Chemical Engineering Science, Vol. 47, No. 4, pp. 705–720, 1992.
Printed in Great Britain.

0009–2509/92 $5.00 + 0.00

MODEL-PREDICTIVE CONTROL OF CHEMICAL PROCESSES

JOHN W. EATON and JAMES B. RAWLINGS[†]
Department of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712, U.S.A.

(*Received for publication* 6 *August* 1991)

Abstract—This paper discusses model-predictive control, a scheme in which an open-loop performance objective is optimized over a finite moving time horizon. Model-predictive control is shown to provide performance superior to conventional feedback control for nonminimum phase systems or systems with input constraints when future set points are known. Stabilizing unstable linear plants and controlling nonlinear plants with multiple steady states are also discussed.

1. INTRODUCTION

For the purposes of this paper, model-predictive control (MPC) is defined as a control scheme in which the controller determines a manipulated variable profile that optimizes some open-loop performance objective on a time interval extending from the current time to the current time plus a prediction horizon. This manipulated variable profile is implemented until a plant measurement becomes available. Feedback is incorporated by using the measurement to update the optimization problem for the next time step. Many methods for updating the optimization problem are possible, such as resetting the model's initial conditions, estimating model parameters and/or states, inferring output disturbances, etc. The exact nature of the update does not matter at this point and several schemes are examined in the following discussion.

The *defining* feature of model-predictive control then is a repeated optimization of an open-loop performance objective over a finite horizon extending from the current time into the future. We emphasize this definition because the distinction is important in what follows and other definitions have appeared in the literature. García *et al.* (1989), for example, define model-predictive control as "that family of controllers in which there is a direct use of an explicit and separately identifiable model." This definition is much too general for our purposes as it includes many classic feedback controllers in which there is no horizon.

The problem to be solved by the model-predictive controller may be stated as

$$\min_{u(t)} \quad \Phi[u(t), x(t), y(t)] \qquad \text{(NLP1)}$$

subject to

$$\frac{dx}{dt} - f(x, u) = 0$$

$$y - g(x, u) = 0$$

$$h(x, u) = 0$$

$$k(x, u) \geqslant 0$$

$$x(t_0) = x_0$$

with

$$t \in [t_0, t_0 + T]$$

in which u is the input vector, y is the output vector, and x is the state vector. The time interval is from the current time, t_0, to some finite time in the future, $t_0 + T$, in which T is the length of the prediction horizon. The scalar functional Φ is the controller's performance objective, the functions f and g determine the plant model, and h and k are equality and inequality constraint functions that may be specified as further performance objectives. The generality of the performance objective, as opposed to standard integral square error between output and set point, provides the opportunity to design the MPC controller for higher level functions such as energy or waste minimization.

Model-predictive control, as defined above, has appeared in several branches of the control literature during the past thirty years. The concept of using an open-loop optimal control computation to synthesize a feedback controller is so natural that it probably occurred to many researchers in the optimal control field in the late 1950s and 1960s. In their textbook on optimal control, Lee and Markus describe the approach while pointing out that current (as of 1967) hardware and software make real-time implementation of the controller difficult. In their review, García *et al.* (1989) cite Propoi (1963) as the first to introduce, explicitly, the finite moving horizon.

In the electrical engineering literature, model-predictive control is usually called receding (or moving) horizon control. Although this name is clearly more descriptive of the general approach, we will also refer to it as model-predictive control since this name has become entrenched in the chemical engineering literature. Kleinman (1970) uses the finite horizon concept to find a state feedback gain that stabilizes linear time-invariant systems. Thomas (1975) formulates a quadratic objective function penalizing only the input with the constraint that the state must be zero at the end of

[†]Author to whom correspondence should be addressed.

the horizon. He shows this formulation results in a state feedback that stabilizes linear time-invariant systems. Kwon and Pearson (1977) generalize these results by considering the linear time-varying system,

$$\dot{x} = A(t)x + B(t)u$$
$$y = C(t)x$$

and using the standard quadratic performance objective,

$$\Phi = \int_{t_0}^{t_0+T} [y^T Q(t) y + u^T R(t) u] \, dt \qquad (1)$$

including the constraint that $x(t_0 + T) = 0$. They show that the receding horizon controller can stabilize linear time-varying systems. Kwon *et al.* (1983) also consider linear time-varying systems with the quadratic objective

$$\Phi = \int_{t_0}^{t_0+T} [x^T Q(t) x + u^T R(t) u] \, dt + x^T F x \Big|_{t_0+T}.$$

They show the solution to this problem is also stabilizing state feedback law. Mayne and Michalska (1990) consider the quadratic objective function with final time constraint for the nonlinear system

$$\dot{x} = f(x, u).$$

They show that, under certain conditions, the receding horizon controller stabilizes the nonlinear system.

The MPC framework is also used in aerospace engineering applications. Most publications concern aircraft trajectory optimization (Brusch, 1974; Johnson, 1975). Attention is presently being given to solving nonlinear, constrained optimal trajectories over finite horizons for real-time guidance (Bless and Hodges, 1990; Jänsch and Paus (1990); Psiaki and Park, 1990).

The use of MPC in the chemical engineering field started in the process industries in the 1970s under the names of "model predictive heuristic control" or "model algorithmic control" (Richalet *et al.*, 1978; Mehra *et al.*, 1982) and "dynamic matrix control" (Cutler and Ramaker, 1979; Prett and Gillette, 1979). The recent review by García *et al.* (1989) covers the chemical engineering literature on MPC and does not need to be repeated here. The conclusions of their study are that MPC has found wide acceptance in industrial practice, has flexible constraint-handling capabilities, and can be adjusted rather easily for robustness.

2. LINEAR SYSTEMS

2.1. *Prediction horizon*

For linear systems, the feature that distinguishes MPC from standard feedback control is the explicit use of a prediction horizon in the control law formulation. In order to see this distinction clearly, consider the standard single-input/single-output block diagram in Fig. 1. One of the characteristics of this structure is that the controller's response to zero error is zero control action,

$$e(t') = 0 \quad t' \in [0, t] \Rightarrow u(t) = 0. \qquad (2)$$

On the other hand, the prediction horizon allows the MPC controller to take control action at the current time in response to a forecast of a future error even though the error up to the current time is zero. In other words, eq. (2) does not hold for linear MPC controllers. We show in the next section that it is precisely this property of the MPC controller that is beneficial for controlling (or, more precisely, scheduling) nonminimum phase plants.

We should also mention internal model control (IMC), another feedback control scheme which shares some of the properties of MPC. The key feature of IMC is that like MPC, a convenient open-loop design is used for the controller (García and Morari, 1982). However, the IMC controller obeys eq. (2), is a standard feedback controller, and can be arranged into the structure shown in Fig. 1.

This article emphasizes the types of behavior that can be achieved with predictive control that are unattainable with standard feedback. This improved performance is only possible because the predictive controller is given information about future constraints and future inputs such as planned set-point changes or forecasts of loads or disturbances. This information is often available: set-point changes and throughput or raw-material changes may be known in advance. Actuator saturation limits are always known in advance. The predictive control structure is nicely suited for using this kind of information, and will make the eventual goal of scheduling and optimization of the entire plant easier to attain.

Because the algorithms for MPC of linear systems have similar features, we will focus attention on one of them, quadratic dynamic matrix control (QDMC), since it has been most fully described in the literature (García and Morshedi, 1986; Ricker, 1985). Satisfaction of hard input constraints, one of the key reasons for the success of QDMC in practice, must be discarded at this juncture to keep the discussion focused on linear systems. Hard input constraints are

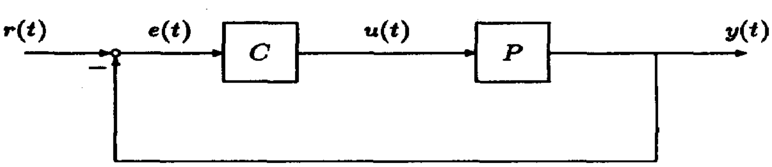

Fig. 1. Standard feedback structure.

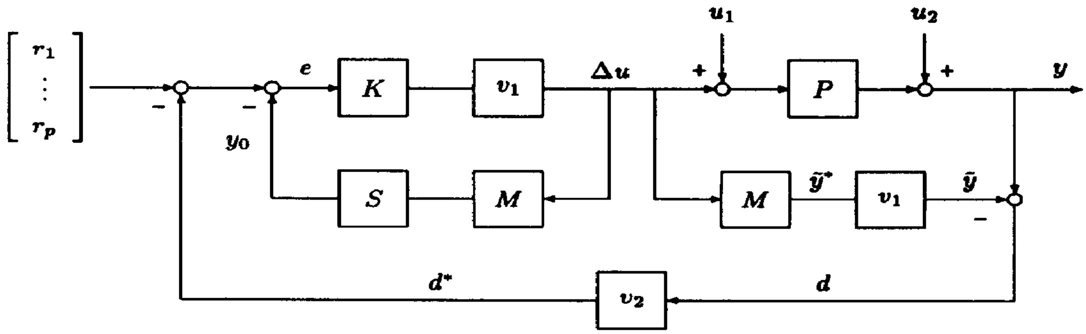

Fig. 2. Block diagram representation of the DMC algorithm.

discussed in Section 3.1. The remainder of this section describes the standard SISO DMC algorithm. The discussion focuses on SISO systems for simplicity of presentation. The features of the algorithm extend naturally to MIMO systems.

