Int. J. Appl. Math. Comput. Sci., 2004, Vol. 14, No. 2, 167–180

AN INFINITE HORIZON PREDICTIVE CONTROL ALGORITHM BASED ON MULTIVARIABLE INPUT-OUTPUT MODELS

MACIEJ ŁAWRYŃCZUK*, PIOTR TATJEWSKI*

* Warsaw University of Technology
Institute of Control and Computation Engineering
ul. Nowowiejska 15/19, 00–665 Warszawa, Poland
e-mail: {lawrynczuk, tatjewski}@ia.pw.edu.pl

In this paper an infinite horizon predictive control algorithm, for which closed loop stability is guaranteed, is developed in the framework of multivariable linear input-output models. The original infinite dimensional optimisation problem is transformed into a finite dimensional one with a penalty term. In the unconstrained case the stabilising control law, using a numerically reliable SVD decomposition, is derived as an analytical formula, calculated off-line. Considering constraints needs solving on-line a quadratic programming problem. Additionally, it is shown how free and forced responses can be calculated without the necessity of solving a matrix Diophantine equation.

Keywords: model predictive control, stability, infinite horizon, singular-value decomposition, quadratic programming

1. Introduction

Model predictive control refers to a control strategy in which a model of the process is used to predict its future behaviour. At each time instant an open-loop optimal control problem, with the current state of the plant as the initial state, is solved. Only the first calculated control move is then applied to the plant, the measurements are updated and the whole procedure is repeated (Camacho and Bordons, 1999; Maciejowski, 2002; Tatjewski, 2002). Such a method makes it possible to control effectively multivariable plants, both linear and nonlinear (Henson, 1998; Morari and Lee, 1999). It is recognised as the only advanced control technique which has made a substantial impact on industrial control problems, largely due to its unique ability to handle hard constraints (Mayne, 2001).

Despite significant advances in nonlinear predictive control (Henson, 1998), algorithms based on linear models, mainly DMC (Dynamic Matrix Control) (Cutler and Ramaker, 1980) and GPC (Generalized Predictive Control) (Clarke *et al.*, 1987a; 1987b), are usually applied to on-line control. In many cases linear models are precise enough, and their identification and validation is significantly simpler than in the case of the nonlinear ones. First and foremost, linear models imply computational simplicity. In the unconstrained case the analytical control law can be calculated beforehand. Imposing constraints results in a convex quadratic programming problem, for which fast and reliable methods are available (Maciejowski, 2002). Both aforementioned algorithms (i.e. DMC and GPC) use input-output models, which are more intuitive and popular than the state-space representation, especially among practitioners.

In the 1980's, when the linear predictive control algorithm gained widespread acceptance in industry (Morari and Lee, 1999), stability was achieved by tuning horizons' lengths and penalty factors. This approach was criticised as "playing games" (Bitmead *et al.*, 1990). In the meantime, however, some stability criteria were obtained. As far as the unconstrained cases are concerned, they were published in (Clarke and Mohtadi, 1989; Clarke *et al.*, 1987a; 1987b; Kwon and Byun, 1989; Rouhani and Mehra, 1982; Scattolini and Bittanti, 1990), while constrained cases were discussed by (Rouhani and Mehra, 1982; Gutman and Hagander, 1985; Sznaider and Damborg, 1990; Zafiriou and Marchal, 1991; Zafiriou, 1990). Unfortunately, their applicability was limited.

Two stabilising modifications to the standard finite horizon problem formulation deserve serious consideration: a terminal constraint and an infinite horizon (e.g. Mayne *et al.*, 2000; Maciejowski, 2002). The first approach, developed in the GPC framework, resulted in the Constrained Receding-Horizon Predictive Control (CRHPC) algorithm (Clarke and Scattolini, 1991; Scokaert and Clarke, 1994) and the Stable Input-Output Receding Horizon Control (SIORHC) algorithm (Chisci and Mosca, 1994). However, imposing additional equality constraints to force the predicted output signal to take exactly a desired set-point value at the end of the prediction horizon, especially when the horizon is short, is

likely to result in a rather aggressive control action. Moreover, undesirable side-effects on the performance and a possible risk of infeasibility are also reported (Scokaert, 1997). The appealing infinite horizon approach was investigated in (Muske and Rawlings, 1993; Rawlings and Muske, 1993), who showed how to reformulate the infinite dimensional optimisation problem as a finite dimensional one in the context of linear state-space models. Stabilising properties were also established in the constrained case (Scokaert and Clarke, 1994). More recently, the same idea was incorporated in the framework of the GPC algorithm and was called GPC$^{\infty}$ (Scokaert, 1997). Yet another interesting algorithm is the Linear Quadratic Gaussian Predictive Control (LQGPC), in which the optimal predictive law is derived using an LQG approach and the infinite horizon is also possible (Ordys *et al.*, 2000).

The purpose of this paper is to develop an infinite horizon algorithm based on multivariable input-output models for stable plants, taking advantage of the work done in (Muske and Rawlings, 1993; Rawlings and Muske, 1993). In the unconstrained case, the control law is derived as an analytical formula, calculated off-line by means of a computationally reliable SVD decomposition. In the constrained case, a quadratic programming approach is used on-line. Additionally, it is shown how free and forced responses can be calculated without solving a matrix Diophantine equation, which is usually employed in the context of GPC algorithms.

The outline of the paper is as follows. In Section 2 the derivation of the algorithm is presented. The infinite horizon performance index is reformulated in terms of a finite number of parameters, two possible approaches to solving the unconstrained optimisation problem are discussed and a closed-form control law is derived. The constrained optimisation problem is then formulated assuming input constraints. Simple methods which make it possible to calculate forced and free responses without solving a matrix Diophantine equation are also described. Simulation results of a chemical reactor are presented in Section 3. In particular, the algorithm is compared with the CRHPC approach. The paper is summarised in Section 4.

2. Infinite Horizon Predictive Control Algorithm

2.1. Process Model

The stable process under consideration has N input and M output variables and is described by the following discrete-time equation:

$$\mathbf{A}(z^{-1})\mathbf{y}(k) = \mathbf{B}(z^{-1})\mathbf{u}(k), \tag{1}$$

where

$$\mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_N(k) \end{bmatrix}, \quad \mathbf{y}(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_M(k) \end{bmatrix} \tag{2}$$

and

$$\mathbf{A}(z^{-1}) = \begin{bmatrix} 1 + a_1^1 z^{-1} + \ldots + a_{na}^1 z^{-na} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 + a_1^M z^{-1} + \ldots + a_{na}^M z^{-na} \end{bmatrix},$$

$$\mathbf{B}(z^{-1}) = \begin{bmatrix} b_1^{1,1} z^{-1} + \ldots + b_{nb}^{1,1} z^{-nb} & \ldots & b_1^{1,N} z^{-1} + \ldots + b_{nb}^{1,N} z^{-nb} \\ \vdots & \ddots & \vdots \\ b_1^{M,1} z^{-1} + \ldots + b_{nb}^{M,1} z^{-nb} & \ldots & b_1^{M,N} z^{-1} + \ldots + b_{nb}^{M,N} z^{-nb} \end{bmatrix}. \tag{3}$$

In this paper a state-space representation of the model (1) is also used:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k). \end{aligned} \tag{4}$$

Although several equivalent state-space formulations can be obtained (Kailath, 1980), a straightforward method is to define the state vector of length $N(nb-1)+Mna$, the elements of which are past input and output values (e.g. Maciejowski, 2002):

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_u(k) \\ \mathbf{x}_y(k) \end{bmatrix}, \tag{5}$$

where

$$\mathbf{x}_u(k) = \begin{bmatrix} u_1(k-nb+1) \\ \vdots \\ u_N(k-nb+1) \\ \vdots \\ u_1(k-1) \\ \vdots \\ u_N(k-1) \end{bmatrix},$$

$$\mathbf{x}_y(k) = \begin{bmatrix} y_1(k-na+1) \\ \vdots \\ y_M(k-na+1) \\ \vdots \\ y_1(k) \\ \vdots \\ y_M(k) \end{bmatrix}. \tag{6}$$

