynomial and on the polynomial degree. By considering the polynomial (6.56) with degree of three, the upper bound for evaluation FLOPS is 14, less than a half of the runtime for the binary search tree. More prominent, however, is the drop in memory consumption. As state above, the explicit MPC solution with 118 regions requires 4368 bytes of memory storage, while to store the polynomial feedback law (6.56), mere 24 bytes of memory are required (6 polynomial coefficients, each of them consuming 4 bytes when represented as floating point numbers).



(a) Optimal control law.



(b) Polynomial with order 3.

Fig. 6.14 Optimal control law and polynomial approximation.

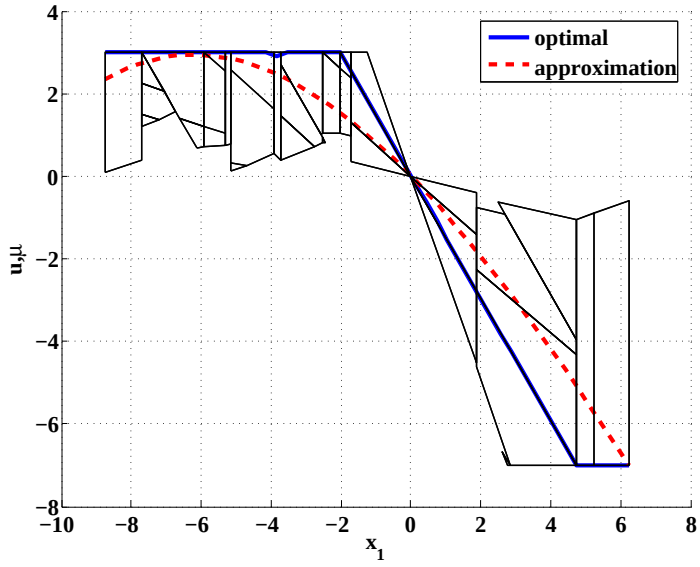


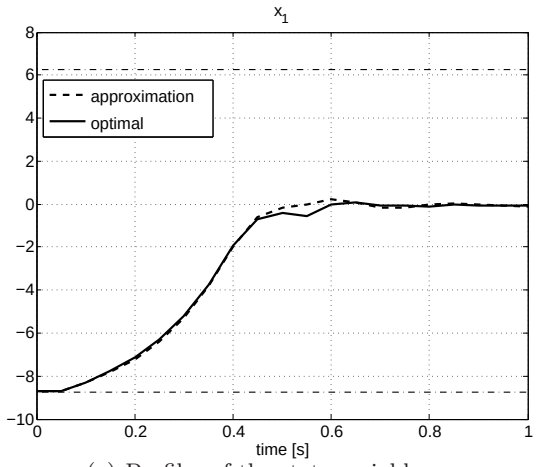
Fig. 6.15 Cross-section of the control laws through $x_2 = 0$.

Experimental Results

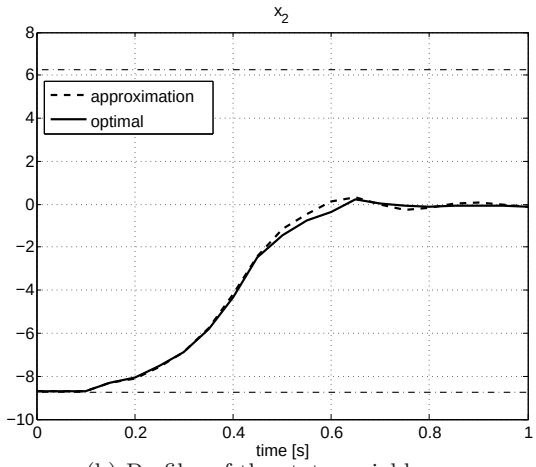
The optimal explicit MPC controller as well as the polynomial feedback strategy have been implemented in real time and obtained results are shown in Figs. 6.16 and 6.17. The plots represent the transition from the initial condition $\boldsymbol{x}_0 = (-8.7, -8.7)^T$ to the origin. Input signal generated by the optimal controller immediately jumps to the upper limit and then gently approaches the origin. In the polynomial controller this effect is different, the controller is slightly slower, but the same stabilising effect is achieved. State and input profiles converge to desired steady state, hence the control objective was met with both approaches. It is interesting to note that a polynomial controller acts better (in the sense of the selected performance criterion (6.55a)) than the optimal one. In particular, (6.55a) evaluates to 146.34 when the optimal MPC controller is used as a feedback, compared to the value of 142.96 for the case where the polynomial controller was used. This small difference can be attributed to the fact that the optimal controller is more sensitive to changes of the states. Nevertheless, the difference is small enough to say that both controllers share roughly the same performance while the approximated controller is significantly cheaper than the optimal one.

Performance of both controllers has not been tested on disturbance attenuation because this effect cannot be fully compensated by any of the used controllers since they do not contain an integration part. Moreover, these ef-

fects are too small to satisfactory evaluate the performance of both controllers while showing their advantages (e.g. constraint satisfaction).



(a) Profiles of the state variable x_1.



(b) Profiles of the state variable x_2.

Fig. 6.16 State profiles for optimal and polynomial controller.

Acknowledgements The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/0537/10 and of the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. It is 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.

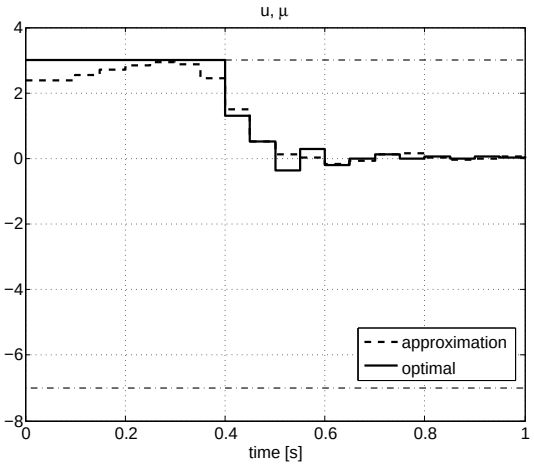


Fig. 6.17 Input profiles for optimal and polynomial controller.

References

Aurenhammer F (1991) Voronoi diagrams – survey of a fundamental geometric data structure. ACM Computing Surveys 23(3):345–405, DOI http://doi.acm.org/10.1145/116873.116880

Baotić M (2005) Optimal Control of Piecewise Affine Systems – a Multi-parametric Approach. Dr. sc. thesis, ETH Zurich, Zurich, Switzerland

Baotić M, Torrisi FD (2003) Polycover. Tech. Rep. AUT03-11, Automatic Control Laboratory, ETHZ, Switzerland

Baotić M, Christophersen F, Morari M (2006) Constrained optimal control of hybrid systems with a linear performance index. IEEE Transactions on Automatic Control 51(12):1903–1919

Baotić M, Christophersen FJ, Morari M (2006) Constrained Optimal Control of Hybrid Systems with a Linear Performance Index. IEEE Trans on Automatic Control In press

Bemporad A, Filippi C (2003) Suboptimal explicit RHC via approximate multiparametric quadratic programming. Journal of Optimization Theory and Applications 117(1):9–38

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

Borrelli F (2003) Constrained Optimal Control Of Linear And Hybrid Systems, Lecture Notes in Control and Information Sciences, vol 290. Springer

Christophersen FJ (2007) Optimal Control of Constrained Piecewise Affine Systems, Lecture Notes in Control and Information Sciences, vol 359. Springer Verlag

Čirka Ľ, Fikar M, Petruš P (2006) IDTOOL 4.0 - A Dynamical System Identification Toolbox for MATLAB/Simulink. In: 14th Annual Conference Proceedings: Technical Computing Prague 2006, pp 29–29, URL http://www.kirp.chtf.stuba.sk/~cirka/idtool/?IDTOOL

Cychowski M, 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, Morari M (2008) Optimal complexity reduction of polyhedral piecewise affine systems. Automatica 44(7):1728–1740

Grieder P, Wan Z, Kothare M, Morari M (2004) Two level model predictive control for the maximum control invariant set. In: American Control Conference, Boston, Massachusetts

Grieder P, Kvasnica M, Baotic M, Morari M (2005) Stabilizing low complexity feedback control of constrained piecewise affine systems. Automatica 41, issue 10:1683–1694

Hardy GH, Littlewood JE, Pólya G (1952) Inequalities, 2nd edn. Cambridge University Press

Huba M, Vrančič D (2007) Constrained control of the plant with two different modes. In: Mikleš J, Fikar M, Kvasnica M (eds) 16th Int. Conf. Process Control, pp paper Le–Tu–5, 205p.pdf

Huba M, Kurčík P, Kamenský M (2006) Thermo-optical device uDAQ28/LT. STU Bratislava, Illkovičova 3, Bratislava, in Slovak.

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

Jones C, Morari M (2009) Approximate Explicit MPC using Bilevel Optimization. In: European Control Conference, Budapest, Hungary

Kulhavý R, Kárný M (1984) Tracking of slowly varying parameters by directional forgetting. In: Proceedings of the 9th IFAC World Congres, Budapest, Hungary

Kvasnica M (2009) Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools. VDM Verlag, Saarbruecken

Kvasnica M, Grieder P, Baotić M (2004) Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/~mpt/

Kvasnica M, Christophersen FJ, Herceg M, Fikar M (2008) Polynomial approximation of closed-form MPC for piecewise affine systems. In: Proceedings of the 17th IFAC World Congress, Seoul, Korea, pp 3877–3882

Kvasnica M, Rauová I, Fikar M (2011) Separation functions used in simplification of explicit mpc feedback laws. In: Huba M, Skogestad S, Fikar M, Hovd M, Johansen TA, Rohal'-Ilkiv B (eds) Preprints of the NIL workshop: Selected Topics on Constrained and Nonlinear Control, STU Bratislava – NTNU Trondheim, pp 48–53

Lazar M, de la Pena DM, Heemels W, Alamo T (2008) On input-to-state stability of min-max nonlinear model predictive control. Systems & Control Letters 57:39–48, DOI 10.1016/j.sysconle.2007.06.013

Löfberg J (2004) YALMIP. Available from http://control.ee.ethz.ch/~joloef/yalmip.php

Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: Stability and optimality. Automatica 36(6):789–814

Mikleš J, Fikar M (2007) Process Modelling, Identification, and Control. Springer Verlag, Berlin Heidelberg

Pannocchia G, Rawlings J, Wright S (2007) Fast, large-scale model predictive control by partial enumeration. Automatica 43:852–860

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

Scibilia F, Olaru S, Hovd M (2009) Approximate explicit linear MPC via Delaunay tessellation. In: Proceedings of the 10th European Control Conference, Budapest, Hungary

Spjøtvold J, Tøndel P, Johansen T (2005) A method for obtaining continuous solutions to multiparametric linear programs. In: Proceedings of IFAC World Congress, IFAC World Congress, accepted for publication

Tøndel P, Johansen T, Bemporad A (2003a) An algorithm for multiparametric quadratic programming and explicit mpc solutions. Automatica 39(3):489–497

Tøndel P, Johansen TA, Bemporad A (2003b) Evaluation of Piecewise Affine Control via Binary Search Tree. Automatica 39(5):945–950

Valencia-Palomo G, Rossiter J (2010) Using Laguerre functions to improve efficiency of multi-parametric predictive control. In: Proc. American Contr. Conf., Baltimore, USA

Vidyasagar M (1993) Nonlinear Systems Analysis, 2nd edn. Prentice Hall

Chapter 7
Predictive Control of Mechatronic Systems with Fast Dynamics

Tomáš Polóni and Gergely Takács and Boris Rohal'-Ilkiv

Abstract This chapter covers the model predictive control of mechatronic systems with fast dynamics. A vibration system and an internal combustion engine are presented as demonstration examples of such systems. High sampling rates are common requirement in both applications, what makes the utilization of the computationally intensive MPC techniques more difficult. A comparison of optimal and sub-optimal MPC strategies providing guaranteed stability and constraint feasibility is presented in the active vibration attenuation of lightly damped mechanical systems. This problem area is connected with another challenging problem: namely how to design a moving horizon observer suitable for monitoring the dynamics of the vibrating system. Such an observer is an essential part of the considered model predictive controllers. In the end of the chapter a vital problem of internal combustion engines; the air-to-fuel ratio control has been analyzed using a multi-model predictive control methodology.

Tomáš Polóni
Institute of Measurement, Automation and Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, e-mail: tomas.poloni@stuba.sk

Gergely Takács
Institute of Measurement, Automation and Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, e-mail: gergely.takacs@stuba.sk

Boris Rohal'-Ilkiv
Institute of Measurement, Automation and Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, e-mail: boris.rohal-ilkiv@stuba.sk

7.1 Introduction

Mechatronic systems represent a very close integration of mechanical and electronic systems together with information technologies. All of this is enhanced with mutual synergistic actions of the individual components. These systems are intended for a wide area of use both in industry and everyday life. Some examples are machine-building, individual mechanical components, machines, automotive engineering and up to the spectrum of precision micro-electro-mechanical components (MEMS).

With increasing the power and miniaturization of microelectronics it is possible to continuously increase the degree of hardware integration as well. This in practice means the direct incorporation of sensors, actuators and microcomputers to electro-mechanical, predominantly nonlinear systems. This hardware integration takes place side by side with an increasing degree of software integration. This process is running on the basis of information processing, which primarily rests upon the development of new advanced control functions for these mechatronic systems.

Apart from the basic feedforward and feedback control functions a new contribution may be made by utilizing a wider knowledge basis, consisting of mathematical models of the controlled plants, algorithms for identification, state and parameters estimation, methods for design of new advanced control algorithms with more sophisticated behaviour, new criteria for looking on the control performance and efficiency of the systems etc... The new and modern solutions of the advanced control structures for the mechatronic systems always lead to a direct, on-line information (signal) processing in real time, which subsequently must be adapted to the features of the mechanical process. At the same time, various basic needs of the systems design such must be respected. Some of these are: limitations in computing possibilities due to real time restrictions; mechanical processes nonlinearities; bounded rates and bounded amplitudes of the actuators; robustness and transparency of resulting control functions algorithms etc...

The present stay-of-the-art (Isermann, 2005) in the development of mechatronic systems control functions is the characteristic usage of direct algorithms of the feedforward and feedback type, with fixed - non-adaptive - parameters setting at the proportional, proportional integration or proportional integration derivative action. The potential nonlinearities are mostly taken into account using in advance elaborated (presetting, non-adaptive) look-up tables or maps.

Very often simple, discrete (two-state) controllers are employed for systems control. For many applications, these solutions fail in ensuring the desired level of control quality for example with respect to reference signals changes, or are not able to sufficiently compensate the outer disturbances caused by variable system loads. Moreover these solutions may not have the sufficient robustness in order to cope with internal and external uncertainties. A serious practical problem of the current mechatronic system control structures

is their insufficient ability to actively respect various limitations, physical and constructional constraints set on the input/manipulated, state or output system variables, design requirements and operational conditions.

The content of this chapter is orientated towards the employment of (mathematical models based) predictive principles for the design and development of new intelligent and robust control functions and algorithms, taking into account specific needs and limitations of the mechatronic systems with fast dynamics. Potential applications are particularly aimed at the usage of mechatronic systems and modules in automotive accessories such as control of combustion engines, anti-lock-braking systems, active suspension, electro-hydraulic brakes etc. . . and in automotive industry such as automated robot assembly lines for cars and various automotive components, for control of MEMS and for usage in other industrial areas also.

The specific mechatronic systems used as examples and analyzed in this chapter are taken from the field of computationally efficient MPC of vibration systems and internal combustion engines. These examples are used to introduce recent improvements and upcoming challenges in the field of efficient MPC.

The first subchapter, that is 7.2 covers recent research on the topic of active vibration attenuation via computationally efficient model predictive control. This specific area is narrowed further to the control of large and under-damped mechanical structures. Such structures are for example helicopter rotor beams, wing surfaces, antenna masts and others. The first part of this section will cover a survey of different MPC algorithms which provide guaranteed stability and constraint feasibility and applied to a vibration attenuation setup. Later on through the use of a laboratory example, vibration damping performance will be matched for the different methods. In addition to that, the off-line and on-line computational properties and some implementation challenges are evaluated as well.

Following this, in subchapter 7.3 the focus will be moved from model predictive control to another very exciting and closely related topic, namely moving horizon observers (MHO). The design of moving horizon observer will be considered for the vibration attenuation system using the least-squares estimation of state and parameters combining with a pre-filtering technique.

Finally in subchapter 7.4 the formulation of multi-model predictive approach to control of air-fuel ratio of an internal combustion engine is described and analyzed. The quality of air-fuel ratio control strongly influences key vehicle attributes such as emissions, fuel economy and drivability. The proper air-fuel ratio control is necessary for efficient conversion of the engine exhaust gases performed by the three-way catalyst. The maximum efficiency of the three way catalyst is reach in a narrow region where the fuel is matched to air quantity in stoichiometric proportion. The main challenges in the design of an air-fuel controller include the engine nonlinear dynamics and variable time delays.

Parts of the subchapters have been originally presented in (Takács and Rohal'-Ilkiv, 2009b), (Polóni et al, 2010) and (Polóni et al, 2007).

7.2 Comparison of Model Predictive Control Methods for Lightly Damped Vibrating Systems

7.2.1 Introduction

Undesirable structural vibrations are present in countless real life engineering applications. Mechanical vibrations may limit the effective system and subsystem lifespan, cause safety concerns, and other issues.

For long years, engineers have been successfully utilizing passive vibration attenuation techniques. Passive structural changes for example the manipulation of structural stiffness and weight are often self-evident, however in many cases not viable. Such passive vibration attenuation treatment is especially impractical if not impossible for systems with dominant low frequency range responses (Preumont, 2002).

The introduction of semi-active suppression methods has brought numerous improvements in comparison with the passive vibration treatment techniques. The absence of expensive and bulky control hardware may even present advantages in comparison with fully active methods. However semi-active vibration suppression is considered to be relatively ineffective when weighted against the fully active approach.

Within the last two decades active vibration suppression has become an attractive way to completely eliminate or significantly reduce unwanted mechanical vibrations. Numerous excellent books and hundreds of publications have appeared since, describing engineering problems where vibration suppression is deemed necessary (Preumont, 2002; Inman, 2006). With the advent of smart materials research, the range of possible actuators has significantly multiplied as well.

The extensive asymmetry of modest actuator capabilities and the substantial range of expected deformations in lightly damped structures suggests an own class engineering problems in vibration attenuation. The class of problems suggested above carries special meaning if model predictive control (MPC) with guaranteed stability is considered as the control algorithm. An excellent example for such lightly damped structures are helicopter rotor beams.

Active rotor beam designs with embedded actuators may stabilize aircraft flight, reduce vibrations and improve fuel efficiency. Examples of lightly damped mechanical structures amongst others are solar panels on satellites, wing surfaces, antenna masts and large manipulator arms on spacecraft.

Amongst different sensor and actuator types, placement optimization procedures available in scientific literature for active vibration attenuation one of the most important part of the control system seems to be slightly overlooked: the control algorithm itself. The well established, albeit simple positive-position feedback (PPF) and strain-rate feedback (SRF) still seems to be the most popular control strategy choice (Sloss et al, 2003; Song et al, 2002).

7.2.1.1 The Need for MPC in Vibration Attenuation

Practical engineering problems and actuators hold inherent limitations, thus control moves must be constrained to prevent issues connected with safety, economy and component life-span. Piezoceramics are amongst the most often utilized smart materials in vibration attenuation. They are utilized both as sensors and more importantly actuators. If a certain maximal voltage limit is exceeded, the piezoelectric material may fail or behave unexpectedly due to depolarization effects. Model predictive control is currently the only control method able to handle process constraints on an algorithmic level (Rossiter, 2003).

Optimal performance along with constraint feasibility and stability guarantees is certainly a positive addition to any controlled system, and that is not any different in the case of active vibration attenuation. One might argue, that mechanical systems such as actively controlled wing surfaces are inherently stable and if subjected to an initial excitation will return to their equilibrium position. However due to the presence of constraints and select type of excitation scenarios the controlled vibrational system may become unstable, that is why the use of MPC with guaranteed stability is highly recommended.

