USING CONTROLLER KNOWLEDGE IN PREDICTIVE CONTROL

M. Fikar [*] **H. Unbehauen** [**] **J. Mikleš** [*]

[*] *Department of Information Engineering and Process Control, FCHPT STU, Radlinského 9, SK-812 37 Bratislava, Slovakia, e-mail: fikar@cvt.stuba.sk, Fax: ++421/2 52.49.64.69*
[**] *Control Engineering Laboratory, Faculty of Electrical Engineering and Information Sciences, Ruhr-University Bochum, D44780 Universitätsstr. 150 Germany, Fax: ++49/234 7094-101*

Abstract: The paper discusses possible ways how to incorporate an existing controller into the predictive control framework. This idea is used in anti-windup framework, where the performance can degrade in some cases. The basic requirements are to meet the performance of the nominal controller in the unconstrained case and to guarantee stability in the constrained case.

Keywords: predictive control, stability, controller design

1. INTRODUCTION

Constraints are omnipresent in real world control systems. When the knowledge about the constraints is neglected, degradation of performance and in some cases even instability may occur. To counteract this, several techniques for constraints handling have emerged recently. The main approaches include anti-windup and bumpless transfer (AWBT) design (Kothare *et al.* 1994, and references therein), reference governor (RG) (Bemporad 1998), and model based predictive control (MBPC) (Clarke *et al.* 1987, Zelinka *et al.* 1999).

In AWBT and RG approaches it is assumed that the controller is known whereas MBPC actually synthesises the controller and thus combines the two steps (nominal controller design + constraints handling) into the one. In some cases this may be considered as a drawback, especially, if a nominal controller was designed to guarantee some special requirements on the closed-loop system not easily attainable with predictive control.

In this paper a combined strategy to deal with constraints is proposed. It is assumed that the plant and the controller descriptions 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 the MBPC framework Similar approaches have been published by (Rossiter and Kouvaritakis 1993, Scokaert *et al.* 1999, Chmielewski and Manousiouthakis 1996).

Two possible design methods are proposed. The first one manipulates the future setpoint sequence. In the second approach, knowledge about the desired closed loop poles is used. Both methods guarantee in the unconstrained case behaviour specified by the choice of the nominal controller. Moreover, in the constrained case stability of the closed-loop system is assured.

2. THE BASIC MATHEMATICAL SETUP

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

$$Ay = Bu, \tag{1}$$

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

$$Fw = G, \tag{2}$$

where F, G are coprime. Here, F specifies a desired class of references (steps, ramps, harmonic signals,...) and G represents the 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 additional term $1/F$ is added before the controlled system (Kučera 1979, Chmúrny *et al.* 1988)

$$u = \frac{1}{F}\tilde{u}. \tag{3}$$

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

Hence, the overall plant 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 are coprime.

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

$$P\tilde{u} = Rw - Qy, \tag{4}$$

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 *Alternate system description*

For the purposes of predictive control, the transfer function description of the controlled system will be transformed into vector-matrix notation. This is a standard procedure in predictive control. Considering the number of predicted outputs into the future being equal N (prediction horizon) this leads to the system description of the form

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{u} + \boldsymbol{f}, \tag{5}$$

where

$$\boldsymbol{y} = [y_{t+1} \ \dots \ y_{t+N}]^T, \tag{6}$$

$$\boldsymbol{u} = [\tilde{u}_t \ \dots \ \tilde{u}_{t+N-1}]^T, \tag{7}$$

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

$$\boldsymbol{G} = \begin{pmatrix} g_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ g_{N-1} & \dots & g_1 \end{pmatrix}. \tag{9}$$

The matrix $\boldsymbol{G}$ and vector $\boldsymbol{f}$ can be calculated as usual from recursive Diophantine equations or by simulating the system recursively (Fikar 1998).