The inputs in the feedback structure shown in Fig. 2 are a sequence of future set points, r, and standard plant input and output disturbances, u_1 and u_2. It is important to note that if the controller is given future set-point information, it will be able to improve performance by acting on estimates of future error. Ricker (1985) provides an example of this feature of the algorithm applied to the control of a continuous evaporator. The future set-point information is also the natural way to define the batch control problem. The future horizon is the part of MPC that allows the same framework to address both batch and continuous problems. In batch problems, however, one often needs to use a nonlinear model because the states may move across a large portion of the state space during the batch run.

The DMC model is the vector of transfer functions

$$M = (qI - S)^{-1} A_1$$

which produces a vector of future outputs due to the sequence of all inputs up to the current Δu. Of these, only the first is normally used as the output of the model. The column vector A_1 is the first column of the "dynamic matrix" of step-response coefficients

$$A \equiv \begin{bmatrix} a_1 & & 0 \\ \vdots & \ddots & \\ a_n & & a_1 \\ \vdots & & \vdots \\ a_p & & a_{p-n+1} \end{bmatrix}$$

which determines the dynamics of the (stable) process model. The row vector $v_1 = [1\ 0 \ldots 0]$ selects the first element of the predicted outputs, $\tilde{y}^*$ for comparison with the plant output, y, and the column vector v_2 projects the current estimate of the disturbance into the future. In the standard DMC algorithm it is assumed that the estimated disturbance will remain constant over the entire prediction horizon and $v_2 = [1 \ldots 1]^T$. This assumption is not necessary, and v_2 may be modified to reflect projected changes in the disturbance if that information is available.

The matrix S has the form

$$S \equiv \left[\begin{array}{c|c} \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} & \mathrm{I} \\ \hline \varepsilon_1 & \varepsilon_2 \ldots \varepsilon_p \end{array}\right]$$

performs a shift and extrapolation of the vector of future output predictions $\tilde{y}^*$. In the standard DMC algorithm, it is assumed that the system will reach steady state by the end of the prediction horizon and the extrapolation vector, ε, is given by $\varepsilon = [0 \ldots 0\ 1]$. Clearly, from the definition of M, the poles of the plant are the eigenvalues of S. The characteristic equation for the eigenvalues can be found from the Laplace expansion of the determinant of $qI - S$

$$\det(qI - S) = q^p - \varepsilon_p q^{p-1} - \cdots - \varepsilon_2 q - \varepsilon_1 = 0.$$

The poles of the standard DMC model are therefore the roots of $q^{p-1}(q-1) = 0$, or $p-1$ poles at the origin and one at $q = 1$ (an integrator). The integrator is simply a result of formulating the DMC model in Δu rather than u [see also Lee *et al.* (1990)].

To see how the transfer function is related to the dynamic matrix, consider Fig. 3. The model output, $\tilde{y}$, is obtained from the sum of $A_1 \Delta u$ and the previous estimate of the output over the entire prediction horizon, which is shifted and extrapolated to account for the horizon moving forward one sample period each time the output is computed.

The gain of the standard QDMC controller, K, is obtained from the solution of the following quadratic program:

$$\min_{\Delta u^*} \Phi, \quad \Phi = (r - y_m)^T \Lambda^T \Lambda (r - y_m) + \Delta u^{*T} \Gamma^T \Gamma \Delta u^* = \tfrac{1}{2} \Delta u^{*T} H \Delta u^* + c^T \Delta u^* \qquad \text{(QP1)}$$

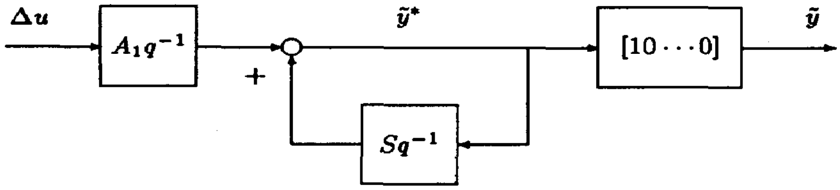

Fig. 3. Block diagram for the DMC model, M.

subject to

$$\begin{bmatrix} \Delta u_l \\ u_l - u_{\text{curr}} \\ \tilde{y}_l \end{bmatrix} \leqslant \begin{bmatrix} I \\ L \\ A \end{bmatrix} \Delta u^* \leqslant \begin{bmatrix} \Delta u_u \\ u_u - u_{\text{curr}} \\ \tilde{y}_u \end{bmatrix}$$

where

$$H = A^T \Lambda^T \Lambda A + \Gamma^T \Gamma$$

$$c^T = e^T A$$

$$e = r - y_0 - d^*$$

$$L = \begin{bmatrix} 1 & & 0 \\ \vdots & \ddots & \\ 1 & \cdots & 1 \end{bmatrix}$$

and y_l and y_u are p-vectors of bounds on the output, u_l and u_u are n-vectors of bounds on the actual value of the input, Δu_l and Δu_u are n-vectors of bounds on the change in the input from one sampling time to the next, and u_{curr} is the current value of the input. The vector y_m represents the best estimate of the plant output over the prediction horizon including disturbance information and the vector y_0 is the model output over the future horizon if all future inputs are zero. The vector d^* is a disturbance estimate over the prediction horizon.

For the strictly linear DMC algorithm, the solution of QP1 is given by

$$\begin{aligned} \Delta u^* &= -H^{-1} c \\ &= -H^{-1} A^T e \\ &= Ke \end{aligned}$$

and $-H^{-1}A^T$ need be evaluated only once and the error, e is simply updated in order to compute the controller output.

With these definitions, the steps of the algorithm are:

(0) Initialize the dynamic matrix, A, the initial estimate of future outputs, $\tilde{y}^*$, and the initial estimate of the output disturbance, d. For simulations, $\tilde{y}^*$ and d are normally initialized to zero. Compute the Hessian for the QP. This will not need to be modified unless either the input or output penalties or the dynamic matrix changes. Initialize the constraint matrix. For the purposes of the current discussion, we will skip this step and focus on the standard unconstrained DMC algorithm.

(1) Update the vector of linear coefficients for the QP objective function. The computation of $A^T(r = y_0 - d^*)$ must be done at each sampling time.

(2) Solve QP1 for the changes in the manipulated variables. If the problem is strictly linear and unconstrained, this is simply a multiplication of the controller matrix, $K = -H^{-1}A^T$, and the error vector, e.

(3) Implement some number of the computed inputs (normally assumed to be just one).

(4) Measure the plant output, y.

(5) Compute the predicted output for the plant over the current prediction horizon based on the inputs used in step 3.

$$\begin{bmatrix} \tilde{y}_1^* \\ \vdots \\ \tilde{y}_p^* \end{bmatrix} \leftarrow A \Delta u + \begin{bmatrix} \tilde{y}_1^* \\ \vdots \\ \tilde{y}_p^* \end{bmatrix}.$$

(6) Estimate future values of the disturbance, d^* from $d = y - \tilde{y}$, and the particular choice of v_2.

(7) Update the predicted output over the future time horizon by setting $y_{0,i} = \tilde{y}_{i+1}^*$ for $i = 1$ to p. Note that a value for $\tilde{y}_{p+1}^*$ is required and that this point lies one sampling time beyond the current prediction horizon. If the plant is stable and the prediction horizon is greater than the control horizon by at least the settling time of the plant then it is reasonable to choose $\tilde{y}_{p+1}^* = \tilde{y}_p^*$. This is what is done in the standard DMC algorithm. In the following sections we will see that this approach does not work well if the prediction horizon is relatively short or the plant is unstable.

(8) Step forward one sampling time and return to step 1.

2.2. *Nonminimum phase plants*

For minimum phase, linear systems, perfect control is possible (in the limit of improper controllers) with standard feedback control. MPC, therefore, offers no performance advantages for minimum phase, linear systems.† It is well-known, however, that nonminimum phase elements (right-half-plane zeros and time delays) limit achievable performance with feedback control [see, for example, p. 48 of Morari and Zafiriou (1989)].

†This is not true minimum phase, linear plants with input constraints (see Section 3.1).

