Minimisation of process response deterioration due to constraints

M. FIKAR*
Control Engineering Laboratory,
Faculty of Electrical Engineering,
Ruhr-University Bochum,
D44780 Universitätsstr. 150
Germany
Telefax: ++49/234 32 14101, E-mail: fikar@esr.ruhr-uni-bochum.de

Technical Report MF0002

http://www.esr.ruhr-uni-bochum.de/~fikar/research/gpc/mf0002.zip

10.07.2000

Abstract

This report investigates the problem of optimal handling of the constraints on signals. Their presence usually lowers performance of the closed-loop system and may even lead to instability. Assuming that the plant model and the controller are given, the task studied is: (i) to find a control sequence satisfying the constraints that minimises the degree of the process output degradation, (ii) to determine optimal transfer from nonlinear constrained mode back to linear controller, (iii) to guarantee stability of the overall scheme.

Keywords:
Constrained control, Predictive control

1 Introduction

Constraints are ubiquitous in real world control systems. Typical constraints are for example finite capacity of pumps, limited speed of motors. As these devices are used as actuators in process control, these constraints are called inputs constraints. The output constraints may include safety requirements as temperature limits in reactors, pressure and liquid level limits in distillation columns.

When the knowledge about the constraints is neglected degradation of performance and in some cases even instability may occur. To counteract this, several techniques of controller design with constraint handling capabilities emerged recently:

1. Anti-windup and bumpless transfer (AWBT) design (Kothare et al. 1994, and references therein): This is a two stage procedure. In the first stage a controller is designed ignoring the constraints. In the second stage an additional feedback loop is designed that acts against constraints so that the control signal is within constraints. The resulting controller remains linear.

*On leave from Department of Process Control, CHTF STU, Radlinského 9, SK-812 37 Bratislava, Slovakia

2. Reference governor (RG) (Bemporad 1998): The approach consists in adding a discrete-time device (RG) to nominal stable closed-loop which filters the desired reference trajectory. The result is that no constraint violation occurs and the compensated control system can operate in a stable way.

3. Model based predictive control (MBPC) (Clarke et al. 1987): This approach comprises prediction of the controlled plant into future based on assumptions about future control actions and minimisation of a given cost function. Input and/or output constraints can easily be added to the problem formulation and the resulting closed-loop control law is solved as an optimisation problem in each sampling step using the receding-horizon strategy. If the controlled plant is linear and no constraints are active, the controller is linear.

The approaches are sorted in order of increasing complexity and computational load. In all cases, it is assumed that the plant model is known. In the first two approaches it is additionally assumed that the controller is known whereas predictive control actually synthesises the controller and thus combines the two steps (nominal controller design + constraints handling) into one step. Moreover, predictive control exploits the plant model for the future predictions and therefore uses more information than the first two methods.

The design criteria imposed on the overall closed-loop system in the AWBT framework and also used here are as follows (Kothare et al. 1994):

1. closed-loop stability,

2. linear performance recovery - when no constraints are active, the nominal controller should act on the system,

3. graceful performance degradation in case of active constraints.

In this report a combined strategy to deal with constraints is proposed. It is assumed that the model of the plant and the controller are known and that the closed-loop without constraints is stable (as in AWBT and RG). The model of the plant is used actively for predictions and thus the whole problem is posed in MBPC framework. As a consequence, the third design criterion of Kothare et al. (1994) can be strengthened to: *optimal* performance degradation in case of active constraints.

1.1 Motivation

The motivation for this work comes from two directions. Both family of methods (AWBT, MBPC) have some advantages and drawbacks. In MBPC the procedure is equivalent to a controller design. However, in many cases a controller is already designed - for example by the use of robust control design for a plant unprecisely known.

AWBT techniques observe only a current state of a plant/controller. The unability to predict the plant behaviour into the future may cause inferiour performance of ABWT compared to MBPC techniques. This claim is illustrated by the means of a simple example.

Consider the controlled system (unstable) of the form

$$y(t) = -2.5y(t-1) + 1.5y(t-2) + 3u(t-1) - u(t-2) \quad (1)$$

where $y(t)$ it the output of the process and $u(t)$ is the manipulated variable. As a controller consider a state dead-beat controller with integral action

$$u(t) = 1.3u(t-1) + 0.3u(t-2) + 0.5w(t) + 0.4y(t) - 1.35y(t-1) + 0.45y(t-2) \quad (2)$$

where $w(t)$ is the setpoint. Simulations of the unconstrained and constrained system with $0 \le u(t) \le 3.15$ are given in Fig. 1 and show that the constrained system is unstable.

As it will be shown later, instability can be prevented in this case if the control action is at $t = 1$ reduced from 1.5 to approximately 0.8. However, at time $t = 1$ nothing shows that the closed-loop will be unstable from $t \ge 2$ because the control signal is within limits. Hence, AWBT controller cannot adjust its gain. The information about possible performance degradation can only be gained if the prediction at $t = 1$ is calculated for future output and input signals.