The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are of dimensions $(N(nb-1)+Mna)\times(N(nb-1)+Mna)$, $(N(nb-1)+Mna)\times N$, $M\times(N(nb-1)+Mna)$, respectively,

$$\mathbf{A} = \left[\begin{array}{c|c} A_{1,1} & \mathbf{0}_{N(nb-1)\times Mna} \\ \hline \mathbf{0}_{M(na-1)\times N(nb-1)} & A_{2,2} \\ \mathbf{b}_{nb} \quad \dots \quad \mathbf{b}_2 & \end{array}\right], \tag{7}$$

where

$$A_{1,1} = \begin{bmatrix} \mathbf{0}_{N(nb-2)\times N} & \mathbf{I}_{N(nb-2)\times N(nb-2)} \\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N(nb-2)} \end{bmatrix},$$

$$A_{2,2} = \begin{bmatrix} \mathbf{0}_{M(na-1)\times M} & \mathbf{I}_{M(na-1)\times M(na-1)} \\ \mathbf{0}_{M\times M} & -\mathbf{a}_{na} \quad \dots \quad -\mathbf{a}_1 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{N(nb-2)\times N} \\ \mathbf{I}_{N\times N} \\ \mathbf{0}_{M(na-1)\times N} \\ \mathbf{b}_1 \end{bmatrix}, \tag{8}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_{M\times(N(nb-1)+M(na-1))} & \mathbf{I}_{M\times M} \end{bmatrix},$$

and the submatrices $\mathbf{a}_i$ and $\mathbf{b}_i$ are of characteristic structures

$$\mathbf{a}_i = \begin{bmatrix} a_i^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & a_i^M \end{bmatrix}, \qquad \mathbf{b}_i = \begin{bmatrix} b_i^{1,1} & \dots & b_i^{1,N} \\ \vdots & \ddots & \vdots \\ b_i^{M,1} & \dots & b_i^{M,N} \end{bmatrix}. \tag{9}$$

2.2. Performance Index Reformulation

The performance index, whose minimisation yields an optimal input profile, is defined over the infinite horizon as follows:

$$J(k) = \sum_{p=1}^{\infty}\left[\sum_{m=1}^{M}\mu_m\big(\hat{y}_m(k+p|k)\big)^2 + \sum_{n=1}^{N}\lambda_n\big(\Delta u_n(k+p-1|k)\big)^2\right], \tag{10}$$

where $\hat{y}_m(k+p|k)$ is the predicted output trajectory, $\Delta u_n(k+p-1|k)$ stands for the future control moves (decision variables), $\mu_m > 0$ and $\lambda_n \geq 0$ denote weighting coefficients. The set point values, temporarily only, are assumed to be zero, which implies a "regulator" problem formulation.

At time instant k an optimal trajectory over the whole infinite horizon is determined. Because at the time step $k+1$ no new information is available, the obtained solution coincides with the "tail" of the previous one (Bellman's principle of optimality). Assuming the disturbance-free case and a perfect model, the decreasing property of the performance index $J(k)$, used as a Lyapunow function, implies stability (e.g. Maciejowski, 2002; Tatjewski, 2002). To obtain a computationally solvable optimisation problem one has to express the performance index (10) in terms of a finite number of parameters. In order to guarantee stability, it is necessary to find a method which makes it possible to calculate the prediction to the infinity.

The control horizon of length H_c is introduced, and hence

$$\Delta u_n(k+p|k) = 0 \quad \text{for} \quad p \geq H_c. \tag{11}$$

The performance index (10) can be then rewritten as

$$J(k) = \sum_{p=1}^{\infty}\sum_{m=1}^{M}\mu_m\big(\hat{y}_m(k+p|k)\big)^2 + \sum_{p=0}^{H_c-1}\sum_{n=1}^{N}\lambda_n\big(\Delta u_n(k+p|k)\big)^2. \tag{12}$$

Defining vectors of lengths M and N, respectively,

$$\hat{\mathbf{y}}(k+p|k) = \begin{bmatrix} \hat{y}_1(k+p|k) \\ \vdots \\ \hat{y}_M(k+p|k) \end{bmatrix}, \qquad \Delta\mathbf{u}(k+p|k) = \begin{bmatrix} \Delta u_1(k+p|k) \\ \vdots \\ \Delta u_N(k+p|k) \end{bmatrix}, \tag{13}$$

the performance index (12) is given by

$$J(k) = \sum_{p=1}^{\infty} \|\hat{\mathbf{y}}(k+p|k)\|^2_{\mathbf{M}_0} + \sum_{p=0}^{H_c-1} \|\Delta\mathbf{u}(k+p|k)\|^2_{\mathbf{\Lambda}_0}, \tag{14}$$

where the diagonal matrices $\mathbf{M}_0$ and $\mathbf{\Lambda}_0$ of dimensions M and N are composed of coefficients μ_m and λ_n, respectively.

As far as stable plants are concerned, the receding horizon predictive control algorithm with performance index (14) and $H_c \geq 1$ is stable, as was proved in (Muske and Rawlings, 1993; Rawlings and Muske, 1993). It also holds true in the case of constraints imposed on inputs, because the parameters of the model, current state of the plant and horizon length H_c do not affect the feasibility of the optimisation problem. The performance index employed in (Muske and Rawlings, 1993) penalises predicted output values with a weighting matrix $\mathbf{M}_0 \geq \mathbf{0}$, control increments with a weighting matrix $\mathbf{\Lambda}_0 \geq \mathbf{0}$ and, additionally, control levels with a positive-definite weighting matrix. In the case of predictive control algorithms based on input-output models, for example DMC or GPC, control levels are not taken into account. The stability results from (Muske and Rawlings, 1993) are still applicable, provided that the matrix $\mathbf{M}_0$ is positive definite and $\mathbf{y}(k) \to \mathbf{0}$ implies $\mathbf{x}(k) \to \mathbf{0}$, as was discussed in (Maciejowski, 2002; Tatjewski, 2002). Thus, it is assumed that $\mu_m > 0$ and $\lambda_n \geq 0$.

In order to express the infinite sum in the performance index (14) in terms of a finite number of parameters, it is split into two parts

$$\sum_{p=1}^{\infty} \|\hat{\mathbf{y}}(k+p|k)\|^2_{\mathbf{M}_0} = \sum_{p=1}^{H_c-1} \|\hat{\mathbf{y}}(k+p|k)\|^2_{\mathbf{M}_0} + \sum_{p=H_c}^{\infty} \|\hat{\mathbf{y}}(k+p|k)\|^2_{\mathbf{M}_0}. \tag{15}$$

In the case of a "regulator" problem, taking into account (11), the input profile has to satisfy the condition

$$u_n(k+p|k) = 0 \quad \text{for} \quad p \geq H_c, \tag{16}$$

otherwise the cost function (14) would be unbounded (e.g. Maciejowski, 2002). Hence, using the state space model (4), we obtain

$$\begin{aligned} \hat{\mathbf{y}}(k+H_c|k) &= \mathbf{C}\mathbf{x}(k+H_c|k), \\ \hat{\mathbf{y}}(k+H_c+1|k) &= \mathbf{C}\mathbf{A}\mathbf{x}(k+H_c|k), \\ \hat{\mathbf{y}}(k+H_c+2|k) &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k+H_c|k), \\ &\vdots \end{aligned} \tag{17}$$

Therefore, the prediction from time instant H_c to infinity in the performance index (15) can be written as

$$\begin{aligned} &\sum_{p=H_c}^{\infty} \|\hat{\mathbf{y}}(k+p|k)\|^2_{\mathbf{M}_0} \\ &\quad = \sum_{p=H_c}^{\infty} \mathbf{x}^T(k+p|k)\mathbf{C}^T\mathbf{M}_0\mathbf{C}\mathbf{x}(k+p|k) \\ &\quad = \mathbf{x}^T(k+H_c|k)\left[\sum_{i=0}^{\infty}(\mathbf{A}^T)^i\mathbf{C}^T\mathbf{M}_0\mathbf{C}\mathbf{A}^i\right] \\ &\qquad \times \mathbf{x}(k+H_c|k). \end{aligned} \tag{18}$$

If the plant is stable, the sum in the brackets is convergent. The matrix

$$\mathbf{M}_{\infty} = \sum_{i=0}^{\infty}(\mathbf{A}^T)^i\mathbf{C}^T\mathbf{M}_0\mathbf{C}\mathbf{A}^i \tag{19}$$

is the solution to the matrix Lyapunov equation

$$\mathbf{A}^T\mathbf{M}_{\infty}\mathbf{A} + \mathbf{C}^T\mathbf{M}_0\mathbf{C} = \mathbf{M}_{\infty} \tag{20}$$

because

$$\begin{aligned} \mathbf{M}_{\infty} &= \mathbf{C}^T\mathbf{M}_0\mathbf{C} + \sum_{i=1}^{\infty}(\mathbf{A}^T)^i\mathbf{C}^T\mathbf{M}_0\mathbf{C}\mathbf{A}^i \\ &= \mathbf{C}^T\mathbf{M}_0\mathbf{C} + \mathbf{A}^T\mathbf{M}_{\infty}\mathbf{A}. \end{aligned} \tag{21}$$

Finally, for (15) and (18) the performance index (14) can be written as a function defined over the finite horizon H_c with an additional penalty term which depends only on the predicted state at the end of the control horizon,

$$\begin{aligned} J(k) = \|\mathbf{x}(k+H_c|k)\|^2_{\mathbf{M}_{\infty}} &+ \sum_{p=1}^{H_c-1} \|\hat{\mathbf{y}}(k+p|k)\|^2_{\mathbf{M}_0} \\ &+ \sum_{p=0}^{H_c-1} \|\Delta\mathbf{u}(k+p|k)\|^2_{\mathbf{\Lambda}_0}. \end{aligned} \tag{22}$$

