Pergamon

Computers chem. Engng Vol. 19, No. 4, pp. 409–421, 1995
Copyright © 1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0098–1354(94)00063–8
0098-1354/95 $9.50+0.00

LIMITATIONS OF DYNAMIC MATRIX CONTROL

P. LUNDSTRÖM,[1]† J. H. LEE[1,]‡, M. MORARI[1,]§ and S. SKOGESTAD[2]
[1] Chemical Engineering 210-41, California Institute of Technology, Pasadena, CA 91125, U.S.A.
[2] Chemical Engineering, University of Trondheim, NTH, N-7034 Trondheim, Norway

(Received 5 June 1992; final revision received 13 May 1994; received for publication 7 June 1994)

Abstract—Dynamic matrix control (DMC) is based on two assumptions which limit the feedback performance of the algorithm. The first assumption is that a stable step response model can be used to represent the plant. The second assumption is that the difference between the measured and the predicted output can be modeled as a step disturbance acting on the output.

These assumptions lead to the following limitations:

1. Good performance may require an excessive number of step response coefficients.
2. Poor performance may be observed for disturbances affecting the plant inputs.
3. Poor robust performance may be observed for multivariable plants with strong interactions.

Limitations 1 and 2 apply when the plant's open-loop time constant is much larger than the desired closed-loop time constant. Limitation 3 is caused by gain uncertianty on the inputs.

In this paper we separate the DMC algorithm into a predictor and an optimizer. This enables us to highlight the DMC limitations and to suggest how they can be avoided. We demonstrate that a new model predictive control (MPC) algorithm, which includes an observer, does not suffer from the listed limitations.

1. INTRODUCTION

Dynamic matrix control (DMC) has been successfully used in industry for more than a decade. Several authors have reported improved control performance by use of DMC as compared to "traditional" control algorithms (Cutler and Ramaker, 1980; Prett and Gillette, 1980; García and Morshedi, 1986). DMC has the ability to deal with constraints, which probably is one of the major reasons for its popularity. It also allows set point changes to be "announced" in advance and it facilitates feedforward control. However, the *feedback* properties of a DMC controller are limited by two restrictive assumptions which are implicit in the algorithm:

A1. A stable step response model can be used to represent the plant.

A2. The difference between the measured and the predicted output can be modeled as a step disturbance acting on the output.

The objective of this paper is to clearly point out these assumptions and to illustrate in which situations they may limit the feedback properties of DMC. Obviously, there are other algorithms in the literature which do not suffer from these deficiencies. In this paper, however, we wanted to adhere to the DMC structure because of its popularity and its capability to handle constraints. We demonstrate that the limitations can be avoided by use of a new observer based algorithm, by Lee *et al.* (1994), which is a *direct* extension of DMC.

DMC belongs to the family of model predictive control (MPC) algorithms. The main idea behind these algorithms is to use an explicit model of the plant to predict the open-loop future behavior of the controlled outputs over a finite time horizon. The predicted behavior is then used to find a finite sequence of control moves which minimizes a particular objective function without violating prespecified constraints. Usually only the first input move is implemented and the procedure is repeated at the next sampling instant.

This algorithm can be separated into two parts, a predictor and an optimizer. By splitting up the algorithm in this manner, similarities with state-observer state-feedback controllers become apparent. In fact, Lee *et al.* (1994) show that unconstrained DMC is equivalent to an optimal state observer (Kalman filter) and linear quadratic feedback, using a receding horizon approach and special assumptions about disturbances and measurement noise.

† Present address: Chemical Engineering, University of Trondheim, NTH, N-734 Trondheim, Norway.

‡ Present address: Department of Chemical Engineering, Auburn University, Auburn, AL 36849-5127, U.S.A.

§ To whom all correspondence should be addressed.

In this paper we use the predictor–optimizer representation of DMC. In this framework the limitations of DMC can be traced to the predictor part of the algorithm. We only consider unconstrained DMC, but the results carry over to the general case with constraints, since the issue of constraints only affects the optimizer.

The limitations we want to illustrate are:

L1. Good performance may require an excessive number step response coefficients.

L2. Poor performance may be observed for "ramp-like" disturbances acting on the plant outputs. In particular, this occurs for input disturbances for plants with large time constants.

L3. Poor robust performance, due to input gain uncertainty (which is always present in practice), may be observed for multivariable plants with strong interactions.

In addition, there is the obvious limitation that the plant has to be stable.

This paper is organized as follows. In Section 2 we present the algorithms we will use. The purpose is to give a coherent overview of the algorithms and to point out the implicit assumption made in DMC. We also use this section to define the nomenclature. Readers not familiar with MPC are referred to García *et al.* (1989). In Section 3 we use a simple single-input–single-output (SISO) plant and a multi-input–multi-output (MIMO) distillation column to illustrate the limitations of DMC and demonstrate that the algorithm by Lee *et al.* (1994) can be used to avoid these limitations. Section 4 contains a Discussion and Section 5, Conclusions.

2. MODEL PREDICTIVE CONTROL

2.1. *Dynamic matrix control*

2.1.1. Modeling the plant. In the original DMC formulation (Cutler and Ramaker, 1980) a step response model of the plant is used to predict the future behavior of the controlled variables.

Let the step response of a SISO system be represented by the sequence:

$$[s_1 s_2 \ldots s_{n-1} s_n s_{n+1} \ldots], \qquad (1)$$

where the kth element is the output at time k caused by a unit step input at time 0. For a stable plant this sequence will asymptotically reach a constant value, i.e. $s_n \approx s_{n+1}$. For a MIMO system with n_u inputs and n_y outputs we get:

$$S_i = \begin{bmatrix} s_{1,1,i} & s_{1,2,i} & \cdots & s_{1,n_u,i} \\ s_{2,1,i} & s_{2,2,i} & \cdots & s_{2,n_u,i} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n_y,1,i} & s_{n_y,2,i} & \cdots & s_{n_y,n_u,i} \end{bmatrix} i = 1, \ldots, n. \qquad (2)$$

The step response model can be represented in the following state space form, which is equivalent to that presented by Li *et al.* (1989):

$$Y(k+1) = MY(k) + S\Delta u(k), \qquad (3)$$

$$y(k) = NY(k), \qquad (4)$$

where

$$\Delta u(k) = u(k) - u(k-1), \qquad (5)$$

$$Y(k) = [y(k)^{\mathrm{T}} y(k+1)^{\mathrm{T}} \ldots y(k+n-1)^{\mathrm{T}}]^{\mathrm{T}} \qquad (6)$$

$$M = \begin{bmatrix} 0 & I_{n_y} & 0 & \ldots & 0 & 0 \\ 0 & 0 & I_{n_y} & \ddots & 0 & 0 \\ & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & I_{n_y} & 0 \\ 0 & 0 & 0 & \ddots & 0 & I_{n_y} \\ 0 & 0 & 0 & \ldots & 0 & I_{n_y} \end{bmatrix} \qquad n \times n_y;$$

$$S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-2} \\ S_{n-1} \\ S_n \end{bmatrix} \qquad (7)$$

and

$$N = [\overbrace{I_{n_y} \quad 0 \quad 0 \quad \ldots \quad 0 \quad 0}^{n \times n_y}]. \qquad (8)$$

$\Delta u(k)$ is a vector of changes in the manipulated inputs at time k. $y(k)$ is the output vector at time k. The vector $Y(k+1)$ represents the dynamic states of the system. Each state, $y(k+l)$, has a special interpretation: it is the future output vector at time $k+l$ assuming constant inputs [i.e. $\Delta u(k+j) = 0$ for $j \geqslant 0$]. The new state vector $Y(k+1)$ is the old vector $Y(k)$ shifted up n_y elements plus the contribution made by the latest input change $\Delta u(k)$.