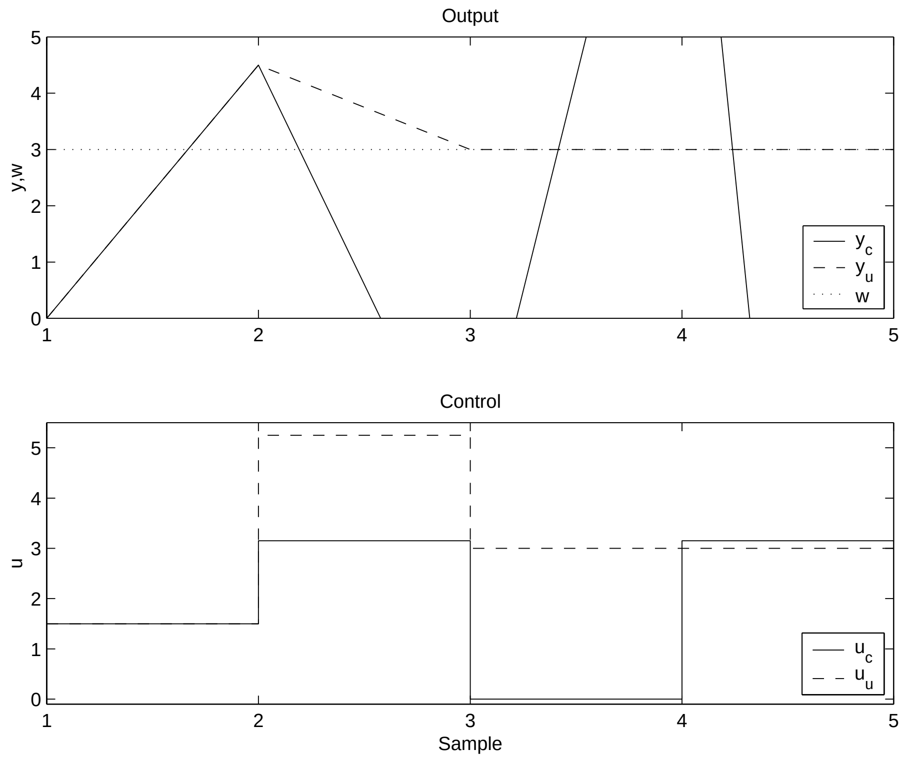

Figure 1: Motivating example, u_u, y_u - unconstrained simulation, u_c, y_c - constrained simulation with $0 \leq u(t) \leq 3.15$

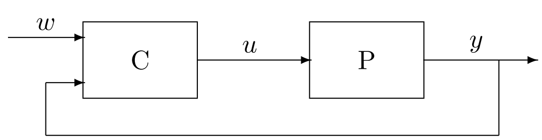

Figure 2: Closed-loop control system

2 The basic setup

Let us consider a time-invariant, single input single output plant expressed in discrete-time form

$$Ay = Bu, \tag{3}$$

where y, u are the process output and manipulated input sequences, respectively. A and B are polynomials in z^{-1} that describe the input-output properties of the plant.

We assume that a class of references w is generated via

$$Fw = G, \tag{4}$$

where $(F, G) = 1$. Here, F specifies desired class of references (steps, ramps, harmonic signals,...,) and G represents initial conditions of the concrete reference.

In order to track the class of references given above (and to reject disturbances of the same class), an integrator is symbolically added to the controlled system

$$\tilde{u} = Fu. \tag{5}$$

If we assume step changes in references, which is most often the case in predictive control, then $F = 1 - z^{-1}$, $G = 1$ and the signal $\tilde{u} = \Delta u$ is a sequence of control increments. However, other specifications for F can also be considered.

Hence, the plant with the integrator is described by the transfer function B/AF and $y, \tilde{u}$ are its output and input sequences, respectively. It is assumed that this plant is free of hidden modes, thus $(AF, B) = 1$.

As a controller, we consider a two-degree-of-freedom (2DoF) configuration described by the equation

$$P\tilde{u} = Rw - Qy, \tag{6}$$

where P, Q, R are controller polynomials that are coprime and $P(0)$ is nonzero. The 2DoF controller has been chosen due to its flexibility. However, any other controller structure could have been chosen.

2.1 Predictive control

The prediction of the plant behaviour into future will be needed. As the plant (3) is linear, its future output response will consist of two parts: forced response due to future control signals and free response resulting from the initial conditions. Choosing the number of predicted outputs into the future being equal N (prediction horizon) yields (Clarke et al. 1987)

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{u} + \boldsymbol{f}, \tag{7}$$

where

$$\boldsymbol{y} = [y_{t+1} \ \dots \ y_{t+N}]^T, \tag{8}$$

$$\boldsymbol{u} = [\tilde{u}_t \ \dots \ \tilde{u}_{t+N-1}]^T, \tag{9}$$

$$\boldsymbol{f} = [f_{t+1} \ \dots \ f_{t+N}]^T, \tag{10}$$

$$\boldsymbol{G} = \begin{pmatrix} g_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ g_{N-1} & \dots & g_1 \end{pmatrix}. \tag{11}$$

The matrix $\boldsymbol{G}$ and vector $\boldsymbol{f}$ can be calculated as usual from recursive Diophantine equations (Clarke et al. 1987) or by simulating the system recursively (Fikar 1998). The latter possibility is as follows.