Defining the vectors of lengths $M(H_c-1)$ and NH_c:

$$\begin{aligned} \hat{\mathbf{y}}(k) &= \begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \vdots \\ \hat{\mathbf{y}}(k+H_c-1|k) \end{bmatrix}, \\ \Delta\mathbf{u}(k) &= \begin{bmatrix} \Delta\mathbf{u}(k|k) \\ \vdots \\ \Delta\mathbf{u}(k+H_c-1|k) \end{bmatrix}, \end{aligned} \tag{23}$$

respectively, the performance index (14) has the following form:

$$J(k) = \|\mathbf{x}(k+H_c|k)\|^2_{\mathbf{M}_\infty} + \|\hat{\mathbf{y}}(k)\|^2_{\mathbf{M}} + \|\Delta\mathbf{u}(k)\|^2_{\mathbf{\Lambda}}, \tag{24}$$

where the diagonal matrices of dimensions $M(H_c-1)$ and NH_c are

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_0 & & \\ & \ddots & \\ & & \mathbf{M}_0 \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_0 & & \\ & \ddots & \\ & & \mathbf{\Lambda}_0 \end{bmatrix}, \tag{25}$$

respectively.

In order to calculate the optimal input profile, the first and the second part of the performance index (24) have to be expressed as functions of $\Delta\mathbf{u}(k)$. As far as practical applications are concerned, nonzero set-point values should be also possible. For linear plants the predicted output $\hat{\mathbf{y}}(k)$ can be split into the forced response $\mathbf{G}\Delta\mathbf{u}(k)$ and the free responses $\mathbf{y}^0(k)$ (e.g. Camacho and Bordons, 1999; Maciejowski, 2002; Tatjewski, 2002). Defining the vectors $\mathbf{y}^0(k)$ and $\mathbf{y}^{\mathrm{ref}}(k)$ of lengths $M(H_c-1)$:

$$\mathbf{y}^0(k) = \begin{bmatrix} y^0_1(k+1|k) \\ \vdots \\ y^0_M(k+1|k) \\ \vdots \\ y^0_1(k+H_c-1|k) \\ \vdots \\ y^0_M(k+H_c-1|k) \end{bmatrix}, \quad \mathbf{y}^{\mathrm{ref}}(k) = \begin{bmatrix} y^{\mathrm{ref}}_1(k) \\ \vdots \\ y^{\mathrm{ref}}_M(k) \\ \vdots \\ y^{\mathrm{ref}}_1(k) \\ \vdots \\ y^{\mathrm{ref}}_M(k) \end{bmatrix}, \tag{26}$$

and the matrix $\mathbf{G}$ of dimension $M(H_c-1)\times NH_c$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_2 & \mathbf{S}_1 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{S}_{H_c-1} & \mathbf{S}_{H_c-2} & \dots & \mathbf{S}_1 & \mathbf{0} \end{bmatrix}, \tag{27}$$

where the submatrices

$$\mathbf{S}_k = \begin{bmatrix} s^{1,1}_k & \dots & s^{1,N}_k \\ \vdots & \ddots & \vdots \\ s^{M,1}_k & \dots & s^{M,N}_k \end{bmatrix} \tag{28}$$

contain step response coefficients of the multivariable process (1), the performance index (24) can be rewritten as

$$\begin{aligned} J(k) = {} & \left\|\mathbf{x}(k+H_c|k) - \mathbf{x}^{\mathrm{ref}}(k)\right\|^2_{\mathbf{M}_\infty} \\ & + \left\|\mathbf{G}\Delta\mathbf{u}(k) + \mathbf{y}^0(k) - \mathbf{y}^{\mathrm{ref}}(k)\right\|^2_{\mathbf{M}} \\ & + \|\Delta\mathbf{u}(k)\|^2_{\mathbf{\Lambda}}. \end{aligned} \tag{29}$$

The penalised state vector in (29) is

$$\mathbf{x}(k+H_c|k) - \mathbf{x}^{\mathrm{ref}}(k) = \begin{bmatrix} \mathbf{x}_u(k+H_c|k) - \mathbf{x}^{\mathrm{ref}}_u(k) \\ \mathbf{x}_y(k+H_c|k) - \mathbf{x}^{\mathrm{ref}}_y(k) \end{bmatrix} \tag{30}$$

with the subvectors $\mathbf{x}^{\mathrm{ref}}_u(k)$ and $\mathbf{x}^{\mathrm{ref}}_y(k)$ of lengths $N(nb-1)$ and Mna, respectively,

$$\mathbf{x}^{\mathrm{ref}}_u(k) = \begin{bmatrix} u^{\mathrm{ref}}_1(k) \\ \vdots \\ u^{\mathrm{ref}}_N(k) \\ \vdots \\ u^{\mathrm{ref}}_1(k) \\ \vdots \\ u^{\mathrm{ref}}_N(k) \end{bmatrix}, \quad \mathbf{x}^{\mathrm{ref}}_y(k) = \begin{bmatrix} y^{\mathrm{ref}}_1(k) \\ \vdots \\ y^{\mathrm{ref}}_M(k) \\ \vdots \\ y^{\mathrm{ref}}_1(k) \\ \vdots \\ y^{\mathrm{ref}}_M(k) \end{bmatrix}. \tag{31}$$

The input values which correspond to the set-points are calculated from the static characteristic of the plant. Using the input-output model (1), assuming that $z^{-1}=1$, we get

$$\begin{bmatrix} u^{\mathrm{ref}}_1(k) \\ \vdots \\ u^{\mathrm{ref}}_N(k) \end{bmatrix} = \mathbf{G}^s \begin{bmatrix} y^{\mathrm{ref}}_1(k) \\ \vdots \\ y^{\mathrm{ref}}_M(k) \end{bmatrix}, \tag{32}$$

where the matrix $\mathbf{G}^s$ of dimension $N\times M$ is

$$\mathbf{G}^s = \mathbf{B}^+(1)\mathbf{A}(1) \tag{33}$$

and "$+$" denotes the Moore-Penrose pseudo-inverse (e.g. Golub and Van Loan, 1989).

Although in the case of stable plants the control horizon can be as short as 1 (Muske and Rawlings, 1993; Rawlings and Muske, 1993), it is convenient to impose a technical condition

$$H_c \geq \max(na, nb-1) \tag{34}$$

which implies that the state vector at the end of the finite control horizon, i.e. $\mathbf{x}(k+H_c|k)$, contains only elements of the vector $\Delta\mathbf{u}(k)$ and future outputs belonging to the vector $\hat{\mathbf{y}}(k)$. The first part of the penalised predicted state vector at the end of the control horizon can be written as

$$\mathbf{x}_u(k+H_c|k) - \mathbf{x}^{\mathrm{ref}}_u(k) = \mathbf{E}\Delta\mathbf{u}(k) + \mathbf{v}_u(k), \tag{35}$$

where the matrix $\mathbf{E}$ of dimension $N(nb-1)\times NH_c$ and the vector $\mathbf{v}_u(k)$ of length $N(nb-1)$ are

$$\mathbf{E}=\begin{bmatrix} \overbrace{\begin{matrix}\mathbf{I} & \dots & \mathbf{I}\\ \mathbf{I} & \dots & \mathbf{I}\\ \vdots & \ddots & \vdots\\ \mathbf{I} & \dots & \mathbf{I}\end{matrix}}^{H_c-nb+2} & \overbrace{\begin{matrix}\mathbf{0} & \dots & \mathbf{0}\\ \mathbf{I} & \dots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{I} & \dots & \mathbf{I}\end{matrix}}^{nb-2} \end{bmatrix},$$

$$\mathbf{v}_u(k)=\begin{bmatrix} u_1(k-1)-u_1^{\mathrm{ref}}(k)\\ \vdots\\ u_N(k-1)-u_N^{\mathrm{ref}}(k)\\ \vdots\\ u_1(k-1)-u_1^{\mathrm{ref}}(k)\\ \vdots\\ u_N(k-1)-u_N^{\mathrm{ref}}(k)\end{bmatrix}, \tag{36}$$

and $\mathbf{I}$ and $\mathbf{0}$ are square matrices of dimensions $N\times N$.