2.1.2. The predictor. The DMC algorithm is illustrated in Fig. 1. The objective of the predictor is to generate a vector, $\mathcal{Y}(k+1|k)$, of predicted open-loop outputs over a horizon of p future time steps, the prediction horizon. This prediction vector is then used as an input to the optimizer.

The DMC predictor is described by the following equations:

$$\tilde{Y}(k) = M\tilde{Y}(k-1) + S\Delta u(k-1), \qquad (9)$$

$$\tilde{y}(k) = N\tilde{Y}(k), \qquad (10)$$

$$\mathcal{Y}(k+1|k) = M_p \tilde{Y}(k) + \mathcal{I}[\hat{y}(k) - \tilde{y}(k)] \qquad (11)$$

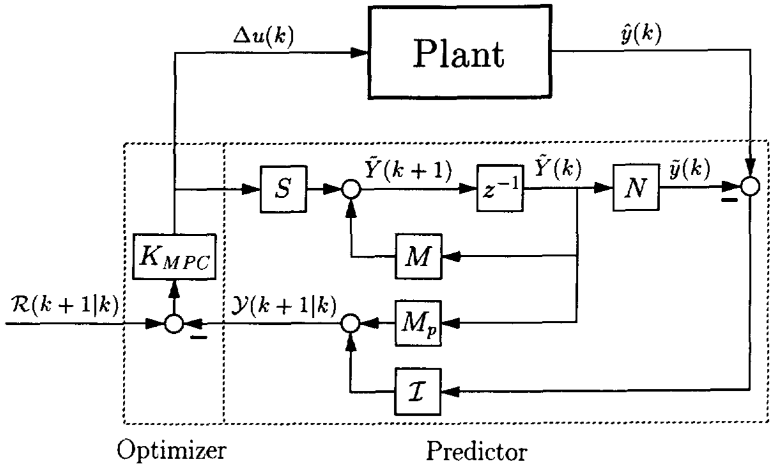

Fig. 1. DMC controller separated into a predictor and an optimizer.

where M_p is the first $p \times n_y$ rows of M and:

$$\mathcal{I} = [\overbrace{I_{n_y}\ I_{n_y}\ \dots\ I_{n_y}\ I_{n_y}}^{p \times n_y}]^{\mathrm{T}} \tag{12}$$

We use $\tilde{}$ to denote that the output is from the model and not from the true plant. $\hat{y}(k)$ is a vector of measured outputs at time k. $\tilde{y}(k)$ and $\hat{y}(k)$ are discontinuous at k_- while $u(k)$ is discontinuous at k_+, i.e. $\hat{y}$ is measured slightly before time k and u is adjusted slightly after time k.

2.1.3. The optimizer. We use the QDMC objective function from García and Morshedi (1986):

$$J = \min_{\Delta\mathcal{U}(k|k)} \{\|\Gamma[\mathcal{Y}_m(k+1)|k) - \mathcal{R}(k+1|k)]\|^2 + \|\Lambda\Delta\mathcal{U}(k|k)\|^2\}, \tag{13}$$

where

$$\Delta\mathcal{U}(k|k) = [\Delta u(k|k)^{\mathrm{T}}\Delta u(k+1|k)^{\mathrm{T}} \dots \Delta u(k+m-1|k)^{\mathrm{T}}]^{\mathrm{T}}, \tag{14}$$

$$\mathcal{Y}_m(k+1|k) = [y_m(k+1|k)^{\mathrm{T}}y_m(k+2|k)^{\mathrm{T}} \dots y_m(k+p|k)^{\mathrm{T}}]^{\mathrm{T}}, \tag{15}$$

and

$$\mathcal{R}(k+1|k) = [r(k+1|k)^{\mathrm{T}}r(k+2|k)^{\mathrm{T}} \dots r(k+p|k)^{\mathrm{T}}]^{\mathrm{T}}. \tag{16}$$

$\Delta\mathcal{U}(k|k)$ is the optimal control sequence computed at time k for m future input moves, where m is the input horizon. $\mathcal{Y}_m(k+1|k)$ is a vector of outputs predicted at time k, over a horizon of p future time steps, including the effect of the m optimal input moves:

$$\mathcal{Y}_m(k+1|k) = \mathcal{Y}(k+1|k) + \mathcal{S}_p^m\Delta\mathcal{U}(k|k), \tag{17}$$

where

$$\mathcal{S}_p^m = \begin{bmatrix} S_1 & 0 & \dots & 0 \\ S_2 & S_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ S_m & S_{m-1} & \dots & S_1 \\ \vdots & \vdots & & \vdots \\ S_p & S_{p-1} & \dots & S_{p-m-1} \end{bmatrix} \tag{18}$$

$\mathcal{R}(k+1|k)$ is a vector describing the desired output trajectory (set points) over p future time steps. Γ and Λ are weighting matrices and are usually chosen to be diagonal.

The least squares solution to this problem is:

$$\Delta\mathcal{U}(k|k) = [(\mathcal{S}_p^m)^{\mathrm{T}}\Gamma^{\mathrm{T}}\Gamma\mathcal{S}_p^m + \Lambda^{\mathrm{T}}\Lambda]^{-1}(\mathcal{S}_p^m)^{\mathrm{T}}\Gamma^{\mathrm{T}}\Gamma \times [\mathcal{R}(k+1|k) - \mathcal{Y}(k+1|k)]. \tag{19}$$

Only the first input move is implemented, and the resulting optimizer is a constant gain matrix, K_{MPC}:

$$\Delta u(k) = [I\ \ 0\dots0]\Delta\mathcal{U}(k|k) = K_{\mathrm{MPC}}[\mathcal{R}(k+1|k) - \mathcal{Y}(k+1|k)], \tag{20}$$

$$K_{\mathrm{MPC}} = [I\ \ 0\dots0][(\mathcal{S}_p^m)^{\mathrm{T}}\Gamma^{\mathrm{T}}\Gamma\mathcal{S}_p^m + \Lambda]^{-1} \times (\mathcal{S}_p^m)^{\mathrm{T}}\Gamma^{\mathrm{T}}\Gamma \tag{21}$$

2.1.4. DMC assumptions. The DMC controller can only be used with stable plants. There are two reasons for this: (1) The internal model [equations (9–10)] can only describe a stable plant; (2) $\hat{y}(k) - \tilde{y}(k)$ can grow unbounded for unstable systems leading to internal instability.

The internal model of the DMC predictor [equations (9–10)] does not yield an estimate of the true plant output. It computes the open-loop *model* output, $\tilde{y}(k)$, for previous input moves, but does not account for the effect of disturbances and model–plant mismatch. This means that generally $\hat{y}(k) - \tilde{y}(k)$ is not zero when there is no steady-state offset and $\hat{y}(k) = 0$. Rather, $-\tilde{y}(k)$ equals the accumulated effect of disturbances and model–plant mismatch.

Equation (11) gives the predicted open-loop output vector, $\mathcal{Y}(k+1|k)$. It is the predicted effect of previous input moves, $M_p\tilde{Y}(k)$, plus a simple bias adjustment given by the mismatch between the measured output, $\hat{y}(k)$, and the output form the internal model, $\bar{y}(k)$.

To achieve good control performance, $\mathcal{Y}(k+1|k)$ should be close to the *true* open-loop output. This requires that n, the number of coefficient matrices in S, is chosen such that $S_n \approx S_{n+1}$, otherwise $M_p\tilde{Y}(k)$ will be in error. It also requires that $\hat{y}(k)-\hat{y}(k)$ stays approximately constant.