When all future control increments are zero ($\boldsymbol{u} = \boldsymbol{0}$) then $\boldsymbol{y} = \boldsymbol{f}$. Hence, simulating the system (3) with given $y_{t-i}, i \geq 0$, $u_{t-i}, i > 0$, and assuming $u_{t+i} = u_{t-1}, i \geq 0$ gives $f_{t+i} = y_{t+i}, i > 0$.

To obtain the matrix of forced responses $\boldsymbol{G}$ consider the case when all initial conditions are zero, i.e. $\boldsymbol{f} = \boldsymbol{0}$. Assume further that the future control increments are the Kronecker delta function, i.e. $\tilde{u}_{t+j} = \delta_{kj}$ where

$$\delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases}. \tag{12}$$

Then, by simulating the system (3) with $y_{t-i} = 0, i \geq 0$, $u_{t-i} = 0, i > 0$, and assuming $\tilde{u}_{t+j} = \delta_{kj}$ gives the column k of $\boldsymbol{G}$ as $\boldsymbol{G}_{k,j} = y_{t+j}, k > 0$.

The cost function that is to be minimised has to take into consideration the three goals defined in the Section 1. Several definitions are possible. Here we will utilise the optimisation of the surface between constrained and unconstrained output trajectories

$$J = \sum_{i=1}^{N} (y^u(t+i) - y(t+i))^2 = (\boldsymbol{y}^u - \boldsymbol{y})^T (\boldsymbol{y}^u - \boldsymbol{y}) \tag{13}$$

where $\boldsymbol{y}^u = [y^u_{t+1} \ \dots \ y^u_{t+N}]^T$ is the trajectory of the unconstrained closed-loop plant output, i. e. future trajectory of the signal y based on equations (3), (5), (6).

3 Methods

In the unconstrained case minimisation of the cost function (13) leads to the optimum with $y^u(t+i) = y(t+i)$ and $J^* = 0$. This can easily be proved as the sequence y^u is calculated for the controlled plant and given controller and hence is admissible.

Consider now the constrained case. In general, the constraints on the signals can correspond to lower and upper hard constraints on the control signal, on the rate of change of the control signal, and to recommended lower and upper limits on the output signal. All these can be transformed into linear inequality constraints on the vector $\boldsymbol{u}$ and generally written as

$$\boldsymbol{A}\boldsymbol{u} \geq \boldsymbol{b} \tag{14}$$

where the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ are of appropriate dimensions.

3.1 GPC

Perhaps the most simple solution within the framework of predictive control is to assume GPC settings taking no penalty on the control increments $\lambda = 0$ and the control horizon $N_u = N$. Substituting (6) into (13) gives quadratic programming problem

$$\begin{aligned} \min_{\boldsymbol{u}} J &= -2\big(\boldsymbol{G}^T(\boldsymbol{y}^u - \boldsymbol{f})\big)^T\boldsymbol{u} + \boldsymbol{u}^T\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{u} \\ & \quad \text{subject to } \boldsymbol{A}\boldsymbol{u} \geq \boldsymbol{b} \end{aligned} \tag{15}$$

The difficulty with this method is lack of stability properties in the constrained case. However, even with this drawback, GPC is actively used in academia and in industry due to its easy implementation and tuning.

3.2 GPC with linear controller

To assure stability also in the constrained case, constraints on terminal states can be used (Fikar and Unbehauen 2000). This is usually accomplished by adding the constraints

$$y(t+N+i-m+1) = w(t+N+i), \quad i = 1, \dots, m \tag{16}$$

where m is the state dimension of the controlled system, in our case $m = \max(\deg(AF), \deg(B))$. This removes m degrees of freedom from the optimisation problem, hence the prediction horizon N must be greater than m.

However, such a terminal constraint does not comply with the linear controller specification that is to be used after the constrained part of the trajectory. Therefore, as a more appropriate strategy, we propose to optimise only the first N_u control increments and to constrain the last $N - N_u \geq m$ steps to be generated by the controller (6).

The sequence of the control increments to be determined can be divided accordingly into optimised and linear part

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_o \\ \boldsymbol{u}_l \end{pmatrix} \tag{17}$$

$$\boldsymbol{u}_o = [\tilde{u}_t \; \dots \; \tilde{u}_{t+N_u-1}]^T \tag{18}$$

$$\boldsymbol{u}_l = [\tilde{u}_{t+N_u} \; \dots \; \tilde{u}_{t+N-1}]^T \tag{19}$$

The linear part $\boldsymbol{u}_l$ can be determined from the controller equation (6). Careful inspection of terms in (6) shows that $\boldsymbol{u}_l$ is a linear combination of $\boldsymbol{y}, \boldsymbol{u}_o$, and $\boldsymbol{w} = [w_{t+N_u-\deg(R)} \; \dots \; w_{t+N-1}]^T$. Hence, it can be written in the matrix form as a sum of free and forced responses

$$\boldsymbol{u}_l = \boldsymbol{G}_{lw}\boldsymbol{w} + \boldsymbol{G}_{ly}\boldsymbol{y} + \boldsymbol{G}_{lo}\boldsymbol{u}_o + \boldsymbol{f}_{lu}, \tag{20}$$

The free and forced responses can be obtained with the procedure described in Section 2.1 applied to the controller equation (6).