For the second part of the penalised state vector, again splitting the predicted output into forced and free responses, we obtain

$$\mathbf{x}_y(k+H_c|k)-\mathbf{x}_y^{\mathrm{ref}}(k)=\mathbf{G}_y\Delta\mathbf{u}(k)+\mathbf{v}_y(k), \tag{37}$$

where the matrix $\mathbf{G}_y$ of dimension $Mna\times NH_c$ is

$$\mathbf{G}_y=\begin{bmatrix} \mathbf{S}_{H_c-na+1} & \mathbf{S}_{H_c-na} & \dots & \mathbf{0}\\ \mathbf{S}_{H_c-na+2} & \mathbf{S}_{H_c-na+1} & \dots & \mathbf{0}\\ \vdots & \vdots & & \vdots\\ \mathbf{S}_{H_c} & \mathbf{S}_{H_c-1} & \dots & \mathbf{S}_1\end{bmatrix} \tag{38}$$

and the vector $\mathbf{v}_y(k)$ of length Mna is

$$\mathbf{v}_y(k)=\begin{bmatrix} y_1^0(k-na+1+H_c|k)-y_1^{\mathrm{ref}}(k)\\ \vdots\\ y_M^0(k-na+1+H_c|k)-y_M^{\mathrm{ref}}(k)\\ \vdots\\ y_1^0(k+H_c|k)-y_1^{\mathrm{ref}}(k)\\ \vdots\\ y_M^0(k+H_c|k)-y_M^{\mathrm{ref}}(k)\end{bmatrix}, \tag{39}$$

where $y_m^0(k+p|k)$ denote free response elements of the plant.

Taking into account (35) and (37), the penalised state vector can be then expressed as

$$\mathbf{x}(k+H_c|k)-\mathbf{x}^{\mathrm{ref}}(k)=\mathbf{G}_0\Delta\mathbf{u}(k)+\mathbf{v}(k), \tag{40}$$

where

$$\mathbf{G}_0=\begin{bmatrix}\mathbf{E}\\ \mathbf{G}_y\end{bmatrix},\quad \mathbf{v}(k)=\begin{bmatrix}\mathbf{v}_u(k)\\ \mathbf{v}_y(k)\end{bmatrix}. \tag{41}$$

Finally, using (40), we obtain the performance index (29) which depends explicitly on the input increments $\Delta\mathbf{u}(k)$:

$$\begin{aligned} J(k)=&\|\mathbf{G}_0\Delta\mathbf{u}(k)+\mathbf{v}(k)\|^2_{\mathbf{M}_\infty}\\ &+\left\|\mathbf{G}\Delta\mathbf{u}(k)+\mathbf{y}^0(k)-\mathbf{y}^{\mathrm{ref}}(k)\right\|^2_{\mathbf{M}}\\ &+\|\Delta\mathbf{u}(k)\|^2_{\mathbf{\Lambda}}. \end{aligned} \tag{42}$$

2.3. Performance Index Minimisation — the Unconstrained Case

Having equated the derivative of $J(k)$ given by (42) to $\mathbf{0}$, we obtain the optimal control increments

$$\begin{aligned} \Delta\mathbf{u}(k)=&\left[\mathbf{G}_0^T\mathbf{M}_\infty\mathbf{G}_0+\mathbf{G}^T\mathbf{M}\mathbf{G}+\mathbf{\Lambda}\right]^{-1}\\ &\times\left[\mathbf{G}^T\mathbf{M}\left(\mathbf{y}^{\mathrm{ref}}(k)-\mathbf{y}^0(k)\right)\right.\\ &\left.-\mathbf{G}_0^T\mathbf{M}_\infty\mathbf{v}(k)\right] \end{aligned} \tag{43}$$

which can be written as

$$\Delta\mathbf{u}(k)=\mathbf{K}(\mathbf{y}^{\mathrm{ref}}(k)-\mathbf{y}^0(k))+\mathbf{L}\mathbf{v}(k), \tag{44}$$

where

$$\begin{aligned} \mathbf{K}&=\left[\mathbf{G}_0^T\mathbf{M}_\infty\mathbf{G}_0+\mathbf{G}^T\mathbf{M}\mathbf{G}+\mathbf{\Lambda}\right]^{-1}\mathbf{G}^T\mathbf{M},\\ \mathbf{L}&=-\left[\mathbf{G}_0^T\mathbf{M}_\infty\mathbf{G}_0+\mathbf{G}^T\mathbf{M}\mathbf{G}+\mathbf{\Lambda}\right]^{-1}\mathbf{G}_0^T\mathbf{M}_\infty. \end{aligned} \tag{45}$$

If the state penalty term is excluded from the performance index, i.e. $\mathbf{M}_\infty=\mathbf{0}$, the obtained formulae (44) and (45) coincide with those for the classical GPC algorithm, namely

$$\Delta\mathbf{u}(k)=\left[\mathbf{G}^T\mathbf{M}\mathbf{G}+\mathbf{\Lambda}\right]^{-1}\mathbf{G}^T\mathbf{M}\left(\mathbf{y}^{\mathrm{ref}}(k)-\mathbf{y}^0(k)\right). \tag{46}$$

Although the solution given by (43) is formally correct, in this paper a computationally safer method is developed, according to the indications given in (Maciejowski, 2002). The reason is that the matrix $\mathbf{G}_0^T\mathbf{M}_\infty\mathbf{G}_0+\mathbf{G}^T\mathbf{M}\mathbf{G}+\mathbf{\Lambda}$, especially in multivariable cases, may suffer from ill-conditioning. Such a phenomenon may result in numerical problems. One can notice that the performance index (42) can be expressed as

$$J(k)=\left\|\begin{bmatrix}\mathbf{S}_\infty(\mathbf{G}_0\Delta\mathbf{u}(k)+\mathbf{v}(k))\\ \mathbf{S}_\mathbf{M}(\mathbf{G}\Delta\mathbf{u}(k)+\mathbf{y}^0(k)-\mathbf{y}^{\mathrm{ref}}(k))\\ \mathbf{S}_\mathbf{\Lambda}\Delta\mathbf{u}(k)\end{bmatrix}\right\|^2, \tag{47}$$

where

$$\mathbf{S}_{\infty}^{T}\mathbf{S}_{\infty}=\mathbf{M}_{\infty},\quad \mathbf{S}_{\mathbf{M}}^{T}\mathbf{S}_{\mathbf{M}}=\mathbf{M},\quad \mathbf{S}_{\mathbf{\Lambda}}^{T}\mathbf{S}_{\mathbf{\Lambda}}=\mathbf{\Lambda}. \tag{48}$$

If matrices $\mathbf{M}$ and $\mathbf{\Lambda}$ are diagonal, as is assumed in the paper, so are $\mathbf{S}_{\mathbf{M}}$ and $\mathbf{S}_{\mathbf{\Lambda}}$. For positive definite ones, Cholesky factorisation should be used. Because the state-space representation given by (7) and (8) results in a positive semidefinite matrix $\mathbf{M}_{\infty}$, the matrix $\mathbf{S}_{\infty}$ is calculated using a singular value decomposition (Golub and Van Loan, 1989). For a symmetric matrix $\mathbf{M}_{\infty}$ we have

$$\mathbf{M}_{\infty}=\mathbf{U}\mathbf{S}\mathbf{U}^{T}, \tag{49}$$

where the diagonal matrix $\mathbf{S}$ is also positive semidefinite and the matrix $\mathbf{U}$ is unitary. Hence

$$\mathbf{S}_{\infty}=\mathbf{S}^{\frac{1}{2}}\mathbf{U}^{T}. \tag{50}$$

The optimal value of the vector $\Delta\mathbf{u}(k)$ is then the least-squares solution of the equation

$$\begin{bmatrix} \mathbf{S}_{\infty}(\mathbf{G}_0\Delta\mathbf{u}(k)+\mathbf{v}(k)) \\ \mathbf{S}_{\mathbf{M}}(\mathbf{G}\Delta\mathbf{u}(k)+\mathbf{y}^{0}(k)-\mathbf{y}^{\mathrm{ref}}(k)) \\ \mathbf{S}_{\mathbf{\Lambda}}\Delta\mathbf{u}(k) \end{bmatrix}=\mathbf{0}. \tag{51}$$

The solution to the control problem is then obtained as

$$\Delta\mathbf{u}(k)=\mathbf{P}\begin{bmatrix} -\mathbf{S}_{\infty}\mathbf{v}(k) \\ \mathbf{S}_{\mathbf{M}}(\mathbf{y}^{\mathrm{ref}}(k)-\mathbf{y}^{0}(k)) \\ \mathbf{0} \end{bmatrix}, \tag{52}$$

where $\mathbf{0}$ is a vector of length NH_c. The matrix $\mathbf{P}$ of dimension $NH_c\times(Mna+N(nb-1)+M(H_c-1)+NH_c)$ is calculated as a Moore-Penrose pseudoinverse

$$\mathbf{P}=\begin{bmatrix} \mathbf{S}_{\infty}\mathbf{G}_0 \\ \mathbf{S}_{\mathbf{M}}\mathbf{G} \\ \mathbf{S}_{\mathbf{\Lambda}} \end{bmatrix}^{+}=\begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 \end{bmatrix}, \tag{53}$$

where the submatrices $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ are of dimensions $NH_c\times(Mna+N(nb-1))$, $NH_c\times M(H_c-1)$, $NH_c\times NH_c$, respectively. From (52) and (53) the optimal input profile is then obtained by means of (44), but the matrices $\mathbf{K}$ and $\mathbf{L}$ are calculated without any inversion, namely,

$$\mathbf{K}=\mathbf{P}_2\mathbf{S}_{\mathbf{M}},\quad \mathbf{L}=-\mathbf{P}_1\mathbf{S}_{\infty}. \tag{54}$$