We formulate these requirements as two implicit assumptions made in the DMC algorithm:

A1. A stable step response model with $S_n \approx S_{n+1}$ can be used to represent the plant.

A2. The difference between the measured and the predicted output can be modeled as a step disturbance acting on the output.

2.2. *DMC with general state space model*

The DMC algorithm can also be derived for a general discrete state space model (Prett and García, 1988) instead of the step response model [equations (3–4)] used in the previous section. We will denote this algorithm DMCss. The only difference between DMC and DMCss is the representation of the internal model. We include DMCss in this paper because it allows us to study DMC without the effects of truncation errors caused by $S_n \neq S_{n+1}$.

Let the plant model be defined by the following equations:

$$x(k+1)=Ax(k)+Bu(k), \tag{22}$$

$$y(k)=Cx(k). \tag{23}$$

Using this model the DMC algorithm can be described by the block diagram in Fig. 1 by making the following substitutions, $\tilde{Y}(k)=\Delta\bar{x}(k)=\Delta\bar{x}(k)-\bar{x}(k-1)$, $M=A$, $S=B$, $N=[0\ldots0]$ and

$$M_p=\begin{bmatrix} CA \\ CA+CA^2 \\ \ddots \\ \Sigma_{i=1}^{p} CA^i \end{bmatrix} \tag{24}$$

2.3. *Observer based model predictive control*

This algorithm is from Lee *et al.* (1994), we will denote it "OBMPC".

Lee *et al.* use the following extended version of the step response model in equations (3–4). The extension is made in order to include measurement noise and general disturbances acting on the plant outputs. It also allows modeling of integrating systems:

$$Y(k+1)=MY(k)+S\Delta u(k)+T\Delta w(k), \tag{25}$$

$$y(k)=NY(k), \tag{26}$$

$$\hat{y}(k)=y(k)+v(k), \tag{27}$$

$\Delta w(k)=w(k)-w(k-1)$ is a vector of changes in disturbances and $v(k)$ is a vector of measurement noise:

$$M=\underbrace{\begin{bmatrix} 0 & I_{n_y} & 0 & \ldots & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{n_y} & \ldots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & I_{n_y} & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & I_{n_y} & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & I_{n_y} & C_u & C_w \\ 0 & 0 & 0 & \ldots & 0 & 0 & A_u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & A_w \end{bmatrix}}_{n\times n_y+\dim\{x_u\}+\dim\{x_w\}}, \tag{28}$$

$$S=\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-2} \\ S_{n-1} \\ S_n \\ B_u \\ 0 \end{bmatrix}; \quad T=\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ B_w \end{bmatrix}, \tag{29}$$

$$N=[I_{n_y} \quad 0 \quad 0 \quad \ldots \quad 0 \quad 0 \quad 0 \quad 0], \tag{30}$$

$$Y(k)=[y(k)^T y(k+1)^T \ldots y(k+n-1)^T \; x_u(k)^T x_w(k)^T]^T. \tag{31}$$

A_u, B_u and C_u constitute a state space description of the residual plant dynamics after n sampling intervals. A_w, B_w and C_w describe the dynamics of the disturbances. x_u and x_w are state vectors for residual plant dynamics and disturbance dynamics, respectively.

This representation allows very general modeling of plant and disturbances. However, we will approximate the residual dynamics with $n_y \times n_u$ first-order systems, each describing the slow response from one input to one output. This approximation gives:

$$A_u\begin{bmatrix} A_{u1} & & \\ & \ddots & \\ & & A_{un_u} \end{bmatrix}; \quad A_{uj}=\begin{bmatrix} a_{uj1} & & \\ & \ddots & \\ & & a_{ujn_y} \end{bmatrix}, \tag{32}$$

$$B_u=\begin{bmatrix} B_{u1} & & \\ & \ddots & \\ & & B_{un_u} \end{bmatrix}; \quad B_{uj}=\begin{bmatrix} b_{uj1} \\ \vdots \\ b_{ujn_y} \end{bmatrix}, \tag{33}$$

$$C_u=[\overbrace{I_{n_y} \quad I_{n_y} \quad \ldots \quad I_{n_y}}^{n_y\times n_u}], \tag{34}$$

We also restrict measurement noise and disturbances to the following special case:

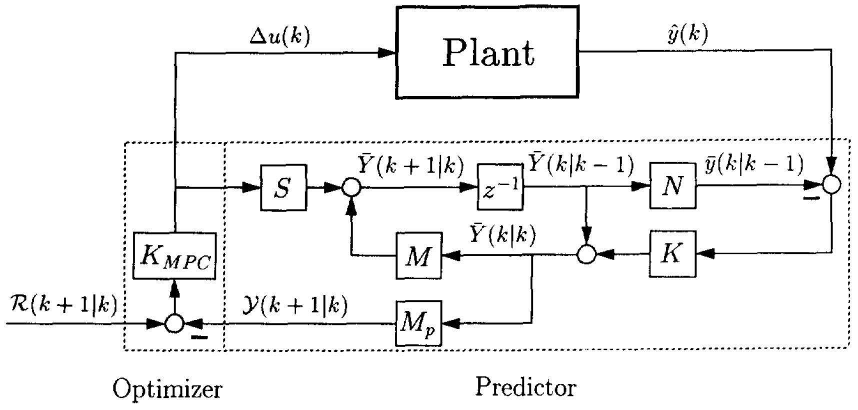

Fig. 2. Observer based MPC controller separated into a predictor and an optimizer.

1. The measurement noise at each output is uncorrelated white noise.
2. The disturbances at the outputs are integrated white noise filtered through first-order dynamics.

For this special case we get the following diagonal covariance matrices;

$$E\{\Delta w(k)\Delta w(k)^{\mathrm{T}}\} = W = \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_{n_y} \end{bmatrix}, \quad (35)$$

$$E\{v(k)v(k)^{\mathrm{T}}\} = V = \begin{bmatrix} V_1 & & \\ & \ddots & \\ & & V_{n_y} \end{bmatrix} \quad (36)$$

and

$$A_{\mathrm{w}} \triangleq \mathcal{A}\ \mathrm{diag}[\alpha_1, \ldots, \alpha_{n_y}\}; \quad B_{\mathrm{w}} = I_{n_y}; \quad C_w = I_{n_y}. \quad (37)$$

For $\alpha_i = 0$, the disturbance at the ith output is integrated white noise ("type 1" disturbance), while $\alpha_i = 1$ yields *double*-integrated white noise ("type 2" disturbance) at the ith output.

2.3.1. The OBMPC predictor. The OBMPC predictor is using an optimal state observer (i.e. Kalman filter), as seen in Fig. 2. This observer is described by the following equations:

$$\bar{Y}(k|k) = \bar{Y}(k|k-1) + K\{\hat{y}(k) - \bar{y}(k|k-1)\}, \quad (38)$$

$$\bar{Y}(k+1|k) = M\bar{Y}(k|k) + S\Delta u(k), \quad (39)$$

where

$$\bar{Y}(k|k-1) = [\bar{y}(k|k-1)^{\mathrm{T}}\bar{y}(k+1|k-1)^{\mathrm{T}} \ldots$$
$$\bar{y}(k+n-1|k-1)^{\mathrm{T}}\bar{x}_{\mathrm{u}}(k|k-1)^{\mathrm{T}}\bar{x}_{\mathrm{w}}(k|k-1)^{\mathrm{T}}]^{\mathrm{T}}, \quad (40)$$