As an example, consider an all-pass plant with a real right-half-plane (RHP) zero at $s = a$, $a > 0$,

$$g(s) = \frac{-s + a}{s + a}.$$

A realization of this plant is given by

$$\frac{dx}{dt} = -ax + u \tag{3}$$

$$y = 2ax - u.$$

Assume that one wants to make a step change in the set point for this plant at $t = \theta$,

$$r(t) = H(t - \theta).$$

The open-loop control action that achieves zero tracking error for this plant on any bounded interval, $t \in [0, B]$, $B > \theta$, is

$$u(t) = \begin{cases} 0, & t \in [0, \theta) \\ 1 - 2e^{a(t-\theta)}, & t \in [\theta, B] \end{cases}. \tag{4}$$

This control action is obviously undesirable because of its exponential growth after $t = \theta$. A feedback controller that produces this control action is easily avoided since such a controller produces a feedback loop that is internally unstable. Such a controller inverts the RHP zero in g and tries to use unstable pole zero cancellation with the plant. If one finds a feedback controller to minimize the integral square error (ISE) of e, and stipulates that the feedback system is internally stable (preventing the RHP zero inversion), the controller is

$$c(s) = \frac{s + a}{2s}.$$

This optimal feedback controller produces the control action and output shown in Fig. 4. The ISE for the controller is $2/a$. In this sense, the RHP zero has limited the achievable performance of any stabilizing feedback controller.

Consider now the MPC controller. Since it does not satisfy eq. (2), one is led to consider what anticipatory control action, $u(t)$, $t \in [0, \theta]$, should be taken if one knows a step change in set point is planned at $t = \theta$. This question becomes clearer if one considers instead the doubly infinite time horizon, $t \in (-\infty, \infty)$, and

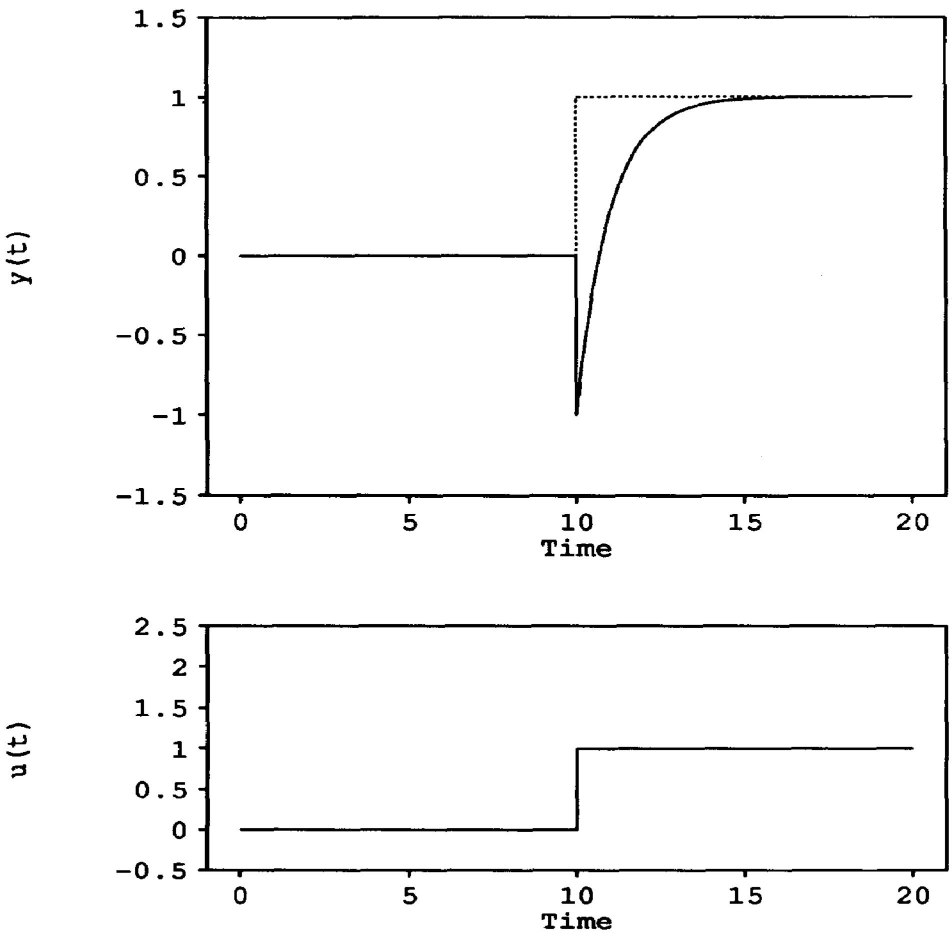

Fig. 4. Optimal ISE response $y(t)$ and input $u(t)$ for the system $(-s + a)/(s + a)$ using an internally stable feedback.

solves the following optimization problem,

$$\min_{u(t)} \Phi = \int_{-\infty}^{\infty} e^2(t)\,\mathrm{d}t. \tag{5}$$

The control action that achieves zero tracking error on the infinite interval is,

$$u(t) = \begin{cases} 2e^{a(t-\theta)}, & t \in [-\infty, \theta) \\ 1, & t \in [\theta, \infty) \end{cases}. \tag{6}$$

This result can be checked using eq. (3) and $x(-\infty) = 0$ as the initial condition. Notice that this control action, shown as the dashed lines in Figs 5 and 6, is small (less than two) for all t, in contrast to eq. (4).

In any practical situation, one has only a finite prediction horizon. In order to examine the performance loss due to the finite horizon, T, $T < \theta$, consider the following approximation to eq. (6).[†]

[†]This input can be shown to be optimal for a class of modified problems. In the modified problems, input or state penalties are added to Φ, or constraints are placed on u in order to remove the undesirable e^{at} term from u for $t > \theta$.

$$u(t) = \begin{cases} 0, & t \in [0, \theta - T) \\ 2e^{a(t-\theta)}, & t \in [\theta - T, \theta) \\ 1, & t \in [\theta, \infty). \end{cases} \tag{7}$$

Substituting this input into eq. (3) with zero initial conditions gives,

$$y(t) = \begin{cases} 0, & t \in [0, \theta - T) \\ -2e^{-a(t+2T-\theta)}, & t \in [\theta - T, \theta) \\ 1 - 2e^{-a(t+2T-\theta)}, & t \in [\theta, \infty). \end{cases} \tag{8}$$

The ISE for this input is

$$\int_0^{\infty} e^2(t)\,\mathrm{d}t = \frac{2}{a} e^{-2aT}. \tag{9}$$

Notice that $T = 0$ (no prediction horizon) in eq. (9) recovers the optimal feedback controller's ISE. Since the ISE decreases exponentially with T, one does not require a long horizon to get a large benefit from MPC. For example, if $a = 1$, one can choose $T = 2.3$ to remove 99% of the error achieved with the best feedback controller. Notice also that the closer the zero is to the origin, the longer the prediction horizon must be to improve performance.

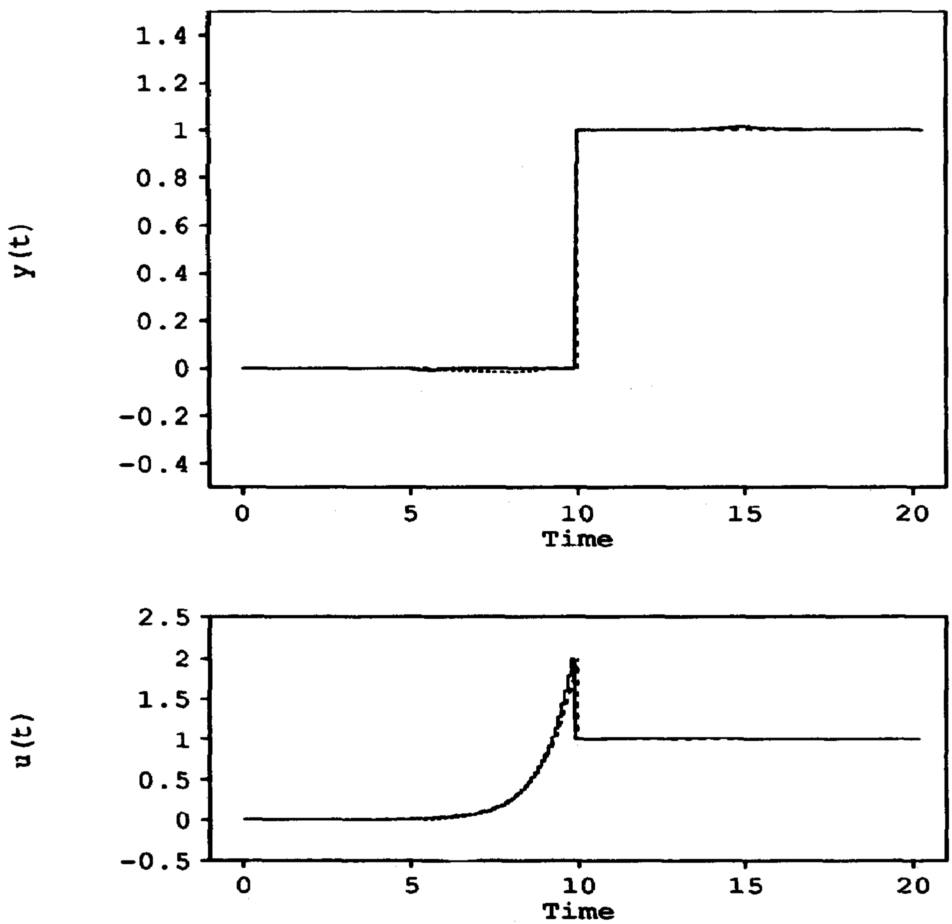

Fig. 5. Ideal [(– – – –), $T = \infty$] and actual [(. . . . .), $T = 5.0$, $C = 4.0$; (———), $T = 5.0$, $C = 1.0$] response, $y(t)$, and input, $u(t)$, for the system $(-s + a)/(s + a)$ using predictive DMC with a sample time of 0.1.