As far as the pseudoinverse in (53) is concerned, the matrix $\mathbf{P}$ it calculated from

$$\mathbf{P}=\mathbf{V}\mathbf{\Sigma}_1^{T}\mathbf{U}^{T},\quad \mathbf{\Sigma}_1=\begin{bmatrix} \frac{1}{\sigma_1} & \dots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \dots & \frac{1}{\sigma_{NH_c}} & \end{bmatrix} \tag{55}$$

using the following SVD decomposition:

$$\begin{bmatrix} \mathbf{S}_{\infty}\mathbf{G}_0 \\ \mathbf{S}_{\mathbf{M}}\mathbf{G} \\ \mathbf{S}_{\mathbf{\Lambda}} \end{bmatrix}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{T},\qquad \mathbf{\Sigma}=\begin{bmatrix} \sigma_1 & \dots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \dots & \sigma_{NH_c} & \end{bmatrix}. \tag{56}$$

2.4. Forced and Free Responses

In the sequel it is shown that the forced and, especially, the free response can be easily calculated without resorting to a matrix Diophantine equation. In the derivation of the GPC algorithm a minimal-variance predictor is applied (Camacho and Bordons, 1999; Clarke *et al.*, 1987a; 1987b; Tatjewski, 2002). As far as the most practical white noise case is concerned, it can be easily proved that the resulting prediction is equivalent to the deterministic one obtained by assuming the so-called "DMC type" disturbance model (Tatjewski, 2002). Prediction errors, resulting from the model's inaccuracies and unmeasured disturbances, are treated collectively. They are calculated using actual (measured) values $y_m(k)$ and values $y_m(k|k-1)$ obtained from the model as follows:

$$d_m(k)=y_m(k)-y_m(k|k-1). \tag{57}$$

Using the model (1), we have

$$d_m(k)=y_m(k)-\sum_{n=1}^{N}\sum_{i=1}^{nb}b_i^{m,n}u_n(k-i)+\sum_{i=1}^{na}a_i^{m}y_m(k-i). \tag{58}$$

It is typical of the "DMC type" disturbance model to assume that disturbances $d_m(k)$ are constant over the prediction horizon. Such an approach turns out to effectively compensate for both step and slowly varying disturbances, which are usually encountered in industry. Moreover, it also provides an integral action which eliminates a steady-state offset. The general prediction equation is then

$$\hat{y}_m(k+p|k)=y_m(k+p|k)+d_m(k), \tag{59}$$

where the quantity $y_m(k+p|k)$ is obtained from the model.

The prediction equation for time instant $k+1|k$ is derived from (1), (58) and (59). We have

$$\begin{aligned}\hat{y}_m(k+1|k) = &\sum_{n=1}^{N}\Big[b_1^{m,n}\Delta u_n(k|k) + b_2^{m,n}\Delta u_n(k-1)\\ &+\cdots+b_{nb}^{m,n}\Delta u_n(k-nb+1)\Big]\\ &+(1-a_1^m)y_m(k)+(a_1^m-a_2^m)y_m(k-1)\\ &+\cdots+(a_{na-1}^m-a_{na}^m)y_m(k-na+1)\\ &+a_{na}^m y_m(k-na). \end{aligned} \tag{60}$$

It can be also rewritten in the form

$$\begin{aligned}\hat{y}_m(k+1|k) = &\sum_{n=1}^{N}\Big[g_0^{m,n}(1)\Delta u_n(k|k)\\ &+g_1^{m,n}(1)\Delta u_n(k-1)\\ &+\cdots+g_{nb-1}^{m,n}(1)\Delta u_n(k-nb+1)\Big]\\ &+f_0^m(1)y_m(k)+f_1^m(1)y_m(k-1)\\ &+\cdots+f_{na}^m(1)y_m(k-na), \end{aligned} \tag{61}$$

where, for $i=0,\ldots,nb-1$,

$$g_i^{m,n}(1) = b_{i+1}^{m,n} \tag{62}$$

and

$$f_i^m(1) = \begin{cases} 1-a_1^m & \text{for} \quad i=0,\\ a_i^m - a_{i+1}^m & \text{for} \quad i=1,\ldots,na-1,\\ a_i^m & \text{for} \quad i=na. \end{cases} \tag{63}$$

The prediction equation for time instant $k+2|k$ depends on $\hat{y}_m(k+1|k)$. We have

$$\begin{aligned}\hat{y}_m(k+2|k) = &\sum_{n=1}^{N}\Big[g_0^{m,n}(1)\Delta u_n(k+1|k)\\ &+\big(f_0^m(1)g_0^{m,n}(1)+g_1^{m,n}(1)\big)\Delta u_n(k|k)\\ &+\cdots+\big(f_0^m(1)g_{nb-2}^{m,n}(1)+g_{nb-1}^{m,n}(1)\big)\\ &\times\Delta u_n(k-nb+2)\\ &+f_0^m(1)g_{nb-1}^{m,n}(1)\Delta u_n(k-nb+1)\Big]\\ &+\big(f_0^m(1)f_0^m(1)+f_1^m(1)\big)y_m(k)\\ &+\big(f_0^m(1)f_1^m(1)+f_2^m(1)\big)y_m(k-1)\\ &+\cdots+\big(f_0^m(1)f_{na-1}^m(1)+f_{na}^m(1)\big)\\ &\times y_m(k-na+1)\\ &+f_0^m(1)f_{na}^m(1)y_m(k-na). \end{aligned} \tag{64}$$

Similarly to (61), it can be rewritten as

$$\begin{aligned}\hat{y}_m(k+2|k) = &\sum_{n=1}^{N}\Big[g_{-1}^{m,n}(2)\Delta u_n(k+1|k)\\ &+g_0^{m,n}(2)\Delta u_n(k|k)\\ &+\cdots+g_{nb-1}^{m,n}(2)\Delta u_n(k-nb+1)\Big]\\ &+f_0^m(2)y_m(k)+f_1^m(2)y_m(k-1)\\ &+\cdots+f_{na}^m(2)y_m(k-na), \end{aligned} \tag{65}$$

where

$$g_i^{m,n}(2) = \begin{cases} g_0^{m,n}(1) & \text{for } i=-1,\\ f_0^m(1)g_i^{m,n}(1)+g_{i+1}^{m,n}(1) & \\ \qquad\qquad \text{for } i=0,\ldots,nb-2,\\ f_0^m(1)g_i^{m,n}(1) & \text{for } i=nb-1, \end{cases} \tag{66}$$

and

$$f_i^m(2) = \begin{cases} f_0^m(1)f_i^m(1)+f_{i+1}^m(1) & \text{for } i=0,\ldots,na-1,\\ f_0^m(1)f_i^m(1) & \text{for } i=na. \end{cases} \tag{67}$$

In general, the output prediction (59) can be expressed as

$$\begin{aligned}\hat{y}_m(k+p|k) = &\sum_{n=1}^{N}\Big[\sum_{i=0}^{p-1}g_{1-p+i}^{m,n}(p)\Delta u_n(k+p-1-i|k)\\ &+\sum_{i=1}^{nb-1}g_i^{m,n}(p)\Delta u_n(k-i)\Big]\\ &+\sum_{i=0}^{na}f_i^m(p)y_m(k-i), \end{aligned} \tag{68}$$

where

$$g_i^{m,n}(p) = \begin{cases} g_{i+1}^{m,n}(p-1) & \text{for } i=1-p,\ldots,-1,\\ f_0^m(p-1)g_i^{m,n}(1)+g_{i+1}^{m,n}(p-1) & \\ \qquad\qquad \text{for } i=0,\ldots,nb-2,\\ f_0^m(p-1)g_i^{m,n}(1) & \text{for } i=nb-1, \end{cases} \tag{69}$$

and

$$f_i^m(p) = \begin{cases} f_0^m(p-1)f_i^m(1)+f_{i+1}^m(p-1) & \\ \qquad\qquad \text{for } i=0,\ldots,na-1,\\ f_0^m(1)f_i^m(1) & \text{for } i=na. \end{cases} \tag{70}$$

The prediction $\hat{y}_m(k+p|k)$ is a linear function of decision variables $\Delta u_n(k+p|k)$, past input values from time instant $k-nb$ to $k-1$ and past output values from $k-na$

to k. From (68), taking into account only future control moves, we obtain the forced response

$$\begin{aligned}\boldsymbol{\Delta} y_m(k+1|k) &= \sum_{n=1}^{N} b_1^{m,n}\Delta u_n(k|k),\\ \boldsymbol{\Delta} y_m(k+2|k) &= \sum_{n=1}^{N}\Big[\big(b_1^{m,n}+b_2^{m,n}-a_1^m b_1^{m,n}\big)\Delta u_n(k|k)\\ &\quad + b_1^{m,n}\Delta u_n(k+1|k)\Big],\\ &\vdots \end{aligned} \tag{71}$$