The real-time implementation of MPC algorithms in vibration damping however comes with a steep price: the two most significant limiting factors are fast sampling times - a common feature of vibration damping applications and the excessive asymmetry in actuator capability and expected deflection ranges in lightly damped structures.

MPC without constraint handling does not require on-line optimization procedures such as quadratic programming (QP), thus its implementation in high sampling rate applications is problem free. In fact the saturated LQ controller as an analogy of an infinite-horizon closed MPC law works very effectively in high sampling rate vibration attenuation systems.

Constrained MPC without feasibility and stability guarantees may be used in high sampling rate applications, and it has been already implemented in active vibration suppression (Hassan et al, 2007). The asymmetry between actuator capabilities and large deflections in lightly damped structures is only an issue if stability and feasibility guarantees are considered. In the work by Wills et al (2008), optimal quadratic-programming based MPC has been applied to a flexible lightly damped beam however without consider-

ing stability or feasibility guarantees. The work achieved significant sampling rates of approximately 5 khz, using a high order model and implemented on a digital signal processing board. The controller effectively damped the first five transversal modes of the vibrating cantilever, while filtering out the uncontrollable resonances. A very different approach has been demonstrated by Niederberger (2005), where a simplified hardware representation of the explicit pre-computed MPC control law has been employed to control a flexible beam; again without stability guarantees.

This section investigates implementation properties of different model predictive control algorithms in vibration control such as on-line running time and damping performance. Contrary to the previously discussed works implementing MPC in vibration control, the methods considered here do include feasibility and stability guarantees. It will be suggested that high sampling rates encountered in vibration control along with considerable actuator and deflection range asymmetry define a class of problems, leading to several practical implementation difficulties in MPC algorithms.

The present section demonstrates experimental vibration damping performance, algorithm execution timing properties, and implementation difficulties of three different MPC approaches on a laboratory model. This laboratory model will emulate the characteristic mechanical properties of lightly damped structures. The algorithms considered here include the traditional infinite horizon dual-mode quadratic programming based MPC (QPMPC), pre-computed optimal multi-parametric MPC (MPMPC) and the efficient albeit sub-optimal Newton-Raphson's MPC (NRMPC). The MPC algorithms will be matched against a simple saturated LQ controller, which will give a good indication on the achievable lower limits of sampling times on the given hardware and provide a base line to compare damping efficiency.

7.2.2 Definition of the Demonstration Problem

A small - scale laboratory model has been created to emulate the mechanical properties of large, lightly - damped flexible vibrating active systems (Takács and Rohal'-Ilkiv, 2009a). An excellent example of a lightly damped flexible structure which may be modelled by a simple beam are helicopter rotor beams. This laboratory model is essentially a clamped cantilever beam with bonded piezoelectric actuators. In the current configuration a laser triangulation based distance sensor is providing position information to the feedback system.

The aim of the controller can be summarized as follows: The applied MPC controller with guaranteed constraint feasibility and stability must minimize the deflection measured at the beam tip, that is minimizing the first mode vibration amplitudes.

Control is carried out through one input signal, therefore the system will be controlled as a single-input, single-output linear system. In order to enable the implementation of the quadratic and multi-parametric programming based MPC methods, the model order and sampling time is limited. The sampling rate necessary to cover the first resonant frequency of 8.127 Hz by a second order model is 100 Hz. The state-space model describing beam dynamics is a simple second order system.

To emulate the difference between actuator capabilities and expected structural deformations, a large range of allowable beam tip deflections is considered. This in turn produces a large region of attraction in the MPC law. While the piezoelectric actuators supplied with voltages meeting polarization limits may generate only a static deflection approximately in the range of ± 0.15 mm, beam resonance measured at the tip in the first mode easily exceeds ± 15 mm. The region of attraction defined by the MPC law must be able to cover this range of deflections, thus all states measured in within this specification must be included in the set of all feasible states - that is the region of attraction.

Vibration damping performance, real-time execution timing properties and implementation possibilities are compared for the following MPC controllers all offering guaranteed stability and constraint feasibility in experiments:

- dual-mode quadratic programming based MPC (QPMPC)
- multi-parametric programming based pre-computed MPC (MPMPC)
- Newton-Raphson's computationally efficient sub-optimal MPC (NRMPC)
- and finally a saturated linear quadratic (LQ) controller (serving as a basis of comparison both for performance and for timing)

According to the problem definition above, all of the MPC algorithms must cover the same region of attraction, running on the same implementation software, utilize the same linear time-invariant state-space prediction model, utilize identical state observers, penalizations and other possible applicable settings.

The algorithms specified above shall be verified in various excitation situations both in the time and frequency domain, effectively comparing timing properties and damping behaviour.

7.2.3 Theoretical Summary

This subsection will give a brief theoretical summary on the basis of the algorithms implemented and tested here. A full treatment of the individual theoretical topics exceeds the scope of this chapter, therefore the reader should refer to the referenced works such as Maciejowski (2002); Rossiter (2003); Mayne et al (2000); Chen and Allgöver (1998); Kouvaritakis et al (2000, 2002); Cannon and Kouvaritakis (2005) ... etc. for more information.

7.2.3.1 Traditional Quadratic Programming Based MPC in Vibration Control

Stability in the traditional quadratic programming (QP) based MPC formulation is guaranteed through suitably formulated state feedback and terminal cost function matrices and the deployment of dual mode predictions: the first mode considers n_c free control moves, while the second mode assumes the LQ control law (Mayne et al, 2000; Chen and Allgöver, 1998). Feasibility of process constraints is ensured beyond the prediction horizon by the inclusion of a constraint checking horizon of n_a steps. These additional constraints define a polytopic set, called the region of attraction. For a given state $\boldsymbol{x}_k$ to be a feasible input to the QP optimization problem, it must be contained within this region of attraction. The fixed state feedback matrix used to ensure stability defines a polytopic terminal set contained within the maximal admissible set. If the actual measured system state $\boldsymbol{x}_k$ is inside this terminal set, the fixed state control law is in effect.

First, let us consider a system described by a linear, time - invariant (LTI) state-space model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{7.1}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k \tag{7.2}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is a state vector, $\boldsymbol{u}_k \in \mathbb{R}^m$ is an input vector and $\boldsymbol{y}_k \in \mathbb{R}^p$ is a an output vector. Matrices $\boldsymbol{A}$,$\boldsymbol{B}$ and $\boldsymbol{C}$ are the state transition matrix, input and output matrix, and integer k denotes sampling instances. Direct feed trough is omitted as in most real feedback control system models. The controlled system is subject to the following constraints:

$$\overline{\boldsymbol{y}} \leq \boldsymbol{y}_k \leq \underline{\boldsymbol{y}} \tag{7.3}$$

$$\overline{\boldsymbol{u}} \leq \boldsymbol{u}_k \leq \underline{\boldsymbol{u}} \tag{7.4}$$

$$\overline{\boldsymbol{x}} \leq \boldsymbol{x}_k \leq \underline{\boldsymbol{x}} \tag{7.5}$$

where the under and over lines denote lower, respectively upper bounds. If we would like to steer system (7.1) into the origin, we may define the following linear quadratic programming problem:

The stability issue of MPC has been comprehensively treated in numerous works like for example Mayne et al (2000); Chen and Allgöver (1998); Rossiter (2003); Maciejowski (2002). This paper assumes the most typical method to guarantee stability: to use the state feedback gain $\boldsymbol{K}$ and terminal cost matrix $\boldsymbol{P}$ as the solution of the unconstrained, infinite horizon quadratic regulation problem (Pistikopoulos et al, 2007):

$$\boldsymbol{K} = -\left(\boldsymbol{R} + \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{A} \tag{7.15}$$

$$\boldsymbol{P} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})^T\boldsymbol{P}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}) + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K} + \boldsymbol{Q} \tag{7.16}$$

To find the solution of the model predictive control problem, perform the following set of operations at each sampling instant:

- Observe or measure actual system state at sample $\boldsymbol{x}_k$.
- Minimize the following cost function with respect to constraints:

$$\min_{\boldsymbol{u}} J(\boldsymbol{u}, \boldsymbol{x}_k) = \sum_{i=0}^{n_c-1} \left(\boldsymbol{x}_{k+i}^T \boldsymbol{Q} \boldsymbol{x}_{k+i} + \boldsymbol{u}_{k+i}^T \boldsymbol{R} u_{k+i}\right) + \boldsymbol{x}_{k+n_c}^T \boldsymbol{P} \boldsymbol{x}_{k+n_c} \tag{7.6}$$

where $\boldsymbol{u} = [u_i, u_{i+1}, \ldots, u_{i+n_c-1}]$ is a vector of predicted control inputs, $\boldsymbol{Q} = \boldsymbol{Q}^T \geq 0$ is a state penalization matrix, $\boldsymbol{R} = \boldsymbol{R}^T \geq 0$ is an input penalization matrix and n_c is a prediction horizon. The typical MPC cost function must be subject to the following constraints:

$$\overline{\boldsymbol{y}} \leq \boldsymbol{y}_i \leq \underline{\boldsymbol{y}}, \qquad i = 1, 2, \ldots, n_c + n_a \tag{7.7}$$

$$\overline{\boldsymbol{u}} \leq \boldsymbol{u}_i \leq \underline{\boldsymbol{u}}, \qquad i = 1, 2, \ldots, n_c + n_a \tag{7.8}$$

$$\overline{\boldsymbol{x}} \leq \boldsymbol{x}_i \leq \underline{\boldsymbol{x}}, \qquad i = 1, 2, \ldots, n_c + n_a \tag{7.9}$$

$$\boldsymbol{x}_{k+0} = \boldsymbol{x}_k \tag{7.10}$$

$$\boldsymbol{x}_{k+i+1} = \boldsymbol{A}\boldsymbol{x}_{k+i} + \boldsymbol{B}\boldsymbol{u}_{k+i}, \qquad i \geq 0 \tag{7.11}$$

$$\boldsymbol{y}_{k+i} = \boldsymbol{C}\boldsymbol{x}_{k+i}, \qquad i \geq 0 \tag{7.12}$$

$$\boldsymbol{u}_{k+i} = \boldsymbol{K}\boldsymbol{x}_{k+i}, \qquad i \geq n_c \tag{7.13}$$

$$\tag{7.14}$$

where $\boldsymbol{K}$ is a stabilizing feedback gain and n_a is the additional constraint checking horizon.

- Apply the first element of the vector of optimal control moves $\boldsymbol{u}$ to the controlled system, and re-start the procedure.

By iterating (7.1) into the future, one may construct prediction matrices and substitute those to the cost function, obtaining the following QP minimization problem:

$$J^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \boldsymbol{u}^T \boldsymbol{H} \boldsymbol{u} + \frac{1}{2} \boldsymbol{x}_k^T \boldsymbol{G} \boldsymbol{x}_k + \boldsymbol{x}_k^T \boldsymbol{F} \boldsymbol{u} \right\} \tag{7.17}$$

subject to the following constraints:

$$\boldsymbol{W_c u} \leq \boldsymbol{w_c} + \boldsymbol{V_c x_k} \tag{7.18}$$

where matrices $\boldsymbol{H}, \boldsymbol{G}, \boldsymbol{F}$ in (7.17) and matrices $\boldsymbol{W_c}, \boldsymbol{w_c}, \boldsymbol{V_c}$ in (7.18) can be uniquely and trivially determined from the prediction matrices; $\boldsymbol{Q}$, $\boldsymbol{R}$ and the relations defined by (7.6)-(7.18).

The formulation and implementation particulars of traditional infinite-horizon dual-mode quadratic programming based model predictive control are well known, therefore only theoretical basics are included here.

7.2.3.2 Multi-parametric MPC

The multi-parametric MPC approach takes advantage of the fact that MPC is a constrained linear piecewise - affine (PWA) problem. For an MPC controller explicit solutions can be calculated off-line by partitioning the state-space and associating a PWA control law with each individual region. This means that at the implementation stage the actual measured or observed state is associated with a region and this is followed by evaluating only a piecewise-linear function. The main drawback of this approach is that off-line computational time and memory requirements grow exponentially with increased problem dimensions. A problem of dimensionality above 10 including the prediction horizon becomes difficult to manage (Maciejowski, 2002). Possible ways how to reduce the complexity problem in explicit model predictive control are much deeper analyzed in Chapter 5.

The constrained quadratic programming MPC problem is solved beforehand, in an off-line regime using multi-parametric programming. The solution assumes the form of a piecewise-affine state feedback control law, which may be represented in a form:

$$\boldsymbol{u}_k(\boldsymbol{x}_k) = \begin{cases} \boldsymbol{K_1}\boldsymbol{x}_k + \boldsymbol{g_1} & \text{if} \boldsymbol{x}_k \in \mathcal{R}_1 \\ \boldsymbol{K_2}\boldsymbol{x}_k + \boldsymbol{g_2} & \text{if} \boldsymbol{x}_k \in \mathcal{R}_2 \\ \boldsymbol{K_3}\boldsymbol{x}_k + \boldsymbol{g_3} & \text{if} \boldsymbol{x}_k \in \mathcal{R}_3 \\ \vdots & \vdots \\ \boldsymbol{K_N}\boldsymbol{x}_k + \boldsymbol{g_N} & \text{if} \boldsymbol{x}_k \in \mathcal{R}_N \end{cases} \tag{7.19}$$

where $\boldsymbol{x}_k$ is the observed system state, this is in fact acting as input to the controller function. Matrices $\boldsymbol{K_i}$ and vectors $\boldsymbol{g_i}$ define a fixed feedback and a shift constant for the given control law. The current measured state is a part of a polyhedral region $\mathcal{R}_i$, the sum of these sets forms a polyhedral partition $\mathcal{P} = \mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \ldots, \mathcal{R}_N$ in state-space. The polyhedral sets $\mathcal{R}_i$ are characterized by intersections of half spaces in hyperspace:

$$\mathcal{R}_i = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{A}_{\mathcal{R}i}\boldsymbol{x} \leq \boldsymbol{b}_{\mathcal{R}i}\} \tag{7.20}$$

These sets intersect only at the boundaries, therefore in case the problem is feasible, the current state $\boldsymbol{x}_k$ unambiguously belongs to one of the partitions. If the partition is found, the state can be associated with a PWL control law. An explicit MPC controller is defined by the following data: $\{\boldsymbol{K_i}, \boldsymbol{g_i}, \boldsymbol{A}_{\mathcal{R}i}, \boldsymbol{b}_{\mathcal{R}i}\}_{i=1}^{N}$.

The off-line multi-parametric optimization process to calculate an explicit MPC controller may be summarized by the following algorithm (Rossiter, 2003):

During the on-line control process, repeated at each sampling interval the set $\mathcal{R}_i$ defining a region corresponding to the actual state is found. Next the function (7.19) corresponding to the polyhedral set $\mathcal{R}_i$ is employed to

To find the solution of the MPMPC problem off-line, given a linear time invariant system and process settings, perform the following task (once):

- For all feasible active sets, define a polyhedral region $\mathcal{R}_i$ such, that in case a given state $\boldsymbol{x}_k \in \mathcal{R}_i$, then the control course $\boldsymbol{u}_k = \boldsymbol{K_i}\boldsymbol{x}_k + \boldsymbol{g}_i$ is optimal and feasible. The sum of regions $\mathcal{R}_i$ is the region of attraction or admissible set $\mathcal{P}$.
- Reduce regions $\mathcal{R}_i$ to prevent overlaps or duplications.
- Store the region look up table $\mathcal{R}_i$ and the corresponding PWL functions for the on-line controller.

calculate the control output. Input to the system is simply the function of the current state $\boldsymbol{u}_k = f(\boldsymbol{x}_k)$. This is a computationally efficient process: with low problem dimensionality currently available hardware is capable of high sampling speeds. The on-line algorithm may be summarized according to:

To find the MPC controller output, evaluate the MPMPC problem on-line. At each sampling instant perform the following tasks:

- Measure or observe current state of the system.
- Identify the index i of the polyhedral region, such that $\boldsymbol{x}_k \in \mathcal{R}_i$.
- Evaluate the PWL function corresponding to the index i: $\boldsymbol{u}_k = \boldsymbol{K_i}\boldsymbol{x}_k + \boldsymbol{g}_i$.

7.2.3.3 Newton-Raphson's Sub-optimal MPC in Vibration Control

The computationally efficient Newton-Raphson based suboptimal MPC controller guarantees stability and constraint feasibility for linear systems. This is realized through the use of an ellipsoidal region of attraction and terminal set to approximate the optimal polyhedral regions. This type of controller has been proposed by Kouvaritakis et al (2000), later optimality has been slighly improved by Kouvaritakis et al (2002). The on-line optimization task is practically reduced to an univariate minimization problem: efficiently solvable by a simple Newton-Raphson algorithm.

Cannon and Kouvaritakis (2005) introduced a method which enables the region of attraction to be maximized while leaving the prediction horizon equal to the prediction model order. The convex problem formulation by Cannon and Kouvaritakis (2005) preserves all the computational advantages of NRMPC while recovering the largest possible region of attraction. This efficient MPC formulation with an optimized maximal admissible set is highly useful for structural vibration control of lightly damped structures.

NRMPC involves augmenting state-space by a perturbation vector $\boldsymbol{f}_k = [c_k \;\; c_{k+1} \;\; \ldots \;\; c_{k+n_c}]^T$. This perturbation vector assumes a zero value with

inactive constraints. Feedback loop optimality is arbitrary: for example a linear quadratic (LQ) state feedback is a good choice. During transients this loop is not optimal any more and the perturbation vector $\boldsymbol{f}_k$ will assume a non-zero value:

$$\boldsymbol{u}_k = \boldsymbol{K}\boldsymbol{x}_k + \boldsymbol{E}\boldsymbol{f}_k \qquad \boldsymbol{x}_{k+1} = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{f}_k \tag{7.21}$$

where $\boldsymbol{\Phi} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})$. In case prediction dynamics is optimized as well, vector $\boldsymbol{E}$ is full, otherwise its role is to select the first element c_k of the perturbation vector $\boldsymbol{f_k}$. Using this formulation we can form an autonomous state-space equation with pre-stabilized dynamics:

$$\boldsymbol{z}_{k+1} = \boldsymbol{\Psi}\boldsymbol{z}_k \qquad \boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Phi} & \boldsymbol{B}\boldsymbol{E} \\ \boldsymbol{0} & \boldsymbol{T} \end{pmatrix} \tag{7.22}$$

where $\boldsymbol{z}_k = [\boldsymbol{x}_k \quad \boldsymbol{f}_k]^T$ is the augmented state vector and "$\boldsymbol{0}$" a matrix of zeros. $\boldsymbol{T}$ acts as a shift matrix without optimizing prediction dynamics. If the optimization of prediction dynamics is considered it is full and a variable of the off-line optimization procedure. To preserve numerical stability of the on - line NRMPC controller algorithm, the size of the region of attraction is limited by enforcing a bound for the predicted cost for each initial condition at all times: $J(\boldsymbol{u}, \boldsymbol{x}_k) \leq \gamma$. If an invariant ellipsoid in the augmented state-space is defined by $\varepsilon_z = \left\{\boldsymbol{z} | \boldsymbol{z}^T \boldsymbol{S} \boldsymbol{z} \leq 1\right\}$, then the invariance condition can be expressed by:

$$\boldsymbol{S} - \boldsymbol{\Psi}^T \boldsymbol{S} \boldsymbol{\Psi} > \frac{1}{\gamma} \begin{bmatrix} \boldsymbol{C}^T & \boldsymbol{K}^T \\ \boldsymbol{0} & \boldsymbol{E}^T \end{bmatrix} \boldsymbol{\Omega} \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{K} & \boldsymbol{E} \end{bmatrix} \tag{7.23}$$

where matrix $\boldsymbol{\Omega}$ is a diagonal block matrix containing $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{I}, \boldsymbol{R})$, with input penalization of $\boldsymbol{R}$.

