Presented at ECC'91, Grenoble, France, July 2-5 1991.

Limitations of Dynamic Matrix Control

Petter Lundström*, Jay H. Lee†, Manfred Morari‡
Chemical Engineering 210-41
California Institute of Technology
Pasadena, CA 91125, USA
Sigurd Skogestad
Chemical Engineering
University of Trondheim, NTH
N-7034 Trondheim, Norway

Abstract

Dynamic Matrix Control (DMC) is based on two assumptions which are limiting 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 a measured and a predicted output can be modeled as a step disturbance acting on the output.

We highlight the DMC limitations and show that a Model Predictive Control (MPC) algorithm based on an observer does not suffer from these 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 ([1], [2]). DMC allows setpoint changes to be "announced" in advance and it facilitates feedforward and constraint handling [2]. These properties do naturally lead to improved performance. However, the *feedback* properties of a DMC controller are limited by two very restrictive assumptions made in the algorithm. It is assumed that:

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

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

Lee *et al.* [4] have recently shown that unconstrained DMC is equivalent to linear quadratic optimal output feedback, under the assumption of integrated white noise disturbances at the output and no measurement noise. They also demonstrate how to represent an unstable plant by use of a step response model.

In this paper we study unconstrained DMC, but our results do carry over to the general case with constraints. We use the results from Lee *et al.* [4] to point out the assumptions made in DMC and to show that they lead to the following limitations in terms of feedback performance:

L1 The sampling interval is determined by the (dominant) time constant of the plant and by computer hardware, although it should be determined by the high frequency behavior of the plant or by constraints in the measurement devices.

L2 Poor performance for "ramp-like" disturbances (on the outputs). In particular, this occurs for input disturbances for plants with large time constants.

L3 Poor performance for plants with strong interactions.

In addition, there is the obvious limitation that the plant has to be stable. In this paper, we illustrate these limitations, and also discuss how we may reformulate the algorithm to avoid them.

2 Model Predictive Control

A detailed description of MPC is not in the scope of this paper, but we do present some of the basic concepts and ideas. Readers not familiar with MPC are recommended to consult for instance García *et al.* [3]. The content of this section is presented in more detail by Lee *et al.* [4].

2.1 The DMC Algorithm

We restrict this presentation to a single input single output system in order to make it somewhat simpler.

*Present address: Chemical Engineering, University of Trondheim, NTH, N-7034 Trondheim, Norway.

†Present address: Department of Chemical Engineering, Auburn University, Auburn, AL 36849-5127, USA

‡Author to whom correspondence should be addressed: phone (818)356-4186, fax (818)568-8743, e-mail MM@IMC.CALTECH.EDU

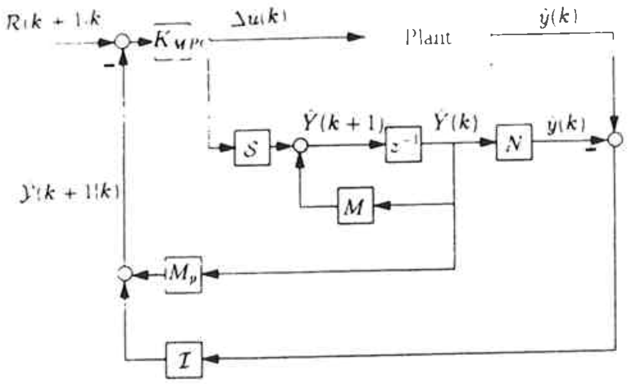

Figure 1: Block diagram of a DMC controller.

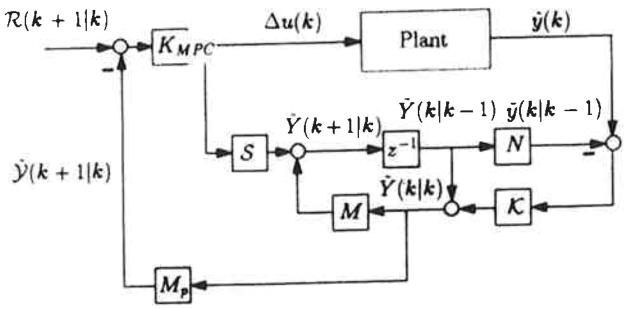

Figure 2: Block diagram of an observer based MPC controller.