Finally, we consider constraints on the signals that may 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 are generally written as

$$\boldsymbol{A}\boldsymbol{u} \geq \boldsymbol{b}, \tag{10}$$

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

3. APPROACH #1

The first predictive control approach consists in a suitable generation of the future setpoint trajectory $\boldsymbol{y}^u = [y^u_{t+1} \ \dots \ y^u_{t+N}]^T$ which is the trajectory of the unconstrained closed-loop plant output, i.e., future trajectory of the signal y based on equations (1)–(4). The cost function then minimises the surface between constrained and unconstrained output trajectories

$$J = \sum_{i=1}^{N} (y^u(t+i) - y(t+i))^2 = (\boldsymbol{y}^u - \boldsymbol{y})^T(\boldsymbol{y}^u - \boldsymbol{y}). \tag{11}$$

In the unconstrained case, minimisation of the cost function given by equation (11) 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 the given controller and hence is admissible.

Perhaps the simplest solution is to assume GPC settings taking no penalty on the control increments $\lambda = 0$ and the control horizon $N_u = N$. Substituting (5) into (11) leads to the quadratic programming problem

$$\begin{aligned} \min_{\boldsymbol{u}} J \ &= -2\big(\boldsymbol{G}^T(\boldsymbol{y}^u - \boldsymbol{f})\big)^T\boldsymbol{u} + \boldsymbol{u}^T\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{u} \\ &\text{subject to } \boldsymbol{A}\boldsymbol{u} \geq \boldsymbol{b}. \end{aligned} \tag{12}$$

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

To assure stability also in the constrained case, the last part of the optimised trajectory will be generated by the linear controller (4). The minimum number m of sampling times follows from the stability requirements and should be 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.

Therefore, we optimise only the first N_u control increments and constrain the last $N - N_u$ steps ($N - N_u \geq m$) to be generated by the controller (4).

The sequence of the control increments to be determined can be divided accordingly into two parts. The first part is optimised and the second one is the linear part, that is,

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_o \\ \boldsymbol{u}_l \end{pmatrix}, \tag{13}$$

$$\boldsymbol{u}_o = [\tilde{u}_t \;\; \dots \;\; \tilde{u}_{t+N_u-1}]^T, \tag{14}$$

$$\boldsymbol{u}_l = [\tilde{u}_{t+N_u} \;\; \dots \;\; \tilde{u}_{t+N-1}]^T. \tag{15}$$

The linear part $\boldsymbol{u}_l$ can be determined from the controller equation (4). Careful inspection of terms in (4) 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 matrix form as a sum of free and forced responses

$$\boldsymbol{u}_l = \boldsymbol{G}_{lw}\boldsymbol{w} + \boldsymbol{G}_{ly}\boldsymbol{y} + \boldsymbol{G}_{lo}\boldsymbol{u}_o + \boldsymbol{f}_{lu}. \tag{16}$$

This vector-matrix representation can be obtained in analogy to the system description (5) assuming the controller equation (4).

Combining (16) with (5) and eliminating intermediate variables yields

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{G}\boldsymbol{u} + \boldsymbol{f} = \boldsymbol{G}_1\boldsymbol{u}_o + \boldsymbol{G}_2\boldsymbol{u}_l + \boldsymbol{f} \\ &= \boldsymbol{G}_y\boldsymbol{u}_o + \boldsymbol{f}_y, \end{aligned} \tag{17}$$

where

$$\boldsymbol{G}_y = (\boldsymbol{I} - \boldsymbol{G}_2\boldsymbol{G}_{ly})^{-1}(\boldsymbol{G}_1 + \boldsymbol{G}_2\boldsymbol{G}_{lo}), \tag{18}$$

$$\boldsymbol{f}_y = (\boldsymbol{I} - \boldsymbol{G}_2\boldsymbol{G}_{ly})^{-1}(\boldsymbol{G}_2[\boldsymbol{G}_{lw}\boldsymbol{w} + \boldsymbol{f}_{lu}] + \boldsymbol{f}) \tag{19}$$

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

In the same manner for $\boldsymbol{u}_l$ we obtain,

$$\begin{aligned} \boldsymbol{u}_l &= (\boldsymbol{G}_{lo} + \boldsymbol{G}_{ly}\boldsymbol{G}_y)\boldsymbol{u}_o + (\boldsymbol{G}_{ly}\boldsymbol{f}_y + \boldsymbol{f}_{lu}) \\ &= \boldsymbol{G}_u\boldsymbol{u}_o + \boldsymbol{f}_u. \end{aligned} \tag{20}$$