It can be expressed as

$$\boldsymbol{\Delta} y_m(k+p|k) = \sum_{j=1}^{p}\sum_{n=1}^{N} s_j^{m,n}\Delta u_n(k+p-j|k). \tag{72}$$

Step response coefficients, comprising the submatrices $\mathbf{S}_k$ (28), resulting matrices $\mathbf{G}_y$ (38) and $\mathbf{G}$ (27), are then obtained recursively from

$$s_j^{m,n} = \sum_{i=1}^{\min(j,nb)} b_i^{m,n} - \sum_{i=1}^{\min(j-1,na)} a_i^m s_{j-i}^{m,n}. \tag{73}$$

From (68) the free response is also derived. Obviously, it is independent of future control moves:

$$\begin{aligned} y_m^0(k+p|k) &= \sum_{n=1}^{N}\sum_{i=1}^{nb} e_i^{m,n}(p)u_n(k-i)\\ &\quad + \sum_{i=0}^{na} f_i^m(p)y_m(k-i), \end{aligned} \tag{74}$$

where

$$e_i^{m,n}(p) = \begin{cases} 0 & \text{for} \quad i=1, nb=1,\\ g_i^{m,n}(p) & \text{for} \quad i=1, nb>1,\\ g_i^{m,n}(p)-g_{i-1}^{m,n}(p) & \text{for} \quad i=2,\dots,nb-1,\\ & \quad i<nb, nb>1,\\ -g_{i-1}^{m,n}(p) & \text{for} \quad i=nb, nb>1. \end{cases} \tag{75}$$

2.5. Closed-Form Control Law

Although unconstrained linear predictive control algorithms result in precomputed off-line control laws, this issue is given little consideration in the literature (Mayne, 2001), but is of fundamental importance as far as practical implementations are concerned.

From the solution to the optimisation problem (44), where $\mathbf{K}$ and $\mathbf{L}$ are given by (45) or (54), the control increment of the r-th input at the current time instant k is

$$\begin{aligned}\Delta u_r(k|k) = &\sum_{p=1}^{H_c-1}\sum_{m=1}^{M} k_{r,(p-1)M+m}\\ &\times\big(y_m^{\mathrm{ref}}(k) - y_m^0(k+p|k)\big)\\ &+\sum_{p=1}^{nb}\sum_{n=1}^{N} l_{r,(p-1)N+n}\big(u_n(k-1)-u_n^{\mathrm{ref}}(k)\big)\\ &+\sum_{p=0}^{na}\sum_{m=1}^{M} l_{r,Nnb+pM+m}\\ &\times\big(y_m^0(k-na+p+H_c|k)-y_m^{\mathrm{ref}}(k)\big). \end{aligned} \tag{76}$$

Inserting the free response (74), eliminating the quantities $u_n^{\mathrm{ref}}(k)$ by means of (32) and after simple arithmetic manipulations, the analytical control law can be expressed as a function of the reference trajectory as well as past input and output values:

$$\begin{aligned} u_r(k|k) &= \sum_{m=1}^{M} k_{r,m}^{y_{\mathrm{ref}}} y_m^{\mathrm{ref}}(k) + \sum_{n=1}^{N}\sum_{j=1}^{nb} k_{r,n}^u(j)u_n(k-j)\\ &\quad + \sum_{m=1}^{M}\sum_{j=0}^{na} k_{r,m}^y(j)y_m(k-j), \end{aligned} \tag{77}$$

where, for $r=1,\dots,N,\ m=1,\dots,M$,

$$k_{r,m}^{y_{\mathrm{ref}}} = \sum_{p=1}^{H_c-1} k_{r,s_1} - \sum_{p=0}^{na} l_{r,s_3} - \sum_{p=1}^{nb}\sum_{n=1}^{N} l_{r,s_2} g_{n,m}^s, \tag{78}$$

and for $r=1,\dots,N,\ n=1,\dots,N,\ j=1,\dots,nb$,

$$k_{r,n}^u(j) = \begin{cases} \displaystyle\sum_{p=0}^{na}\sum_{m=1}^{M} l_{r,s_3} e_j^{m,n}(s_4) - \sum_{p=1}^{H_c-1}\sum_{m=1}^{M} k_{r,s_1} e_j^{m,n}(p)\\ \qquad \displaystyle+\sum_{p=1}^{nb} l_{r,s_3} + 1 & \text{for} \quad j=1, r=n,\\ \displaystyle\sum_{p=0}^{na}\sum_{m=1}^{M} l_{r,s_3} e_j^{m,n}(s_4) - \sum_{p=1}^{H_c-1}\sum_{m=1}^{M} k_{r,s_1} e_j^{m,n}(p)\\ \qquad \displaystyle+\sum_{p=1}^{nb} l_{r,s_3} & \text{for} \quad j=1, r\neq n,\\ \displaystyle\sum_{p=0}^{na}\sum_{m=1}^{M} l_{r,s_3} e_j^{m,n}(s_4) - \sum_{p=1}^{H_c-1}\sum_{m=1}^{M} k_{r,s_1} e_j^{m,n}(p)\\ & \text{for} \quad j\neq 1, r\neq n, \end{cases} \tag{79}$$

and for $r = 1, \ldots, N,\ m = 1, \ldots, M,\ j = 0, \ldots, na$,

$$k_{r,m}^{y}(j) = \sum_{p=0}^{na} l_{r,s_3} f_j^m(s_4) - \sum_{p=1}^{H_c-1} k_{r,s_1} f_j^m(p) \qquad (80)$$

and, where the auxiliary indices are $s_1 = (p-1)M + m$, $s_2 = (p-1)N + n$, $s_3 = Nnb + pM + m$, $s_4 = p - na + H_c$.

The implementation steps of the unconstrained version of the algorithm are as follows. At first, for a given input-output model (1), the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, which give the state-space formulation (4), are obtained according to (7), (8), and the matrix $\mathbf{G}^s$ is found from (33). The control horizon H_c is fixed using the condition (34) and the weighting coefficients $\mu_m > 0$ and $\lambda_n \geq 0$ are chosen according to the needs. The symmetric matrix $\mathbf{M}_\infty$ is then obtained as a solution to the discrete-time Lyapunow equation (20). Step response coefficients $s_k^{m,n}$, comprising the matrices $\mathbf{G}$ (27) and $\mathbf{G}_y$ (38), are obtained from (73). Free response quotients $e_j^{m,n}(p)$, $f_j^m(p)$ are calculated by means of (75), (63) and (70). The matrices $\mathbf{K}$ and $\mathbf{L}$ are then calculated according to (45) or, as is recommended, to (54). Finally, the coefficients $k_{r,m}^{y_{\mathrm{ref}}}$, $k_{r,n}^{u}(j)$, $k_{r,m}^{y}(j)$ are determined from (78)–(80). During on-line control, the current values of the input variables are calculated from the closed-form control law (77), which, for a given model of the plant, is always a polynomial of the same order, regardless of the length of the finite control horizon and the weighting coefficients.

2.6. Performance Index Minimisation — the Constrained Case

In the constrained case the performance index (42) is minimised subject to the following constraints:

$$\begin{aligned} u_n^{\min} \leq u_n(k+p|k) \leq u_n^{\max}, \\ -\Delta u_n^{\max} \leq \Delta u_n(k+p|k) \leq \Delta u_n^{\max} \end{aligned} \qquad (81)$$

for $n = 1, \ldots, N,\ p = 0, \ldots, H_c - 1$. Such a problem is of the quadratic-programming type:

$$\begin{aligned} \min_{\mathbf{x}} J(k) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{f}^T\mathbf{x}, \\ \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \\ \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned} \qquad (82)$$

where

$$\begin{aligned} &\mathbf{x} = \Delta\mathbf{u}(k), \quad \mathbf{x}_{\min} = -\Delta\mathbf{u}_{\max}, \quad \mathbf{x}_{\max} = \Delta\mathbf{u}_{\max}, \\ &\mathbf{H} = 2\left[\mathbf{G}_0^T\mathbf{M}_\infty\mathbf{G}_0 + \mathbf{G}^T\mathbf{M}\mathbf{G} + \mathbf{\Lambda}\right], \\ &\mathbf{f} = 2\left[\mathbf{G}^T\mathbf{M}(\mathbf{y}^0(k) - \mathbf{y}^{\mathrm{ref}}(k)) + \mathbf{G}_0^T\mathbf{M}_\infty\mathbf{v}(k)\right], \end{aligned} \qquad (83)$$

and

$$\mathbf{A} = \begin{bmatrix} -\mathbf{J} \\ \mathbf{J} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \end{bmatrix}. \qquad (84)$$