To simplify matters, let us consider symmetric constraints over the input in the form $|\boldsymbol{u}_k| \leq \bar{\boldsymbol{u}}$. The feasibility condition for ε_z is then defined by:

$$\begin{bmatrix} \bar{\boldsymbol{u}}^2 & \begin{bmatrix} \boldsymbol{K} & \boldsymbol{E} \end{bmatrix} \\ * & \boldsymbol{S} \end{bmatrix} \geq 0 \tag{7.24}$$

where the symmetric part of the matrix is denoted by $*$.

Cannon and Kouvaritakis (2005) introduces a non-linear transformation of variables, which allows the invariance and feasibility conditions to be modified in a way that the optimization of prediction dynamics is possible. The transformation preserves the convexity of the optimization problem. Instead of treating expressions $\boldsymbol{E}$ and $\boldsymbol{T}$ as selection and shift matrices, we may include them in the off-line optimization as variables.

The augmented invariant set ε_z is described by matrix $\boldsymbol{S}$ which according to Cannon and Kouvaritakis (2005) may be expressed in the following way:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{X}^{-1} & \boldsymbol{X}^{-1}\boldsymbol{W} \\ \boldsymbol{X}^{-1}\boldsymbol{W}^{T} & \bullet \end{bmatrix} \qquad \boldsymbol{S}^{-1} = \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{V} \\ \boldsymbol{V}^{T} & \bullet \end{bmatrix} \tag{7.25}$$

$$\boldsymbol{N} = \boldsymbol{W}\boldsymbol{T}\boldsymbol{V}^{T} \qquad \boldsymbol{M} = \boldsymbol{E}\boldsymbol{V}^{T} \tag{7.26}$$

The $\bullet$ symbol denotes blocks of $\boldsymbol{S}$ and $\boldsymbol{S}^{-1}$ uniquely determined by $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{W}$ and $\boldsymbol{V}$.

Using relations (7.25) and (7.26) in the original invariance and feasibility conditions (7.23) respectively (7.24) result in the modified invariance and feasibility conditions with optimized prediction dynamics:

$$\begin{bmatrix} \gamma\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{\Omega}^{1/2}\begin{bmatrix} \boldsymbol{C}\boldsymbol{Y} & \boldsymbol{C}\boldsymbol{X} \\ \boldsymbol{K}\boldsymbol{Y}+\boldsymbol{M} & \boldsymbol{K}\boldsymbol{X} \end{bmatrix} \\ * & \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X} & \boldsymbol{Y} \end{bmatrix} & \begin{bmatrix} \boldsymbol{\Phi}\boldsymbol{Y}+\boldsymbol{B}\boldsymbol{M} & \boldsymbol{\Phi}\boldsymbol{X} \\ \boldsymbol{N}+\boldsymbol{\Phi}\boldsymbol{Y}+\boldsymbol{B}\boldsymbol{M} & \boldsymbol{\Phi}\boldsymbol{X} \end{bmatrix} \\ * & * & \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X} & \boldsymbol{Y} \end{bmatrix} \end{bmatrix} \geq 0 \tag{7.27}$$

$$\begin{bmatrix} \bar{\boldsymbol{u}}^2 & \begin{bmatrix} \boldsymbol{K}\boldsymbol{Y}+\boldsymbol{M} & \boldsymbol{K}\boldsymbol{X} \end{bmatrix} \\ * & \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X} & \boldsymbol{Y} \end{bmatrix} \end{bmatrix} \geq 0 \tag{7.28}$$

As it has been previously noted, the "shift" matrices $\boldsymbol{T}$ and $\boldsymbol{E}$ are full and may be computed from the following relation:

$$\boldsymbol{T} = \boldsymbol{W}^{-1}\boldsymbol{N}\boldsymbol{V}^{-T} \qquad \boldsymbol{E} = \boldsymbol{M}\boldsymbol{V}^{-1} \tag{7.29}$$

In the NRMPC formulation, the set of stabilizable states or in other words the region of attraction is simply the projection of the augmented ellipsoid ε_z into the $\mathbb{X}$ subspace. The invariant ellipsoidal target set is defined as the intersection of ε_z with the $\mathbb{X}$ subspace. Here the LQ control law is optimal with leaving the perturbation vector $\boldsymbol{f}_k = 0$. This ellipsoidal terminal set is defined by $\boldsymbol{X}$, while the region of attraction through $\boldsymbol{Y}$.

Maximizing the volume of the terminal set and region of attraction defined by $\boldsymbol{X}$ and $\boldsymbol{Y}$ is the aim of the off line optimization algorithm, which can be summarized according to Cannon and Kouvaritakis (2005):

Optimization of prediction dynamics in NRMPC recovers the maximal volume ellipsoidal admissible set. Note that the volume oft he region of attraction in independent of the prediction horizon used. This is simply true, because the matrix term $\boldsymbol{Y}$ is independent on the prediction horizon thus it is enough to set the horizon equal to the order of the model used $n_c = n_x$.

The on-line algorithm utilizes the results of the off-line algorithm 7.2.3.3 and can be summarized according to Kouvaritakis et al (2002):

Off-line NRMPC procedure: Run the following procedure once.

- Calculate an LQ optimal feedback matrix, without considering constraints.
- Maximize the volume of the region of attraction defined by the projection of the augmented invariant ellipsoid with $\mathbb{X}$ subspace and the target set defined by the intersection of the augmented invariant ellipsoid with $\mathbb{X}$ subspace by solving:

$$\max\left(-\log\det\begin{bmatrix}\boldsymbol{Y} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{X}\end{bmatrix}\right) \tag{7.30}$$

  subject to constraints defined by the invariance condition (7.27) and the feasibility condition (7.28). This is a semi-definite programming problem in the variables $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{N}$ and $\boldsymbol{M}\ldots$
- To determine $\boldsymbol{W}$ and $\boldsymbol{V}$, factorize $\boldsymbol{X}$ and $\boldsymbol{Y}$.
- Using relation (7.29) for $\boldsymbol{N}$ and $\boldsymbol{M}$ solve for $\boldsymbol{T}$ and $\boldsymbol{E}\ldots$

On-line procedure: At each sampling instant k perform the following minimization:

$$\min_{f} \boldsymbol{f}_k^T \boldsymbol{f}_k \qquad \text{s.t.} \qquad \boldsymbol{z}_k^T \boldsymbol{S} \boldsymbol{z}_k \leq 1 \tag{7.31}$$

From this the control signal $\boldsymbol{u}_k$ is evaluated, and the procedure restarted at the next sampling instant.

Partitioning $\boldsymbol{S}$ allows the reformulation of the optimization constraint in (7.31) similarly to the structure of $\boldsymbol{z}$:

$$\boldsymbol{z}_k^T \boldsymbol{S} \boldsymbol{z}_k = \boldsymbol{x}_k^T \hat{\boldsymbol{S}}_{11} \boldsymbol{x}_k + 2\boldsymbol{f}_k^T \hat{\boldsymbol{S}}_{21} \boldsymbol{x}_k + \boldsymbol{f}_k^T \hat{\boldsymbol{S}}_{22} \boldsymbol{f}_k \leq 1 \tag{7.32}$$

This is a simple univariate optimization problem, solvable by Lagrange's method for constrained extrema:

$$\boldsymbol{f}_k = \lambda \Delta \hat{\boldsymbol{S}}_{21} \boldsymbol{x}_k \tag{7.33}$$

$$\begin{aligned}\Phi(\lambda) = {} & \hat{\boldsymbol{S}}_{12}[\boldsymbol{\Delta}\hat{\boldsymbol{S}}_{22}^{-1}\boldsymbol{\Delta} - \hat{\boldsymbol{S}}_{22}^{-1}]\hat{\boldsymbol{S}}_{21}\boldsymbol{x}_k + \\ & + \boldsymbol{x}_k^T \hat{\boldsymbol{S}}_{11} \boldsymbol{x}_k - 1 = 0\end{aligned} \tag{7.34}$$

where $\boldsymbol{\Delta} = (\boldsymbol{I} - \lambda \hat{\boldsymbol{S}}_{22})^{-1}$ and λ is the unique real root of $\Phi(\lambda)$. Newton - Raphson's root searching algorithm is then utilize to find λ, . Usually no more than 10 iterations are needed for the NR procedure to converge to the root of $\Phi(\lambda)$, which is in turn used to compute the perturbation vector $\boldsymbol{f}$.

The optimality of the NRMPC algorithm may be improved by implementing an extension introduced by Kouvaritakis et al (2002). The NRMPC algorithm utilized in the experiments featured in this section does not make use of this extension. Simulation studies with the model of the vibrating system did not prove a substantial improvement over the method based on the original formulation.

7.2.4 Experimental Setup

7.2.4.1 Hardware Description

A small scale laboratory model has been created to test the performance of various MPC algorithms with constraint feasibility and stability guarantees on lightly damped vibrating active structures. The laboratory device is featured in Fig. 7.1. This experimental system consists of an aluminum beam with one end clamped and fixed to a base and the other allowed to vibrate freely. The beam has the dimensions of $550 \times 40 \times 3\,\text{mm}$ and is made of commercially pure aluminum.

The actuating elements are single layer MIDÉ QP16n piezoelectric transducers marked as PZT 1 & 2 in Fig. 7.1. These transducers are electrically connected counter-phase and receive the same electric signal from a MIDÉ EL-1225 wide bandwidth amplifier. The polarization voltage of the transducers is $\pm 120\,\text{V}$. To prevent actuator damage, this constraint is incorporated in the MPC law.

Deflections at the beam tip are measured using a Keyence LK-G82 industrial grade laser triangulation system. Manufacturer given sensor accuracy is $\pm 0.05\%$ with the resolution of $0.2\mu\text{m}$ in the range of $80 \pm 15\,\text{mm}$. The Keyence LK-G3001V central processing unit provides analogue outputs to the A/D input of the measurement card. Measurements from this sensor are directly used in the controller feedback.

A 16 bit resolution and 100 kS/s sampling speed National Instruments DAPC6030 laboratory measurement card is installed in a personal computer, providing A/D and D/A conversion to the controller. The controller itself is implemented using the xPC Target rapid software prototyping platform.

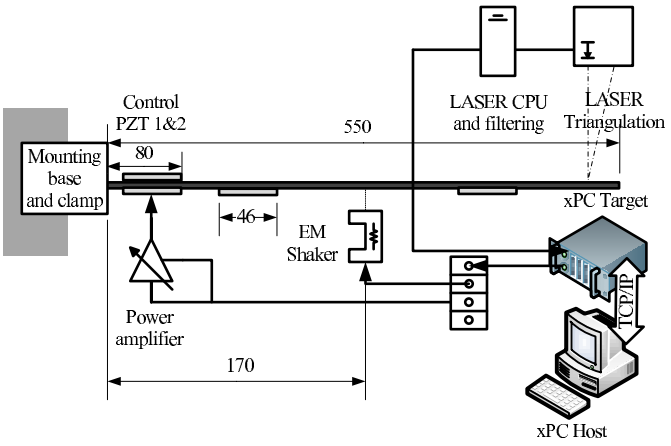


Fig. 7.1 Hardware configuration scheme.

7.2.4.2 System Identification Procedure

All MPC algorithms considered here assume the use of the same linear time-invariant (LTI) state-space system defined by (7.1). Current states are observed using a Kalman filtered output measurement, utilizing the LTI system defined by (7.1). The state-space model assumed here is second order, containing the dynamics of the first vibration mode.

The experimentally identified state-space model represents the input-output relationship between the actuator voltage and beam tip deflections directly in millimeters. The experimental identification procedure used a $\pm 120\,\mathrm{V}$ peak amplitude chirp signal to the two transducers in actuator mode via the amplifiers. This signal covered the $0.5 - 20\,\mathrm{Hz}$ frequency range in 300 seconds. The time domain deflection data and input voltage has been converted into the frequency domain utilizing Fast Fourier Transformation. Unnecessary frequency ranges have been discarded and the signal has been de-trended and filtered as well.

The first resonant frequency of the cantilever beam is located at approximately $8.1\,\mathrm{Hz}$, therefore model sampling has been set to $T_s = 0.01$. This sampling is sufficient to control the first mode. The state-space model has been obtained through the subspace iteration method of Ljung. (1999). The experimentally identified model used to generate predictions is:

$$\boldsymbol{A} = \begin{bmatrix} 0.867 & 1.119 \\ -0.214 & 0.870 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 9.336E^{-4} \\ 5.309E^{-4} \end{bmatrix} \tag{7.35}$$

$$\boldsymbol{C} = [-0.553 \quad -0.705]$$

7.2.4.3 Controller Implementation

All experiments featured in this paper have common features, the controllers are implemented with the same or equivalent settings. This is essentially required so that the sampling time and vibration damping comparison is meaningful. The controllers utilize the same state-space model (7.35) for prediction, QP and MP based MPC assumes a $n_c = 75$ steps long prediction horizon, while NR based MPC utilizes a $n_c = n_x$ horizon according to the theoretical considerations presented in 7.2.3.3. The common $n_c = 75$ steps horizon has been set up according to the maximal task execution time of the QPMPC algorithm. Since the QPMPC algorithm is the least computationally efficient of the three considered methods, this horizon acts as a common basis of comparison.

State penalty matrices in all MPC methods and in the LQ controller computation have been set to $\boldsymbol{Q} = \boldsymbol{C}^T\boldsymbol{C}$, which diretly penalizes output: the

beam deflections. Input penalty has been determined to be $\boldsymbol{R} = 10E^{-4}$, which is an ideal compromise between performance and and controller aggressiveness.

All MPC methods assume $\underline{\boldsymbol{u}} = \overline{\boldsymbol{u}} = 120\,\mathrm{V}$ constraints on the input. Sampling time is set to $T_s = 0.01\,\mathrm{s}$. All controllers have been implemented in Matlab / Simulink while the resulting Simulink block scheme has been transferred onto the same target computer running the xPC Target environment. The block schemes are identical in every case, except the controller algorithm itself. In addition to the controller, these block schemes contain means for A/D and D/A data conversion a Kalman filter routine utilizing the same system model, and means for data logging.

Quadratic programming is performed via the 2.0 version of the qpOASES open-source C++ active set strategy (Ferreau, 2006; Ferreau et al, 2008). This quadratic programming solver implements theoretical features enabling a computational efficiency gain for MPC applications. The qpOASES solver is loaded and compiled via its Simulink interface, then prediction and constraint matrices are passed on to it from the Matlab workspace.

The current latest 2.6.2 version of the Multi-Parametric Toolbox (MPT) for Matlab by Kvasnica et al (2004); Kvasnica (2009) has been utilized to implement the optimal multi-parametric programming MPC on the experimental device. The MPT Toolbox allows for convenient explicit controller computation including transferring the controller regions from Matlab to a C header code. A sequential search algorithm calls this C code header and finds the PWA law according to the region containing the observed system state. Using the common conventions and settings introduced at the beginning of this section, the stable and invariant controller has been computed in 2923 seconds which is over 48 minutes and is defined over 11601 polytopic regions.

The off-line NRMPC procedure is implemented in the Matlab m-file scripting language. The controller matrices for the on-line run are evaluated by the maximization problem defined in (7.30) which is constrained by the linear matrix inequalities (LMI) defining invariance (7.27) and feasibility (7.28). This is a semi-definite optimization problem (SDP), which is passed through the parser YALMIP (L ofberg, 2004) into the SDP solver (Sturm, 1999). The on - line NRMPC procedure summarized by Algorithm 7.2.3.3 has been implemented using a custom S-Function block written in the C programming language. Matrix and vector operations are computed using the familiar Basic Linear Algebra Subprograms (BLAS) library (Dongarra, 2002).

7.2.5 Off-line Properties

7.2.5.1 On the Horizon Length of Constrained MPC with Stability Guarantees

A linear, second order state-space model sampled at $T_s = 0.01\,\mathrm{s}$ does not render the implementation of a MPC controller difficult. However this is true only if the MPC law is either unconstrained, or it is constrained without stability or feasibility guarantees. If stability and constraint feasibility is guaranteed as in the case of the QPMPC or MPMPC formulation introduced earlier, the allowable states are contained in a limited subset of the state-space, the so-called region of attraction or maximal admissible set.

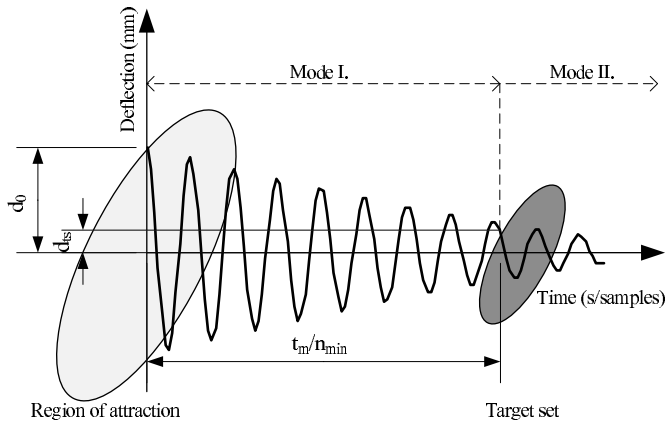


Fig. 7.2 Illustration of the time t_m required to enter the target set, when starting from a given initial condition in the region of attraction.

This means that all expected system states corresponding to the variations in the output, must be contained in the region of attraction. Given a fixed system model and penalization matrices, one can enlarge this region through increasing the prediction horizon of the controller. The minimal necessary prediction horizon in MPC with guaranteed stability and constraint feasibility is basically the number of steps necessary to drive system state from a given initial condition inside the terminal set.

A large asymmetry between the static effect of the actuators and the range of expected changes also translates to the asymmetry in the volume of the admissible set of states and the target set in stabilized MPC control. The time required to steer the initial system state corresponding to the largest expected output into the target set divided by the sampling period is a good indicator of the necessary minimal prediction horizon.

This idea is illustrated in Fig. 7.2. The decaying sinusoid suggests a lightly damped but controlled vibration of the beam. This obviously requires an extensive settling time in comparison with the sampling period. The ellipse on the left represents the region of attraction. All expected system states must be included in it. There is a certain point in the free vibration of the beam tip, where system states enter the terminal set. This is illustrated by the smaller ellipse on the left. The minimal necessary prediction horizon for stable MPC with constraint feasibility guarantees can be understood as the time t_m required for the state to reach the target set, in sample periods.

Even though this idea can transferred to any physical system with a long (forced) settling time vs. sampling period ratio and limited control action in comparison to the effort necessary to drive the system into equilibrium, lightly damped vibrating structures are especially susceptible to this issue.

For example the laboratory device considered here uses piezoelectric actuators, with a limited force deformation effect. The (LQ) controlled settling time is near 2 seconds. Assuming that approximately 1.5 seconds are necessary to reach the target set with a $T_s = 0.01\,\mathrm{s}$ sampling; one might expect approximately a $n_c = 150$ steps long minimal necessary prediction horizon for stabilized MPC control!

Let us now assume an exponential decay of vibration amplitudes. Amplitudes d_t of a a freely vibrating system for a given time can be approximated using:

$$d_t = d_0 e^{-\zeta \omega_n t} \tag{7.36}$$

where d_0 is the initial deflection, t is the time in seconds since the initial conditions had affected the system, ζ is the damping ratio and ω_n is the first or dominant natural frequency of the vibrating system.

If MPC control is used with constraint feasibility and stability guarantees, there is a certain amplitude level d_{ts} under which the system state enters the target set. According to relation (7.36) the minimal prediction horizon n_{min} for initial deflection d_0 can be approximated:

$$n_{min} = \frac{-\lg(\frac{d_{ts}}{d_0})}{2\pi\zeta f_n T_s} \tag{7.37}$$

Here f_n is the dominant mechanical eigenfrequency, T_s the sampling rate considered for control and the rest of the variables as defined for (7.36).