Combining (20) with (7) and eliminating intermediate variables yields for $\boldsymbol{y}$

$$\begin{aligned}\boldsymbol{y} &= \boldsymbol{G}\boldsymbol{u} + \boldsymbol{f} \\ &= \boldsymbol{G}_1\boldsymbol{u}_o + \boldsymbol{G}_2\boldsymbol{u}_l + \boldsymbol{f} \\ &= \boldsymbol{G}_y\boldsymbol{u}_o + \boldsymbol{f}_y,\end{aligned} \tag{21}$$

where

$$\boldsymbol{G}_y = (\boldsymbol{I} - \boldsymbol{G}_2\boldsymbol{G}_{ly})^{-1}(\boldsymbol{G}_1 + \boldsymbol{G}_2\boldsymbol{G}_{lo}), \tag{22}$$

$$\boldsymbol{f}_y = (\boldsymbol{I} - \boldsymbol{G}_2\boldsymbol{G}_{ly})^{-1}(\boldsymbol{G}_2[\boldsymbol{G}_{lw}\boldsymbol{w} + \boldsymbol{f}_{lu}] + \boldsymbol{f}). \tag{23}$$

As $\boldsymbol{G}_2$ is zero on and above the main diagonal, the inverse matrix exists.

In the same manner for $\boldsymbol{u}_l$ yields

$$\begin{aligned}\boldsymbol{u}_l &= (\boldsymbol{G}_{lo} + \boldsymbol{G}_{ly}\boldsymbol{G}_y)\boldsymbol{u}_o + (\boldsymbol{G}_{ly}\boldsymbol{f}_y + \boldsymbol{f}_{lu}) \\ &= \boldsymbol{G}_u\boldsymbol{u}_o + \boldsymbol{f}_u.\end{aligned} \tag{24}$$

The constraint description (14) holds for both components $\boldsymbol{u}_o, \boldsymbol{u}_l$. Substituting for $\boldsymbol{u}_l$ from (24) gives

$$\begin{aligned}\boldsymbol{A}\boldsymbol{u} &\geq \boldsymbol{b} \\ (\boldsymbol{A}_1\ \boldsymbol{A}_2)\begin{pmatrix}\boldsymbol{u}_o \\ \boldsymbol{u}_l\end{pmatrix} &\geq \boldsymbol{b} \\ (\boldsymbol{A}_1 + \boldsymbol{A}_2\boldsymbol{G}_u)\boldsymbol{u}_o &\geq \boldsymbol{b} - \boldsymbol{A}_2\boldsymbol{f}_u\end{aligned} \tag{25}$$

The resulting quadratic programming problem will be obtained by substituting (21) into (13). Neglecting the constant term and using the inequality constraint (25) yields

$$\begin{aligned}\min_{\boldsymbol{u}_o} J \quad = \quad & -2\left(\boldsymbol{y}^u - \boldsymbol{f}_y\right)^T \boldsymbol{G}_y\boldsymbol{u}_o + \boldsymbol{u}_o^T\boldsymbol{G}_y^T\boldsymbol{G}_y\boldsymbol{u}_o \\ & \text{subject to } (\boldsymbol{A}_1 + \boldsymbol{A}_2\boldsymbol{G}_u)\boldsymbol{u}_o \geq \boldsymbol{b} - \boldsymbol{A}_2\boldsymbol{f}_u\end{aligned} \tag{26}$$

3.3 Implementation

Unconstrained output trajectory

There are two principal ways how to calculate the desired unconstrained output trajectory $\boldsymbol{y}^u$

1. receding horizon approach - $\boldsymbol{y}^u$ is calculated based on actual initial conditions, i.e. the whole unconstrained output trajectory is recalculated in each sample,

2. model reference approach - $\boldsymbol{y}^u$ is determined by simulating the unconstrained closed-loop system in parallel and only y^u_{t+N} is pushed to the trajectory stack in each sample.

Clearly, in the unconstrained case both approaches coincide. In the constrained case the receding horizon approach may calculate unrealistic trajectories, because during active constraints the linear controller will not be active. Moreover, model reference approach will better capture the original goal - to minimise the surface between constrained and unconstrained trajectories.

Output horizon

The choice of the output horizon N depends on how long the constraint handling is active. It must be chosen sufficiently large so that infeasibility problems do not occur with the stabilising GPC method. Here, stability constraint dictates its lower bound as

$$N \geq N_u + m, \quad m = \max(\deg(AF), \deg(B)) \tag{27}$$

The same rule can be used for the basic GPC method as otherwise instability could happen.