We also exclude feedforward of the same reason, although feedforward is a standard feature in DMC.

In DMC [1] the plant is assumed to be stable and the controller is using a step response model to predict future behavior (assumption A1). Sending a unit step input into the plant at time 0 gives a sequence of step response coefficients:

$$[S_1\ S_2\ S_3 \ldots] \tag{1}$$

where the k^{th} element is the output at time k. For a stable plant this sequence will reach a constant value, $S_n = S_{n+1} = \cdots$, after a sufficiently large number of coefficients. The step response model of the system can be represented in the following standard state space form.

$$\begin{aligned} \tilde{Y}(k+1) &= M\tilde{Y}(k) + \mathcal{S}\Delta u(k) \\ \tilde{y}(k) &= N\tilde{Y}(k) \end{aligned} \tag{2}$$

where

$$\tilde{Y}(k) = [\tilde{y}(k)\ \tilde{y}(k+1)\ \ldots\ \tilde{y}(k+n-1)]^T \tag{3}$$

$$M = \left.\begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix}\right\} n \tag{4}$$

$$\mathcal{S} = [S_1\ S_2\ \ldots\ S_{n-1}\ S_n]^T \tag{5}$$

and

$$N = [1\ 0\ \ldots\ 0] \tag{6}$$

The DMC algorithm is illustrated in Fig.1. The model in Eq.(2) is used to compute $\tilde{y}(k)$, and this predicted output is compared with the measured output, $\hat{y}(k)$. The assumed step disturbance $(\hat{y}(k) - \tilde{y}(k))$ is projected p steps into the future by

$$\mathcal{I} = [1\ \ldots\ 1]^T. \tag{7}$$

$\tilde{Y}(k)$ is shifted one step forward by matrix M_p, the first p rows of M. By adding these two projections we get:

$$\tilde{\mathcal{Y}}(k+1|k) = M_p\tilde{Y}(k) + \mathcal{I}[\hat{y}(k) - \tilde{y}(k)] \tag{8}$$

$\tilde{\mathcal{Y}}(k+1|k)$ is the predicted outputs if no future input moves were made. $M_p\tilde{Y}(k)$ is the effect of past input moves on future outputs and $\mathcal{I}[\hat{y}(k) - \tilde{y}(k)]$ is the assumed effect of unmeasured disturbances. $\tilde{\mathcal{Y}}(k+1|k)$ is then compared with $\mathcal{R}(k+1|k)$, the desired output trajectory. In unconstrained DMC the difference between desired and predicted trajectories is finally multiplied by K_{MPC} (defined below, Eq.10) and the new input move $\Delta u(k)$ is found. In the constrained case, K_{MPC} is exchanged with an optimization routine.

In this paper we use the following (QDMC) objective function [2]:

$$J = \min_{\Delta\mathcal{U}(k|k)} \{\|\Gamma[\tilde{\mathcal{Y}}_m(k+1|k) - \mathcal{R}(k+1|k)\|^2 + \|\Lambda\Delta\mathcal{U}(k|k)\|^2\} \tag{9}$$

where Γ and Λ are weighting matrices and are usually chosen to be diagonal. This objective function leads to

$$K_{MPC} = [\mathrm{I}\ 0\ \ldots\ 0](\mathcal{S}^{uT}\Gamma^T\Gamma\mathcal{S}^u + \Lambda^T\Lambda)^{-1}\mathcal{S}^{uT}\Gamma^T\Gamma \tag{10}$$

where

$$\mathcal{S}^u = \begin{bmatrix} S_1 & 0 & \ldots & 0 \\ S_2 & S_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ S_m & S_{m-1} & \ldots & S_1 \\ \vdots & \vdots & & \vdots \\ S_p & S_{p-1} & \ldots & S_{p-m-1} \end{bmatrix} \tag{11}$$

2.2 Observer Based MPC

In Fig.2 the well known state-observer state-feedback controller is shown. The states of this controller are updated by *direct* use of the measurement, through a filter $\mathcal{K}$, and not only by the input moves as in Fig.1.