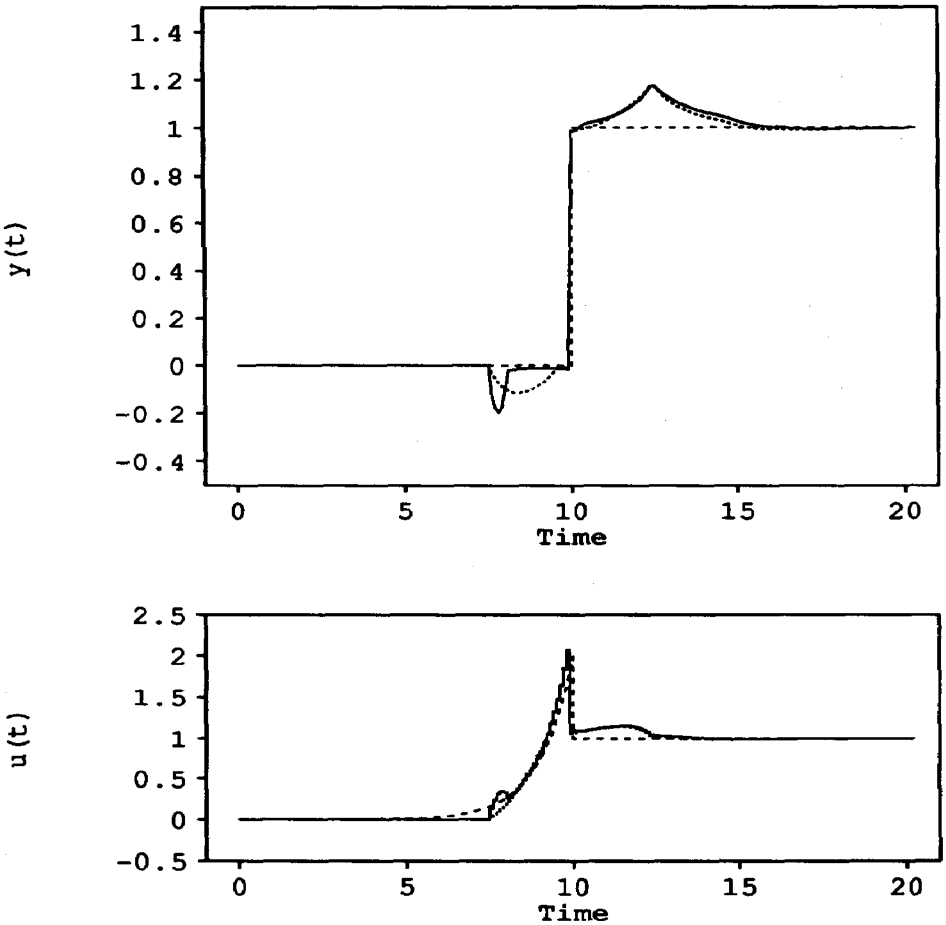

Fig. 6. Ideal [(– – – –), $T = \infty$] and actual [(. . . . .), $T = 2.5$, $C = 2.0$; (———), $T = 2.5$, $C = 0.5$] response, $y(t)$, and input, $u(t)$, for the system $(-s + a)/(s + a)$ using predictive DMC with a sample time of 0.1.

In designing the MPC controller, one should be careful to define the performance objective so the controller approximates eq. (6) rather than eq. (4) as the control action for plants with RHP zeros. There are several ways to accomplish this, most of which involve penalizing the input (or states). One could use eq. (1), for example, for the objective function. The controller would be stabilizing with even a modest penalty, R. The only criticism of this approach is that there is no real justification for the selection of R. In process control applications, R certainly does not reflect the cost of control action. An alternative approach is given in García and Morshedi (1986). They choose the control horizon, C, to be less than T. This is equivalent to placing an *infinite* penalty on the rate of change of the last control moves in the horizon. This has two benefits. First, the need for selecting a value of R is removed. Second, the demand that the control action become constant after sufficient time has elapsed, forces the controller to avoid the input in eq. (4), and the system is stable. In practice, it is also probably easier to find a good value of C than to choose an appropriate R. Mehra and Rouhani (1980) offer a different approach to design for RHP zeros involving a second process model. Their approach does not seem nearly as direct and simple as García and Morshedi's.

Figures 5 and 6 show the performance of the standard DMC algorithm for several choices of C and T. In all cases C is less than T and the controller is stabilizing. Notice that the standard DMC algorithm does a reasonably good job of finding eq. (7) as the input. The deviations in the interval $7 < t < 9$ in Fig. 6 are directly attributable to making $C < T$. The parameterization of the input for $C < T$ [i.e. u constant for $t \in (t_0 + C, t_0 + T)$] prevents the controller from finding eq. (7) when the time of the set-point change falls between the ends of the control horizon and the prediction horizon. The optimal u has its large negative jump in this time interval but the DMC controller has u parameterized as a constant. Figure 7 shows what the DMC algorithm forecasts for the input when the time of the set-point change is between the control and prediction horizons. After some time passes, the jump in $u(t)$ occurs for $t < t_0 + C$ and the DMC parameterization of the input can track eq. (7) exactly. The effects of the small time of suboptimal behavior of the DMC algorithm can be reduced by

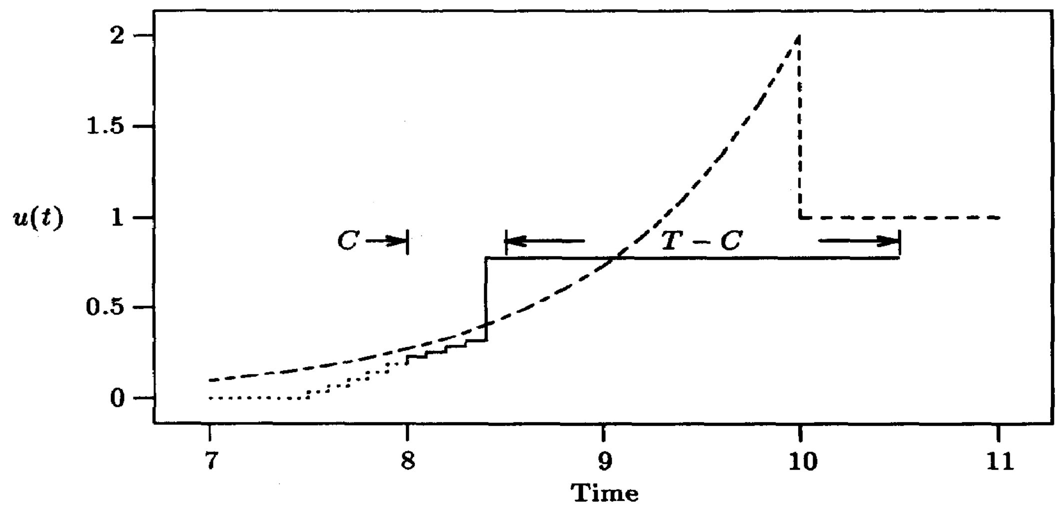

Fig. 7. Performance of DMC at a given sample time with $T = 2.5$ and $C = 0.5$ for the system $(-s + a)/(s + a)$. (————) represents the predicted optimal inputs, of which only one will be implemented. (. . . . .) shows the past inputs which have been used, and (– – – – –) shows the ideal input.

making the horizon larger (compare Figs 5 and 6). Finally, a more flexible solution is to add a penalty term of the form $R(u - u_{\mathrm{ref}})^2$ to Φ in which u_{ref} is given by eq. (7) and R is adjustable. In this framework, the controller could have $C = T$, and thus be able to track eq. (7) very closely, and still be stabilizing.

The other nonminimum phase element, the time delay, is trivial to handle because, unlike the linear system with a RHP zero, the delayed system only requires a finite horizon for perfect tracking. Consider a plant with delay, θ_p,

$$g(s) = e^{-\theta_p s}.$$

Perfect tracking of r is obviously achieved with $\theta > \theta_p$.

$$u(t) = \begin{cases} 0, & t \in [0, \theta - \theta_p) \\ 1, & t \in [\theta - \theta_p, \infty). \end{cases} \tag{10}$$

The only restriction is that the horizon must be at least as long as the delay. If it is not, the step in u occurs at $\theta - T$ (when the MPC controller first "sees" the step set-point change in its prediction horizon) instead of at $\theta - \theta_p$ and the ISE is

$$\int_0^\infty e_2^2(p)\,\mathrm{d}p = \begin{cases} \theta_p - T, & T < \theta_p \\ 0, & T \geqslant \theta_p. \end{cases}$$

All reasonable MPC controllers compute the u in eq. (10) without any special tuning. Indeed, one of the strengths of MPC is its inherent time delay compensation.