$\bar{y}(k+1|k)$ is the estimate of $y(k+1)$ based on measurements up to and including time k. The predicted output vector (the input to the optimizer) is:

$$\mathcal{Y}(k+1|k) = M_p\bar{Y}(k|k). \quad (41)$$

For the special noise and disturbance case defined in the previous section, the optimal filter gain K in equation (38) is parametrized as follows (Lee *et al.*, 1994):

$$K = \begin{bmatrix} I_{n_y} \\ I_{n_y} \\ I_{n_y} \\ \vdots \\ I_{n_y} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} (f_{\mathrm{a}})_1 & & \\ & \ddots & \\ & & (f_{\mathrm{a}})_{n_y} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ I_{n_y} \\ I_{n_y} + \mathcal{A} \\ \vdots \\ \sum_{i=0}^{n-2} \mathcal{A}^i \\ 0 \\ \mathcal{A}^{n-1} \end{bmatrix} \begin{bmatrix} (f_{\mathrm{b}})_1 & & \\ & \ddots & \\ & & (f_{\mathrm{b}})_{n_u} \end{bmatrix}, \quad (42)$$

$$(f_{\mathrm{b}})_i = \frac{\alpha_i(f_{\mathrm{a}})_i^2}{1 + \alpha_i - \alpha_i(f_{\mathrm{a}})_i} \quad 1 \leqslant i \leqslant n_y. \quad (43)$$

The adjustable parameters, $(f_{\mathrm{a}})_i$, are determined by the disturbance-to-noise ratio for the ith output, W_i/V_i:

$$(f_{\mathrm{a}})_i \to 0 \quad \text{for} \quad W_i/V_i \to 0, \quad (44)$$

$$(f_{\mathrm{a}})_i \to 1 \quad \text{for} \quad W_i/V_i \to \infty. \quad (45)$$

Hence, we may implement the Kalman filter *without* solving a Riccati equation, and $(f_{\mathrm{a}})_i$ and α_i may be used as on-line tuning parameters.

2.3.2. State-observer state-feedback interpretation. The unconstrained OBMPC described above is a state-observer state-feedback controller using a receding horizon approach. The optimal state-observer is defined by equations (38–39) and the linear quadratic state feedback gain is $K_{\mathrm{MPC}}M_p$. The closed-loop dynamics is determined by the following state transition matrix:

$$\begin{bmatrix} Y(k) \\ Y(k)-\bar{Y}(k|k) \end{bmatrix} = \begin{bmatrix} M-SK_{\text{MPC}}M_p & SK_{\text{MPC}}M_p \\ 0 & M-KNM \end{bmatrix} \times \begin{bmatrix} Y(k-1) \\ Y(k-1)-\bar{Y}(k-1|k-1) \end{bmatrix}. \tag{46}$$

The eigenvalues of $M-SK_{\text{MPC}}M_p$ are the regulator poles and the eigenvalues of $M-KNM$ are the observer poles.

If the measurements are noise-free and the disturbances are random steps acting on the plant outputs, then an unconstrained DMC controller where $S_n = S_{n+1}$ is equivalent to the unconstrained OBMPC controller. That is, for this special case:

$$M_p\bar{Y}(k/k) = M_p\bar{Y}(k) + \mathcal{I}[\hat{y}(k)-\tilde{y}(k)], \tag{47}$$

and DMC is an optimal state-observer state-feedback controller.

3. LIMITATIONS OF DYNAMIC MATRIX CONTROL

3.1. Limitation 1: Good performance may require an excessive number of step response coefficients

In the previous section we stated that the DMC step response model requires $S_n \approx S_{n+1}$. In this section we demonstrate the consequence of sacrificing this requirement.

Assume that a high closed-loop bandwidth is desired for the plant described by the following model:

$$P(s) = \frac{100}{100s+1}e^{-s}, \tag{48}$$

In order to achieve the desired bandwidth a short sampling interval is required. (A common rule is to use $\Delta T \leqslant 2\pi/10\omega_B$, where ω_B is the closed-loop bandwidth, e.g. Middleton, 1991.) We select $\Delta T =$ 1 min. According to common practice (e.g. Cutler and Ramaker, 1980), the truncation error should not be larger than about 5%, which in our case yields 300 step coefficients. However, there is always a practical limit on the number of coefficients (state) that can be used in the internal model, since a large number of coefficients leads to an excessive use of computer memory and a high computational load.

Consider the case of selecting $n = 30$, which is a typical industrial choice (e.g. Cutler and Ramaker, 1980). The effect of this truncation on feedback control is demonstrated in Fig. 3 which shows the response to a unit step disturbance acting on the plant output at time $t = 10$. The simulation is performed with a dead beat DMC controller (controller A, Table 1). The truncated step response causes an erroneous prediction (a "jump") $n-1$ sampling intervals after the disturbance occurred. The error here is so large that it leads to instability.

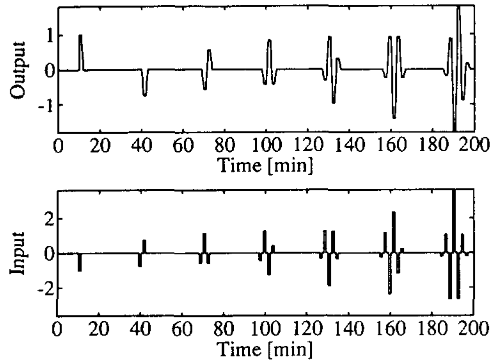

Fig. 3. Effect of truncation. Response for the SISO plant [equation (48)] with DMC controller A (Table 1). A unit step disturbance acts on the plant output at $t = 10$.

From this example we conclude that heavily truncated models cannot be used, and thereby the computer hardware may restrict the choice of sampling interval and the achievable closed-loop bandwidth. This is especially important for plants with a large open-loop time constant.

3.2. Avoiding limitation 1

Limitation 1 may avoided by using a state space model which has no truncation error. For example, the DMCss algorithm requires only 2 states to represent equation (48) for $\Delta T = 1$, one state for the first-order transfer function and one for the delay. More states are needed if ΔT is less than the delay.

The OBMPC controller can also be used to avoid limitation 1. Instead of truncating the response after

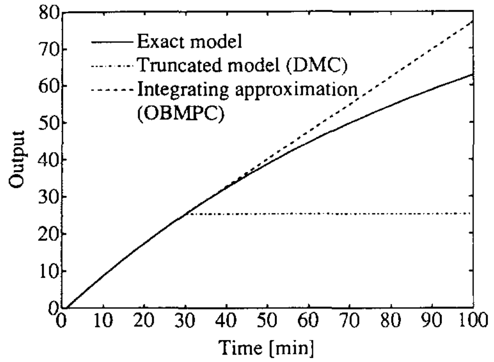

Fig. 4. Open-loop response to unit step in u at $t = 0$ for different models of $100/(100s+1)\,e^{-s}$. The DMC step response model is truncated at $n = 30$, $\Delta T = 1$ [equations (3–4)]. The OBMPC model uses $A_u = 1$, $B_u = s_{n+1} - s_n$, $n = 30$ and $\Delta T = 1$ [equations (25–26)].

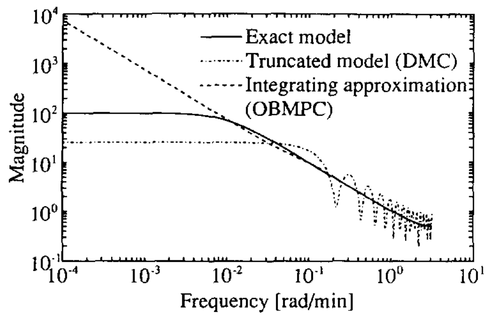

Fig. 5. Frequency response for different models of $100/(100s+1)\,e^{-s}$. The DMC step response model is truncated at $n=30$, $\Delta T=1$ [equations (3–4)]. The OBMPC model uses $A_u=1$, $B_u=s_{n+1}-s_n$, $n=30$ and $\Delta T=1$ [equations (25–26)].