7.2.5.2 Off-line Computational Time

Off - line computational time for the QP, MP and NR based MPC controllers is demonstrated in Fig. 7.4 . Clearly off-line computational time is not an issue for the QP nor the NR based MPC controllers. The time to set up the controllers in the off line regime is limited to constructing prediction matrices

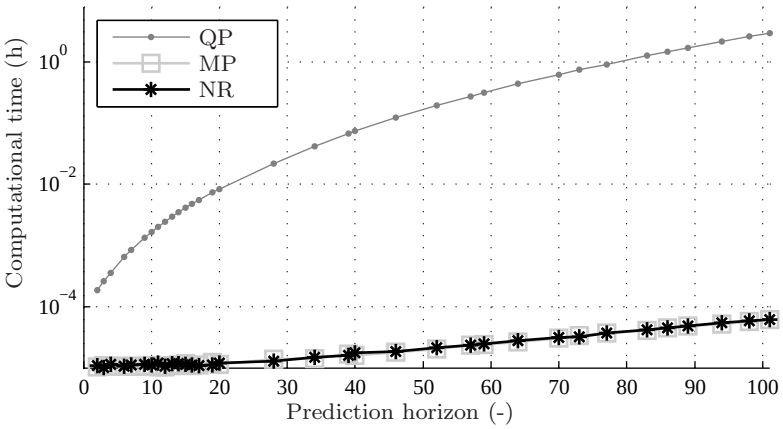


Fig. 7.3 Off-line computational time in hours, required to compute controller structure for a given horizon length.

in the case of QP while increasing horizon length has no effect on the off-line NR problem. Assuming generic hardware, off-line computational time for both QP and NR controllers is under 1 seconds even for a $n_c = 100$ steps long horizon.

The multi-parametric programming based MPC controller pre-computes controller regions and the associated PWL law. This requires CPU time which is exponentially increasing with the prediction horizon. For a problem with $n_c = 100$ steps prediction horizon, the off line optimal MP problem takes approximately 3 hours to compute.

A $\pm 20mm$ maximal allowable deflection is not an unusual requirement for the type of vibration attenuation system featured here. According to Fig. 7.4 this requires a prediction horizon over $n_c = 150$ steps, however the off-line MPMPC computation fails over $n_c = 168$ steps due to memory issues. This horizon requires an approximately 36 hours long computational time assuming MPMPC, while using interpolation we may estimate an excessive 7 days long off-line computation time to ensure a ± 30 mm deflection range. This is clearly beyond reasonable limits for most applications, while the implementation of such large look-up tables is an issue as well.

7.2.6 On-line Properties

Te on-line properties of the QP, MP and NR based MPC algorithms have been evaluated in three different experimental excitation situations. A vibration damping performance analysis and task execution time comparison

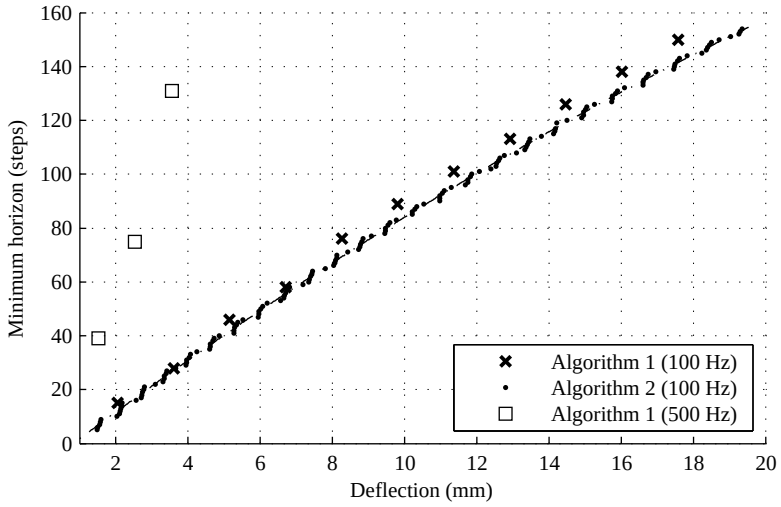


Fig. 7.4 Minimal necessary horizon vs. possible deflection range.

is demonstrated in the time domain for a beam deflected from its equilibrium. In the frequency domain the beam has been subjected to a disturbance caused by a laboratory modal shaker supplied with a narrow band chirp signal and a pseudo random binary signal. Vibration damping performance and computational time is demonstrated for the frequency domain experiments as well.

7.2.6.1 Initial Deflection Experiment

The end of the smart cantilever beam has been deflected 5 mm away from its equilibrium position, then left to vibrate under control without any further outside excitation. The cantilever beam would settle to its equilibrium even without control, however the controlled response is approximately an order of magnitude faster.

The response of the beam to the type of excitation described above is featured in Fig. 7.5. The different MPC methods in question are contrasted to saturated LQ control. This gives a basis of comparison both in damping performance and on-line computation time requirements.

Vibration of the beam tip is demonstrated on 7.5(a), where one may observe no substantial difference between any the individual MPC methods. The worst damping performance is associated with saturated LQ. The saturated LQ controlled beam settles slower than either one of the MPC controlled responses. It may be concluded, that from a practical viewpoint the controlled

vibration response is not distinguishable, all stable MPC methods perform very similarly.

The difference between the individual MPC methods is slightly better demonstrated on Fig. 7.5(b), where the voltage passed onto the actuators is shown. As it is expected, saturated LQ produces more aggressive control moves than the MPC methods. This is especially visible after the initial saturated stage has passed. The QP and MP based MPC methods should perform completely identically according to theory, and in fact the experimental difference in this test is negligible.

The sub-optimality of Newton-Raphson based MPC is clear and dominant on the voltage output figure: instead of the saturated behavior resembling a square signal visible at the beginning of the test, NRMPC produces a less optimal output approximating but never fully reaching the constraints. This experiment suggests that there are no substantial differences in the damping performance for the three investigated MPC methods.

Computational times for the corresponding sample periods are featured in Fig. 7.6, where the horizontal axis denotes time samples and the vertical axis shows task execution times (TET). Minimal, maximal and average computational time for both inside and outside the target set is presented in Table 7.1 as well.

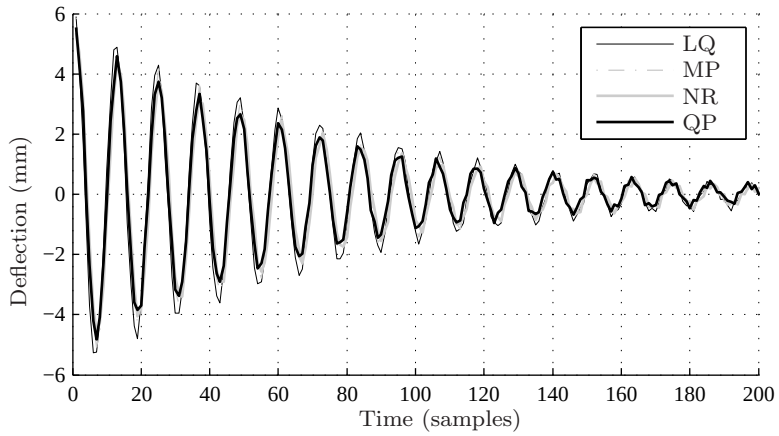
Traditional quadratic programming based MPC utilizes almost all the sampling period to complete its calculations. The first section of the graph is computationally more intensive. As the system states move into the target set, a stable level of short computational times is needed to evaluate the problem. With the experimental setup and requirements demonstrated here QPMPC is on the limit of practical implementability.

The two remaining controllers require significantly shorter task execution times. Multi-parametric MPC achieves more than two orders of magnitude better computational times than QP even when the constraints are active. This in fact indicates substantial reserve in implementability.

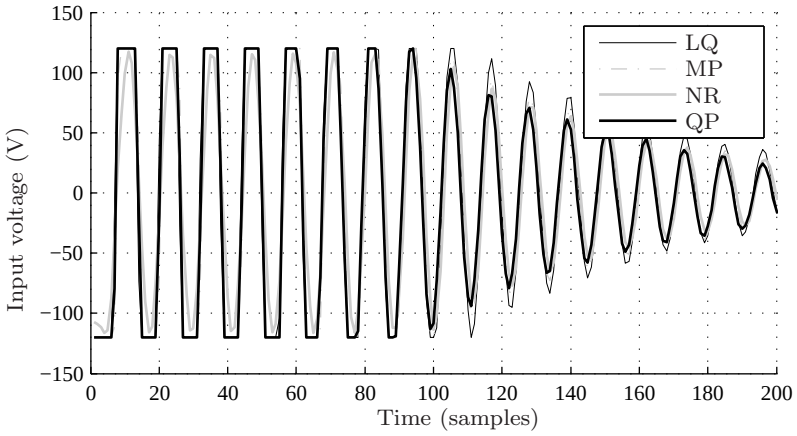
Saturated LQ is also shown in Fig. 7.6, and serves as a baseline for the computational times. Since all three MPC methods require computational time for state observation, accessing input and output ports and data logging; the TET response of the LQ controller demonstrated here may be regarded as an absolute minimal computational time floor for the given hardware config-

Table 7.1 Task execution time summary for the initial deflection test in micro seconds. ("t.s." denotes target set)

	t_{min} (μs)	t_{max} (μs)	t_{avg} (μs)	
			Outside t.s.	Inside t.s.
QP	715	8975	6815	716
MP	14	77	42	14
NR	14	17	16	14
LQ	15	16	15	15



(a) Beam deflection.



(b) Actuating voltage.

Fig. 7.5 Response of the beam tip after an initial deflection of $5mm$ is shown on (a), while the corresponding actuating signal is featured on (b).

uration. The laboratory measurement card requires 12 microseconds for data transfer, which is included both on 7.6 and Table 7.1.

Taking into account these observations, is interesting to note the execution time graph for the NRMPC controller. Here NR shows no significant increase of computational time even when compared to a simple saturated LQ.

After the system state enters the target set, all three controllers need shorter execution periods than that required during control with active constraints. If constraints are not engaged, both MPMPC and NRMPC needs comparable computation time to the LQ controller.

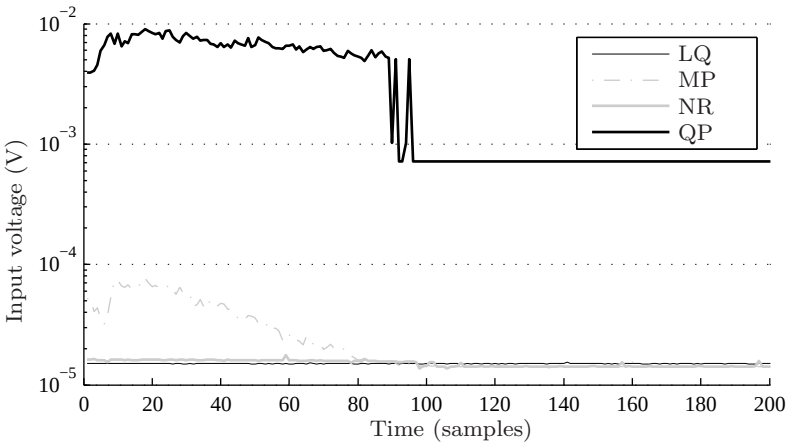


Fig. 7.6 Task execution time (TET) required to compute the previous step in seconds.

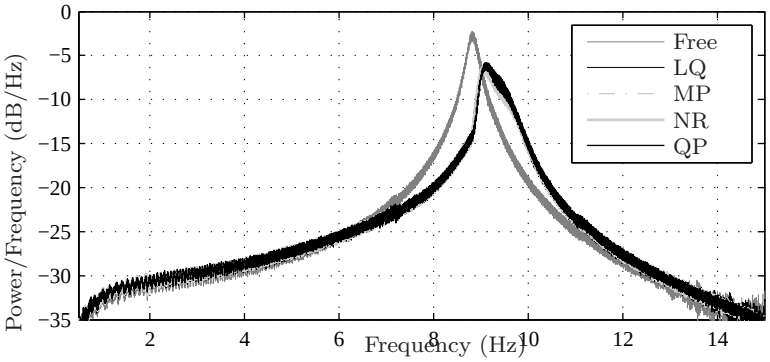
7.2.6.2 Chirp Signal Experiment

The vibration attenuation performance of the MPC controllers of interest is also compared in the frequency domain. For this experiment a Bruel&Kjær Type 4810 electrodynamic modal shaker is utilized as a source of mechanical excitation in the bandwidth of interest. A chirp input signal with the bandwidth of $0-20\,\text{Hz}$ has been supplied to the shaker to excite the beam in the vicinity of the first resonant frequency.

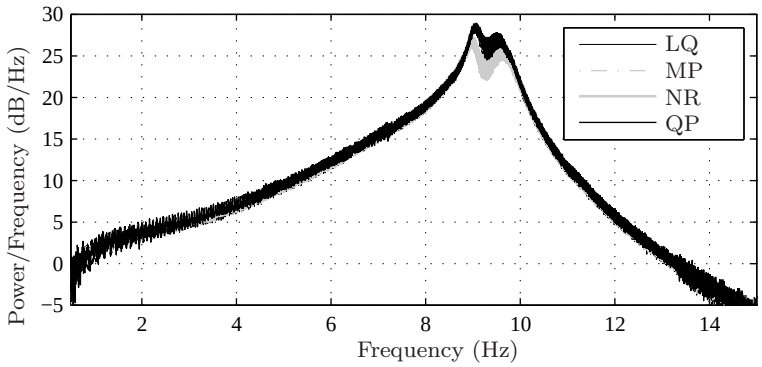
Fig. 7.7 shows the result of the chirp signal excitation experiment. The periodogram of the beam tip deflection signal is featured in Fig. 7.7(a), where both the controlled and free response is observable. Similarly to the experiment presented in 7.2.6.1 there is no observable vibration attenuation performance difference for the various types of MPC algorithm. It may be concluded, that all MPC methods of interest provide a practically identical vibration damping performance in the bandwidth of interest.

Maximal power-frequency signal amplitude with the corresponding resonant frequency and the absolute deflection is demonstrated in Table 7.4. All three types of controllers reduce maximal deformation amplitudes approximately to $d = 3.3\,\text{mm}$. Since the active beam is essentially stiffened by the controllers, resonant frequency of the controlled response is shifted to higher values than the resonant frequency for the free response.

The periodogram of the actuator voltage supplied by the controllers into the actuators is indicated in Fig. 7.7(a). Note that there is no substantial difference in the system voltage input for the QP, MP and saturated LQ controlled experiments. The NRMPC signal periodogram is less dominant,



(a) Beam deflection.



(b) Actuating voltage.

Fig. 7.7 Narrow band periodogram of the beam tip deflection signal is shown on (a), while the periodogram of the corresponding actuating signal is featured on (b).

this is due to the fact that NR provides sub-optimal control in comparison to QP or optimal MP based MPC.

Table 7.2 Task execution time summary for the resonant and non-resonant controlled beam response to the chirp test in microseconds. Constraints are active inside the resonant area, while outside resonance the system state is located within the target set.

$(\mu s) \rightarrow$	t_{min}	t_{max}	t_{avg}	t_{min}	t_{max}	t_{avg}
	Resonance $(9.0 - 9.6\,\text{Hz})$			Outside res. $(0 - 8\,\text{Hz})$		
QP	4925	8739	6405	725	732	728
MP	25	48	32	23	28	24
NR	25	30	26	24	28	24
LQ	24	28	24	23	28	24

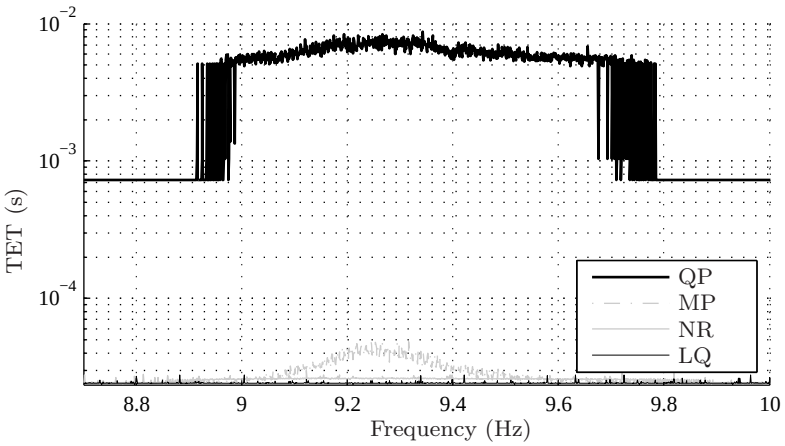


Fig. 7.8 Time required to compute one control step.

Task execution times for the chirp excitation experiment are indicated in Fig. 7.8. Minimal, maximal and average computational times are summarized in Table 7.2 for within and outside the resonant area. The execution time values are shown in the area surrounding the first resonant frequency, as this is the region where the disturbance introduced by the shaker may cause the system states to leave the target set.

The conclusions from the time domain experiment in 7.2.6.1 are valid for this case as well. Quadratic programming based MPC is running with execution times close to the sampling period Ts. Optimal MPMPC requires significantly lower sampling times than QP, while NR based MPC remains the fastest of them all closely approaching the minimal possible task execution times set by the LQ controller.

7.2.6.3 Pseudo-Random Binary Signal

This experimental excitation situation utilizes a pseudo-random binary signal (PRBS) supplied to the modal shaker, which has been itnroduced in 7.2.6.2. This PRBS signal assumes two voltage levels, adjusted in order for the shaker to drive system states to levels where the quadratic programming MPC algorithm requires execution times closely matching but not exceeding the sampling period of the controller.

Periodogram of the beam tip deflection signal for each examined MPC controller is demonstrated in Fig. 7.9. Saturated LQ controller and free response is indicated as well. Again, there is no significant difference between the vibration damping performance of these methods. The analysis given for the chirp signal test in 7.2.6.2 is valid here as well. Maximal signal amplitudes

and resonant frequencies, including maximal deflections for the individual controller schemes are featured in Table 7.4.

Fig. 7.10 shows a sample portion of the task execution times for the individually investigated controllers. A computational time summary for a 100 second long pseudo-random test is indicated in Table 7.3. Execution times are very similar to the experiments featured in 7.2.6.1 and 7.2.6.2, therefore the analysis is not repeated here.

Table 7.3 Task execution time summary for a 100 second long pseudo-random test in micro seconds.

	t_{min} (μs)	t_{max} (μs)	t_{avg} (μs)	$\pm t_{std}$ (μs)
QP	722	9789	2784	2603
MP	24	76	26	5
NR	24	29	25	1
LQ	23	28	24	<1

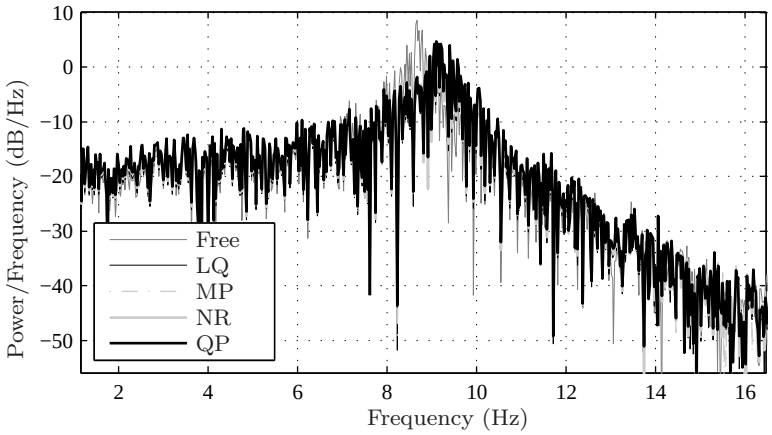


Fig. 7.9 Periodogram of the beam tip deflection signal subject to a pseudo-random binary signal.