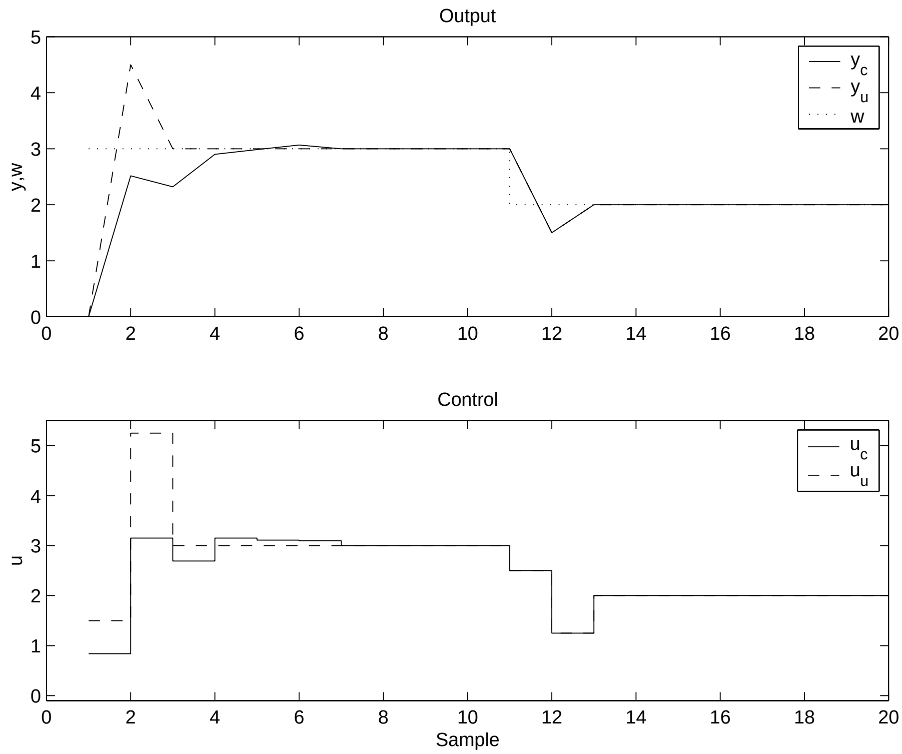

Figure 3: Proposed constrained controller for the motivating example

Control horizon

In general, the calculated constrained control trajectory consists of two parts - at first the constrained part followed by the unconstrained part that brings the controlled system to the desired setpoint value. It would be most natural if this division would correspond to the division of $\boldsymbol{u}_o$ and $\boldsymbol{u}_l$, given by the choice of the control horizon N_u. This makes N_u variable and results in a mixed integer quadratic programming problem that can be solved as follows:

Start with $N_u = 0$ and test whether the constraints are satisfied. If not, increase N_u until feasibility is attained with the upper bound given by (27).

Another possibility is to fix N_u at a constant value. The larger its value, the lower the value of the cost function can be obtained. However, after returning back from the constrained part of the control trajectory, the plant will still be for some steps under optimisation instead of the desired linear controller. As the optimising controller and linear controller design criteria differ (error surface minimisation versus nominal controller design) this may not be wanted. Theoretically, with an asymptotically stabilising controller, optimising phase will take infinitely long. Practically however, after some steps in the unconstrained regime, the control actions will coincide as the trajectory $\boldsymbol{y}^u$ will be matched reasonably well. We note that a compromise has to be found whether one should insist on the linear controller actions as soon as possible or the performance degradation due to constraints should be minimised. It may happen that a very early transfer to the linear controller may cause undesirable behaviour of the closed-loop system.

4 Discussion and simulation results

The role of the user parameters will be illustrated by the means of simulation examples with the system from the motivating example in the Section 1.1. We consider step changes in the references.

Fig. 3 shows clearly that stability is restored in the constrained case during the first setpoint

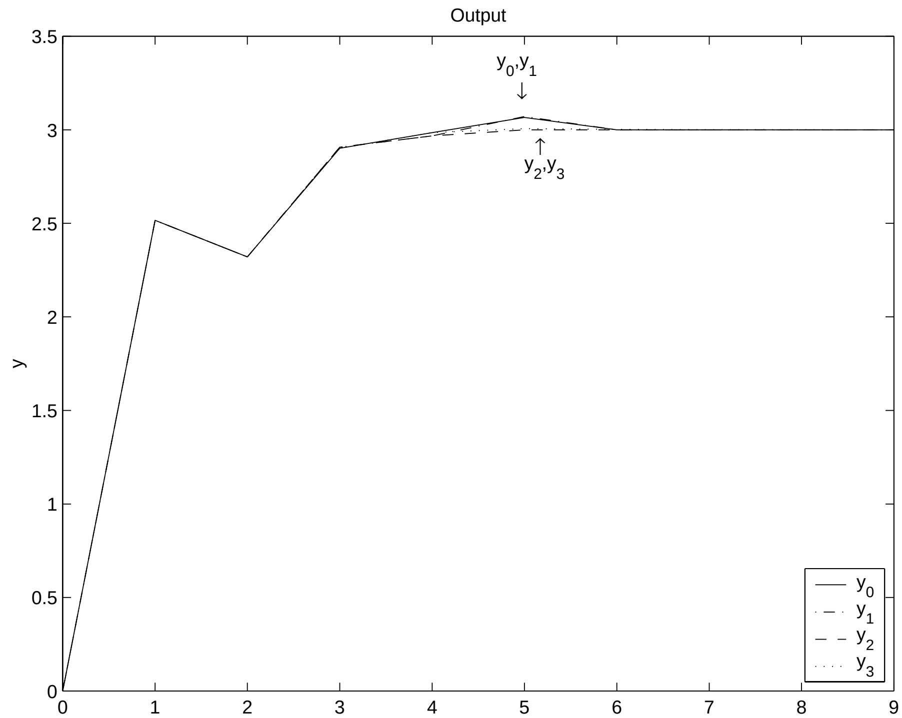

Figure 4: Comparison of different strategies for the choice of N_u. See explanation of the legend in the text.