The constraint description (10) holds for both components $\boldsymbol{u}_o, \boldsymbol{u}_l$. Substituting for $\boldsymbol{u}_l$ from (20) we get,

$$\begin{aligned} \boldsymbol{A}\boldsymbol{u} &\geq \boldsymbol{b} \\ (\boldsymbol{A}_1 \;\; \boldsymbol{A}_2)\begin{pmatrix} \boldsymbol{u}_o \\ \boldsymbol{u}_l \end{pmatrix} &\geq \boldsymbol{b} \\ (\boldsymbol{A}_1 + \boldsymbol{A}_2\boldsymbol{G}_u)\boldsymbol{u}_o &\geq \boldsymbol{b} - \boldsymbol{A}_2\boldsymbol{f}_u. \end{aligned} \tag{21}$$

The resulting quadratic programming problem will be obtained by substituting (17) into (11).

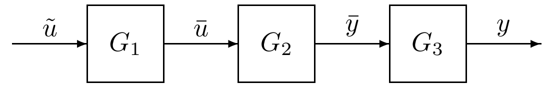

Fig. 1. System decomposition

Neglecting the constant term and using the inequality constraint (21) leads to the minimisation problem

$$\begin{aligned} \min_{\boldsymbol{u}_o} J \;=\; & -2\,(\boldsymbol{y}^u - \boldsymbol{f}_y)^T\, \boldsymbol{G}_y\boldsymbol{u}_o + \boldsymbol{u}_o^T\boldsymbol{G}_y^T\boldsymbol{G}_y\boldsymbol{u}_o \\ & \text{subject to } (\boldsymbol{A}_1 + \boldsymbol{A}_2\boldsymbol{G}_u)\boldsymbol{u}_o \geq \boldsymbol{b} - \boldsymbol{A}_2\boldsymbol{f}_u. \end{aligned} \tag{22}$$

4. APPROACH #2

The second approach uses the fact that any predictive controller can be described as the 2DoF controller given by (4) that generates the closed-loop system characterised by the closed-loop pole polynomial M.

Let us decompose the plant B/AF into three subsystems G_1, G_2, G_3 in a series (see Fig. 1) where

$$G_1 = \frac{M}{A^+}, \quad G_2 = \frac{B^-}{FA^-}, \quad G_3 = \frac{B^+}{M}, \tag{23}$$

where the superscript $^+$ denotes the stable and the superscript $^-$ the anti-stable part of a polynomial. We assume that M is a stable polynomial. From the decomposition it follows that the sequences $\tilde{u}, y$ are generated from the internal signals $\bar{u}, \bar{y}$ filtered by the stable transfer functions

$$\tilde{u} = \frac{A^+}{M}\bar{u}, \quad y = \frac{B^+}{M}\bar{y}. \tag{24}$$

We will now apply the method CRHPC (Constrained Receding Horizon Predictive Control) (Clarke and Scattolini 1991) to the transfer function G_2 and the signals $\bar{u}, \bar{y}$.

To this end, let us define the number of degrees of freedom $n \geq 0$, the output horizon N, the control horizon N_u, and the system state dimension m as

$$N_u = \deg(FA^-) + n, \tag{25}$$

$$N = \deg(B^-) + n, \tag{26}$$

$$m = \max(\deg(FA^-), \deg(B^-)), \tag{27}$$

and the vectors

$$\bar{\boldsymbol{u}}^T = (\bar{u}_t \;\; \dots \;\; \bar{u}_{t+N_u-1}), \tag{28}$$

$$\bar{\boldsymbol{y}}^T = (\bar{y}_{t+1} \;\; \dots \;\; \bar{y}_{t+N-1}), \tag{29}$$

$$\bar{\boldsymbol{y}}_1^T = (\bar{y}_{t+N} \;\; \dots \;\; \bar{y}_{t+N+m-1}), \tag{30}$$

$$\bar{\boldsymbol{w}}^T = (w_{t+1} \;\; \dots \;\; w_{t+N-1})M(1)/B^+(1), \tag{31}$$

$$\bar{\boldsymbol{w}}_1^T = (\bar{w}_{t+N} \;\; \dots \;\; \bar{w}_{t+N+m-1})M(1)/B^+(1) \tag{32}$$