Lee *et al.*, [4] showed that if we use the step response model from Eq.(2) in the structure of Fig.2 and make the same disturbance assumptions as in DMC (a step disturbance acting on the output and no measurement noise), then the optimal $\mathcal{K} = \mathcal{I}$, and the controllers in Fig.1 and Fig.2 are equivalent. This means that a DMC controller is an *optimal* state observer state feedback controller for these very special assumptions. (Lee *et al.*, [4]).

Lee *et al.* also showed how an unstable processes can be represented by a step response model. A step response model of an integrating system may be written on the form of Eq.(2) with M replaced by:

$$M^I = \left.\begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix}\right\} n \tag{12}$$

The last row of M^I shows that the step response has a constant slope at a sufficiently large n:

$$S_n - S_{n-1} = S_{n+1} - S_n \tag{13}$$

which gives

$$S_{n+1} = 2S_n - S_{n-1} \tag{14}$$

Assuming a *ramp* disturbance on the output and white measurement noise, leads to an observer with the optimal $\mathcal{K}$:

$$\mathcal{K} = f_a \mathcal{I} + f_b \begin{bmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ n-1 \end{bmatrix} \tag{15}$$

where f_a and f_b are found from a Riccati equation. However, [4] suggest that f_a may be used as a tuning parameter, reflecting the relative size of noise and ramp disturbance, and f_b computed from

$$f_b = 2 - f_a - 2\sqrt{1 - f_a} \tag{16}$$

This choice of f_a and f_b gives an observer with real eigenvalues and a filter $\mathcal{K}$ close to the optimal one. The advantage with this method is that $\mathcal{K}$ is found *without* solving a Riccati equation.

Both the controller in Fig.1 and Fig.2 may use a state space representation which is not based on the step response model but on the "A,B,C and D"-matrices. We will use "DMC" to denote a step response based controller of the form in Fig.1. "DMCss" denotes that the "A,B,C,D" representation is used. With "MPCi" we will mean an observer based controller with an integrating step response model. "MPCss" denotes an observer based controller with the "A,B,C,D" representation.

3 Example Processes

SISO example

We use a simple plant described by the following continuous first order plus dead time model.

$$P(s) = \frac{100}{100s + 1} e^{-s} \tag{17}$$

Note that the dominant time constant is much larger than the time delay. We do not claim this to be a typical plant controlled by DMC. Our reason for using it is simply that it is highlighting the limitations with the algorithm.

MIMO example

The multivariable example process is a distillation column. The model is from Skogestad and Morari [7] and is denoted "column A" in their paper. The column is described by the following equation:

$$\begin{bmatrix} dy_D \\ dx_B \end{bmatrix} = \begin{bmatrix} \frac{g_{11}}{1+\tau_1 s} e^{-\theta s} & \left(\frac{g_{11}+g_{12}}{1+\tau_2 s} - \frac{g_{11}}{1+\tau_1 s}\right) e^{-\theta s} \\ \frac{g_{21}}{1+\tau_1 s} g_L(s) e^{-\theta s} & \left(\frac{g_{21}+g_{22}}{1+\tau_2 s} - \frac{g_{21}}{1+\tau_1 s}\right) e^{-\theta s} \end{bmatrix} \begin{bmatrix} dL \\ dV \end{bmatrix} \tag{18}$$

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

$$g_L(s) = \frac{1}{(1 + (\theta_L/n_T)s)^{n_T}} \tag{19}$$

Reflux, L, and boilup, V, are manipulated inputs, top composition, y_D, and bottom composition, x_B, are controlled outputs. We use the following parameter values; $g_{11} = 0.878, g_{12} = -0.864, g_{21} = 1.082, g_{22} = 1.096, \tau_1 = 194$ min, $\tau_2 = 15$ min, $\theta = 1$ min. $\theta_L = 2.46$ min and n_T=5. Skogestad and Morari [7] 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 Eq.(18) we assume the delays to be known and equal to 1 minute for each transfer function. We do this only because known delays fit better into the MPC framework.