change. During the second setpoint change actions of the nominal and constrained controller coincide as no constraint violation occurs. Stabilising GPC method was used. The parameters were $N = 10$ and the control horizon N_u was minimised. The nominal controller was the state dead-beat controller generated by equation (2). The desired output trajectory was generated with the model reference approach.

Minimised $N_u = 4$ was found at time $t = 1$ and corresponds to the number of sampling times after which the manipulated input is within constraints. After that, it decreased each sample and at $t = 5$ the nominal controller control law took over.

Four different strategies for the choice of N_u were investigated in Fig. 4 showing the output trajectory for the first setpoint change. Here, y_0, y_1 represent strategies that optimise only if the unconstrained problems violate the limits. The scenario with y_0 solves also minimisation of N_u (as in the previous simulation) whereas y_1 is calculated using fixed $N_u = 4$. Similarly, y_2, y_3 are strategies that optimise always. In the case of y_2, minimum N_u is searched with the constraint $N_u > 0$ whereas for y_3 fixed $N_u = 4$ is used. We note that all four strategies produced the same control actions for the second setpoint change that did not violate the constraints.

The comparison of trajectories and computation of the run-time cost function values showed that all trajectories are the same during active constraints and differ only in the way how the unconstrained output trajectory is approached. y_2, y_3 produce slightly lower value of the cost function and the controlled output tracks sooner the unconstrained trajectory. This was expected and explained in the previous section. It is on the user whether the smallest cost value is desired or the nominal controller performance is preferred.

The constrained part of the trajectory is dependent on the choice of the nominal controller. Consider for example the case when the closed-loop poles are set to $1 - 1/3z^{-1}$. These settings represent the so-called tracking error dead-beat when the closed-loop poles are placed on the locations of stable numerator zeros of the system. The result is the fastest possible output tracking (Fikar and Unbehauen 2000).

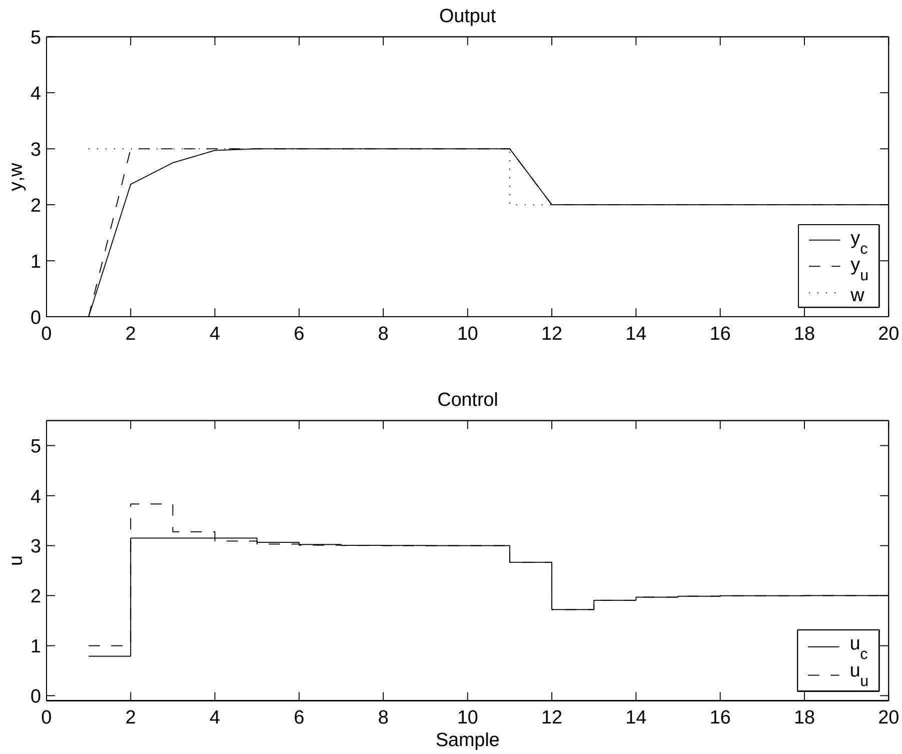

Figure 5: Choice of a different controller with $M = 1 - 1/3z^{-1}$

The results shown in Fig. 5 indicate that in the unconstrained case one step is needed to reach desired setpoint. The constrained trajectories differ significantly from those in Fig. 3. For example the control signal is on the upper level constraint during the first four samples whereas in the first simulation it moved up and down at $t = 3$.

Several experiments were performed to compare the basic and stabilising GPC strategies. It was observed that as long as the parameters for the basic GPC are "suitably" chosen (prediction horizon N sufficiently large), the results were practically identical. Of course, the differences would be bigger for "difficult" systems (as in (Bitmead et al. 1990, p. 102)).