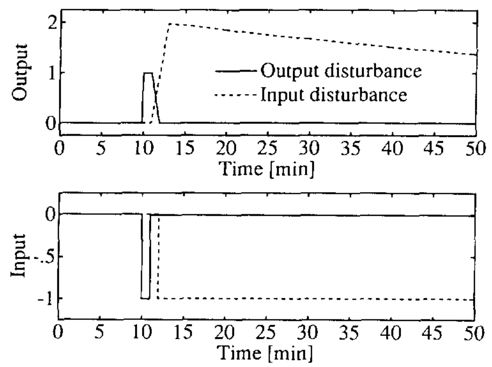

Fig. 7. Responses for the SISO plant [equation (48)] with DMCss controller B (Table 1). Solid curves: unit step disturbance acting on the plant *output* at $t=10$. Dashed curves: unit step disturbance acting on the plant *input* at $t=10$.

n time steps (as would be the case with a DMC step response model) we may use A_u and B_u in equation (32–33) to model the slow dynamics. This way we can reduce the number of states required to represent the plant and thereby allow a short sampling interval. A_u and B_u allow us to use any first-order model of the slow dynamics. We could obtain an exact model of the plant in equation (48) by using $A_u=e^{-\Delta T/100}$ and $B_u=s_{n+1}-s_n$. However, in this example we will approximate the slow dynamics with an integrator, and select $A_u=1$, $B_u=s_{n+1}-s_n$ and $n=30$.

In Fig. 4 we compare the open-loop model response to a unit step in u at time 0 for the exact model [equation (48)] (solid curve) with the truncated DMC model [equations (3–4)] (dash–dot curve) and the OBMPC model [equations (25–26)] with the residual dynamic approximation given above (dashed curve). The last model gives a large error as time increases, but does not have the abrupt change at $t=n\Delta T$ which is characteristic for the truncated DMC model. In the frequency domain (Fig. 5), the truncated DMC model is poor both at high and low frequencies. The integrating OBMPC model on the other hand, yields excellent agreement with the exact model at high frequencies, but displays large deviation at low frequencies.

Simulations with DMC and OBMPC (controllers A and D) are shown in Fig. 6. The disturbance is a unit step on the plant output. We conclude that the rough integrating approximation of the residual dynamics is better than the truncated model. Note that controller D is tuned for *ramp* disturbances, $\alpha=1$ (to take care of the low-frequency mismatch) and some measurement noise, $f_a=0.5$ (to achieve high-frequency robustness).

3.3. Limitation 2: Poor response for ramp-like disturbances

The DMC performance may be very poor for disturbances which do not act as steps on the output. Figure 7 shows the responses for the plant in equation (48) to a unit step acting on the plant output and input, respectively. A DMCss controller is used in both simulations in order to avoid truncation effects.

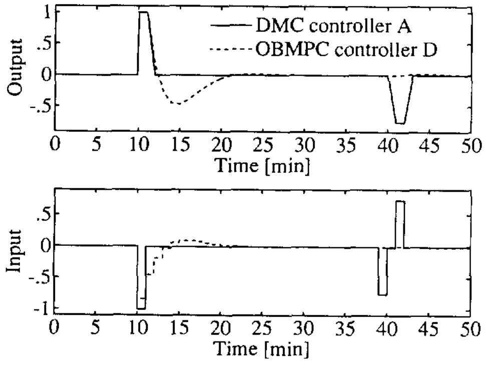

Fig. 6. Responses for the SISO plant [equation (48)] with different controllers (Table 1). A unit step disturbance acts on the plant output at $t=10$.

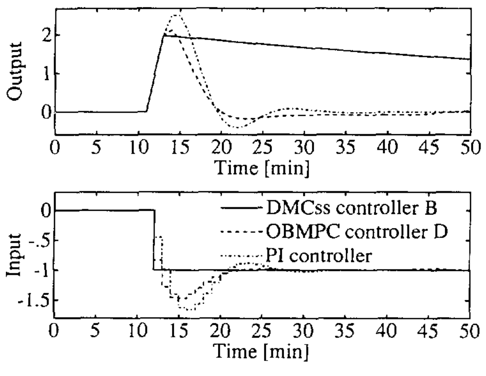

Fig. 8. Responses for the SISO plant [equation (48)] with different controllers (Table 1). A unit step disturbance acts on the plant input at $t=10$.

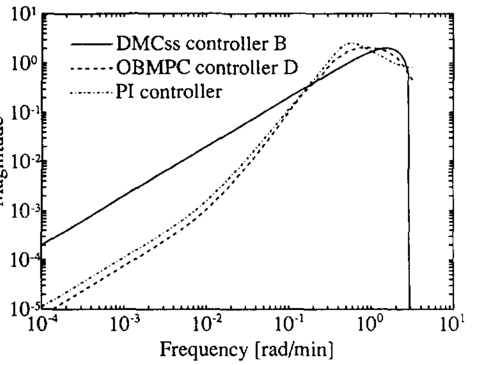

Fig. 9. Sensitivity function vs frequency for the SISO plant [equation (48)] with different controllers (Table 1).

The controller is tuned for dead beat control (controller B) and the *output* disturbance (solid curve) is rejected in one sampling interval, since the disturbance is in accordance with assumption A2. The response to the *input* disturbance (dashed curve) is extremely sluggish. The reason is that a step disturbance on the input results in a slow, ramp-like disturbance on the output. In this case assumption A2 does not hold and the output prediction used by the algorithm is incorrect which results in poor performance. The response cannot be improved by a different tuning since a dead beat controller gives the highest feedback gain of any choice of Γ, Λ, p and m for a given ΔT.

3.4. *Avoiding limitation 2*

An observer based MPC algorithm makes it possible to avoid the output step disturbance assumption A2 which causes limitation 2. To demonstrate this we compare the dead beat DMCss response (controller B) with the OBMPC response (controller D). We also included a PI controller in this comparison to demonstrate the performance of a very simple controller. The PI controller is tuned according to Ziegler–Nichols rules taking into account an extra delay of half the sampling time (Table 1).

Responses to a unit input disturbance are shown in Fig. 8. The DMCss response is sluggish, while the other controllers perform well. Actually, the PI controller is almost as good as OBMPC for this simple plant.

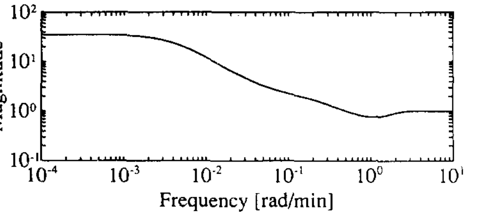

Fig. 10 Magnitude vs frequency plot of the (1,1)—RGA element of the distillation column [equation (49)]. The interactions are large at low frequencies $(\lambda_{1,1} \approx 35.1)$, but not at high frequencies $(\lambda_{1,1} \approx 1.0)$.

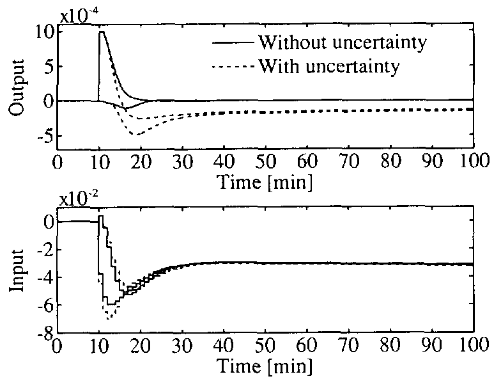

Fig. 11. Responses for the distillation column [equation (49)] with DMCss controller C (Table 1). A 0.001 step disturbance acts on y_D at $t = 10$. Uncertainty as defined in equation (51).