Simulations

In the simulations we use as "plant", a discrete version of Eq.17/18 with sampling time equal to 0.1 minutes. The sampling time of the controller and its internal model (or observer) is denoted ΔT. Unless otherwise stated we shall use a K_{MPC} tuned for dead beat control, i.e. there is no penalty on input moves ($\Gamma = 1$, $\Lambda = 0$, $m = p = 2$) and $\Delta T = 1$. We have chosen this tuning for two reasons; it generates clear and illustrative time responses; it gives maximum feedback gain for a given ΔT.

4 Limitations of DMC

4.1 Limitation 1: Poor bandwidth if large dominant time constant

DMC is using a stable step response model to represent the plant (assumption A1). In practice this representation may cause problems because sometimes an excessive number of coefficients is needed for good performance, but of computational reasons, there is a practical limit on the maximum number of coefficients that can be used.

Consider the SISO model in Eq.17. The effect of truncation error is demonstrated in Fig.3. The truncated step response causes an erroneous prediction n sampling intervals after the disturbance occurred. The error in this simulation is so large that it eventually

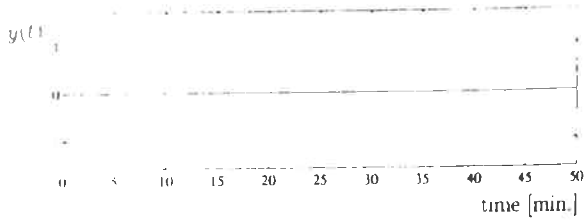

Figure 3: Response to a unit step disturbance at $t = 10$, acting on y. Solid curve: DMCss which gives no truncation; Dashed curve: DMC with a step response model truncated at $n = 30$.

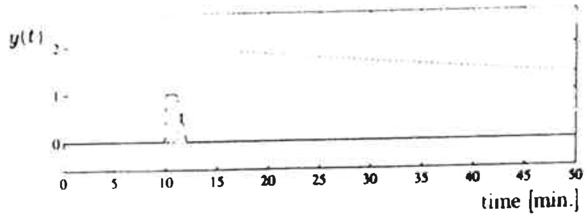

Figure 4: Response to a unit step at $k = 10$ acting on the output (solid curve) and the input (dashed curve), DMCss used in both simulations.

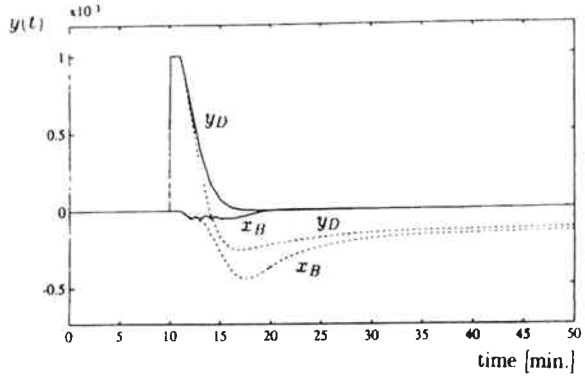

Figure 5: Response to a step disturbance acting on y_D at $k = 10$. DMCss with $\Gamma = 100 * I$, $\Lambda = I$, $m = 5$, $p = 10$ and $\Delta T = 1$ is used. Solid curves: No input uncertainty. Dashed curves: 20% input uncertainty.

will lead to instability. This shows that it is necessary that $S_n \approx S_{n+1}$ when a stable step response model is used to represent the plant.

An other requirement for good performance is that the sampling time, ΔT, is 5 to 10 times shorter than the desired closed loop time constant. The distillation column in Eq.18 has a large dominant time constant ($\tau_1 = 194$ min). In order to achieve a closed loop time constant of 20 min a sampling interval of not more than 4 min is required. This would demand about 150 step response coefficients, as compared to 30 which is commonly used in DMC.