The polynomial formulation of the system G_2 is rewritten in the vector-matrix form similar as in (5)

$$\begin{pmatrix} \bar{\boldsymbol{y}} \\ \bar{\boldsymbol{y}}_1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{G} \\ \boldsymbol{G}_1 \end{pmatrix} \bar{\boldsymbol{u}} + \begin{pmatrix} \bar{\boldsymbol{f}} \\ \bar{\boldsymbol{f}}_1 \end{pmatrix} \tag{33}$$

Finally, let us introduce the following cost function with the weighting matrices $\boldsymbol{W}_e > 0$ and/or $\boldsymbol{W}_u > 0$

$$J(\bar{\boldsymbol{u}}) = \bar{\boldsymbol{e}}^T \boldsymbol{W}_e \bar{\boldsymbol{e}} + \bar{\boldsymbol{u}}^T \boldsymbol{W}_u \bar{\boldsymbol{u}}, \tag{34}$$

where $\bar{\boldsymbol{e}} = \bar{\boldsymbol{w}} - \bar{\boldsymbol{y}}$.

The stable predictive controller is then defined as the following optimisation problem: Minimise (34) subject to the inequality constraints (10), the equality constraints

$$\bar{\boldsymbol{y}}_1 = \bar{\boldsymbol{w}}_1, \tag{35}$$

$$\bar{u}_{t+N_u+j} = 0, \quad j = 0, \dots, N - N_u + m, \tag{36}$$

and the system equality constraint (33). The actual control increment $\tilde{u}(t)$ is calculated from (24).

It can easily be shown that the predictive controller without degrees of freedom ($n = 0$) is equivalent to a 2DoF controller with given closed-loop poles M. For the case of $n > 0$, the task is to construct a controller (and a cost function) such that in the unconstrained case the control actions are those of the nominal controller with $n = 0$.

This can be obtained by minimising the cost function

$$J = \sum_{i=1}^{\infty} \tilde{e}^2(t+i), \tag{37}$$

where

$$\tilde{e} = \frac{B^+}{M_1} \bar{e} \tag{38}$$

and the stable polynomial M_1 is given by the spectral factorisation equation (Kučera 1979)

$$B^* B = M_1^* M_1. \tag{39}$$

Although the infinite horizon cost function (37) is minimised, the optimisation problem has only a finite number of variables. For further details, see (Fikar and Unbehauen 1999).

5. SIMULATION RESULTS

In this section some of the properties of the proposed algorithms are studied by means of simulations. Let us consider the class of step-change references and the following unstable discrete-time system

$$G = \frac{z^{-2}}{(1 + 3z^{-1})^2}, \tag{40}$$

and control constraints $\Delta u \leq 5$, $u \geq -1$. For the controller consider 2DoF dead-beat controller with integral action of the form

$$u_t = 6u_{t-1} - 5u_{t-2} + w_t - 22y_t - 24y_{t-1} + 45y_{t-2} \tag{41}$$

The first simulation given in Fig. 2 shows the performance of this nominal controller in the unconstrained case (subscripts u) and in the constrained case (subscripts c). It is interesting to notice in both cases the same control action at the time $t = 1$ because it is within the constraints. However, this has the consequence that the system states are moved into an unstabilisable region with respect to the constraints.

Thus, the example shows a case where even the best anti-windup strategy would be unsuccessful because it lacks information about the future behaviour of the system.

The second simulation compares the proposed approaches and the results are shown in Fig. 3. One can observe that in both cases the closed loop is stabilised in the first part of the trajectory. During the second step change the control actions are within constraints and both predictive controllers generate the same actions as those of the nominal unconstrained controller.

The small differences in the first part are caused by different design aims while dealing with the constraints. Recall that the first controller tries to minimise the difference between unconstrained and constrained output trajectories whereas the second one gives more importance to the location of the desired closed-loop poles.

6. CONCLUSIONS

The article has discussed two different ways how to implement a given controller within the framework of predictive control. Of course, identical performance can only be achieved in the unconstrained case.

The advantage of the proposed methods has been shown by the simulation examples where for a given controller, conventional anti-windup methods cannot stabilise the constrained closed-loop system due to the lack of their predictive properties. More precisely, to guarantee stability for certain systems, a modification of the control trajectory has to be performed some steps before the constraints are reached.