The multivariable extension of the desirable properties of MPC for handling nonminimum phase elements is straightforward. Several papers in the Second Shell Process Control Workshop (Prett *et al.*, 1989), for example, implement MPC algorithms on a process with three inputs, two loads, and seven outputs in which 24 of the 35 transfer functions have time delays.

2.3. *Unstable plants*

Clarke *et al.* (1987) handle unstable plants with predictive control by using a CARIMA plant model in which the unstable poles are included explicitly in the model. Other researchers have demonstrated stabilization of unstable plants with various MPC algorithms (Brosilow and Cheng, 1987; Brengel and Seider, 1989; Hidalgo and Brosilow, 1990; Patwardhan *et al.*, 1990; Mayne and Michalska, 1990; Lee *et al.*, 1990; Schmid and Biegler, 1990).

The standard DMC algorithm, however, is unable to handle unstable processes due to an unstable transfer function between u_1 and d in Fig. 2.

To represent an unstable process with the DMC model, one may choose ε to be an extrapolation formula that approximates the unstable growth of the output due to past inputs. Within the framework of the DMC feedback structure, however, one can only choose ε to describe an integrator, not any arbitrary unstable plant. This approximation may work in practice for some systems, but will fail if the plant–model mismatch becomes too large. In order to overcome this deficiency and stabilize arbitrary unstable plants, one must design a feedback that avoids the internal stability problems of the DMC algorithm.

To demonstrate the ability of this modified algorithm to handle unstable processes, consider the plant with a pole at $s = 1$ (i.e. q strictly outside the unit circle).

$$\frac{\mathrm{d}x}{\mathrm{d}t} = x + u$$

with

$$x(t = 0) = 0.$$

The sample time for the simulation is 0.1, with control and prediction horizons both equal to 1.0. Figure 8

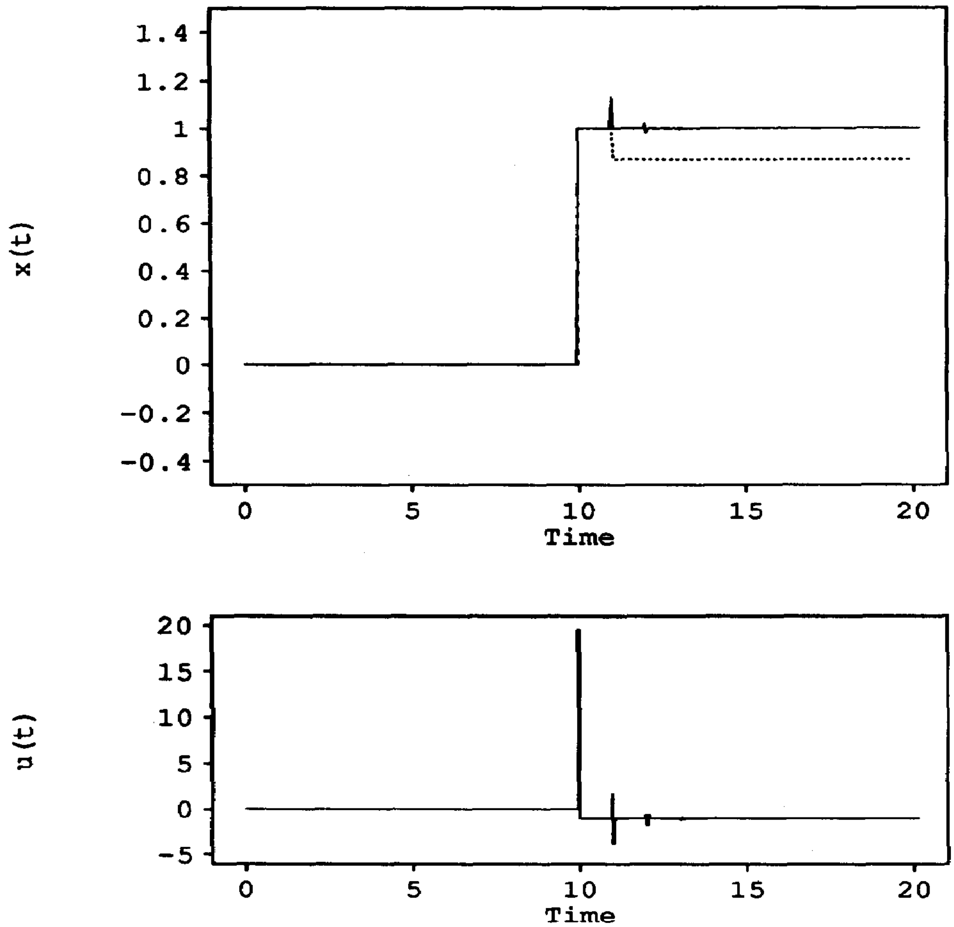

Fig. 8. Performance of DMC with an integrating model for the plant $\dot{x} = x + u$. (———) shows the actual input and output, (. . . . .) shows the model prediction at each sample time, and (– – – – –) is the desired behavior.

shows the system response and control action for the unconstrained system. In the plot of the output, $x(t)$, the solid line shows the plant output, the dotted line shows the model prediction at each sample time, and the dashed line is the desired behavior. The difference between the plant and model prediction at steady state is due to model error caused by the specific choice of the extrapolation formula, which in this case is $\varepsilon = [0 \ldots 0 - 1\ 2]$. This represents an integrating process and the model will have two poles at $q = 1$ (again, one arises from the fact that the DMC model is in written in terms of Δu).

The spike in the input at $t = 10.0$ is due to the fact that the system is not constrained and there are no penalties for large changes in the input. This undesirable behavior can be eliminated by the addition of input constraints but is not discussed here because the resulting system would be nonlinear. The smaller spikes are due to the model error introduced by choosing an extrapolation formula for an integrator in the model when the plant has a pole at $s = 1$. Note that the spikes occur at intervals equal to the length of the prediction horizon. The effect of this model error could also be minimized by the introduction of input constraints or penalties.

3. NONLINEAR SYSTEMS

Two types of nonlinearities are considered in this section and the performance of MPC is examined for each. The first involves input constraints for a linear process model and the second involves an unstable nonlinear process which exhibits multiple steady-state behavior.

The QDMC algorithm described in Section 2.1 is applied to the first example, with step 2 of the algorithm replaced by a solution of the full quadratic program QP1.

For the second example, the full nonlinear open-loop optimal control problem, NLP1, is solved at each sampling time. A portion of the predicted input is implemented and the plant output is measured. This information may be used with the process model to estimate a disturbance as in the QDMC algorithm or to update the values of the model parameters.

Patwardhan *et al.* (1990) and Eaton and Rawlings (1990) discuss additional details of this algorithm.

3.1. *Input constraints*

The following example demonstrates the algorithm's ability to handle input constraints in a way that is different from standard feedback control with anti-windup. The predictive controller is able to forecast the violation of constraints in the future. This feature allows current action to be taken to minimize the errors caused by constraints that are predicted to become active in the future [see also Ricker (1985)].

To demonstrate this feature we will examine the performance of the QDMC algorithm on the simple first-order system

$$\frac{dx}{dt} = -x + u$$

with

$$x(t = 0) = 0$$

with the nonlinearity introduced by the additional requirement that $-0.1 \leq u(t) \leq 1.1$. The sample time used in the simulations is 0.1. In both cases shown in Fig. 9 the prediction horizon is 5.0. The dashed line shows the controller action and the system response for a control horizon of 1.0. Note that the controller is able to take action ahead of the desired set-point change and achieves performance superior to that of a feedback controller.

The negative control action that begins near $t = 5.0$ is caused by the difference in the control and prediction horizons. Once again we see that although the control action is optimal at each sampling time for the chosen horizons, the overall sequence of inputs that is implemented is not particularly desirable for some choices of C and T. This is again due to the fact that the controller is asked to predict the required control action for a changing profile while also being forced to predict one relatively long constant control move over the final portion of the prediction horizon.