7.2.7 Conclusion

This subsection has demonstrated practical implementation properties and experimental verification of damping performance and cycle execution timing of various stable MPC algorithms with guaranteed stability and feasibility;

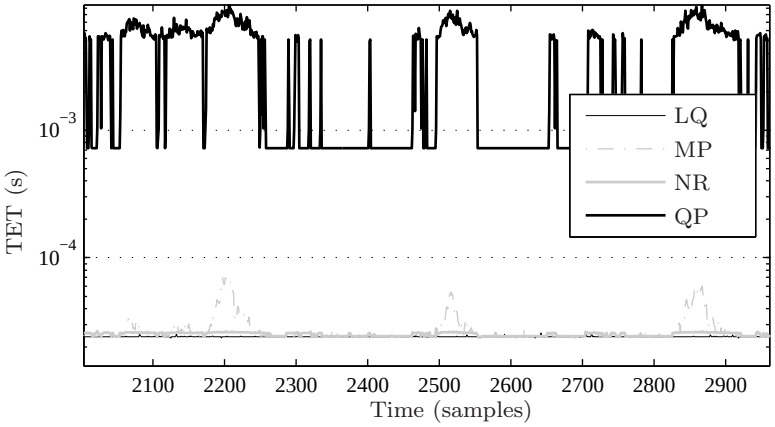


Fig. 7.10 Sample portion of a typical on-line computational time requirement.

Table 7.4 Summary of the damping performance analysis of the tests performed using both the chirp and PRB signal excitation. Amplitude is represented as signal power / frequency and is denoted by $A\,(\mathrm{dB/Hz})$, first mode resonance frequency is denoted by $f\,(\mathrm{Hz})$ and the absolute maximal beam tip deflection is marked by $d\,(\mathrm{mm})$ in the table.

	A(dB/Hz)	f(Hz)	d(mm)	A(dB/Hz)	f(Hz)	d(mm)
	Chirp test			PRBS test		
Free	-2.29	8.82	4.3	8.60	8.67	5.6
QP	-5.82	9.13	3.4	4.76	9.11	4.7
MP	-5.94	9.09	3.3	4.41	9.12	4.4
NR	-6.25	9.08	3.3	4.70	9.14	4.7
LQ	-5.87	9.094	3.4	4.41	9.15	4.3

applied to lightly damped flexible vibrating structures. Results of experiments performed on the laboratory device demonstrated no substantial difference between the vibration damping performance of the four considered stable MPC algorithms. This not only true for the theoretically identical dual-mode QP MPC and optimal MP MPC, but also in the case where optimality has been traded for simplicity and in turn computational efficiency. The damping performance comparison suggests that in practice, the computationally efficient but sub-optimal methods like minimum - time explicit MPC or Newton-Raphson's MPC may be implemented without a considerable loss of performance.

The main practical distinction between the four algorithms is the computational time required to complete one cycle. From the performed tests it is evident that QP based MPC has been on the verge of implementability even for this fairly simple case. Despite the fact that the solver utilized in this algorithm realization claims to be specifically designed for the needs of MPC,

it is highly unlikely to be usable for problems of increased dimensionality or shortened sampling periods. On the other hand, the optimal and in its outputs theoretically identical pre-computed explicit MP MPC requires extensive calculations in the off-line regime. Problems of higher dimensionality are unlikely to be successfully implemented due to the likely failure of the off-line computations.

In short we may conclude that given their current algorithmic formulation neither QP nor MP based optimal MPC with stability and feasibility guarantees may be recommended for the active vibration damping of lightly damped structures.

Alternatively the small loss of theoretical performance does not present problems in practice, therefore the sub-optimal stable NR MPC method considered in this article may be recommended for vibration attenuation purposes. NR MPC showed damping capabilities comparable to its truly optimal counterparts. The on-line execution times featured in the experimental results suggest that there is a reserve for either increasing problem dimensionality or shortening sampling time. Nevertheless, let us not forget about the possible drawbacks of NRMPC, namely that optimality decreases steeply in Newton-Raphson's MPC if the order of the prediction model is increased. This is due to the fact that in higher dimensions the true polytopic region of attraction respectively the target set can not be effectively approximated by a hyper-ellipsoidal shape.

7.3 Pre-Filtered Moving Horizon Observer for Vibration Dynamics

7.3.1 Introduction

High-performance embedded controllers open the possibilities for application of numerical methods to solve the problems of modeling and control of vibrating systems. Fast vibration dynamics is an interesting challenge for computing hardware, software and mechanical design of beam and cantilever mechatronics (Fuller et al, 1996). Algorithms based on a model of the dynamics are becoming the standard approach for control and monitoring of vibrating systems. One of the most applied model structures is the state-space model. If the full state vector is not completely measurable it is necessary to estimate it using a state observer. The observer algorithms do not need to be restricted only to the state estimation problem. With a standard augmentation of the model one can estimate the parameters of the model by declaring them as states. Through this approach, the problem of joint estimation of system states and parameters is considered.

The classical approach to determine the state and parameters is in vibration mechanics the Kalman filter and its modified version for nonlinear systems, the Extended Kalman Filter (EKF) (Gelb et al, 2001). The foundation of such filtration is the model of vibrating structure based on the lumped parameter assumption of rigid, shape invariable mass (Corigliano and Mariani, 2004; Ghosh et al, 2007). The application of a dynamic model for the purpose of filtration based on the principles of continuum mechanics is proposed by Ohsumi and Nakano (2002). The typical filtration application of state and parameters is the control (Gao and Lu, 2006), diagnostics and monitoring of vibrating system (Hoa and Ma, 2007). The above mentioned EKF method uses a linearized model to approximate nonlinear vibration dynamics with the assumption of sequentially uncorrelated Gaussian noise distribution. If the noise is correlated or does not have the Gaussian distribution, application of the Extended Kalman filter can cause divergence of the estimated states and parameters. Moreover, the method is sensitive to initial condition of the estimate. In this case the approaches based on probabilistic Bayesian Particle Filter (PF) methods with the application of stochastic Monte Carlo simulations lead to more accurate estimates of state and parameters of the nonlinear vibration dynamics (Ching et al, 2006; Namdeo and Manohar, 2007; Sajeeb et al, 2009).

The objective and novel contribution of this study is the numerical application of least-squares estimation of state and parameters of vibrating system by combining a pre-filtering EKF with an Moving Horizon Observer (MHO). The MHO is the alternative to statistical methods (PF) and minimum-variance (EKF) methods though it needs no statistical assumption about the sources of uncertainty (Moraal and Grizzle, 1995; Alessandri et al, 2008). Pre-filtration in the arrival cost using variants of the Kalman filter (Rao et al, 2003; López-Negrete et al, 2009; Qu and Hahn, 2009; Ungarala, 2009) is shown to improve the accuracy of the observer.

The physical wave equation of a one-mode oscillator (mass-spring-damper system) is further considered to represent the model which the MHO uses for estimation. The lumped parameter model often describes the vibration dynamics sufficiently where this model was experimentally applied in relation to EKF and PF in Jones et al (1995) and Namdeo and Manohar (2007); Uchino and Ohta (1986) respectively.

7.3.2 Basic Model Formulation

The structural vibration model can be written as

$$M_0\ddot{q} + C_0\dot{q} + K_0 q = L_0 u \tag{7.38}$$

where M_0 is the mass matrix, C_0 is the damping matrix, K_0 is the stiffness matrix, L_0 is the transition matrix, q is the displacement vector and u is the excitation force. Conventionally, the state-space equation of the problem can be represented as

$$\dot{x}_s = Ax_s + Bu \tag{7.39}$$

where $x_s = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$, $A = \begin{bmatrix} 0 & I \\ -M_0^{-1}K_0 & -M_0^{-1}C_0 \end{bmatrix}$,
$B = \begin{bmatrix} 0 \\ M_0^{-1}L_0 \end{bmatrix}$

An augmented state vector $x \in \mathbb{R}^{2n_q+n_p}$ can be defined

$$x = \begin{bmatrix} x_s \\ p \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \\ p \end{bmatrix} \tag{7.40}$$

where $p \in \mathbb{R}^{n_p}$ is the vector of uncertain model parameters (e.g. stiffness, damping). The number of modes is n_q, and n_p is the total number of model parameters to be identified. For the purpose of parameter identification the vibration dynamics (7.38) is described by a general time-invariant state-space equations

$$\dot{x}_s = \tilde{f}_c(x_s, p, u) \tag{7.41}$$

$$\dot{p} = 0 \tag{7.42}$$

where $\tilde{f}_c : \mathbb{R}^{2n_q} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_u} \to \mathbb{R}^{2n_q}$ represents the augmented dynamics. The model which was linear-in-the-parameters becomes nonlinear by declaring the unknown model parameters as additional states of the system. Even a system without any mechanical nonlinearity leads to a nonlinear filtering problem.

Eq. (7.41) and (7.42) can be combined as

$$\dot{x} = f_c(x, u) \tag{7.43}$$

where $f_c : \mathbb{R}^{2n_q+n_p} \times \mathbb{R}^{n_u} \to \mathbb{R}^{2n_q+n_p}$. The observation equation may be written as

$$y = h_c(x, u) + v \tag{7.44}$$

where $y \in \mathbb{R}^{n_y}$ is a vector of measurements and $h_c : \mathbb{R}^{2n_q+n_p} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ is a continuous measurement function. The measurement errors are modeled with the noise term $v \in \mathbb{R}^{n_y}$. The most frequent situation encountered in practice is when the system is governed by continuous-time dynamics and the measurements are obtained at discrete time instances. For the problem formulation we consider the numerically discretized dynamic nonlinear system described by the equations

$$x_{t+1} = f(x_t, u_t) \tag{7.45}$$
$$y_t = h(x_t, u_t) + v_t \tag{7.46}$$

for $t = 0, 1, \ldots$, where $x_t \in \mathbb{R}^{n_x}$ is the state vector and $u_t \in \mathbb{R}^{n_u}$ is the control vector. The state vector is observed through the measurement equation (7.46) where $y_t \in \mathbb{R}^{n_y}$ is the observation vector and $v_t \in \mathbb{R}^{n_y}$ is a measurement noise vector.

The state dynamics given by Eq. (7.43) (or discretized by Eq. (7.45)) is a deterministic formulation. A common procedure is to include the process noise vector in Eq. (7.43) which would account for the stochastic behavior. For a sake of simplicity the process noise will not be considered in this study, however in practical situations this might be an important part of the dynamic equations to account for uncertainty in the inputs or unmodeled dynamics.

7.3.3 Extended Kalman Filter

The EKF is perhaps the most often applied algorithm for the estimation of state and parameters of nonlinear dynamic systems (Gelb et al, 2001) and it will be here considered as the benchmark algorithm. The following algorithm is in the literature known as continuous-discrete or hybrid EKF (Gelb et al, 2001). The dynamic system is given by (7.45) and (7.46) where the white noise has the normal Gaussian distribution $v_t \sim N(0, R_t)$. The initial condition of the state vector is $x_0 \sim N(\hat{x}_0^+, P_0^+)$. The estimate of the state vector at $t = 0$ begins with the initial state vector estimate and with the initial covariance matrix of the initial state vector estimate error

$$\hat{x}_0^+ = E[x_0] \tag{7.47}$$
$$P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \tag{7.48}$$

From time instance $t-1$, the dynamic system (7.43) is simulatively propagated one step ahead as

$$\hat{x}_t^- = f(\hat{x}_{t-1}^+, u_{t-1}) \tag{7.49}$$

where $t = 1, 2, \ldots$. This one step computation gives an a priori state estimate. The time update of the covariance matrix estimate is given by

$$\dot{P} = Z(\hat{x})P + PZ^T(\hat{x}) \tag{7.50}$$

where

$$Z(\hat{x}) = \left.\frac{\partial f_c(x)}{\partial x}\right|_{x=\hat{x}} \tag{7.51}$$

The covariance matrix estimate of state vector $\hat{x}_t^-$ estimation error is achieved by simulative propagation of Eq. (7.50)

$$P_t^- = g(P_{t-1}^+, Z(\hat{x}_{t-1}^+)) \tag{7.52}$$

The EKF gain matrix is in time instant t

$$K_t = P_t^- L_t^T [L_t P_t^- L_t^T + M_t R_t M_t^T]^{-1} \tag{7.53}$$

and the measurement y_t is used for updating the a posteriori estimate

$$\hat{x}_t^+ = \hat{x}_t^- + K_t[y_t - h(\hat{x}_t^-)] \tag{7.54}$$

The covariance matrix a posteriori estimate is updated as

$$P_t^+ = [I - K_t L_t] P_t^- [I - K_t L_t]^T + K_t M_t R_t M_t^T K_t^T \tag{7.55}$$

where

$$L_t = \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t = \hat{x}_t^-} \tag{7.56}$$

$$M_t = \left.\frac{\partial h(x_t)}{\partial v_t}\right|_{x_t = \hat{x}_t^-} \tag{7.57}$$

7.3.4 Moving Horizon Estimation Algorithm

The statistics of the measurement noise v_t is assumed unknown. The function composition as the application of one function to the results of another like $f(f(x_{t-N}, u_{t-N}), u_{t-N+1})$ and $h(f(x_{t-N}, u_{t-N}), u_{t-N+1})$ can be written as $f^{u_{t-N+1}} \circ f^{u_{t-N}}(x_{t-N})$ and $h^{u_{t-N+1}} \circ f^{u_{t-N}}(x_{t-N})$ respectively, where "$\circ$" denotes function composition. The $N+1$ subsequent measurements of the outputs Y_t and inputs U_t up to time t are

$$Y_t = \begin{bmatrix} y_{t-N} \\ y_{t-N+1} \\ \vdots \\ y_t \end{bmatrix}, U_t = \begin{bmatrix} u_{t-N} \\ u_{t-N+1} \\ \vdots \\ u_t \end{bmatrix}. \tag{7.58}$$

where $t = N+1, +2, \ldots$. For Y_t the following algebraic map is defined

$$Y_t = H_t(x_{t-N}, U_t) = \begin{bmatrix} h^{u_{t-N}}(x_{t-N}) \\ h^{u_{t-N+1}} \circ f^{u_{t-N}}(x_{t-N}) \\ \vdots \\ h^{u_t} \circ f^{u_{t-1}} \circ \ldots \circ f^{u_{t-N}}(x_{t-N}) \end{bmatrix} \tag{7.59}$$

The a priori state estimate used in the arrival cost at the beginning of the horizon is declared as $\bar{x}_{t-N|t}$, for which two alternatives are considered.

7.3.4.1 Simulative Propagation (Alt. 1)

The $\bar{x}_{t-N|t}$ vector in a time instant t is computed for the time instance $t-N$ by simulative propagation (Butcher, 2003; Pytlak, 1999) of function f. The initial condition of such one-step simulation is given by the last optimal state vector estimate $\hat{x}_{t-N-1|t-1}$ that is not a part of a receding window anymore

$$\bar{x}_{t-N|t} = f(\hat{x}_{t-N-1|t-1}, u_{t-N-1}) \tag{7.60}$$

7.3.4.2 Pre-filtration (Alt. 2)

The $\bar{x}_{t-N|t}$ vector is computed in a time instant t for the time instance $t-N$ by pre-filtration with EKF (Rao et al, 2003). The EKF is running at the beginning of horizon on the output data y_{t-N} which were measured in $t-N$ time instance. This is the information which corrects the one-step simulation

$$\hat{x}^{-}_{t-N|t} = f(\hat{x}_{t-N-1|t-1}, u_{t-N-1}) \tag{7.61}$$

The a priori state estimate at the beginning of the horizon is computed as

$$\bar{x}_{t-N|t} = \hat{x}^{-}_{t-N|t} + K_{t-N|t}[y_{t-N} - h(\hat{x}^{-}_{t-N|t})] \tag{7.62}$$

The covariance matrix is computed as

$$\begin{aligned} P^{+}_{t-N|t} = {} & [I - K_{t-N|t}L_{t-N|t}]P^{-}_{t-N|t}[I - K_{t-N|t}L_{t-N|t}]^T + \\ & + K_{t-N|t}M_{t-N|t}R_t M^T_{t-N|t}K^T_{t-N|t} \end{aligned} \tag{7.63}$$

The other matrix computations necessary for the pre-filtration are done via regular EKF equations as explained in Section 7.3.3 (index t changes to $t-N|t$ and index $t-1$ changes to $t-N-1|t-1$). The only difference is that the EKF equations here are applied for the first time instance $t-N$ of receding window.

Define the N-information vector at time t

$$I_t = [y^T_{t-N}, \ldots, y^T_t, u^T_{t-N}, \ldots, u^T_t]^T \tag{7.64}$$

The observer design problem is to reconstruct the vector x_{t-N} based on the information vector I_t. The basic formulation of such a problem is defined as the inverse mapping of Eq. (7.59). The unique existence and continuity of solution depends on the function H_t. If the Eq. (7.59) does not have unique solution, the problem is ill-posed according to definitions of Tikhonov and

Arsenin (1977). The solution of vector x_{t-N} is in the case of uniform observability formulated on an over-determined set of algebraic equations where there are more equations than unknowns for which $n_x \leq N n_y$. The formulation can be also under-determined if there is no persistence of excitation or the system is not observable. From the existence point of view of solution for vector x_{t-N} under noisy measurements, the computation is formulated as an optimization problem.

The cost function of the optimization problem is in the meaning of the least-squares method defined as

$$J_{LS}(\hat{x}_{t-N|t}, I_t) = \|\hat{x}_{t-N|t} - \bar{x}_{t-N|t}\|_Q^2 + \|\hat{Y}_t - Y_t\|^2 \tag{7.65}$$

with

$$\hat{Y}_t = H_t(\hat{x}_{t-N}, U_t) = \begin{bmatrix} h^{u_{t-N}}(\hat{x}_{t-N}) \\ h^{u_{t-N+1}} \circ f^{u_{t-N}}(\hat{x}_{t-N}) \\ \vdots \\ h^{u_t} \circ f^{u_{t-1}} \circ \ldots \circ f^{u_{t-N}}(\hat{x}_{t-N}) \end{bmatrix} \tag{7.66}$$

The cost function (7.65) comprises of two squared norms where the first norm is weighted by the Q matrix. The given formulation contains an arrival cost (Alessandri et al, 2008; Rao et al, 2003). The schematic time sequence of the a priori state estimate vector, state estimate vectors, output estimate vectors, input and output vectors on N-horizon are in Fig. 7.11. The MHO algorithm, schematically shown in Fig. 7.12 consists of three main computation parts: Datapool, Simulative propagation (if Alt. 1) or Pre-filtration (if Alt. 2) and Optimizer with N-step Model Simulation and Cost function minimization blocks. The main computation engine is the optimization algorithm that performs the cost function minimizations. The MHO algorithm

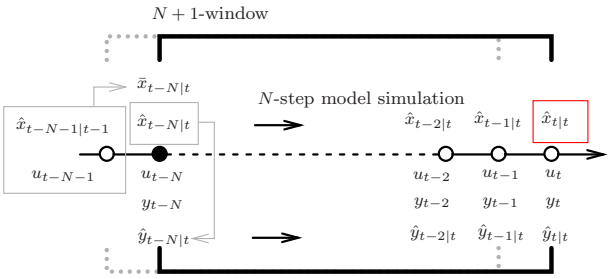


Fig. 7.11 Time sequences of state, input and output variables in $N+1$ Moving Horizon window

with pre-filtration can be summarized into following steps:

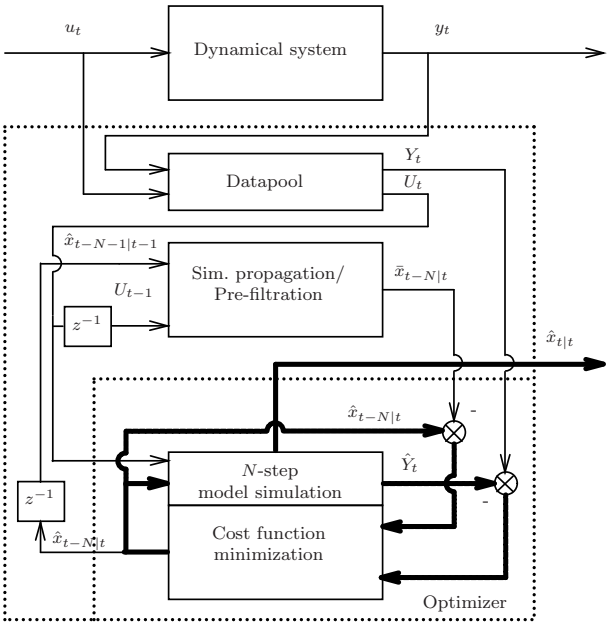


Fig. 7.12 Algorithm scheme of Moving Horizon Observer where z^{-1} is a one sample delay operator

Step 0. Do the initial Datapool loading with measurement data and input data

Step 1. Obtain the actual output measurement y_t and update the Datapool

Step 2. If the current time instance is $t = N + 1$, set the initial values for $\hat{x}^+_{0|N}$ and $P^+_{0|N}$ (as in the case of EKF Eq. (7.47),(7.48)). Then according to Fig. 7.11: $\hat{x}^+_{0|N} = \hat{x}_{0|N}$ (this value is set by the user)

Else for $t > N + 1$: $\hat{x}^+_{t-N-1|t-1} = \hat{x}_{t-N-1|t-1}$ (this value is set from the last optimization run)

Step 3. Compute the a priori estimate with Eq. (7.61)

Step 4. Numerically integrate $P^-_{t-N|t}$ (as in the case of EKF Eq. (7.50) through Eq. (7.52))

Step 5. Compute the EKF gain matrix $K_{t-N|t}$ (as in the case of EKF according to Eq. (7.53))

Step 6. Compute the a posteriori state estimate $\hat{x}^+_{t-N|t}$ with Eq. (7.62) where $\hat{x}^+_{t-N|t} = \bar{x}_{t-N|t}$ (Block in Fig. 7.12: Pre-filtration)

Step 7. Minimize the cost function (7.65) to compute the optimal state vector at the very beginning of receding window $\hat{x}_{t-N|t}$. In the minimization routine the model is used through the Eq. (7.66)

Step 8. Use the model to simulatively propagate the state from the beginning of receding window to the end of receding window as

$$\hat{x}_{t|t} = f^{u_{t-1}} \circ f^{u_{t-2}} \circ \ldots \circ f^{u_{t-N}}(\hat{x}_{t-N|t})$$

End of loop; Go to Step 1.