The conclusion is that a stable step response model imposes a tradeoff between truncation error and sampling time. Since the truncation error can not be too large we have to accept infrequent sampling, and thereby the closed loop bandwidth is determined by the open loop dominant time constant.

4.2 Limitation 2: Poor response for ramp-like disturbances

The DMC algorithm is based on the assumption that unmeasured disturbances act as steps on the outputs (assumption A2). Figure 4 shows the responses for the SISO process (Eq.17). The output disturbance (solid curve) is rejected in one sampling interval, as expected since the disturbance is in accordance with the assumption (A2) and we are using a dead beat controller. However, the response to the input disturbance (dashed curve) is very sluggish. Remember that a dead beat controller gives the highest feedback gain of any choice of Γ, Λ, p and m, and the response can thereby not be improved by a different tuning. Rather, this sluggish response is an effect of the disturbance assumption built into the predictor part of the algorithm.

4.3 Limitation 3: Poor response for interactive MIMO plants

In this section we show that there are situations in which a DMC controller does not perform well even if the disturbance actually *is* a step acting on the outputs.

There is always a certain mismatch between a real process and a model. The mismatch is caused not only by uncertain model parameters or model structure, but also by uncertainty in actuators and measurement devices. So although we assume that the actual *model* is perfect, there will still be some uncertainty. MIMO systems, as opposed to SISO systems, introduce a special problem here because the gain of a multivariable process does not only vary with frequency, but also with "direction". If a plant is ill-conditioned irrespectively of scaling, then the control performance is strongly affected of input uncertainty, especially if the controller is trying to invert the plant [8]. The DMC controller is such a controller, especially if the penalty weight on input moves is low. Since there always is some input uncertainty, it should be clear that a DMC controller is potentially bad when used with an ill-conditioned plant.

4.3.1 Effect of Input Uncertainty

We use the distillation model in Eq.18. In the simulations with uncertainty we use $\Delta L_{\text{actual}} = 1.2\Delta L_{\text{computed}}$ and $\Delta V_{\text{actual}} = 0.8\Delta V_{\text{computed}}$.

Responses to a step disturbance acting on y_D at $k = 10$ are shown in Fig.5. Uncertain inputs leads to extremly sluggish disturbance rejection although the disturbance is in accordance with the DMC disturbance assumption. The reason to this is that the actual input moves do not have the predicted effect on the outputs since the prediction is based on computed and not actual input moves.

This can also be demonstrated in a plot of the singular values of the sensitivity function as a function of frequency. Such a plot is shown in Fig.6. The solid

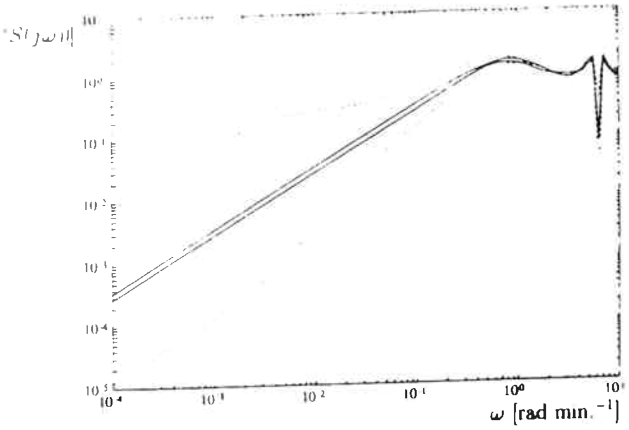

Figure 6: Sensitivity function vs. frequency for the DMC controller. Solid curves: No input uncertainty. Dashed curves: 20% input uncertainty.

curves (no uncertainty) lie close to each other, which shows that the sensitivity function is well-conditioned. Since the plant in it self 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 from the great difference between solid and dotted curves. The damping is severely affected by the uncertainty. The actual minimum damping, $\overline{\sigma}(S_{\text{actual}})$, is less than 10% for frequencies over ω = 0.002rad/min. This means that it takes very long time to reject disturbances acting in the direction of the minimum damping.