The solid line shows the system response for a control horizon equal to the prediction horizon. Note that the performance is again improved by choosing

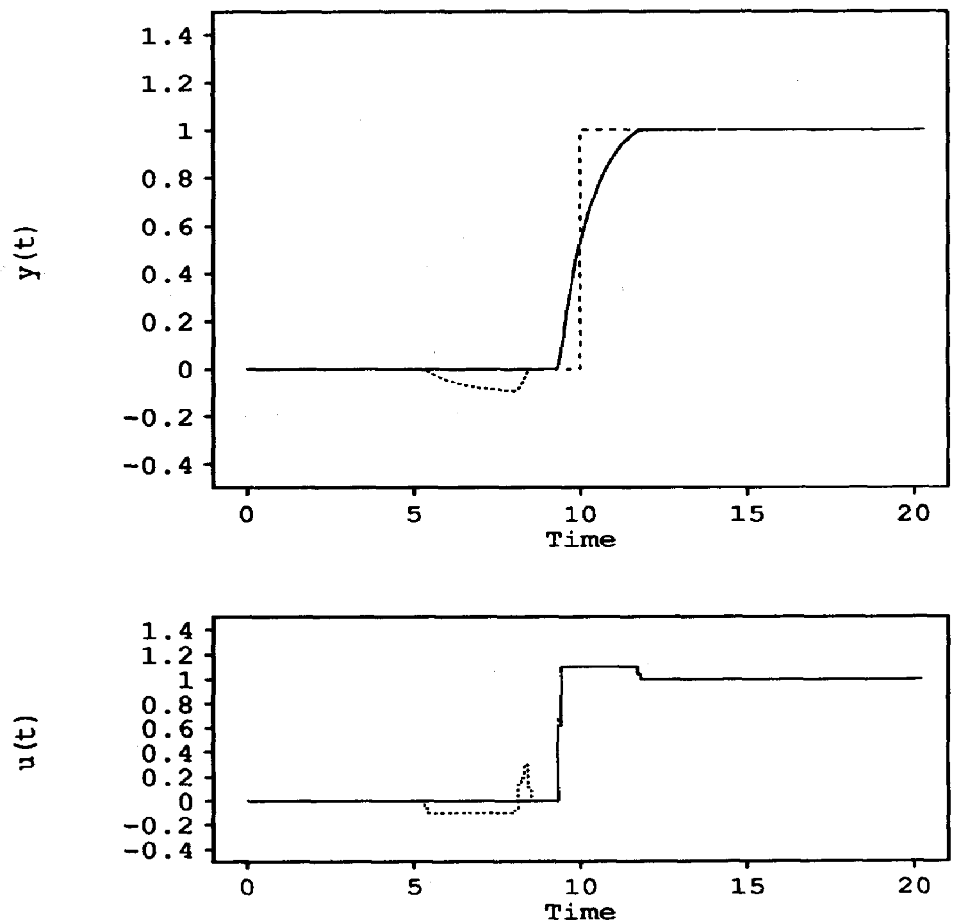

Fig. 9. Performance of QDMC for $G(s) = 1/(s + 1)$ for a sampling time of 0.1 and input constraint $-0.1 \leq u(t) \leq 1.1$ for two choices of prediction and control horizons [(——), $T = 5.0$, $C = 5.0$; (. . . . .), $T = 5.0$, $C \sim 1.0$].

$C = T$ because the controller is always able to predict the proper control action.

The optimal bang-bang profile for this problem is

$$u(t) = \begin{cases} 0, & t \leq t_1 \\ u_{max}, & t_1 < t < t_2 \\ 1, & t \geq t_2 \end{cases}$$

$$t_1 = \theta - \log\left(\frac{2u_{max}}{2u_{max} - 1}\right)$$

$$t_2 = t_1 + \log\left(\frac{u_{max}}{u_{max} - 1}\right)$$

where θ is the time of the step change in the set point, u_{max} is the upper bound on the input, and t_1 and t_2 are the switching times. For this particular example, $t_1 = 9.39$ and $t_2 = 11.79$, which is close to the solution that the QDMC controller finds, provided that the control and prediction horizons are sufficiently long.

The ISE for MPC and standard feedback (with a perfect anti-windup strategy) are shown as a function of u_{max} in Fig. 10. Notice that MPC is superior regardless of the value of u_{max} and that the ratio of the ISEs approaches 0.25 in the limit of large u_{max}.

3.2. *Unstable plants*

As an example of an unstable nonlinear process, consider the following model of a continuous bioreactor with substrate inhibition (Agrawal and Lim, 1984). For some regions of the parameter space, this system exhibits multiple and unstable steady-state behavior. The state equations are

$$\frac{dx}{dt} = (\mu - D)x$$

$$\frac{ds}{dt} = (s_f - s)D - \frac{\mu x}{y}$$

$$\mu = \frac{\mu_{max} s}{k_m + s + k_1 s^2}$$

Table 1. Steady-state values of the states (biomass, x and substrate, s) for a single value of manipulated variable, D

	Open-loop stable steady state	Open-loop unstable steady state
x	1.5122	0.9951
s	0.1746	1.5301
D	0.3	0.3

with

$$x(t = 0) = x_0$$

$$s(t = 0) = s_0.$$

The dilution rate, D is manipulated to control the cell mass concentration, x. The other state and parameters are the substrate concentration, s, the specific growth rate, μ, the yield of cell mass y, and the substrate concentration in the feed stream, s_f. The values of the parameters used in the simulation of the plant are $y = 0.4$, $s_f = 4.0$, $\mu_{max} = 0.53$, $k_m = 0.12$, and $k_1 = 0.4545$. Steady-state solutions of the model equations for these parameter values are given in Table 1. The values of the parameters μ_{max} and k_m used in the model equations are 20% greater than the values used in the simulation of the plant.

For the simulations shown here, the following performance objective is chosen to obtain a specified cell mass concentration,

$$\Phi = \int_t^{t+T} (x_{set} - x)^2 \, dt' \tag{11}$$

and the open-loop optimal control step is solved by approximating the model equations using orthogonal collocation on finite elements. The resulting nonlinear program is solved using NPSOL, a Fortran implementation of a successive quadratic programming algorithm (Gill *et al.*, 1986). For the specific simulations shown subsequently, the prediction horizon

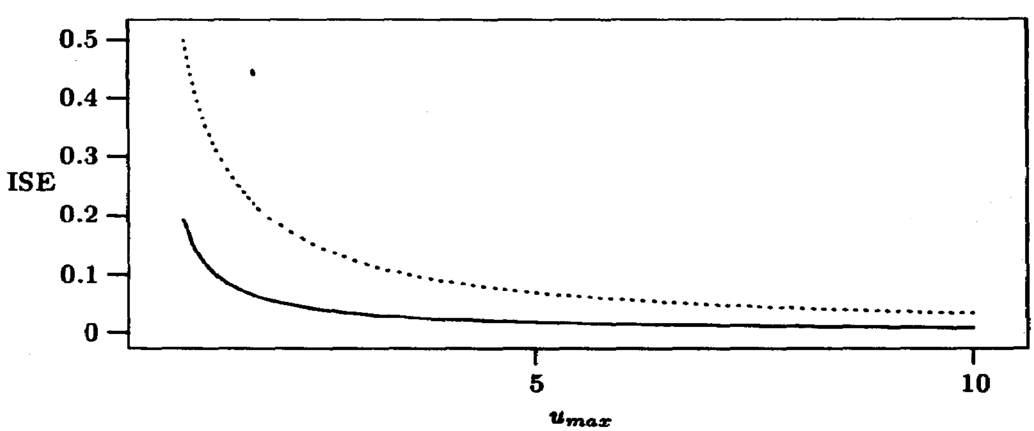

Fig. 10. Integral square error for predictive control (———) and optimal feedback control (. . . . .) for the first-order system with input constraints.

and control horizon are both 0.6, and the sample time is 0.2.

Figure 11 shows the response of the system to a set-point change from the unstable to the stable steady state. In this case, the general approach is very much like the standard DMC algorithm—the controller provides a sequence of optimal inputs and a small number of these are used to drive the plant. A measurement is obtained, compared to the output predicted by the model, and the difference is fed back to the controller. The model equations are allowed to evolve separately from the plant without any difficulty because the parameter error is small and the set point is a stable stationary point.

While this approach is acceptable for open-loop stable linear systems, it is not particularly well-suited to nonlinear systems. Even if the only steady state in the entire operating region is stable, the performance will suffer if the model is allowed to move too far from the plant because, for nonlinear systems, the plant's behavior will vary throughout the operating region.

To see how this can happen, even for the same small errors in model parameters, consider Fig. 12, which shows the behavior of the same system when moving from the stable to the unstable set point. Even though the closed-loop system is input–output stable, there are unexpected deviations near $t = 5.0$, and the model prediction bears no resemblance to the plant output.