It can be effectively solved on-line by means of the available software packages. Matrix $\mathbf{J}$ is of dimension $NH_c \times NH_c$,

$$\mathbf{J} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \ldots & \mathbf{I} \end{bmatrix}, \qquad (85)$$

where $\mathbf{I}$ and $\mathbf{0}$ are square matrices of dimensions $N \times N$ and the following vectors are of lengths NH_c:

$$\begin{aligned} \mathbf{u}_{\min} &= \begin{bmatrix} u_1^{\min} \\ \vdots \\ u_N^{\min} \\ \vdots \\ u_1^{\min} \\ \vdots \\ u_N^{\min} \end{bmatrix}, \quad \mathbf{u}_{\max} = \begin{bmatrix} u_1^{\max} \\ \vdots \\ u_N^{\max} \\ \vdots \\ u_1^{\max} \\ \vdots \\ u_N^{\max} \end{bmatrix}, \\ \mathbf{u}(k-1) &= \begin{bmatrix} u_1(k-1) \\ \vdots \\ u_N(k-1) \\ \vdots \\ u_1(k-1) \\ \vdots \\ u_N(k-1) \end{bmatrix}. \end{aligned} \qquad (86)$$

3. Simulation Results

The process under consideration is a stirred tank reactor (Camacho and Bordons, 1999), which can be described for small signals by the following transfer matrix (time constants in min):

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1+0.7s} & \dfrac{5}{1+0.3s} \\ \dfrac{1}{1+0.5s} & \dfrac{2}{1+0.4s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}, \qquad (87)$$

where the manipulated variables u_1 and u_2 are the feed flowrate and the flow of the coolant in the jacket, respectively. The controlled variables y_1 and y_2 are the effluent concentration and the reactor temperature. Having discretised the model with a sampling time of 0.03 min, we obtain

$$\mathbf{A}(z^{-1}) = \begin{bmatrix} a_{1,1} & 0 \\ 0 & a_{2,2} \end{bmatrix},$$

where $a_{1,1} = 1-1.862885z^{-1}+0.866877z^{-2}$ and $a_{2,2} = 1 - 1.869508z^{-1} + 0.873715z^{-2}$.

$$\mathbf{B}(z^{-1}) = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix}, \tag{88}$$

where,

$$\begin{aligned} b_{1,1} &= 0.041951z^{-1} - 0.037959z^{-2}, \\ b_{1,2} &= 0.475812z^{-1} - 0.455851z^{-2}, \\ b_{2,1} &= 0.058235z^{-1} - 0.054027z^{-2}, \\ b_{2,2} &= 0.445130z^{-1} - 0.136097z^{-2}. \end{aligned}$$

During all the experiments carried out, the weighting coefficients μ_1, μ_2 were set to 1 and 5, respectively, and $\lambda_1 = \lambda_2$ to 0.5.

For comparison, at first the unconstrained CRHPC algorithm (Clarke and Scattolini, 1991) was considered. The length of the control horizon has to satisfy the condition $H_c \geq 5$, which results in stable dead-beat control. Figure 1 depicts simulation results. As one might expect, a short horizon combined with terminal equality constraints results in excessive input profiles and a huge overshoot, which is unacceptable in practice. To overcome such obstacles, the horizon was set to 20. It made it possible to significantly reduce the amplitude of the input variables, but the closed-loop behaviour is much slower.

Figure 2 depicts simulation results of the unconstrained infinite horizons predictive control algorithm, with the control horizons $H_c = 5$ and $H_c = 20$ and the same changes in the set-points as in the previous experiment. The controller was implemented according to the procedure described in Section 2.5, the current values of the control variables were calculated by means of the closed-form control law (77). When compared with the CRHPC algorithm, its behaviour is appealing, i.e. the amplitudes of the input profiles and the overshoot are much smaller. Because the infinite prediction horizon is used and the plant is fast, the results are similar for both control horizons.

Finally, to reduce rapid changes in input variables, the following constraints were taken into account:

$$\begin{aligned} &u_1^{\min} = -2, \quad u_1^{\max} = 2, \quad \Delta u_1^{\max} = 0.5, \\ &u_2^{\min} = -0.5, \quad u_2^{\max} = 0.5, \quad \Delta u_2^{\max} = 0.25. \end{aligned} \tag{89}$$

At each time step, the quadratic programming problem was solved as described in Section 2.6, the closed-form free response was calculated using (74). The simulation results of the infinite horizon algorithm are presented in Fig. 3. Again, as in the unconstrained case, the input and output variables for $H_c = 5$ and $H_c = 20$ are hardly distinguishable.

4. Summary

The paper presented a stabilising infinite horizon predictive control algorithm for multivariable input-output models. Using the theoretical background presented by (Muske and Rawlings, 1993;

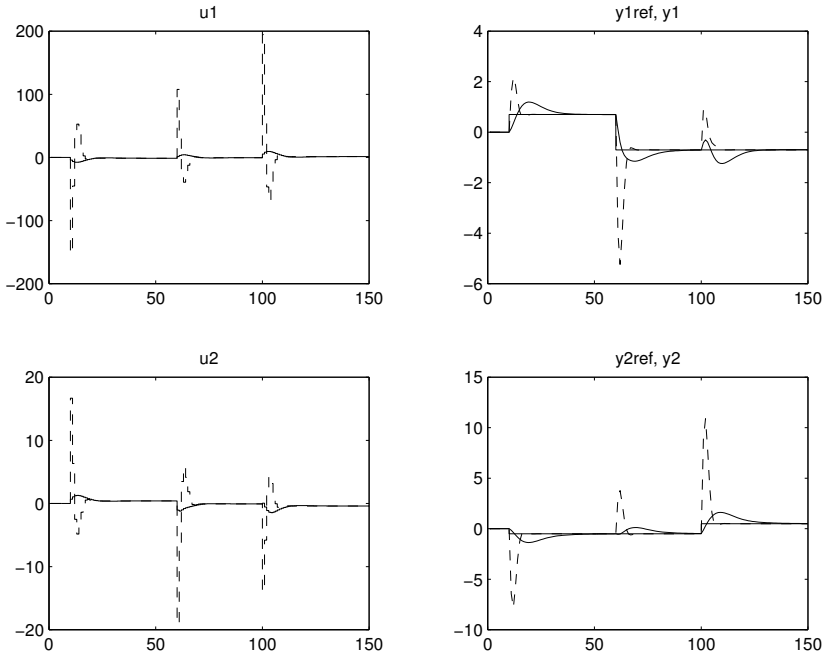

Fig. 1. Simulation results of the unconstrained CRHPC algorithm: $H_c = 5$ (dashed) and $H_c = 20$ (solid).

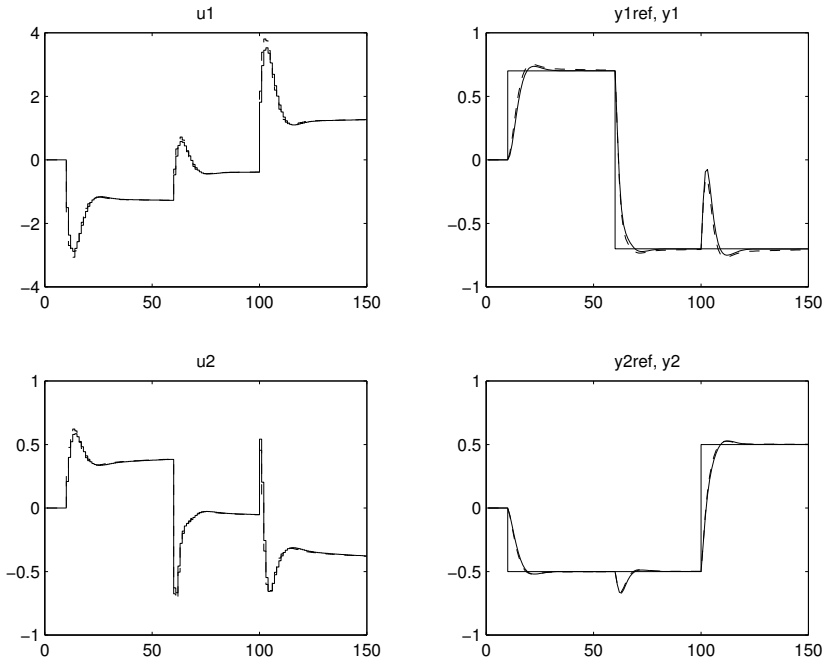

Fig. 2. Simulation results of the unconstrained infinite horizon predictive control algorithm: $H_c = 5$ (dashed) and $H_c = 20$ (solid).