5 Conclusions

This report has investigated a problem of optimal handling of the constraints on input/output signals. Predictive control framework was used as the basis for the proposed technique and three design criteria on the overall closed-loop were imposed: stability, nominal linear controller recovery in unconstrained case, and minimisation of the performance degradation in the constrained case.

It is shown that the design criteria can be respected by a very simple change of the well-known GPC method. The modification consists in preparation of the desired trajectory to be followed being the output of the unconstrained closed-loop system. Further, the cost function minimises only squared tracking error.

If a formal requirement of the constrained stability is desired, a modified method has been proposed. It consists of the combination of the GPC method followed by the linear controller. This not only assures stability, but also reduces the number of the optimised variables with a compromise of slightly more complex implementation.

If one wishes that the linear controller should take over as soon as possible after the constrained part of the trajectory, the problem is defined as mixed integer quadratic program that can be solved iteratively by increasing the number of the optimised variables and checking feasibility of the linear constraints. This can be done effectively with interior-point methods developed in the framework of linear programming.

If the computational load for such an algorithm is considered as too heavy, one can specify conservatively a fixed number of optimised variables that makes the problem feasible. An advantage of this approach is lower degree of performance degradation. On the other hand, the nominal controller actions will be generated only after the unconstrained and constrained output trajectories are sufficiently close.

For simplicity, only singlevariable case was considered here. However, generalisation of this idea to multivariable systems is straightforward.

Acknowledgments

The work has been supported by the Alexander von Humboldt Foundation that has enabled the first author to carry out his post-doctoral research in Bochum. This support is very gratefully acknowledged.

References

A. Bemporad. Reference governor for constrained nonlinear systems. 43(3):415–419, 1998. 2

R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control*. 1990. 9

D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control - part I. The basic algorithm. *Automatica*, 23(2):137 – 148, 1987. 2, 4

M. Fikar. Predictive control – an introduction. Technical report KAMF9801, Department of Process Control, FCT STU, Bratislava, Slovakia, 1998. 4

M. Fikar and H. Unbehauen. Pole placement based approach to stable linear predictive control. *Automatica*, 2000. (submitted). 5, 8

M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett. A unified framework for the study of anti-windup designs. *Automatica*, 30(12):1869–1883, 1994. 1, 2

A Symbolic derivation for the example

Forced and free responses for the controlled system

At first, the equation (3) is rewritten in so that the input is $\tilde{u}$ rather than u

$$FAy = B\tilde{u} \tag{28}$$

hence

$$y(t) = -1.5y(t-1) + 4y(t-2) - 1.5y(t-3) + 3\tilde{u}(t-1) - \tilde{u}(t-2) \tag{29}$$

Let us specify $N = 4$. For the predictions follows

$$\begin{aligned}
y_{t+1} &= -1.5y_t + 4y_{t-1} - 1.5y_{t-2} + 3\tilde{u}_t - \tilde{u}_{t-1} \\
y_{t+2} &= -1.5y_{t+1} + 4y_t - 1.5y_{t-1} + 3\tilde{u}_{t+1} - \tilde{u}_t \\
y_{t+3} &= -1.5y_{t+2} + 4y_{t+1} - 1.5y_t + 3\tilde{u}_{t+2} - \tilde{u}_{t+1} \\
y_{t+4} &= -1.5y_{t+3} + 4y_{t+2} - 1.5y_{t+1} + 3\tilde{u}_{t+3} - \tilde{u}_{t+2}
\end{aligned} \tag{30}$$

Now, substitute in the right hand sides of the predictions for y_{t+i}

$$\begin{aligned}
y_{t+1} &= 3\tilde{u}_t - \tilde{u}_{t-1} - 1.5y_t + 4y_{t-1} - 1.5y_{t-2} \\
y_{t+2} &= -5.5\tilde{u}_t + 3\tilde{u}_{t+1} + 1.5\tilde{u}_{t-1} + 6.25y_t - 7.5y_{t-1} + 2.25y_{t-2} \\
y_{t+3} &= 20.25\tilde{u}_t - 5.5\tilde{u}_{t+1} + 3\tilde{u}_{t+2} \\
&\quad -6.25\tilde{u}_{t-1} - 16.875y_t + 27.25y_{t-1} - 9.375y_{t-2} \\
y_{t+4} &= -56.875\tilde{u}_t + 20.25\tilde{u}_{t+1} - 5.5\tilde{u}_{t+2} + 3\tilde{u}_{t+3} \\
&\quad +16.875\tilde{u}_{t-1} + 52.5625y_t - 76.875y_{t-1} + 25.3125y_{t-2}
\end{aligned} \tag{31}$$