7.3.5 Simulations

In the following section the simulation of MHO and EKF described in above sections for oscillating mass-spring-damper system with one-degree-of-freedom will be presented. In both studied approaches (EKF, MHO), the propagation of filter dynamics in Eq. (7.49), (7.52) and the propagation of observer dynamics in Eq. (7.60), (7.61), (7.66) is required through the numerical simulation. In this experiment the Matlab function `ode23` is used which is explicit Runge-Kutta method (Pytlak, 1999; Butcher, 2003).

7.3.5.1 Model of Mass-Spring-Damper System

For an SDOF vibration system, the equation of motion may be represented as follows

$$m\ddot{q}(t) + b\dot{q}(t) + kq(t) = F(t) \tag{7.67}$$

The state-space model consists of an ordinary differential equation system with the displacement $q = x_1$, the speed $\dot{q} = x_2$ and no external force (free vibration)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{7.68}$$

Denoting $x = [x_1, x_2, x_3, x_4]^T$, with $k = x_3$ and $b = x_4$, (7.68) can be rewritten in the augmented form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{1}{m} x_3 x_1 - \frac{1}{m} x_4 x_2 \\ \dot{x}_3 &= 0 \\ \dot{x}_4 &= 0 \end{aligned} \tag{7.69}$$

where only the displacement is measurable

$$y_t = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T + v_t \tag{7.70}$$

7.3.5.2 Initialization of Simulation, MHO and EKF

The system described by Eq. (7.68) is simulated to generate data. The noisy measurement of displacement is generated through Eq. (7.70). The true parameters considered in simulation are $m = 1$, $k = 1$, $b = 0.01$. The initial true joint state and parameter vector at the beginning of simulation is then $[x_1, x_2, x_3, x_4]^T = [1, 0, 1, 0.01]^T$.The initial MHO state estimate $\hat{x}_{0|N}$, which is not a part of a receding window, has the first displacement term set directly from measurement. The other terms are considered as initially unknown and are set to 0.5. Considering the initial datapool loading with $N + 1$ measurements, the first estimated state vector with the MHO in $t = N + 1$ is at the beginning of receding window ($\hat{x}_{t-N|t}$) and further computed for the end of receding window ($\hat{x}_{t|t}$). The horizon size N is heuristically chosen long enough to capture at least one full oscillation period. The initial state estimate of EKF $\hat{x}_0^+$ has the first displacement term set directly from measurement. The other terms of $\hat{x}_0^+$ are considered as initially unknown and are set to 0.5. The EKF is N-times pre-iterated in order to use the access to the same data information as the MHO has in datapool buffer. With this initial conditions the MHO and EKF qualitatively compare their first state estimates for the same time instance $t = N + 1$. The other possibility (not applied in this paper) to compare the EKF and MHO from very first measurement instance would be to apply the MHO with growing horizon until $t = N+$

7.3.5.3 Gaussian White Noise Experiment (Exp. 1.):

In this first experiment, the noise is generated with Gaussian distribution (band-limited-white-noise) with the variance $R_t = \sigma^2 = 0.01$

$$v_t \sim N(0, \sigma^2) \tag{7.71}$$

The measured noisy data have overall signal-to-noise ratio $SNR = 10$.

7.3.5.4 Correlated Noise Experiments (Exp. 2a., Exp. 2b.):

In the second and third experiment sequentially correlated, sometimes referred as colored noise, will be assumed. The noise is given by the filtration of band-limited-white-noise input $e_t \sim N(0, \sigma^2)$, $\sigma^2 = 0.01$ with the filter (Hansen and Snyder, 1997)

$$v_t = \frac{0.5}{1.0 - 1.75z^{-1} + 0.81z^{-2}} e_t \tag{7.72}$$

This filter produces the colored noise which in addition to the base displacement signal produces the signal with average signal-to-noise ratio $SNR = 5$.

Two different noise realization sequences are produced by the filter, given by Eq. (7.72). The first realization is used in *Exp. 2a.* and the second realization in *Exp. 2b.*

7.3.6 Extended Kalman Filter Setup

In order to compute the continuous part of the EKF (7.50), the Jacobian matrix (7.51) is required to propagate Eq. (7.50) through Eq. (7.52). The partial derivative matrices for the mass-spring-damper system are

$$\begin{aligned} Z(\hat{x}) &= \left.\frac{\partial f_c(x)}{\partial x}\right|_{x=\hat{x}} \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{m}x_3 & -\frac{1}{m}x_4 & -\frac{1}{m}x_1 & -\frac{1}{m}x_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{x=\hat{x}} \\ L_t &= \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t=\hat{x}_t^-} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\ M_t &= \left.\frac{\partial h(x_t)}{\partial v_t}\right|_{x_t=\hat{x}_t^-} = 1 \end{aligned} \tag{7.73}$$

In the numerical simulation with Eq. (7.52), the argument of the Jacobian matrix $Z(\hat{x})$ "continuously" changes the values from $\hat{x} = \hat{x}_{t-1}^+$ to $\hat{x} = \hat{x}_t^-$. The values of the argument $\hat{x}$ are changed within the simulative propagation step which is much smaller than the filter sampling interval. The initial conditions as explained in section 7.3.5.2 are set as follows.

7.3.6.1 Exp. 1.

The vector $\hat{x}_0^+ = [1.052, 0.5, 0.5, 0.5]^T$ is the initial state estimate and the initial covariance matrix with the measurement variance is $P_0^+ = diag(0.1, 0.1, 0.1, 0.1)$, $R = 0.01$. The pre-iterated covariance matrix after $N = 10$ steps is

$$P_{10}^+ = \begin{bmatrix} 0.0062 & 0.0053 & 0.0024 & -0.0039 \\ 0.0053 & 0.0142 & -0.0017 & -0.0107 \\ 0.0024 & -0.0017 & 0.0038 & 0.0015 \\ -0.0039 & -0.0107 & 0.0015 & 0.0086 \end{bmatrix} \tag{7.74}$$

7.3.6.2 Exp. 2a., 2b.

The initial covariance matrix with the measurement variance is $P_0^+ = diag(0.3, 0.3, 0.3, 0.3)$, $R = 0.1$. The initial state estimate for *Exp. 2a.* is $\hat{x}_0^+ = [1.352, 0.5, 0.5, 0.5]^T$. The pre-iterated covariance matrix for *Exp. 2a.*

after $N = 10$ steps is

$$P_{10}^{+} = \begin{bmatrix} 0.0182 & -0.0051 & 0.0287 & 0.0145 \\ -0.0051 & 0.0187 & -0.0000 & -0.0375 \\ 0.0287 & -0.0000 & 0.0627 & 0.0095 \\ 0.0145 & -0.0375 & 0.0095 & 0.0873 \end{bmatrix} \tag{7.75}$$

The vector $\hat{x}_0^+ = [0.892, 0.5, 0.5, 0.5]^T$ is the initial state estimate for *Exp. 2b.*

7.3.7 Moving Horizon Observer Setup

To minimize the cost function (7.65) Matlab unconstrained optimization function `fminunc` is called. The following equation for the Q matrix is motivated by Rao et al (2003)

$$Q = RP^{-1} \tag{7.76}$$

7.3.7.1 Exp 1./Simulative Propagation (Alt. 1)

The moving horizon window is set to $N = 10$. The Q matrix is time-invariant since $P = P_{10}^{+}$ (Eq. (7.74)) during the whole simulation, $R = 0.01$. The initial value of a priori state vector for $t = N + 1$ is $\hat{x}_{0|N} = [1.052, 0.5, 0.5, 0.5]^T$.

7.3.7.2 Exp 1./Pre-filtration (Alt. 2)

The Q matrix is time-varying since $P = P_{t-N|t}^{+}$ according to Eq. (7.63), $R = 0.01$. The initial value of a priori state vector for $t = N + 1$ is $\hat{x}_{0|N} = [1.052, 0.5, 0.5, 0.5]^T$ and $P_{0|N}^{+} = diag(0.1, 0.1, 0.1, 0.1)$.

7.3.7.3 Exp. 2a., 2b./Simulative Propagation (Alt. 1)

This method was not applied for the correlated noise experiments, due to its poor slow convergence for the Gaussian noise experiment.

7.3.7.4 Exp. 2a., 2b./Pre-filtration (Alt. 2)

The moving horizon window is set to $N = 10$. The Q matrix is time-varying since $P = P_{t-N|t}^{+}$, Eq. (7.63), with the initial value $P_{0|N}^{+} = diag(0.3, 0.3, 0.3, 0.3)$ and $R = 0.1$. The initial value of a priori state vector for $t = N+1$ for *Exp. 2a.* is $\hat{x}_{0|N} = [1.352, 0.5, 0.5, 0.5]^T$ and for *Exp. 2b.* is $\hat{x}_{0|N} = [0.892, 0.5, 0.5, 0.5]^T$.

7.3.8 Simulation Results and Discussion

The system dynamics is perturbed by the initial state deviation from its equilibrium without any external force input leaving the system to respond freely. Such free oscillatory response with decaying trend is providing sufficient self excitation needed for the observer to successfully converge. The quality of the algorithms is evaluated by the Root Mean Square Error (RMSE) computed for each state and parameter as

$$RMSE_{x_j} = \sqrt{\sum_{i=1}^{n} \frac{(x_j - \hat{x}_{j,i})^2}{n}} \tag{7.77}$$

where $j = 1, 2, 3, 4$ and $n = 100$.

7.3.8.1 Gaussian White Noise Experiment (Exp. 1.):

The MHO observer estimation is run with two different settings for the a priori state vector computation according to the Simulative propagation (Alt. 1) and Pre-filtration (Alt. 2) procedures. For better distinction of the different settings, the estimated states of the system are presented by their error from the true states. The displacement error is shown in Fig. 7.13 and the speed error is shown in Fig. 7.14. The estimation of spring and damping constants is in Fig. 7.15 and 7.16 respectively. The results show comparable convergence of states and parameters for the Pre-filtered MHO and the EKF with pre-iterations. The Simulative propagation did not give satisfactory results mainly due to its non-adaptivity of the Q matrix, which is given by (7.76). The results for different MHO settings and the results of EKF are summarized in Table 7.5.

RMSE.10^{-2}	x_1	x_2	x_3	x_4
EKF	**0.2953**	**0.3616**	0.1466	0.1520
Pre-filtered MHO	0.4392	0.4895	**0.1234**	**0.1382**
Sim. propag. MHO	2.2722	1.9598	1.3277	3.6531

Table 7.5 Root Mean Square Error in *Exp. 1.*

7.3.8.2 Correlated Noise Experiments (Exp. 2a., Exp. 2b):

The magnitude of noise is slightly higher to what one can expect from regular sensor dynamics, however the correlated character sets a challenging problem for the EKF and the MHO. The magnitude and character of colored noise

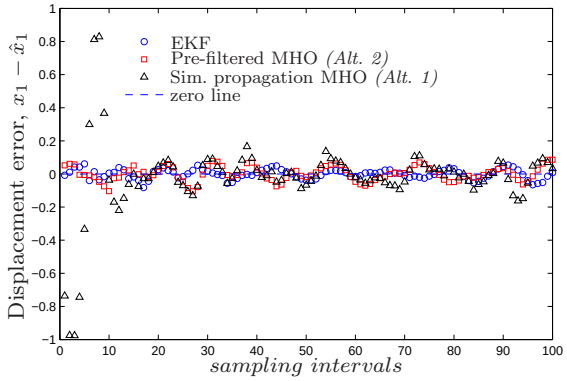


Fig. 7.13 *Exp. 1.*: Displacement error between true state and the estimate

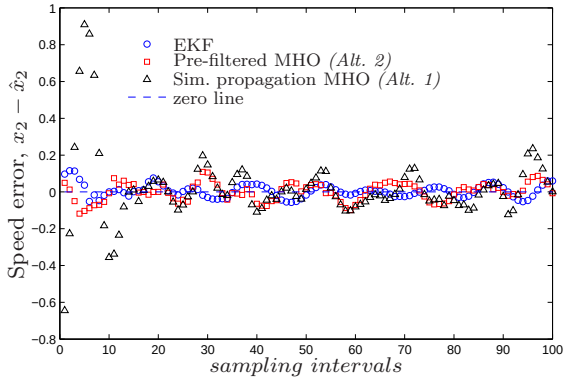


Fig. 7.14 *Exp. 1.*: Speed error between true state and the estimate

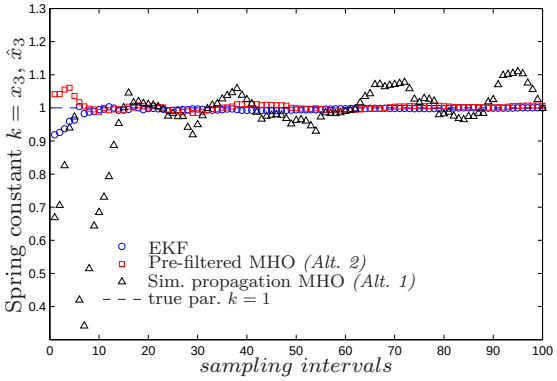


Fig. 7.15 *Exp. 1.*: Estimation of spring constant

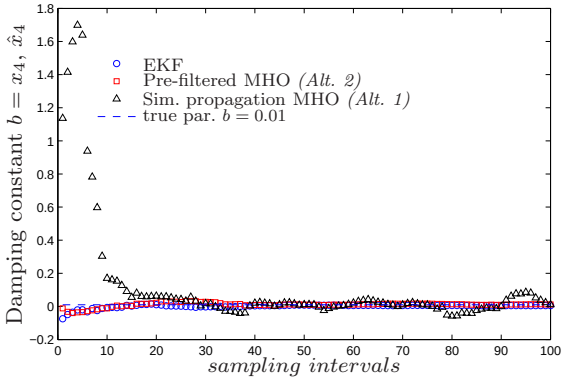


Fig. 7.16 *Exp. 1.*: Estimation of damping constant

may cause disturbance to the estimation algorithms to successfully converge. In this experiment the Pre-filtered MHO algorithm showed improved robust convergence ability compared to the EKF. The EKF turned out to be very sensitive to converge from the initial estimate or from reasonable surrounding set of initial estimates, compared to the more robust MHO algorithm where even large initial state estimate error can be corrected at the very beginning by the measured information contained in the Datapool. The displacement estimate is shown in Fig. 7.17 and the speed estimate is shown in Fig. 7.18. The estimation of spring and damping constants is in Fig. 7.19 and 7.20 respectively. The sensitivity of the pre-iterated EKF is demonstrated on these figures where the method eventually diverges on noisy data and identifies incorrect states in *Exp. 2a.*. The RMSE results are summarized in Table 7.6. The results of the *Exp. 2b.* are shown in Figs. 7.21–7.24 and summarized in Table 7.7.

RMSE.10^{-2}	x_1	x_2	x_3	x_4
EKF	5.8142	6.0577	1.8004	4.3752
Pre-filtered MHO	**2.0800**	**3.6054**	**1.5023**	**1.4773**

Table 7.6 Root Mean Square Error in *Exp. 2a.*

RMSE.10^{-2}	x_1	x_2	x_3	x_4
EKF	3.0661	3.5506	1.2009	**0.7403**
Pre-filtered MHO	**2.7170**	**3.2849**	**0.8892**	0.9224