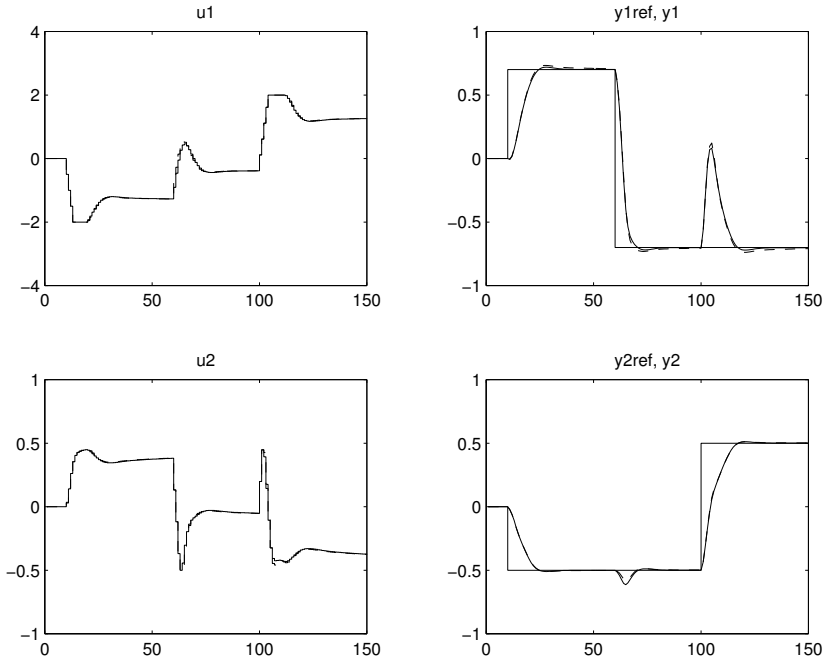

Fig. 3. Simulation results of the constrained infinite horizon predictive control algorithm: $H_c = 5$ (dashed) and $H_c = 20$ (solid).

Rawlings and Muske, 1993) in the state-space context, the infinite horizon performance index was reformulated as a finite horizon one with a penalty term. Two versions of the controller were discussed: the unconstrained one, for which an analytical stabilising control law can be calculated beforehand using a numerically reliable SVD decomposition, and the constrained one, which results in a quadratic programming problem to be solved on-line. It is also shown how forced and free responses can be easily calculated without solving a matrix Diophantine equation, assuming the "DMC-type" disturbance model.

A small computational burden and a simple implementation procedure are the advantages of the described algorithm. In fact, when compared with the standard GPC and DMC algorithms, the only additional task is to solve the discrete-time Lyapunow equation, the closed-

form control law and the optimisation problem being of comparable complexity. Moreover, when compared with a terminally constrained predictive control scheme, for example of a CRHPC type (Clarke and Scattolini, 1991), the algorithm reveals its superiority. The lack of equality constraints makes the solution to the optimisation problem easier from the numerical point of view. First and foremost, terminal constraints, particularly combined with relatively short horizons, result in highly excessive, practically unacceptable, input profiles and big overshoots. The presented approach is also computationally less demanding in comparison with other stabilising algorithms, for example LQGPC (Ordys *et al.*, 2000).

The presented algorithm is different from GPC$^\infty$ (Scokaert, 1997): the state vector is defined in a different way—it comprises not only past output, but also input values, the "DMC type" disturbance model is assumed, and hence the model of the plant is of the ARX rather than the CARIMA type. Because only stable processes are considered in the paper, the derivation of the algorithm is much simpler in comparison with GPC$^\infty$. However, the presented algorithm can be easily modified to deal with unstable plants. It is only necessary to decompose the model into its stable and unstable parts by means of an eigenvalue-eigenvector (Jordan) decomposition and to take into account additional equality constraints to force the unstable modes to be at zero (in the case of a "regulator" problem formulation) at the end of the finite control horizon while only the stable modes should contribute to the performance index. It is possible to reduce the number of decision variables beforehand using the constraints, as was done in GPC$^\infty$, or it may be convenient to let the optimisation routine decide how to use the equality constraints, especially in multivariable cases (e.g. Maciejowski, 2002).

Acknowledgment

The work presented in this paper was supported by a statutory research grant from the Faculty of Electronics and Information Technology, Warsaw University of Technology.

References

Bitmead R.R., Gevers M. and Wertz V. (1990): *Adaptive Optimal Control—The Thinking Man's GPC.* — Englewood Cliffs: Prentice Hall.

Camacho E.F. and Bordons C. (1999): *Model Predictive Control.* — London: Springer.

Chisci L. and Mosca E. (1994): *Stabilizing I-O receding horizon control of CARMA plants.* — IEEE Trans. Automat. Contr., Vol. 39, No. 3, pp. 614–618.

Clarke D.W. and Scattolini R. (1991): *Constrained receding-horizon predictive control.* — Proc. IEE, Part D, Vol. 138, No. 4, pp. 347–354.

Clarke D.W. and Mohtadi C. (1989): *Properties of generalized predictive control.* — Automatica, Vol. 25, No. 6, pp. 859–875.

Clarke D.W., Mohtadi C. and Tuffs P.S. (1987a): *Generalized predictive control – I. The basic algorithm.* — Automatica, Vol. 23, No. 2, pp. 137–148.

Clarke D.W., Mohtadi C. and Tuffs P.S. (1987b): *Generalized predictive control – II. Extensions and interpretations.* — Automatica, Vol. 23, No. 2, pp. 149–160.

Cutler C.R. and Ramaker B.L. (1980): *Dynamic matrix control – A computer control algorithm.* — Proc. Joint. Automat. Contr. Conf., San Francisco.

Golub G.H. and Van Loan C.F. (1989): *Matrix Computations.* — Baltimore: The Johns Hopkins University Press.

Gutman P. and Hagander P. (1985): *A new design of constrained controllers for linear systems.* — IEEE Trans. Automat. Contr., Vol. 30, No. 1, pp. 22–33.

Henson M.A. (1998): *Nonlinear model predictive control: current status and future directions.* — Comput. Chem. Eng., Vol. 23, No. 2, pp. 187–202.

Kailath T. (1980): *Linear Systems.* — Englewood Cliffs: Prentice Hall.

Kwon W.H. and Byun D.G. (1989): *Receding horizon tracking control as a predictive control and its stability properties.* — Int. J. Contr., Vol. 50, No. 5, pp. 1807–1824.

Maciejowski J.M. (2002): *Predictive Control with Constraints.* — Englewood Cliffs: Prentice Hall.

Mayne D.Q. (2001): *Control of constrained dynamic systems.* — Europ. J. Contr., Vol. 7, Nos. 2–3, pp. 87–99.

Mayne D.Q., Rawlings J.B., Rao C.V. and Scokaert P.O.M. (2000): *Constrained model predictive control: Stability and optimality.* — Automatica, Vol. 36, No. 6, pp. 789–814.

Morari M. and Lee J.H. (1999): *Model predictive control: past, present and future.* — Comput. Chem. Eng., Vol. 23, No. 4/5, pp. 667–682.

Muske K.R. and Rawlings J.B. (1993): *Model predictive control with linear models.* — AIChE J., Vol. 39, No. 2, pp. 262–287.

Ordys W.A., Hangstrup M.E. and Grimble M.J. (2000): *Dynamic algorithm for linear quadratic Gaussian predictive control.* — Int. J. Appl. Math. Comput. Sci., Vol. 10, No. 2, pp. 227–244.

Rawlings J.B. and Muske K.R. (1993): *The stability of constrained receding horizon control.* — IEEE Trans. Automat. Contr., Vol. 38, No. 10, pp. 1512–1516.

Rouhani R. and Mehra R.K. (1982): *Model algoritmic control (MAC); Basic theoretical properties.* — Automatica, Vol. 18, No. 4, pp. 401–414.

Scattolini R. and Bittanti S. (1990): *On the choice of the horizon in long-range predictive control – some simple criteria.* — Automatica, Vol. 26, No. 5, pp. 915–917.

Scokaert P.O.M. (1997): *Infinite horizon generalized predictive control.* — Int. J. Contr., Vol. 66, No. 1, pp. 161–175.

Scokaert P.O.M. and Clarke D. W. (1994): *Stabilising properties of constrained predictive control.* — Proc. IEE, Part D, Vol. 141, No. 5, pp. 295–304.

Sznaier M. and Damborg M.J. (1990): *Heuristically enhanced feedback control of constrained discrete-time linear systems.* — Automatica, Vol. 26, No. 3, pp. 521–532.

Tatjewski P. (2002): *Advanced Control of Industrial Processes, Structures and Algorithms.* — Warszawa: Akademicka Oficyna Wydawnicza Exit (in Polish).

Zafiriou E. (1990): *Robust model predictive control of processes with hard constraints.* — Comput. Chem. Eng., Vol. 14, Nos. 4/5, pp. 359–371.

Zafiriou E. and Marchal A.L. (1991): *Stability of SISO quadratic dynamic matrix control with hard output constraints.* — AIChE J., Vol. 37, No. 10, pp. 1550–1560.

Received: 16 October 2002
Revised: 2 February 2004