4.4 Avoiding limitations 1 and 2

We have already stated that limitation 1 may be avoided by using a state space model in the DMC algorithm. With an observer based algorithm we may use an ***integrating step response model*** to represent a ***stable*** plant and thereby allow a short sampling interval. The integrating model will give a large mismatch at low frequencies, but at high frequencies the agreement is good. With an observer based algorithm we may use a high feedback gain to correct for the low frequency mismatch. Such a correction can not be achieved with the DMC algorithm due to the disturbance assumption.

An observer based algorithm does also make it possible to avoid the step disturbance assumption which is causing limitation 2. We will demonstrate this by comparing the DMCss responses with responses from MPCi and MPCss. We have also included a PI controller to demonstrate the performance of a very simple controller. Figure 7 shows the responses to a unit step input disturbance at k = 10. The MPCi controller is using f_a = 0.8 (Eq.15) and f_b is computed using Eq.(16). MPCss is tuned for unit step input disturbance, and is therefore giving dead-beat response. The PI controller is tuned according to the Ziegler-Nichols criteria (K_c = 0.45 and τ_I = 5.0). The intention with this simulation is not to find "the best simulation", but to demonstrate that DMCss suffer from limitation 2, while MPCi and MPCss do not.

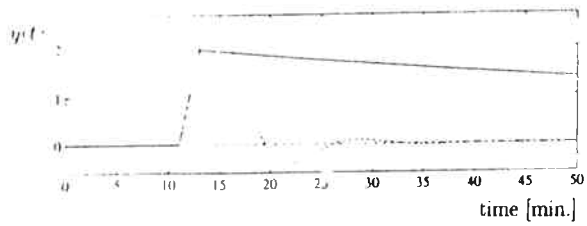

Figure 7: Responses to a unit step input disturbance at k = 10. Solid curve: DMCss, Dotted curve: MPCi, Dashed-dotted curve:MPCss, Dashed curve: PI-controller.

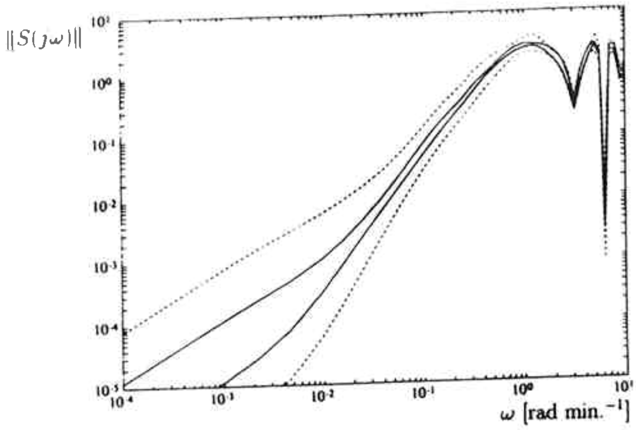

Figure 8: Sensitivity function vs. frequency for the observer based controller using an integrating step response model and a filter for ramp disturbances acting on plant outputs. Solid curves: No input uncertainty. Dashed curves: 20% input uncertainty.

4.5 Avoiding limitation 3

There are two different ways to deal with the problem demonstrated in Fig.6:

1 Use a controller that does not correct for the directionality of the plant.

2 Increase the gain at those frequencies where the damping is low.

Fig. 8 shows the sensitivity function for MPCi with n = 30, f_a = [1 1] and f_b from Eq.16. A quantitative comparison of this plot and the plot in Fig. 6 is not fair since they have different bandwidths and different peak values for frequencies above the bandwidth. However, a qualitative comparison is very illustrative. MPCi is not insensitive to uncertainty. Instead it is using a high gain at low frequencies to compensate for the effect of uncertainty. If the plant had a high RGA over all frequencies, i.e. was sensitive to input uncertainty also at high frequencies, this approach could not be used. Note that the nominal sensitivity function is well-conditioned for high frequencies but not for low. The controller is trying to invert the plant, but it does not succeed at low frequencies because of model-plant-mismatch.

Simulations using DMC, MPCi and MPCss are in agreement with the results obtained from sensitivity plots.