Table 7.7 Root Mean Square Error in *Exp. 2b.*

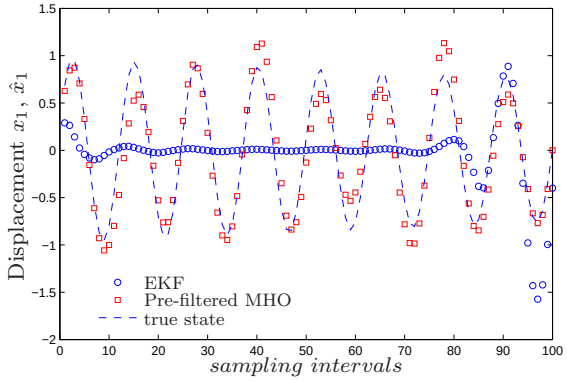


Fig. 7.17 *Exp. 2a.*: True state of displacement and the estimate

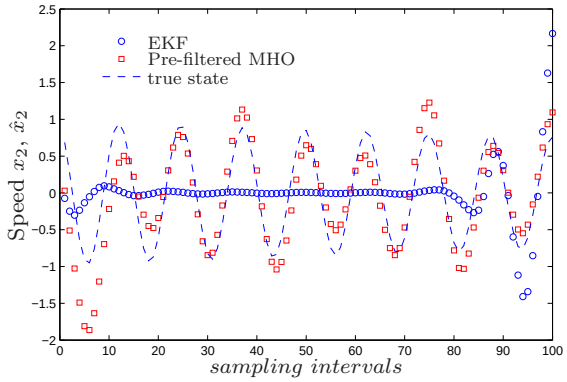


Fig. 7.18 *Exp. 2a.*: True state of speed and the estimate

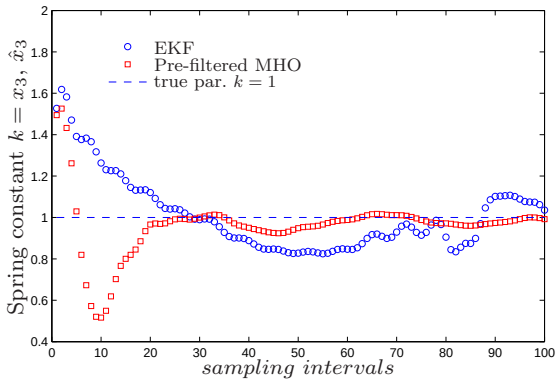


Fig. 7.19 *Exp. 2a.*: Estimation of spring constant

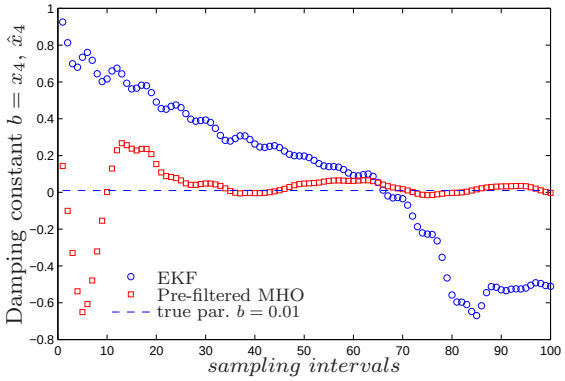


Fig. 7.20 *Exp. 2a.*: Estimation of damping constant

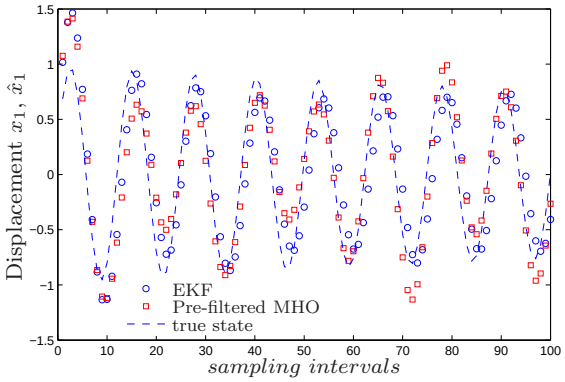


Fig. 7.21 *Exp. 2b.*: True state of displacement and the estimate

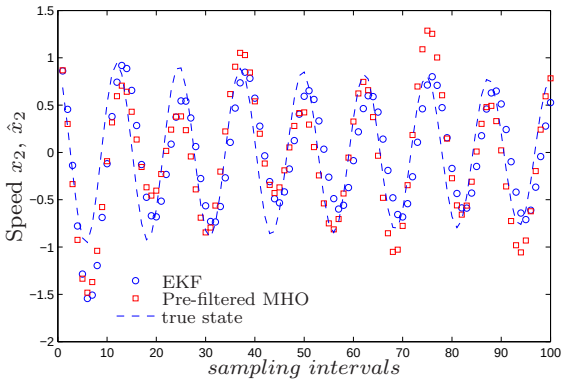


Fig. 7.22 *Exp. 2b.*: True state of speed and the estimate

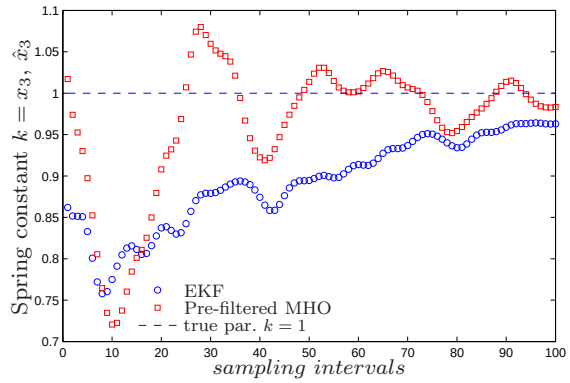


Fig. 7.23 *Exp. 2b.*: Estimation of spring constant

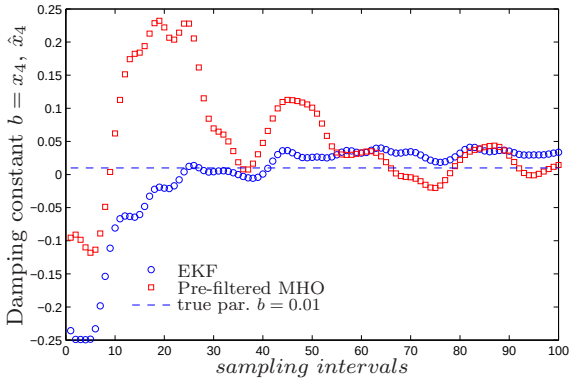


Fig. 7.24 *Exp. 2b.*: Estimation of damping constant

7.3.9 Conclusion

The MHO and EKF estimation algorithms are tested in three different numerical experiments. In the first experiment the white noise and in the second and third experiment the correlated noise are assumed to superpose on the true displacement signal. The corrupted displacement is the on-line measured information used by the MHO and EKF to compute the estimation of states: displacement, speed and parameters: spring constant, damping constant. The experiments indicate that no method is always better. In the second experiment the MHO demonstrates robustness with an ability to safely converge and extract the dynamic information about states and parameters. The further advantage of MHO is that it can directly handle constraints on states and parameters. This was not applied here (although $x_4 > 0$ is evident), but the

application of constraints is straightforward and would improve the performance. Also modeling the colored noise would further improve the estimates. The recursive Prediction Error Methods could in this case be considered as suitable alternative to the EKF. The application of proposed algorithms in embedded controllers depends on reliable function minimization routines. The recent computationally fast and efficient methods of function minimization based on Sequential Quadratic Programming for real-time applications are proposed in Diehl et al (2009).

7.4 Predictive Control of Air-Fuel Ratio in Spark Ignition Engines

7.4.1 Introduction

The problem of air-fuel ratio (AFR) control is one of the main parts of the more complex emission reduction strategy for combustion engines. The mixture quality is essential for efficiency of a three-way catalytic converter and therefore proper control techniques are needed to fulfil emission legislations. During the last twenty years different control methodologies were developed from simple to more sophisticated "model (observer)-based" ones. In advanced control methods the model plays the most important role in the control strategy (Muske, 2006). A classical approach to modeling problem of AFR is based on linear observer theory where physical models of the process are part of state estimator (Powell et al, 1998; Guzzella and Onder, 2010). A review of observers based on physical laws related to "gray-box" models can be found in (Hendricks and Luther, 2001). Another promising branch of control model-based strategies relies on "black-box" modeling principles where identified models are used. From the field of nonlinear approximation theory many different nonlinear model structures have been applied to engine emission control problems. One of the most popular approaches to combustion engine modeling is based on neural network principles for their flexibility (Nelles, 2001). Especially, the AFR modeling problem was solved by radial basis function observer in Manzie et al (2002), by Chebyshev polynomial network in Gorinevsky et al (2003) and recently a simulator of AFR dynamics based on recurrent neural network was proposed by Arsie et al (2006). The purpose of this study is to design the AFR predictive controller based on linear parameter varying model (LPV) of the AFR and a simulative verification of its ability to maintain stoichiometric mixture during transients throughout different operating regimes of the 2.8 liter engine.

7.4.2 Model Structure

This section describes the model structure. First, a general weighted linear local model with single input single output (SISO) structure is presented. Specifically, composite local linear ARX models with weighted validity (Murray-Smith and Johansen, 1997) are identified to model AFR nonlinear dynamics. The global AFR model is then validated against measured data (Polóni et al, 2008). Weighted linear local models (LLM) have already been used in engine emission NOx control applications as an extension of radial basis function network sometime referred to as local linear neuro-fuzzy tree network (Hafner et al, 1999; Isermann and Müller, 2003) and also in diesel engine drivetrain modeling (Johansen et al, 1998). Below it is shown how this structure can be applied for modeling of AFR dynamics of the engine.

7.4.2.1 Weighted Linear Local Model Network Structure

The basic principle of this nonlinear modeling technique is partitioning the operating regimes. For these operating regimes LLMs are defined. The transition between particular local models is fluent due to smooth interpolation (weighting) functions. In this case the local models will be linear ARX models with weighted parameters in an operating point $\phi \in \Phi \subset \mathbb{R}^{n_\phi}$,

$$\sum_{h=1}^{n_M} \rho_h(\phi(k)) A_h(q) y(k) = \sum_{h=1}^{n_M} \rho_h(\phi(k)) B_h(q) u(k) + \\ + \sum_{h=1}^{n_M} \rho_h(\phi(k)) c_h + e(k) \tag{7.78}$$

defining polynomials A_h and B_h

$$\begin{aligned} A_h(q) &= \quad 1 + a_{h,1} q^{-1} + \ldots + a_{h,n_y} q^{-ny} \\ B_h(q) &= b_{h,1+d_h} q^{-1-d_h} + \ldots + b_{h,n_u+d_h} q^{-nu-d_h} \end{aligned} \tag{7.79}$$

where $a_{h,i}$,$b_{h,(j+d_h)}$,c_h are the h-*th* local function parameters and d_h is the delay. The parameters n_M and n_γ stand for the number of local models and size of the regression vector (7.85) respectively. Here q^{-1} is the time shift operator, i.e. $q^{-i} y(k) = y(k-i)$. The Gaussian local model validity function $\{\tilde{\rho_h} : \Phi \to (0,1)\}_{h=1}^{n_M}$ is defined by the vector of center $\boldsymbol{c}_{c,h} \in \mathbb{R}^{n_\phi}$ and by the scaling matrix $\boldsymbol{M}_h$

$$\tilde{\rho_h}(\phi(k)) = e^{-(\phi(k)-\boldsymbol{c}_{c,h})^T \boldsymbol{M}_h (\phi(k)-\boldsymbol{c}_{c,h})} \tag{7.80}$$

$$\boldsymbol{M}_h = \begin{pmatrix} \frac{1}{\sigma^2_{h,1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma^2_{h,2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma^2_{h,n_\phi}} \end{pmatrix} \tag{7.81}$$

The function $\tilde{\rho_h}$ can be considered as degree of fulfilment (even though it is not a combination of antecedent fuzzy sets). To achieve a partition of unity, local model validity functions are normalized to get the weighting functions used which is based on Takagi-Sugeno fuzzy inference (Takagi and Sugeno, 1985).

$$\rho_h(\phi(k)) = \frac{\tilde{\rho_h}(\phi(k))}{\sum_{h=1}^{n_M} \tilde{\rho_h}(\phi(k))} \tag{7.82}$$

That means in any operating point $\sum_{h=1}^{n_M} \rho_h(\phi(k)) = 1$. For simulation of the model (7.78) following equation has to be considered

$$\begin{aligned} y_s(k) = \sum_{h=1}^{n_M} \rho_h(\phi(k)) \Big(& - \sum_{i=1}^{ny} \hat{a}_{h,i} q^{-i} y_s(k) + \\ & + \sum_{j=1}^{nu} \hat{b}_{h,(j+d_h)} q^{-j-d_h} u(k) + \hat{c}_h \Big) \end{aligned} \tag{7.83}$$

Introducing the estimated parameter vector $\hat{\theta}_h$ and the regression vector $\gamma(k)$ with $d_{max} = max\{d_h\}_{h=1}^{n_M}$

$$\begin{aligned} \hat{\theta}_h = [& \hat{a}_{h,1}, \hat{a}_{h,2}, \ldots, \hat{a}_{h,ny}, \{0,0,\ldots,0\}_{d_h}, \\ & \hat{b}_{h,1+d_h}, \hat{b}_{h,2+d_h}, \ldots, \hat{b}_{h,nu+d_h}, \\ & \{0,0,\ldots,0\}_{d_{max}-d_h}]^T \end{aligned} \tag{7.84}$$

$$\begin{aligned} \gamma(k) = [& -y_s(k-1), -y_s(k-2), \ldots, -y_s(k-ny), \\ & u(k-1), u(k-2), \ldots, u(k-nu-d_{max})]^T \end{aligned} \tag{7.85}$$

equation (7.83) becomes,

$$y_s(k) = \gamma^T(k) \sum_{h=1}^{n_M} \rho_h(\phi(k)) \hat{\theta}_h + \sum_{h=1}^{n_M} \rho_h(\phi(k)) \hat{c}_h \tag{7.86}$$

The offset term c_h of the local ARX model can be computed from the system's steady state values $y_{e,h}, u_{e,h}$. Given a parameter estimate $\hat{\theta}_h$, the estimate of c_h is defined as follows

$$\hat{c}_h = y_{e,h} + y_{e,h} \sum_{i=1}^{ny} \hat{a}_{h,i} - u_{e,h} \sum_{j=1}^{nu} \hat{b}_{h,j} \tag{7.87}$$

A block diagram illustrating Eq. (7.86) can be seen in Fig. 7.25. There are several possibilities how to estimate the parameters and weights of model

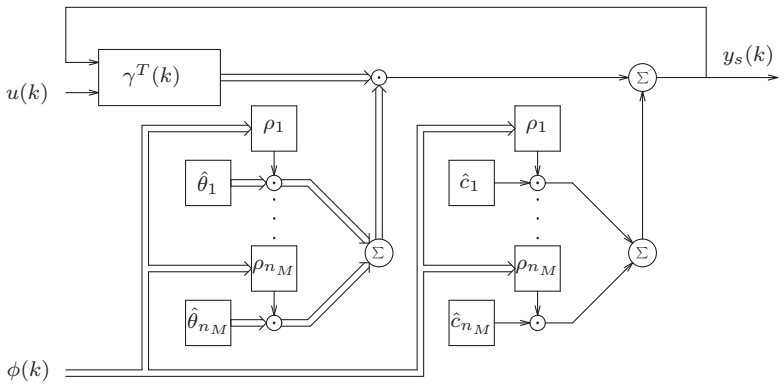


Fig. 7.25 Weighted ARX local model network structure

(7.78). This is discussed in Johansen and Foss (1993) and Takagi and Sugeno (1985).

7.4.2.2 Air-Fuel Ratio Model Structure

In this section we briefly introduce the AF ratio model structure, see Polóni et al (2008) for further details. The nonlinear (parameter varying) model is needed mainly due to nonlinear throttle characteristic Heywood (1988) and delay-varying AFR dynamics. The dynamic model of AFR is based on a definition of a mixture as a ratio of air and fuel quantities in time instance (k). Since $\lambda(k)$ is a non-dimensional ratio the air and fuel quantities can be expressed in any physical units, even relative ones. It is convenient to express these quantities in the meaning of relative mass densities ($[g/cylinder]$) telling us how much mass of air (or fuel) is concentrated per volume of one cylinder. The relative mass density of the mixture consists of relative air density $m_a(k)$ and relative fuel density $m_f(k)$ that define the mixture quality in a time instance (k). The effect of mixture formation is transformed from the discrete event process (one combustion cycle) to continuous changes of AFR information due to mixing dynamics in the exhaust manifold. To scale the AFR at one for stoichiometric mixture ($\lambda_{st} = 1$), we divide the ratio by the value of theoretical stoichiometric coefficient for gasoline fuel $L_{th} \approx 14.64$, so the ratio is defined

$$\lambda(k) = \frac{1}{L_{th}} \frac{m_a(k)}{m_f(k)} [-] \tag{7.88}$$

The $m_a(k)$ and $m_f(k)$ information can be indirectly measured with a delay at the confluence point (Fig. 7.26). To model $\lambda(k)$, two different subsystems with independent inputs are considered. The air-path subsystem (m_a) with a throttle position (t_r) input as a disturbance variable (DV) and the fuel-path

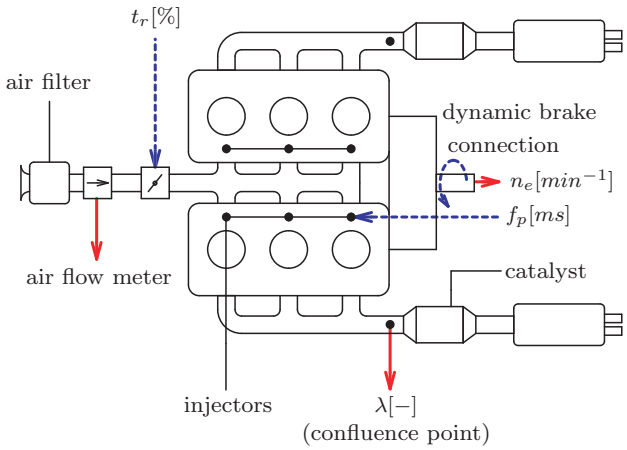


Fig. 7.26 Engine setup with input/output relations; dashed arrows - inputs, solid arrows - outputs

subsystem (m_f) with an injection pulse width (u_f) input as a manipulated variable (MV). The other DV is the engine speed (n_e) which is implicitly included in the model to define the operating point together with t_r. In accordance with the general model structure presented in Section 7.4.2.1 the key variables are defined in Table 7.8. In the operating point vector $\phi(k)$

Table 7.8 Symbol connection between general expression and the model

general symbol	air-path model	fuel-path model	operating point
$y(k)$	$m_a(k)$	$m_f(k)$	
$u(k)$	$t_r(k)$	$u_f(k)$	
$\gamma(k)$	$\gamma_a(k)$	$\gamma_f(k)$	
$\hat{\theta}_h$	$\hat{\theta}_{a,h}$	$\hat{\theta}_{f,h}$	
$\rho_h(\phi(k))$	$\rho_{a,h}(\phi(k))$	$\rho_{f,h}(\phi(k))$	
$\hat{c}_h$	$\hat{c}_{a,h}$	$\hat{c}_{f,h}$	
$\phi(k)$			$[n_e(k), t_r(k-\delta)]^T$

the parameter δ represents the throttle position delay. To simulate the AFR dynamics we combine (7.86) with (7.88)

$$\lambda_s(k) = \frac{1}{L_{th}} \left[\frac{\gamma_a^T(k) \sum_{h=1}^{n_A} \rho_{a,h}(\phi(k)) \hat{\theta}_{a,h} + \sum_{h=1}^{n_A} \rho_{a,h}(\phi(k)) \hat{c}_{a,h}}{\gamma_f^T(k) \sum_{h=1}^{n_F} \rho_{f,h}(\phi(k)) \hat{\theta}_{f,h} + \sum_{h=1}^{n_F} \rho_{f,h}(\phi(k)) \hat{c}_{f,h}} \right] \quad (7.89)$$

The weighting functions considered for the global AFR model are shown in Fig.s 7.27 and 7.28.

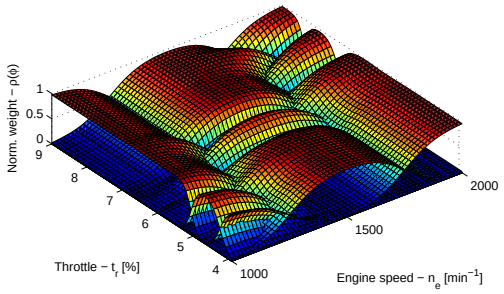


Fig. 7.27 Weighting function - air path

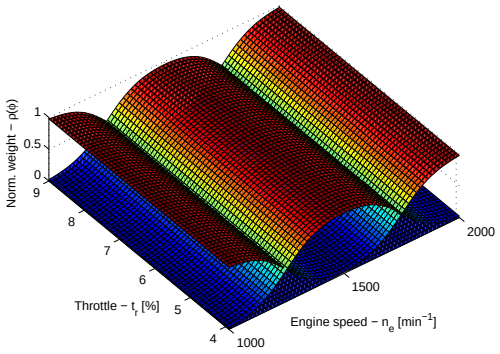


Fig. 7.28 Weighting function - fuel path

7.4.3 Predictive Controller Design

The applied control strategy is based on the knowledge of an internal model[1] (IM) of air-path, predicting the change of air flow through cylinders, and consequently, setting the profile of desired values of the objective function on the control horizon. The second modelled subsystem of the fuel-path is an explicit component of the objective function where the amount of the fuel is a function of optimized control action.