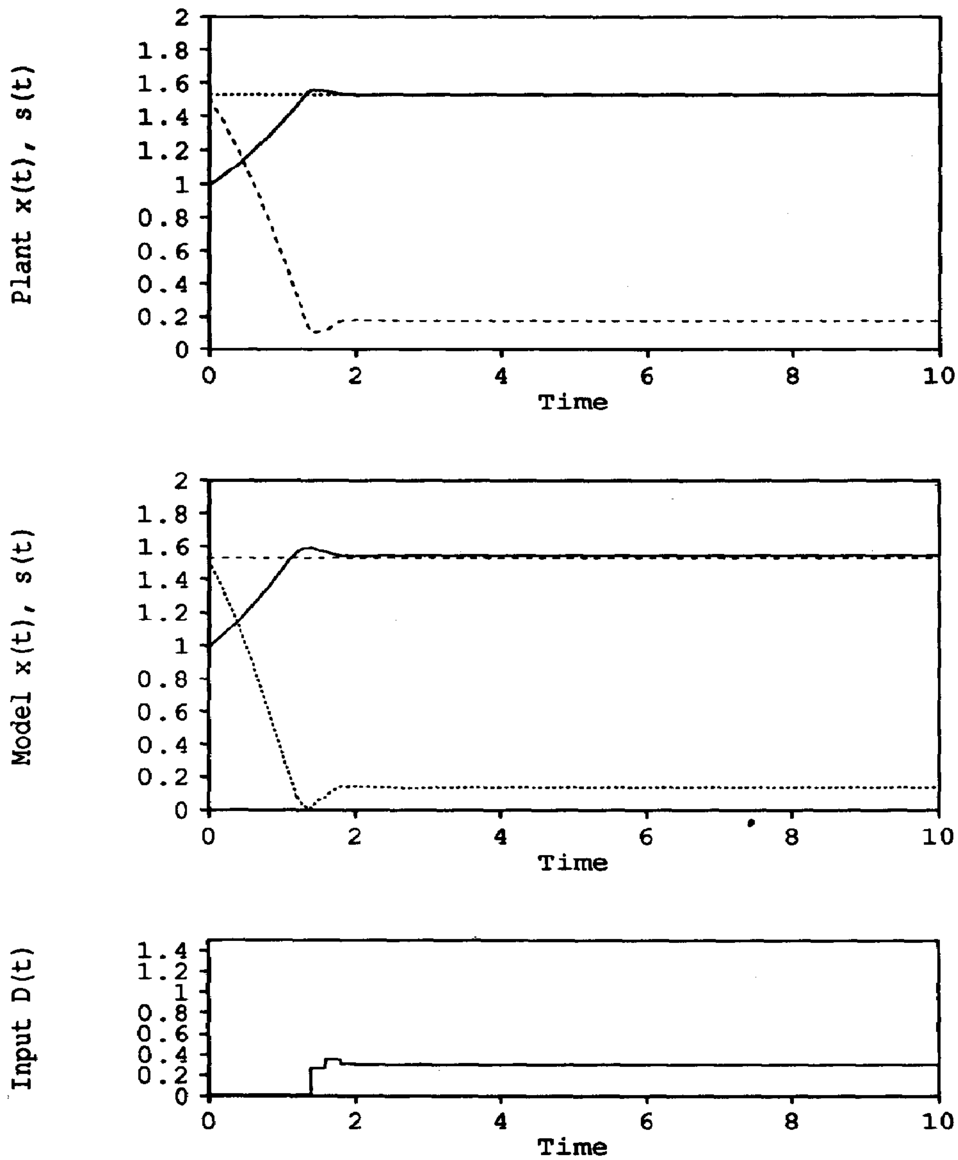

Fig. 11. Performance of nonlinear model-predictive controller without resetting the model initial condition for an open-loop stable target. Biomass (——) and substrate (. . . . .) concentrations (top and middle) and dilution rate (bottom) for the parameters given in Table 1.

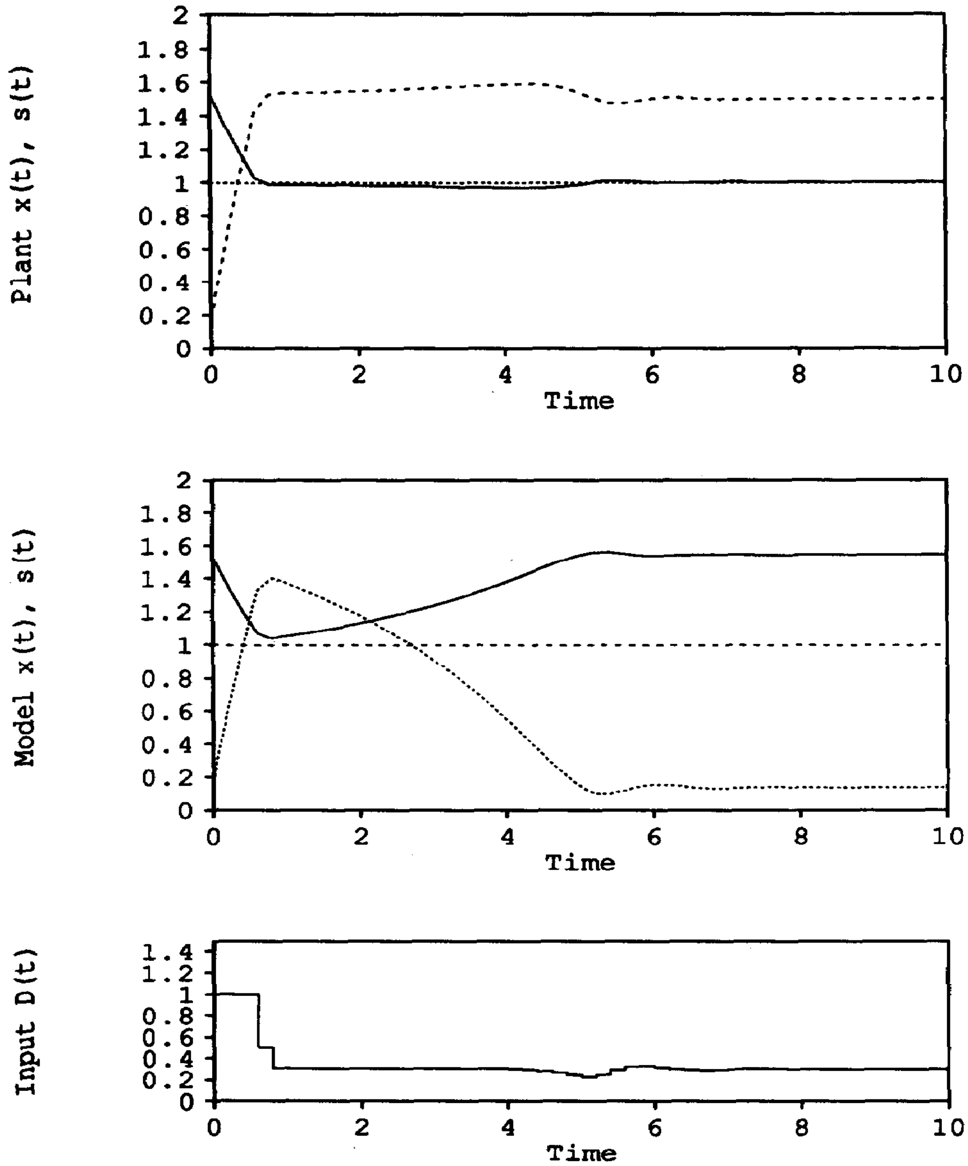

Fig. 12. Performance of nonlinear model-predictive controller without resetting the model initial condition for an open-loop unstable target. The model prediction reaches a different (stable) steady state than the plant. Biomass (———) and substrate (. . . . .) concentrations (top and middle) and dilution rate (bottom) for the parameters given in Table 1.

In fact, the model has moved away from the intended set point and has come to rest on the other *stable* steady state. Had this sytem not had a second stable steady state, the difference between the model prediction and the plant output would have grown until the controller eventually failed from numerical difficulties.

In order to overcome this problem, the control scheme is modified to reset the initial condition of the model to the measurements of the states at each time step. As Fig. 13 shows, the resulting system is stable, and the model does not wander off to some other part of the operating region. Although both states are assumed to be measured and measurement noise is assumed negligible this is not always true in practice. In such cases, it is reasonable to assume that the model initial conditions may be obtained from a state estimation scheme.

4. CONCLUSIONS

The model-predictive control framework is versatile; it handles nonlinearities, input constraints, nonminimum phase and unstable plants, batch and continuous problems, and complex performance objective functions. It is being used widely in the process industries. The controller is straightforward to design