5 Discussion

In this paper we have studied limitations of the feedback properties of unconstrained DMC. We have not explicitly considered constraints. However, constraints will only affect the optimization part of the algorithm, K_{MPC} will be replaced by a quadratic (QDMC) or linear (LDMC) programming routine. All the limitations are caused by the DMC predictor, which means that they are present also in constrained DMC (QDMC, LDMC), since these algorithms are using the same type of predictor.

In DMC the mismatch between measured and predicted outputs is modeled as a step disturbance acting on the outputs. We have called this "the step disturbance assumption" although it is not really an assumption. Instead it is *imposed* by the actual structure of the algorithm and can not be avoided in DMC. This means that DMC can not use knowledge about the nature of unmeasured disturbances. García and Morshedi (1984/86) are not clear on this point. They say that this "assumption" is made "in the absence of any additional knowledge" of unmeasured disturbances, as if there were a real option to choose another assumption.

We use an observer based predictor in order to avoid the limitations imposed by the step disturbance assumption. This modification does only affect the predictor part of the algorithm, and can therefore easily be incorporated in a controller which handles constraints. The other good properties with DMC, setpoint scheduling and feedforward, are also preserved in the modification.

The observer may use a stable step response model, an integrating step response model or a state space model of the plant. If a stable step response model is used, it will impose the same limitations on ΔT as in the DMC structure.

There are some advantages with using a step response model in the observer. The model is physically intuitive and gives a Kalman filter which also can be intuitively understood. Even more important, a "close-to-optimal" filter can be computed without solving a Riccati equation. A step response model is also easy to obtain from simple experiments.

The DMC algorithms in Cutler and Ramaker (1979/80) ("original" DMC), Prett and Gillette (1979/80) (DMC with least squares satisfaction of input constraints), Morshedi *et al.* (1985) (linear programming optimization, LDMC) and García and Morshedi (1984/86) (quadratic programming, QDMC) do all compute $\tilde{y}(k+1|k)$ as described above. That is, they have the same prediction part as standard DMC. The results in section 3 show the severe limitations imposed by using this predictor.

6 Conclusions

The predictor part of the DMC algorithms is based on the assumption that the difference between measured and predicted outputs are caused by step disturbances on the outputs. Essentially, what the predictor does is to adjust the bias of the predicted outputs at each sampling interval. This makes the predictor simple to understand, but at the same time it seriously restricts the prediction capabilities and thereby the overall control behavior. In contrast, with a Kalman filter the difference between measured and predicted outputs are used to update all the states. This gives a much more flexible algorithm, and the prediction capabilities, at least when applied for control, may be markedly improved.

Standard DMC uses a step response model of the plant to predict the effect of past input moves on future outputs. The control performance is degenerated if this model is truncated.

References

[1] Cutler, C.R. and B.L. Ramaker (1979/80). *AIChE National Meeting*, Houston, Texas (1979); also *Proc. Joint Automatic Control Conf.*, San Francisco, California, Paper WP5-B (1980).

[2] García, C.E. and A.M. Morshedi (1984/86). *Proc. Am. Control Conf.*, San Diego, California (1984); also *Chem. Eng. Commun.*, **46**, 73-87 (1986).

[3] García, C.E., D.M. Prett and M. Morari (1989). *Automatica*, **25**, 3, 335-348.

[4] Lee, J.H., M. Morari and C.E. García (1990). Submitted to *Automatica*.

[5] Morshedi, A.M., C.R. Cutler and T.A. Skrovanek (1985). *Proc. Am. Control Conf.*, Boston, Massachusetts, pp. 199-208.

[6] Prett, D.M. and R.D. Gillette (1979/80). *AIChE National Meeting*, Houston, Texas (1979); also *Proc. Joint Automatic Control Conf.*, San Francisco, California (1980).

[7] Skogestad, S. and M. Morari (1988). *AIChE Journal*, **33**, 10, 1620-1635.

[8] Skogestad, S., M. Morari and J.C. Doyle (1988). *IEEE Automat. Control*, **33**, 1092-1105. (Also see corrections to μ-optimal controller in **34**, 6, 672).