Therefore

$$\begin{aligned}
\boldsymbol{y} &= \begin{pmatrix} 3 & 0 & 0 & 0 \\ -5.5 & 3 & 0 & 0 \\ 20.25 & -5.5 & 3 & 0 \\ -56.875 & 20.25 & -5.5 & 3 \end{pmatrix} \boldsymbol{u} \\
&\quad + \begin{pmatrix} -\tilde{u}_{t-1} - 1.5y_t + 4y_{t-1} - 1.5y_{t-2} \\ 1.5\tilde{u}_{t-1} + 6.25y_t - 7.5y_{t-1} + 2.25y_{t-2} \\ -6.25\tilde{u}_{t-1} - 16.875y_t + 27.25y_{t-1} - 9.375y_{t-2} \\ 16.875\tilde{u}_{t-1} + 52.5625y_t - 76.875y_{t-1} + 25.3125y_{t-2} \end{pmatrix}
\end{aligned} \tag{32}$$

Symbolic derivation of the equations (20)–(24)

Consider $N_u = 1$. From (2) follows

$$\begin{aligned}
\tilde{u}_{t+1} &= 0.3\tilde{u}_t + 0.5w_{t+1} + 0.4y_{t+1} - 1.35y_t + 0.45y_{t-1} \\
\tilde{u}_{t+2} &= 0.3\tilde{u}_{t+1} + 0.5w_{t+2} + 0.4y_{t+2} - 1.35y_{t+1} + 0.45y_t \\
\tilde{u}_{t+3} &= 0.3\tilde{u}_{t+2} + 0.5w_{t+3} + 0.4y_{t+3} - 1.35y_{t+2} + 0.45y_{t+1}
\end{aligned} \tag{33}$$

Note that expression for $\tilde{u}_t$ was not given as it is assumed that it is optimised.

Now, substitute in the right hand sides of the predictions for $\tilde{u}_{t+i}$

$$\begin{aligned}
\tilde{u}_{t+1} &= 0.3\tilde{u}_t + 0.5w_{t+1} + 0.4y_{t+1} - 1.35y_t + 0.45y_{t-1} \\
\tilde{u}_{t+2} &= 0.09\tilde{u}_t + 0.15w_{t+1} + 0.5w_{t+2} \\
&\quad -1.23y_{t+1} + 0.4y_{t+2} + 0.045y_t + 0.135y_{t-1} \\
\tilde{u}_{t+3} &= 0.027\tilde{u}_t + 0.045w_{t+1} + 0.15w_{t+2} + 0.5w_{t+3} \\
&\quad +0.081y_{t+1} - 1.23y_{t+2} + 0.4y_{t+3} + 0.0135y_t + 0.0405y_{t-1}
\end{aligned} \tag{34}$$

Grouping the terms yields

$$\begin{aligned}
\boldsymbol{u}_l &= \begin{pmatrix} 0.3 \\ 0.09 \\ 0.027 \end{pmatrix} \tilde{u}_t + \begin{pmatrix} 0.5 & 0 & 0 \\ 0.15 & 0.5 & 0 \\ 0.045 & 0.15 & 0.5 \end{pmatrix} \begin{pmatrix} w_{t+1} \\ w_{t+2} \\ w_{t+3} \end{pmatrix} \\
&\quad + \begin{pmatrix} 0.4 & 0 & 0 & 0 \\ -1.23 & 0.4 & 0 & 0 \\ 0.081 & -1.23 & 0.4 & 0 \end{pmatrix} \begin{pmatrix} y_{t+1} \\ y_{t+2} \\ y_{t+3} \\ y_{t+4} \end{pmatrix} + \begin{pmatrix} -1.35y_t + 0.45y_{t-1} \\ 0.045y_t + 0.135y_{t-1} \\ 0.0135y_t + 0.0405y_{t-1} \end{pmatrix}
\end{aligned} \tag{35}$$

Introducing this equation into (21), (24) gives

$$\begin{aligned}
\begin{pmatrix} y_{t+1} \\ y_{t+2} \\ y_{t+3} \\ y_{t+4} \end{pmatrix} &= \begin{pmatrix} 3 \\ -1 \\ 0 \\ 0 \end{pmatrix} \tilde{u}_t + \begin{pmatrix} -\tilde{u}_{t-1} - 1.5yt + 4y_{t-1} - 1.5y_{t-2} \\ 0.3\tilde{u}_{t-1} + 1.5w_{t+1} + 0.4y_t - 1.35y_{t-1} + 0.45y_{t-2} \\ -0.5w_{t+1} + 1.5w_{t+2} \\ -0.5w_{t+2} + 1.5w_{t+3} \end{pmatrix} \\
\begin{pmatrix} \tilde{u}_{t+1} \\ \tilde{u}_{t+2} \\ \tilde{u}_{t+3} \end{pmatrix} &= \begin{pmatrix} 1.5 \\ -4.0 \\ 1.5 \end{pmatrix} \tilde{u}_t \\
&\quad + \begin{pmatrix} -0.4\tilde{u}_{t-1} + 0.5w_{t+1} - 1.95y_t + 2.05y_{t-1} - 0.6y_{t-2} \\ 1.35\tilde{u}_{t-1} + 0.75w_{t+1} + 0.5w_{t+2} + 2.05y_t - 5.325y_{t-1} + 2.025y_{t-2} \\ -0.45\tilde{u}_{t-1} - 2w_{t+1} + 0.75w_{t+2} + 0.5w_{t+3} - 0.6y_t + 2.025y_{t-1} - 0.675y_{t-2} \end{pmatrix}
\end{aligned}$$