The difference between the controllers is also illustrated in Fig. 9, showing the sensitivity function vs frequency. The DMCss controller yields a sensitivity function with slope 1 for frequencies below the bandwidth. This shape of the sensitivity function is optimal for step disturbances and is a consequence of assumption A2. However, for "ramp-like" disturbances we need a stronger disturbance suppression at low frequencies. With a DMC controller this can only be achieved by increasing the bandwidth of the closed-loop system, since the shape of the sensitivity function is fixed (due to A2). The maximum bandwidth for a given ΔT is obtained by using $\Lambda = 0$ (dead-beat) and if the resulting suppression of low-frequency disturbances is not enough, then a smaller ΔT has to be used. With an OBMPC controller we may use a to adjust the disturbance suppression.

Figure 9 also shows that the sensitivity function for DMCss controller B goes to zero at $\omega = \pi / \Delta T$ rad/min. This is due to the dead-beat tuning and makes the controller very sensitive to high frequency uncertainty, e.g. dead time uncertainty.

3.5. *Limitation 3: Poor response for interactive MIMO plants*

In this section we will show that there are cases with model-plant mismatch when a DMC controller does not perform well even when the disturbance actually *is* a step acting on the output.

There is always a certain mismatch between a real process and a model. The mismatch can have various sources: uncertainty in the model parameters and the model structure, inaccuracies of the actuators and measurement devices, etc. Multivariable systems introduce a special problem here because

Table 1. Tuning parameters for controllers

	Type	Γ	Λ	ΔT (min)	m	p	n	a_{uij}	b_{uij}	α_i	$(f_a)_i$
A	DJC	1	0	1	3	4	30				
B	DMCss	1	0	1	3	4	—				
C	DMCss	$I_{2\times 2}$	$0.02I_{2\times 2}$	1	5	10	—				
D	OBMPC	1	0	1	3	4	30	1	$s_{n+1}-s_n$	1	0.5
E	OBMPC	$I_{2\times 2}$	$0.125I_{2\times 2}$	1	5	10	30	0.995	$s_{i,j,n+1}-s_{i,j,n}$	0.995	0.9
F	OBMPC	$I_{2\times 2}$	$0_{2\times 2}$	1	5	10	30	0.995	$s_{i,j,n+1}-s_{i,j,n}$	0.995	0.22

PI $C_{PI}(z)=\dfrac{K_c\left(\dfrac{\Delta T}{\tau_I}\right)}{(z-1)}+K_c$; $K_c=0.45$, $\tau_I=5.0$ min, $\Delta T=1$ min

the "gain" of a multivariable process varies not only with frequency, but also with "direction". Skogestad *et al.* (1988) show that if a plant is ill-conditioned irrespective of scaling, then the control performance is strongly affected by input uncertainty, in particular, when the controller is trying to invert the plant. The DMC controller is such a controller, especially, if the penalty weight on the input moves is low. Since there is always *some* input uncertainty, it should be clear that a DMC controller is potentially bad when used for an ill-conditioned plant.

3.5.1. MIMO example. We use a distillation column as an example process. The model is from Skogestad and Morari (1988) and is denoted "column A" in their paper. The column is described by the following equations:

$$\begin{bmatrix} \mathrm{d}y_D \\ \mathrm{d}x_B \end{bmatrix}=\begin{bmatrix} \dfrac{k_{11}}{1+\tau_1 s}\mathrm{e}^{-\theta s} & \left(\dfrac{k_{11}+k_{12}}{1+\tau_2 s}-\dfrac{k_{11}}{1+\tau_1 s}\right)\mathrm{e}^{-\theta s} \\ \dfrac{k_{21}}{1+\tau_1 s}g_L(s)\,\mathrm{e}^{-\theta s} & \left(\dfrac{k_{21}+k_{22}}{1+\tau_2 s}-\dfrac{k_{21}}{1+\tau_1 s}\right)\mathrm{e}^{-\theta s} \end{bmatrix} \times\begin{bmatrix} \mathrm{d}L \\ \mathrm{d}V_B \end{bmatrix}, \tag{49}$$

where $g_L(s)$ expresses the liquid flow dynamics:

$$g_L(s)=\frac{1}{[1(\theta_L/n_T)s]^{n_T}}, \tag{50}$$

θ_L is the overall liquid lag from the top to the bottom of the column. n_T in equation (50) should be equal to the number of trays in the column, but we use $n_T=5$ to avoid a model of unnecessary high order. Reflux L and boilup V_B are manipulated inputs, top composition, y_D and bottom composition x_B are controlled outputs. We use the following parameter values; $k_{11}=0.878$, $k_{12}=-0.864$, $k_{21}=1.082$, $k_{22}=1.096$, $\tau_1=194$ min, $\tau_2=15$ min, $\theta=1$ min, $\theta_L=2.46$ min and $n_T=5$. Skogestad and Morari do not include any specified delays in their model, instead they use a norm bounded uncertainty description to cover the effect of delays and other unmodeled high frequency dynamics. In equation (49) we assume the delays to be known and equal to 1 min for each transfer function. We do this only because known delays fit better into the MPC framework.

Skogestad *et al.* (1990) demonstrate that a frequency dependent relative gain array (RGA) (Bristol, 1966) is a useful tool to check how sensitive a plant is to input uncertainty. Figure 10 shows the (1,1) RGA element, $\lambda_{1,1}$, of the distillation column, as a function of frequency. $\lambda_{1,1}$ is high (35.1) at low frequencies but falls to one at higher frequencies. This shows that a DMC controller may have problems with low-frequency input uncertainty.

3.5.2. Effect of input uncertainty. We assume that there is 20% uncertainty in the input moves. From a singular value analysis, one can determine that the worst steady state effect is obtained when the uncertainties in ΔL and ΔV_B act in opposite directions (Skogestad *et al.*, 1988). In the simulations we use:

$$\Delta L_{\text{actual}}=1.2\Delta L_{\text{computed}} \text{ and}$$
$$\Delta V_{\text{Bactual}}=0.8\Delta V_{\text{Bcomputed}}. \tag{51}$$

Responses for controller C (Table 1) to a 0.001 step disturbance acting on y_D are shown in Fig. 11. Errors in the input gains lead to very sluggish disturbance rejection although the disturbance is in accordance with the DMC disturbance assumption. The reason for this slow settling is that the effect of the errors in the manipulated inputs is similar to the effect of input step disturbances. Note that this sluggish behavior cannot be improved by alternative choices of tuning parameters.

This can also be demonstrated in a plot of the singular values of the sensitivity function (Fig. 12). Both the solid curves (no uncertainty) have slope 1 which is a consequence of the disturbance assumption. They also lie close to each other, which shows that the sensitivity function is well-conditioned. Since the plant itself is ill-conditioned we can conclude that the controller is compensating for the directionality of the plant. Such a controller is basically inverting the plant and the system should be sensitive to input uncertainty. Indeed, this is the case as seen both from the simulation in Fig. 11 and from the large difference between solid and dotted

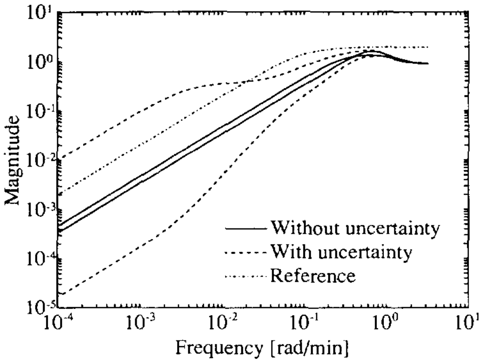

Fig. 12. Maximum and minimum singular values of the sensitivity function for the distillation column [equation (49)] with DMCss controller C (Table 1). Uncertainty as defined in equation (51).