Two possible methods how to incorporate the controller into the predictive control have been presented. In the first approach, the future setpoint sequence has been manipulated and the cost function mininising the difference between constrained and unconstrained output trajectories has been chosen. In the second approach, terminal equality constraints leading to the desired closed-loop pole locations have been specified.

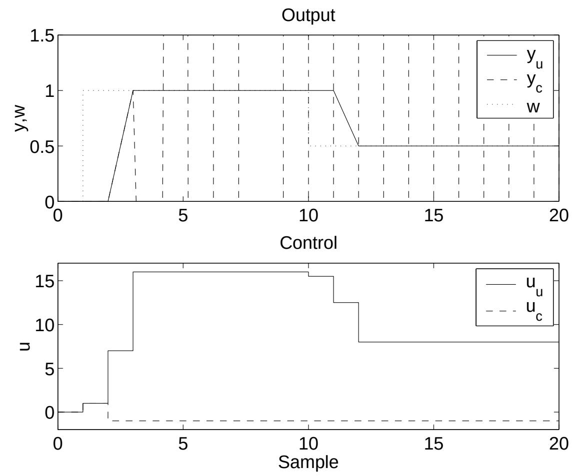

Fig. 2. Nominal controller performance

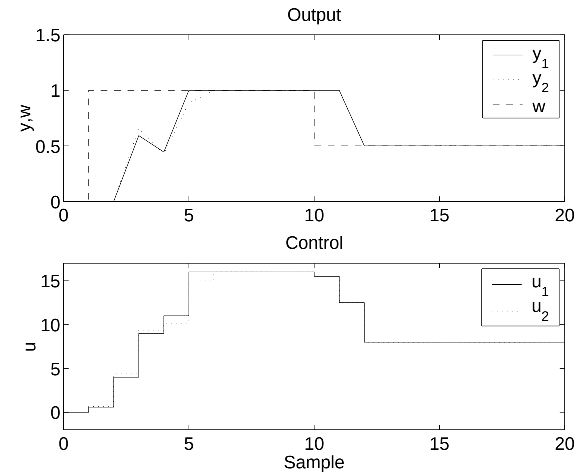

Fig. 3. Performance of the proposed controllers

In both approaches, stability of the constrained closed-loop system is guaranteed as well as a bumpless transfer between constrained and unconstrained modes. The behaviour of the controllers in the constrained mode is not the same due to their different design formulations.

Acknowledgements

The work has been supported by the Alexander von Humboldt Foundation and by the grant agency VEGA MŠ SR (1/135/03, 1/8108/01). This support is very gratefully acknowledged.

7. REFERENCES

Bemporad, A. (1998). Reference governor for constrained nonlinear systems. *IEEE Trans. Automatic Control* **43**(3), 415–419.

Chmielewski, D. and V. Manousiouthakis (1996). On constrained infinite-time linear quadratic optimal
control. *Systems Control Lett.* **29**(3), 121 – 129.

Chmúrny, D., R. Prokop and M. Bakošová (1988). *Automatic Control of Technological Processes*. Alfa. Bratislava. (in Slovak).

Clarke, D. W. and R. Scattolini (1991). Constrained receding-horizon predictive control. *IEE Proc. D* **138**(4), 347 – 354.

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

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

Fikar, M. and H. Unbehauen (1999). Some new results in linear predictive control. Technical report MF9902. Control Engineering Laboratory, Faculty of Electrical Engineering, Ruhr-University Bochum, Germany.

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

Kučera, V. (1979). *Discrete Linear Control: The Polynomial Equation Approach*. Wiley. Chichester.

Rossiter, J. A. and B. Kouvaritakis (1993). Constrained stable generalised predictive control. *IEE Proc. D* **140**(4), 243 – 254.

Scokaert, P. O. M., D. Q. Mayne and J. B. Rawlings (1999). Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Automatic Control* **44**(3), 648–654.

Zelinka, P., B. Rohal̆-Ilkiv and A.G. Kuznetsov (1999). Experimental verification of stabilising predictive control. *Control Engineering Practice* **7**(5), 601–610.