[1] Implying from IM strategy, we write y_s in (7.85) and (7.95) as internally simulated outputs

7.4.3.1 Linear Predictive Model

The predictive control can come out from several model structures of the system that lead to different computation algorithms of the control action (Maciejowski, 2002). The proposed MPC stands on linearised process model similarly used in Roubos et al (1999) and Mollov et al (2004). 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

$$\begin{aligned} x_{(a,f)}(k+1) &= A_{(a,f)}(\phi)x_{(a,f)}(k) + B_{(a,f)}(\phi)u_{(a,f)}(k) \\ m_{s,(a,f)}(k) &= C_{(a,f)}x_{(a,f)}(k) \end{aligned} \tag{7.90}$$

This is a non-minimal SS representation whose advantage is, that no state observer is needed. The individual vectors and matrices of equation (7.90) are defined as follows[2]

$$A_{(a,f)}(\phi(k)) = \begin{bmatrix} -a_1(\phi(k)) & -a_2(\phi(k)) & \cdots & -a_{ny-1}(\phi(k)) & -a_{ny}(\phi(k)) & b_2(\phi(k)) & \cdots \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} \cdots & b_{nu-1+d_{max}}(\phi(k)) & b_{nu+d_{max}}(\phi(k)) & c(\phi(k)) \\ \cdots & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 \\ \ddots & \vdots & \vdots & \vdots \\ \cdots & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 \\ \ddots & \vdots & \vdots & \vdots \\ \cdots & 1 & 0 & 0 \\ \cdots & 0 & 0 & 1 \end{bmatrix}_{(a,f)} \tag{7.91}$$

[2] For a more compact notification we write ϕ instead of $\phi(k)$ in all the equations

$$x_{(a,f)}(k) = \begin{bmatrix} y_s(k) \\ y_s(k-1) \\ \vdots \\ y_s(k-ny+1) \\ u(k-1) \\ u(k-2) \\ \vdots \\ u(k-nu-d_{max}) \\ 1 \end{bmatrix}_{(a,f)} \tag{7.92}$$

$$B_{(a,f)}(\phi) = \begin{pmatrix} b_1(\phi)\ 0\ 0 \ldots 0\ 1\ 0 \ldots 0\ 0 \end{pmatrix}^T_{(a,f)} \tag{7.93}$$

$$C_{(a,f)} = \begin{pmatrix} 1\ 0 \ldots 0\ 0\ 0 \ldots 0\ 0\ 0 \end{pmatrix}_{(a,f)} \tag{7.94}$$

The parameters of multi-ARX models are scheduled by operating point $\phi(k)$ according to (7.86) and final weighted parameters are displayed in matrices $A_{(a,f)}$ and $B_{(a,f)}$ for both subsystems. The control of the fuel pulse width is tracking of the air mass changing profile on a prediction horizon from IM of the air-path, with the amount of injected fuel mass. Due to tracking offset elimination, the SS model of the fuel-path (7.90) (index f) is written in augmented SS model form to incorporate integral action

$$\tilde{x}_f(k+1) = \tilde{A}_f(\phi)\tilde{x}_f(k) + \tilde{B}_f(\phi)\Delta u_f(k) \tag{7.95}$$

or

$$\begin{bmatrix} x_f(k+1) \\ u_f(k) \end{bmatrix} = \begin{bmatrix} A_f(\phi) & B_f(\phi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_f(k) \\ u_f(k-1) \end{bmatrix} + \\ + \begin{bmatrix} B_f(\phi) \\ 1 \end{bmatrix} \Delta u_f(k)$$

$$m_{s,f}(k) = \tilde{C}_f \tilde{x}_f(k) + D_f \Delta u_f(k) \tag{7.96}$$

or

$$m_{s,f}(k) = \begin{bmatrix} C_f & D_f \end{bmatrix} \tilde{x}_f(k) + D_f \Delta u_f(k)$$

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

$$\underrightarrow{m}_a(k) = \Gamma_a(\phi) x_a(k) + \Omega_a(\phi)\, \underrightarrow{t}_r(k-1) \tag{7.97}$$

Due to unprecise modeling (IM strategy), biased predictions of the air mass future trajectory and consequently fuel mass might occur. This error can be compensated by the term $L[\hat{m}_f(k) - m_{s,f}(k)]$ in fuel mass prediction equation ($\underrightarrow{m}_f$)

$$\begin{aligned}\underrightarrow{m}_f(k) = \Gamma_f(\phi)\tilde{x}_f(k) + \Omega_f(\phi)\Delta\underrightarrow{u}_f(k-1) + \\ + L[\hat{m}_f(k) - m_{s,f}(k)]\end{aligned} \tag{7.98}$$

The matrices of free response Γ_a, Γ_f and forced response Ω_a, Ω_f are computed from models (7.90) and (7.95) respectively (Maciejowski, 2002). Since there is only $\lambda(k)$ measurable in equation (7.88), the value of $m_a(k)$ needs to be substituted using IM of the air-path, then

$$\hat{m}_f(k) = \frac{1}{L_{th}} \frac{m_{s,a}(k)}{\lambda(k)} \tag{7.99}$$

The estimate $\hat{m}_f(k)$ is used to compensate for possible bias errors of predicted $\underrightarrow{m}_f(k)$ in (7.98).

7.4.3.2 Computation of the Control Action

The controller indirect setpoint is $\lambda = \lambda_{st} = 1$, and from (7.88) we define the control objective

$$m_f(k) - \frac{m_a(k)}{L_{th}} = 0 \tag{7.100}$$

The objective function for the AFR problem is then defined and written for chosen prediction horizon N in matrix formulation

$$J = \left[\underrightarrow{m}_f - \frac{\underrightarrow{m}_a}{L_{th}}\right]^T Q \left[\underrightarrow{m}_f - \frac{\underrightarrow{m}_a}{L_{th}}\right] + \Delta\underrightarrow{u}_f^T R \Delta\underrightarrow{u}_f \tag{7.101}$$

The control action computation stands on a minimization of the objective function

$$\Delta\underrightarrow{u}_f = \arg\min_{\Delta\underrightarrow{u}_f} J \tag{7.102}$$

subject to (7.95) and (7.98). For the sake of simplicity, the correction of the bias in (7.98) is omitted and analytical solution for constraint free case (Rossiter, 2003) is

$$\begin{aligned}\Delta\underrightarrow{u}_f = -\left[\Omega_f^T Q \Omega_f + R\right]^{-1} \cdot \\ \cdot \left[\tilde{x}_f^T \Gamma_f^T Q \Omega_f - \left[\frac{\underrightarrow{m}_a}{L_{th}}\right]^T Q \Omega_f\right]^T\end{aligned} \tag{7.103}$$

Incremental controller can be expressed in the meaning of receding horizon as

$$u_f(k) = u_f(k-1) + \Delta u_f(k) \tag{7.104}$$

7.4.4 Simulation

The ability to control the mixture concentration at stoichiometric level is demonstrated through the simulation of an experimentally validated model Polóni et al (2008). The control scheme is shown in Fig. 7.29. In the simulation the sudden changes of throttle position with changing load (see engine speed (n_e)) were considered to shift the operating regime. The nonlinear character mainly caused by the throttle can be seen at m_a, especially on different system gain in speed regimes around 1000 and 2000 min^{-1}. Simulation results are displayed in Fig. 7.30. The predicted $\overrightarrow{\lambda}(k)$ computed from the prediction of setpoint profile $\frac{\overrightarrow{m_a}}{L_{th}}(k)$, tracked by predicted fuel mass $\overrightarrow{m_f}(k)$, are depicted in Fig. 7.31. The middle graph in Fig. 7.31 also shows the record of all computed λ predictions, from the beginning to the (k) period of AFR control simulation.

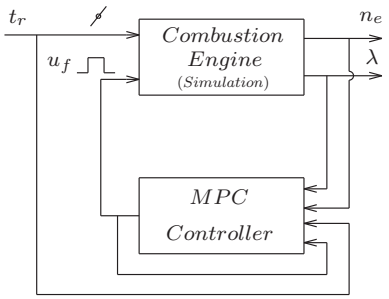


Fig. 7.29 Control scheme

7.4.5 Conclusion

In this article, we present preliminary design of a predictive controller for SI-engine air-fuel ratio. The control as well as the prediction are based on an ARX model network where the knowledge of physical phenomena is included a priori into assumptions that are utilized to design the model structure. The results are acceptable from the simulation point of view. However one has to expect worse results in real situation, particularly in λ peak overshoots. The control is based on internal model (IM) simulation strategy, with throttle position measurement, without mass air flow sensor or intake manifold pressure sensor. For future real time applications the algorithm is expressed in a simple analytical form (without constraints) which brings rather lower computational demands on hardware.

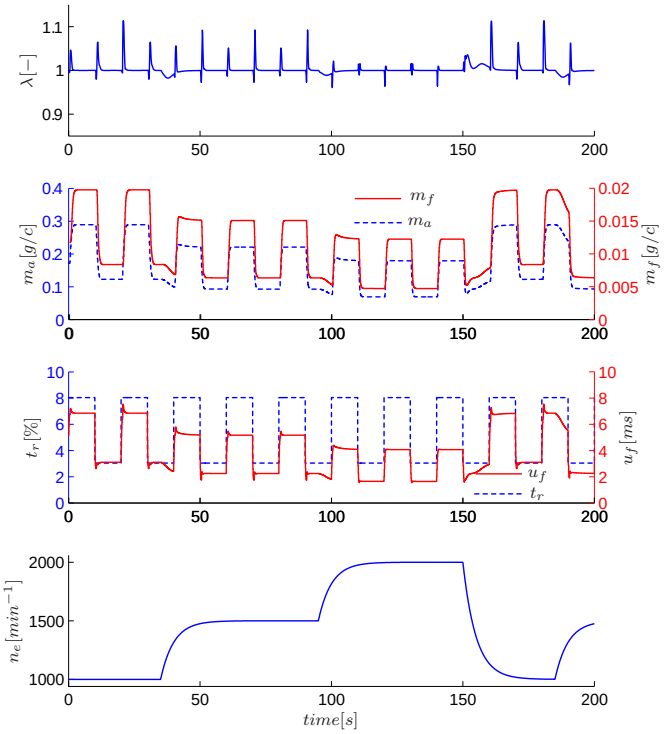


Fig. 7.30 Simulation of the engine air-fuel ratio control

Acknowledgements The authors gratefully acknowledge the financial support granted by the Slovak Research and Development Agency under the contracts APVV-0160-07 and LPP-0118-09. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This grant is co-financed from the state budget of the Slovak Republic.

References

Alessandri A, Baglietto M, Battistelli G (2008) Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. Automatica 44(7):1753–1765, DOI http://dx.doi.org/10.1016/j.automatica.2007.11.020

Arsie I, Pianese C, Sorrentino M (2006) A procedure to enhance identification of recurrent neural networks for simulating air-fuel ratio dynamics in si engines. En-

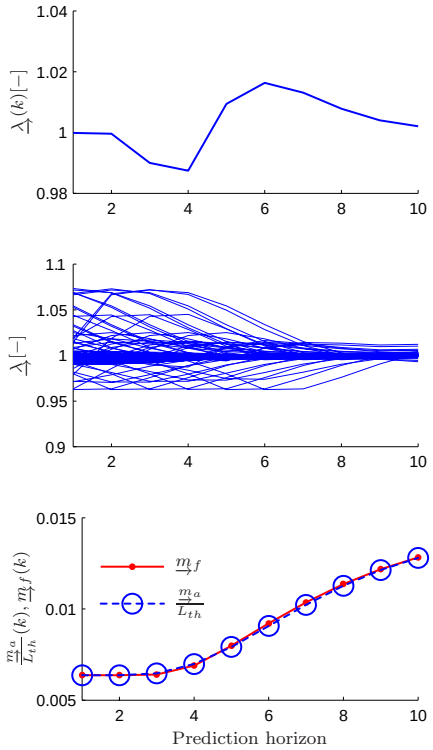


Fig. 7.31 Tracking fuel mass based on air mass setpoint on the prediction horizon ((*k*) period)

gineering Applications of Artificial Intelligence 19:65–77
Butcher J (2003) Numerical Methods for Ordinary Differential Equations. John Wiley&Sons, Ltd
Cannon M, Kouvaritakis B (2005) Optimizing Prediction Dynamics for Robust MPC. IEEE Transactions on Automatic Control 50(11):1892–1597
Chen H, Allgöver F (1998) A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34(10):1205–1217
Ching J, Beck JL, Porter KA (2006) Bayesian state and parameter estimation of uncertain dynamical systems. Probabilistic Engineering Mechanics 21:81–96
Corigliano A, Mariani S (2004) Parameter identification in explicit structural dynamics: performance of the extended kalman filter. Comput Methods Appl Mech Engrg 193:3807–3835
Diehl M, Ferreau HJ, Haverbeke N (2009) Nonlinear Model Predictive Control, Springer, chap Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, pp 317–417. Lecture Notes in Control and Information Sciences
Dongarra J (2002) Basic linear algebra subprograms technical forum standard. International Journal of High Performance Applications and Supercomputing 16:1–111

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, 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

Fuller C, Elliott S, Nelson P (1996) Active Control of Vibration. Academic Press Limited

Gao F, Lu Y (2006) A kalman-filter based time-domain analysis for structural damage diagnosis with noisy signals. Journal of Sound and Vibration 297:916–930

Gelb A, Joseph F Kasper J, Raymond A Nash J, Price CF, Arthur A Sutherland J (2001) Applied optimal estimation. The. M.I.T. Press

Ghosh SJ, Roy D, Manohar C (2007) New forms of extended kalman filter via transversal linearization and applications to structural system identification. Comput Methods Appl Mech Engrg 196:5063–5083

Gorinevsky D, Cook J, 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

Guzzella L, Onder CH (2010) Introduction to Modeling and Control of Internal Combustion Engine Systems, 2nd edn. Springer

Hafner M, Schuler M, Nelles O (1999) Dynamical identification and control of combustion engine exhaust. In: Proceedings of the American Control Conference, San Diego, California, pp 222–226

Hansen CH, Snyder SD (1997) Active Control of Noise and Vibration. E & FN Spon, an imprint of Chapman & Hall

Hassan M, Dubay R, Li C, Wang R (2007) Active vibration control of a flexible one-link manipulator using a multivariable predictive controller. Mechatronics 17(1):311–323

Hendricks E, Luther JB (2001) Model and observer based control of internal combustion engines. In: Proc. MECA (Modeling, Emissions and Control in Automotive Engines), Salerno, Italy

Heywood JB (1988) Internal Combustion Engine Fundamentals. McGraw-Hill

Hoa CC, Ma CK (2007) Active vibration control of structural systems by a combination of the linear quadratic gaussian and input estimation approaches. Journal of Sound and Vibration 301:429–449

Inman DJ (2006) Vibration with control. Wiley & Sons

Isermann R (2005) Mechatronic systems: Innovative products with embedded control. In: Proceedings of the 16th IFAC World Congress, Prague, Czech republic, paper Code: Tu-M02-PL/1

Isermann R, Müller N (2003) Design of computer controlled combustion engines. Mechatronics 13(10):1067–1089

Johansen TA, Foss BA (1993) Constructing narmax models using armax models. International Journal of Control 58(5):1125–1153

Johansen TA, Hunt KJ, Gawthrop PJ, Fritz H (1998) Off-equilibrium linearisation and design of gain-scheduled control with application to vehicle speed control. Control Engineering Practice 6(2):167–180

Jones NP, Shi T, Ellis JH, Scanlan RH (1995) System-identification procedure for system and input parameters in ambient vibration surveys. Journal of Wind Engineering and Industrial Aerodynamics 54/55:91–99

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

Kouvaritakis B, Cannon M, Rossiter J (2002) Who needs QP for linear MPC anyway? Automatica 38:879–884

Kvasnica M (2009) Real - Time Model Predictive Control via Multi - Parametric Programming: Theory and Tools, 1st edn. VDM Verlag

Kvasnica M, Grieder P, Baotić M (2004) Multi-Parametric Toolbox (MPT). Online, available: http://control.ee.ethz.ch/

Ljung L (1999) System Identification: Theory for the User, 2nd edn. PTR Prentice Hall, Upper Saddle River, NJ.

L ofberg J (2004) YALMIP: A toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference, Taipei, Taiwan

López-Negrete R, Patwardhan SC, Biegler LT (2009) Approximation of arrival cost in moving horizon estimation using a constrained particle filter. In: Rita Maria de Brito Alves CAOdN, Evaristo Chalbaud Biscaia J (eds) 10th International Symposium on Process Systems Engineering: Part A, Computer Aided Chemical Engineering, vol 27, Elsevier, pp 1299 – 1304, DOI 10.1016/S1570-7946(09)70607-8

Maciejowski J (2002) Predictive Control with Constraints, 1st edn. Prentice Hall

Manzie C, Palaniswami M, Ralph D, Watson H, 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

Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: Stability and optimality. Automatica 36:789–814

Mollov S, Babuška R, Abonyi J, Verbruggen HB (2004) Effective optimization for fuzzy model predictive control. IEEE Trans on Fyzzy Systems 12(5):661–675

Moraal PE, Grizzle JW (1995) Observer design for nonlinear systems with discrete-time measurement. IEEE Transactions on automatic control 40(3):395–404

Murray-Smith R, Johansen TA (1997) Multiple Model Approaches to Modelling and Control. Taylor&Francis

Muske KR (2006) A model-based si engine air fuel ratio controller. Minneapolis, Minnesota, USA, pp 3284–3289

Namdeo V, Manohar C (2007) Nonlinear structural dynamical system identification using adaptive particle filters. Journal of Sound and Vibration 306:524–563

Nelles O (2001) Nonlinear System Identification. Springer

Niederberger D (2005) Hybrid Systems: Computation and Control, vol 3414, Publisher Springer / Heidelberg, Berlin, chap Design of Optimal Autonomous Switching Circuits to Suppress Mechanical Vibration, pp 511–525

Ohsumi A, Nakano N (2002) Identification of physical parameters of a flexible structure from noisy measurement data. IEEE Transactions on Instrumentation and Measurement 51(5):923–929

Pistikopoulos EN, Georgiadis MC, Dua V (eds) (2007) Multi-Parametric Model-Based Control, vol 2., 1st edn. Wiley-VCH Verlag GmbH & Co., Weinheim, Germany

Polóni T, Rohal'-Ilkiv B, Johansen TA (2007) Multiple arx model-based air-fuel ratio predictive control for si engines. In: 3rd IFAC workshop on advanced fuzzy and neural control, Valenciennes, France

Polóni T, Johansen TA, Rohal'-Ilkiv B (2008) Modeling of air-fuel ratio dynamics of a gasoline combustion engine with arx network. Transaction of the ASME (Journal of Dynamic Systems, Measurement, and Control) 130(061009)

Polóni T, Rohal'-Ilkiv B, Johansen TA (2010) Damped one-mode vibration model state and parameter estimation via moving horizon observer. In: IFAC Symposium on Mechatronic Systems, Boston, Massachusetts, USA

Powell JD, Fekete NP, Chang CF (1998) Observer-based air-fuel ratio control. Control Systems Magazine,IEEE 18(5):72–83

Preumont A (2002) Vibration Control of Active Structures, 2nd edn. Kluwer Academic Publishers

Pytlak R (1999) Numerical Methods for Optimal Control Problems with State Constraints. Springer-Verlag

Qu CC, Hahn J (2009) Computation of arrival cost for moving horizon estimation via unscented kalman filtering. Journal of Process Control 19(2):358 – 363, DOI 10.1016/j.jprocont.2008.04.005

Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. IEEE Transactions on Automatic Control 48(2):246–258

Rossiter JA (2003) Model-based predictive control: a practical approach, 1st edn. CRC Press LCC

Roubos JA, Mollov S, Babuška R, Verbruggen HB (1999) Fuzzy model-based predictive control using takagi-sugeno models. International Journal of Approximate Reasoning 22(1):3–30

Sajeeb R, Manohar C, Roy D (2009) A conditionally linearized monte carlo filter in non-linear structural dynamics. International Journal of Non-Linear Mechanics 44:776–790

Sloss J, Bruch J, Sadek I, Adali S (2003) Piezo patch sensor/actuator control of the vibrations of a cantilever under axial load. Composite Structures 62:423–428

Song G, Qiao PZ, Bibienda WK, Zhou GP (2002) Active vibration damping of composite beam using smart sensors and actuators. Journal of Aerospace Engineering 15(3):97–103

Sturm JF (1999) Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software - Special issue on Interior Point Methods 11-12:625–653

Takács G, 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, Oxford, United Kingdom

Takács G, 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 International Conference on Control and Applications, Cambridge, United Kingdom.

Takagi T, Sugeno M (1985) Fuzzy identification of systems and its application to modeling and control. IEEE Trans Systems, Man and Cybernetics 15:116–132

Tikhonov AN, Arsenin VY (1977) Solutions of Ill-posed Problems. Wiley

Uchino E, Ohta M (1986) A new methodological trial on state estimation of linear structure vibration model with noisy power observation mechanism of non-gaussian type. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP

Ungarala S (2009) Computing arrival cost parameters in moving horizon estimation using sampling based filters. Journal of Process Control 19(9):1576 – 1588, DOI 10.1016/j.jprocont.2009.08.002

Wills AG, Bates D, Fleming AJ, Ninness B, Moheimani SOR (2008) Model predictive control applied to constraint handling in active noise and vibration control. IEEE Transactions on Control Systems Technology 16(1):3–12