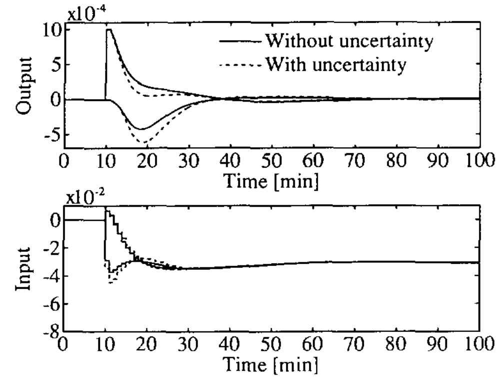

Fig. 13. Responses for the distillation column [equation (49)] with OBMPC controller E (Table 1). A 0.001 step disturbance acts on y_D at $t=10$. Uncertainty as defined in equation (51).

curves in Fig. 12. The dash-dot curve in Fig. 12 is included as a reference. It is an upper bound on sensitivity functions which achieve about 20 min closed-loop time constant and a maximum sensitivity peak of 2.

By comparing Figs 10 and 12 we see that there is an excellent agreement between the predicted effect of uncertainty, based on the RGA-plot, and the actual effect seen in the sensitivity plot. [However, this sensitivity plot is only showing how the control performance deteriorate for this specific input error (+20% in L and −20% in V_B), and there may be an even larger effect for other error combinations of the same magnitude, i.e. the plot is not necessarily showing the "worst case" of a norm-bounded uncertainty.]

3.6. *Avoiding limitation 3*

There are two different ways to deal with the problem caused by input uncertainty demonstrated in Fig. 12:

1. Use a controller that does not correct for the directionality of the plant.
2. Increase the gain at those frequencies where the disturbance suppression is poor.

The first method is suggested in Skogestad *et al.* (1988). It gives a controller with somewhat sacrificed nominal performance, but the performance is much less sensitive to uncertainty because the controller does not correct for directionality.

The second approach will work if the uncertainty only causes problem at low frequencies. With this approach the controller is still sensitive to uncertainty, but this is counteracted by increasing the controller gain at low frequencies to make the nominal response much better than what is nominally needed.

We will now demonstrate the two approaches, using OBMPC controllers E and F (Table 1). In both cases the residual plant dynamics for each input–output pair is approximated by $b_{ij}/(z-a_{ij})$ where $b_{ij}=s_{i,j,n+1}-s_{i,j,n}$ and $a_{ij}=0.995$, i.e. a first-order response with a time constant approximately equal to 200 min and a gain determined at the "truncation" step. The disturbance is assumed to have the same dynamics as the plant, i.e. $a_i=0.995$.

Figures 13 and 14 show responses for OBMPC controllers E and F, respectively, for the same output disturbance as used in Fig. 11. The OBMPC controllers perform well despite the uncertainty [equation (51)] and suppress the disturbance much faster than DMCss does (Fig. 11). The response of controller E is nearly unaffected by the uncertainty. Controller F yields an almost perfectly decoupled response when there is no uncertainty (i.e. x_B is not affected by the disturbance in y_D), while the response with input error clearly demonstrates interaction between the two loops.

In the case of an *input* disturbance the difference between the OBMPC controllers and the DMCss

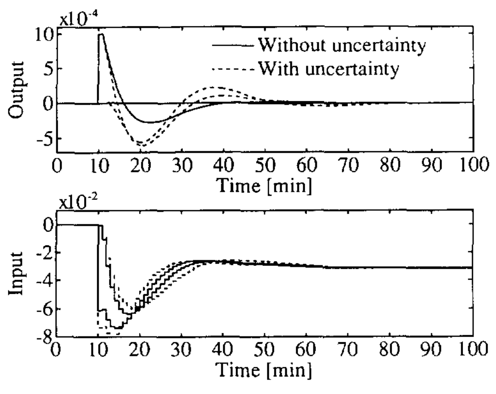

Fig. 14. Responses for the distillation column [equation (49)] with OBMPC controller F (Table 1). A 0.001 step disturbance acts on y_D at $t=10$. Uncertainty as defined in equation (51).

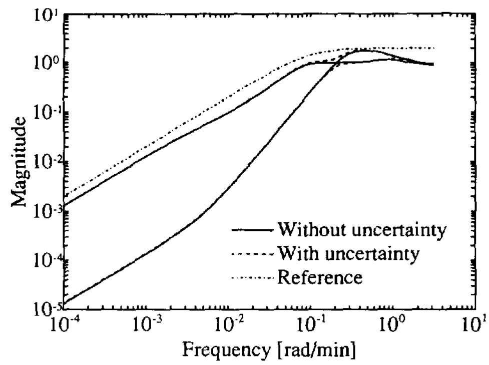

Fig. 15. Maximum and minimum singular values of the sensitivity function for the distillation column [equation (49)] with OBMPC controller E (Table 1). Uncertainty as defined in equation (51).

controller would be even larger, because of limitation 2.

The sensitivity plot for controller E is shown in Fig. 15. This controller is using a high input weight and a large disturbance-to-noise ratio. The plot shows that this controller does *not* try to invert the plant; the solid curves (no uncertacnty) do not lie close to each other. We can also conclude that it is insensitive to uncertainty since the dotted curves [uncertainty defined in equation (51)] lie close to the solid curves.

Figure 16 shows the sensitivity plot for controller F. This controller has no weight on the inputs (dead beat K_{MPC}) but is tuned for substantial measurement noise ($f_a = 0.22$). From the plot we see that this controller *is* inverting the plant for frequencies above $\omega \approx 0.01$ rad/min. Below this frequency it is still trying to invert the plant, but it does not succeed, since the true plant and the model with approximated residual dynamics are slightly different. This controller is sensitive to input uncertainty in the sense that low frequency controller gain is so

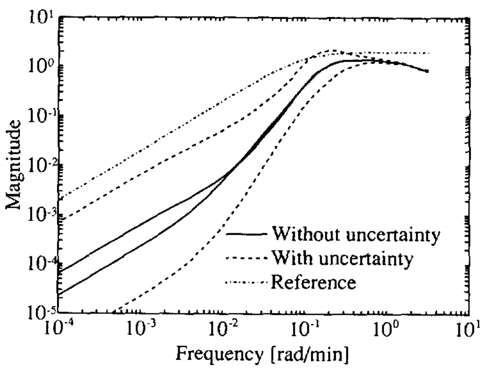

Fig. 16. Maximum and minimum singular values of the sensitivity function for the distillation column [equation (49)] with OPMPC controller F (Table 1). Uncertainty as defined in equation (51).

high that even *with* uncertainty the performance is satisfactory, except over a short frequency range.

Although controller F yields satisfactory performance, we may conclude that the plant is rather sensitive to input uncertainty also at frequencies above the bandwidth. Thus, the best tuning approach for this plant is approach 1 above, used for tuning controller E.

4. DISCUSSION

We have studied feedback limitations of unconstrained DMC with a quadratic objective function [equation (13)]. There are several variants of DMC: "original" DMC (Cutler and Ramaker, 1980), DMC with least squares satisfaction of input constraints (Prett and Gillette, 1980), DMC with constrained linear programming optimization (LDMC) (Morshedi *et al.*, 1985), DMC with constrained quadratic programming optimization (QDMC) (García and Morshedi, 1986). These variants use different optimizers but the predictor is the same for all of them. Both the limiting assumptions (A1 and A2), which we have studied, are implicit in the predictor and will not be avoided by modifying the optimizer, so the results in this paper hold for all these algorithms. The results also carry over to the general case with constraints, since the issue of constraints only affects the optimizer.

The requirement $S_n \approx S_{n+1}$ of assumption 1, can be avoided within the DMC framework (as defined by Fig. 1) by using a general state space model instead of a step response model. However, the plant still has to be stable to ensure internal stability. Assumption A2 cannot be avoided unless the *constant* matrix $\mathcal{I}$ is exchanged with a transfer function *states*. This is most clearly seen in the DMCss algorithm where the input to $\mathcal{I}$ is $\hat{y}(k)$. $\hat{y}(k)$ cannot be "filtered" by $\mathcal{I}$ since it has no states, no knowledge of previous measurements. Using a gain different from 1 in $\mathcal{I}$ would not filter $\hat{y}(k)$, but rather introduce a steady state offset.

In this paper we have excluded the feedforward part of the algorithms, although feedforward control is a standard feature of MPC. It is simple to include feedforward in the algorithms by introducing measured disturbances as inputs to the predictor. However, this does not affect our results, the feedback limitations are still present.

In the following we discuss some of the results and our choice of example processes and controller tunings.

The SISO example [Equation (48)] has a time constant much larger than the time delay. This parameter choice is made on purpose to demonstrate limitations L1 and L2, since they are especially important when the time constant of the open-loop plant is large, compared to the desired closed-loop time constant. In the SISO simulations we use controllers where the optimizer is tuned for dead-beat control. The reason for this tuning is that we want to produce clear illustrative simulations where the controller action is easy to understand. In real applications one should always use a nonzero input weight Λ to achieve some robustness to noise and high-frequency model–plant mismatch.

All the simulations presented in this paper are *without* measurement noise. Again the reason is that we want to show clear illustrative simulations. (We have also performed the simulations with noise and it does not change our results.)

We have demonstrated that a truncated step response model may cause severe model–plant mismatch, both at high and low frequencies. Low-frequency mismatch is not critical as long as the "sign" of the process is correct, but high-frequency mismatch may yield an unstable system (Fig. 3). With the OBMPC controller truncation is avoided. In controller D we use a rough approximation of the residual dynamics (an integrator) to show that even this approximation is better than truncation.

The sensitivity plots for the distillation column (Figs 12, 15 and 16) show singular values of the sensitivity function for no uncertainty and for *one specific* case of uncertainty. However, we have used the structural singular value, μ, to check that the controllers will perform well also for other cases of uncertainty (see Skogestad *et al.*, 1988, for instance).

We have not put any effort into finding the best approximation of the residual dynamics, although that may have improved the OBMPC responses further, instead the simplest choices of A_u and B_u are used throughout the paper. Hovd *et al.* (1991) discuss how to choose A_u and b_u in an optimal fashion.

5. CONCLUSIONS

We have shown that there are situations where the feedback performance of DMC is poor irrespective of tuning. This poor prformance is due to the two assumptions (A1 and A2) made in the *predictor* part of the algorithm. This explains why the performance cannot be improved by different tuning—different tuning only affects the *optimizer* part of the algorithm.

The OBMPC algorithm by Lee *et al.* (1994) allows us to avoid the limitations of DMC and still preserve all the attractive properties of the DMC algorithm.

Acknowledgement—Partial support from the National Science Foundation is gratefully acknowledged.

NOMENCLATURE

A = State matrix
$\mathcal{A}$ = Disturbance state matrix, equation (37)
a = Element in state matrix
B = Input matrix
b = Element input matrix
C = Output matrix
f_a = Tuning parameter in filter gain K
f_b = Tuning parameter in filter gain K
$\mathcal{F}$ = Equation (12)
K = Kalmal filter gain
K_c = PI-controller gain
K_{MPC} = MPC feedback gain
k_{ij} = Gain, equation (49)
L = Reflux
M = Matrix in step response model, equation (3)
m = Input horizon
N = Matrix in step response model, equation (4)
n = Model horizon
n_u = Number of inputs
n_y = Number of outputs
p = Prediction horizon
$\mathcal{R}$ = Set point vector
S = Step response coefficient matrix, equation (3)
s = Step response coefficient
T = Disturbance input matrix, equation (25)
ΔT = Sampling time
t = Time (min)
$\mathcal{U}$ = Optimal control sequence
u = manipulated input
V = Noise covariance matrix
V_B = Boilup
v = Measurement noise
W = Disturbance covariance matrix
w = Disturbance
x = State variable
x_B = Bottom composition (kmol/kmol)
Y = Output vector
$\hat{Y}$ = Dynamic states of DMC predictor
$\bar{Y}$ = Dynamic states of OBMPC predictor
$\mathcal{Y}$ = Predicted output vector
y = Controlled output
$\hat{y}$ = Measured outut
y_D = Top composition (kmol/kmol)
z = Shift operator

Greek symbols

α = Parameter in disturbance model, equation (37)
Γ = Output weighting matrix, equation (13)
$\Delta = \Delta u(k) = u(k) - u(k-1)$
θ = Time delay (min)
Λ = Input weighting matrix, equation (13)
λ_{11} = (1,1) RGA element
τ_I = PI-controller integral time constant (min)
τ_1, τ_2 = Time constant (min), equation (49)
ω = Frequency (rad/min)
ω_B = Closed-loop bandwidth (rad/min)

Abbreviations

DMC = Dynamic matrix control
DMCss = Dynamic matrix control with state space model
MIMO = Multi-input–multi-output
MPC = Model predictive control
OBMPC = Observer based model predictive control
QDMC = quadratic programming dynamic matrix control
RGA = Relative gain array
SISO = Single input–single output

REFERENCES

Bristol E. H. On a new measure of interactions for multivariable process control. *IEEE Automat. Control* **AC-11**, 133–134 (1966).

Cutler C. R. and B. L. Ramaker, Dynamic matrix control—a computer control algorithm. *Proc. Joint Automatic Control Conf.*, San Francisco, CA, Paper WP5-B (1980).

García C. E. and A. M. Morshedi, Quadratic programming solution of dynamic matrix control (QDMC). *Chem. Engng Commun.* **46**, 73–87 (1986).

García C. E., D. M. Prett and M. Morari, Model predictive control: theory and practice—a survey. *Automatica* **25**, 335–348 (1989).

Hovd M., J. H. Lee and M. Morari, Model requirements for model predictive control. *European Control Conf.*, Grenoble, France (1991).

Lee J. H., M. Morari and C. E. García, State-space interpretation of model predictive control. *Automatica* **30**, 707–717 (1994).

Li S., K. Y. Lim and D. G. Fisher, A state space formulation for model predictive control. *AIChE Jl* **35**, 241–249 (1989).

Middleton R. H. Trade-offs in linear control system design. *Automatica* **27**, 281–292 (1991).

Morshedi A. M., C. R. Cutler and T. A. Skrovanek, Optimal solution of dynamic matrix control with linear programming techniques (IDMC). *Proc. Am. Control Conf.*, Boston, MA, pp. 199–208 (1985).

Prett D. M. and C. E. García, *Fundamental Process Control.* Butterworths, Stoneham, MA (1988).

Prett D. M. and R. D. Gillette, *Proc. Joint Automatic Control Conf.*, San Francisco, CA (1980).

Skogestad S. and M. Morari, Understanding the Dynamic Behavior of Distillation Columns. *AIChE Jl* **33**, 1620–1635 (1988).

Skogestad S., M. Morari and J. C. Doyle, Robust control of ill-conditioned plants: high-purity distillation. *IEEE Automat. Control* **33**, 1092–1105 (1988).

Skogestad S., P. Lundström and E. W. Jacobsen, Selecting the best distillation control configuration. *AIChE Jl* **36**, 753–764 (1990).