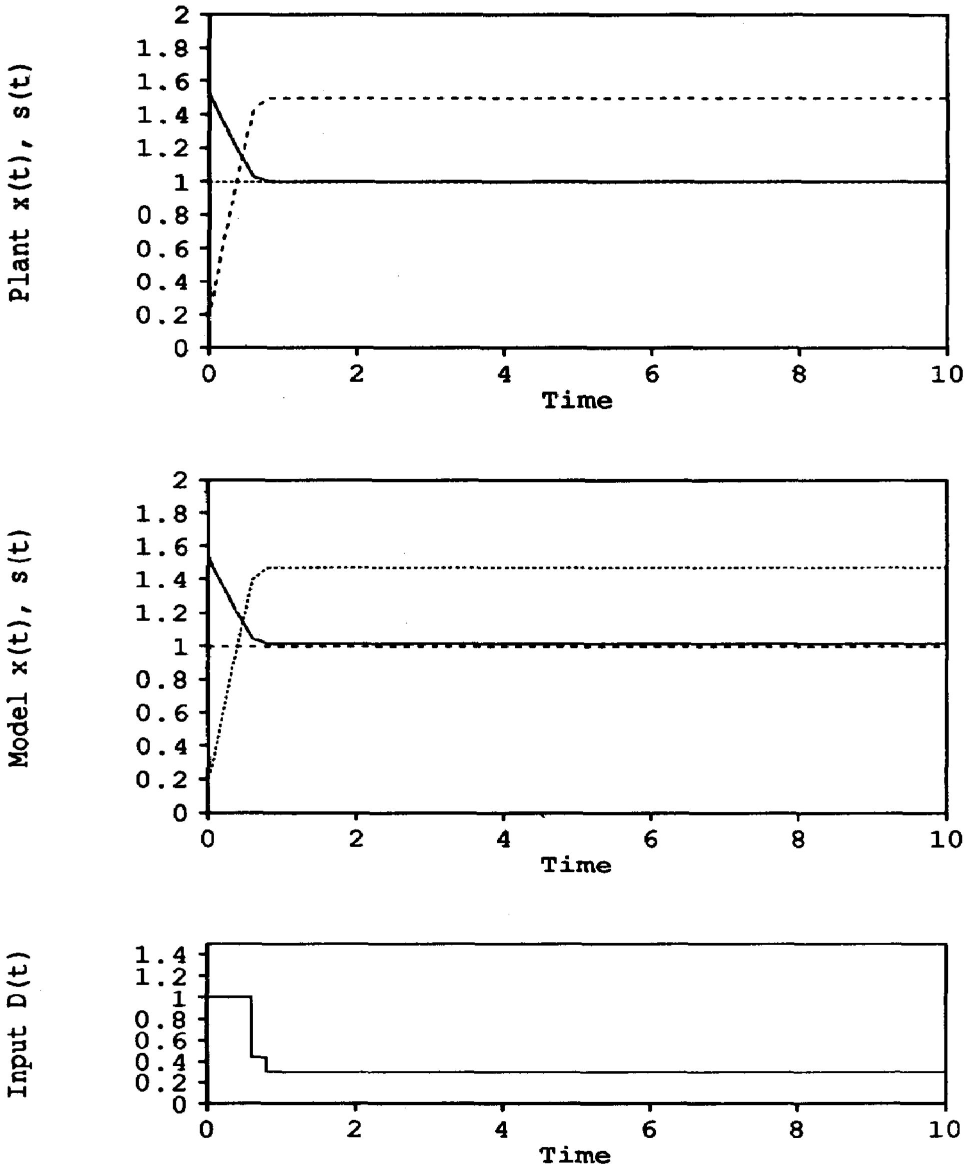

Fig. 13. Performance of nonlinear model-predictive controller with resetting the model initial condition for an open-loop unstable target. The model prediction remains near the plant output. Biomass (———) and substrate (. . . . .) concentrations (top and middle) and dilution rate (bottom) for the parameters given in Table 1.

since it optimizes an open-loop performance objective.

The prediction horizon allows one to include in the control system any planned set-point changes and forecasts of disturbances. This feature was shown to be beneficial for nonminimum phase systems and systems with input constraints.

One approach that may be used to stabilize open-loop unstable plants is to reset the model initial conditions to the plant output after each measurement. This also prevents the plant and model outputs from moving to *different* steady states in nonlinear systems with multiple steady states.

In principle, for a particular plant and a controller performance objective, one can find the best input trajectory, using optimal control methods. The advantage of MPC is that it does this automatically for a wide assortment of problems and builds in the necessary feedback to handle the modelling errors.

Acknowledgements—The authors would like to thank Professor K. J. Åström for several insightful discussions during his one-year visit to the University of Texas. We would also like to acknowledge an anonymous reviewer for several helpful comments. Financial support from Shell Development, Du Pont, Kodak, and the National Science Foundation under Grant No. CTS-8957123 is also gratefully acknowledged.

NOTATION

a	real RHP zero location
A	dynamic matrix
A_1	first column of the dynamic matrix
c	controller transfer function
C	control horizon
d	estimated disturbance in the DMC algorithm
d^*	vector of projected disturbances over the prediction horizon in the DMC algorithm
D	dilution rate
e	error, $r - y$ in Fig. 1, error vector, $r - y_0 - d^*$ in Fig. 2
F	final-time state penalty parameter
g	plant transfer function
H	unit step function, Hessian matrix
k_1	constant in the substrate inhibition model
k_m	constant in the substrate inhibition model
K	feedback gain matrix
M	plant model
P	plant
q^{-1}	backward shift operator
Q	state (or output) penalty parameter
r	set point
R	input penalty parameter
s	Laplace transform variable, substrate concentration
S	shift matrix in the DMC algorithm
t	time
T	prediction horizon
u	manipulated variable or input
x	state, cell mass concentration
y	plant output
$\tilde{y}$	model output
$\tilde{y}^*$	vector of model outputs over the prediction horizon (Fig. 2)
y_0	vector of projected outputs over the prediction horizon (Fig. 2)

Greek letters

Γ	output penalty parameter in the DMC algorithm
ε	extrapolation vector in the matrix S in the DMC algorithm
θ	time of set-point change
θ_p	time delay
Λ	input penalty parameter in the DMC algorithm
μ	specific growth rate
Φ	controller performance objective function

REFERENCES

Agrawal, P. and Lim, H. C., 1984, *Analyses of Various Control Schemes for Continuous Bioreactors*, Vol. 31, pp. 61–90. Springer, Berlin.

Bless, R. R. and Hodges, D. H., 1990, Finite element solution of optimal control problems with inequality constraints, in *Proceedings of the* 1990 *American Control Conference*, pp. 242–247, May 1990.

Brengel, D. D. and Seider, W. D., 1989, Multistep nonlinear predictive controllers, in *Proceedings of the* 1989 *American Control Conference*, pp. 1100–1101, June 1989.

Brosilow, C. and Cheng, C. M., 1987, Model predictive control of unstable systems. AIChE National Meeting, New York, November 1987.

Brusch, R. G., 1974, A nonlinear programming approach to space shuttle trajectory optimization. *J. Optimization Theory Appl.* **13**, 94–118.

Clarke, D. W., Mohtadi, C. and Tuffs, P. S., 1987, Generalized predictive control—Part I. The basic algorithm. *Automatica* **23**, 137–148.

Cutler, C. R. and Ramaker, B. L., 1979, Dynamic matrix control—a computer control algorithm. AIChE National Meeting, Houston, TX, April 1979.

Eaton, J. W. and Rawlings, J. B., 1990, Feedback control of chemical processes using on-line optimization techniques. *Comput. Chem. Engng* **14**, 469–479.

García, C. E. and Morari, M., 1982, Internal model control. 1. A unifying review and some new results. *Ind. Engng Chem. Process Des. Dev.* **21**, 308–323.

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

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

Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H., 1986, User's guide for SOL/NPSOL (Version 4.0): a Fortran package for nonlinear programming. Technical Report SOL 86-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

Hidalgo, P. M. and Brosilow, C. B., 1990, Nonlinear model predictive control of styrene polymerization at unstable operating points. *Comput. Chem. Engng* **14**, 481–494.

Jänsh, C. and Paus, M., 1990, Aircraft trajectory optimization with direct collocation using movable gridpoints, in *Proceedings of the* 1990 *American Control Conference*, pp. 262–267, May 1990.

Johnson, I. L., 1975, Optimization of the solid-rocket assisted space shuttle ascent trajectory. *J. Spacecraft* **12**, 765–769.

Kleinman, D. L., 1970, An easy way to stabilize a linear constant system. *IEEE Trans. autom. Control* **15**, 692.

Kwon, W. H., Bruckstein, A. M. and Kailath, T., 1983, Stabilizing state-feedback design *via* the moving horizon method. *Int. J. Control* **37**, 631–643.

Kwon, W. H. and Pearson, A. E., 1977, A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans. autom. Control* **22**, 838–842.

Lee, E. B. and Markus, L., 1967, *Foundations of Optimal Control Theory*. Wiley, New York.

Lee, J. H., Morari, M. and García, C. E., 1990, State–space interpretation of model predictive control. *Automatica* (submitted).

Mayne, D. Q. and Michalska, H., 1990, Receding horizon control of nonlinear systems. *IEEE Trans. autom. Control* **35**, 814–824.

Mehra, R. K. and Rouhani, R., 1980, Theoretical considerations on model algorithmic control for nonminimum phase systems, in *Proceedings of the Joint Automatic Control Conference.*

Mehra, R. K., Rouhani, R., Eterno, J., Richalet, J. and Rault, A., 1982, Model algorithmic control: review and recent development, in *Engineering Foundation Conference on Chemical Process Control* II, pp. 287–310.

Morari, M. and Zafiriou, E., 1989, *Robust Process Control.* Prentice-Hall, Englewood Cliffs, NJ.

Patwardhan, A. A., Rawlings, J. B. and Edgar, T. F., 1990, Nonlinear model predictive control. *Chem. Engng Commun.* **87**, 123–141.

Prett, D. M., García, C. E. and Ramaker, B. L. (Eds), 1989, *The Second Shell Process Control Workshop: Solutions to the Shell Standard Control Problem.* Butterworths, Boston.

Prett, D. M. and Gillette, R. D., 1979, Optimization and constrained multivariable control of a catalytic cracking

unit. AIChE National Meeting, Houston, TX, April 1979.
Propoi, A. I., 1963, Use of linear programming methods for synthesizing sampled-data automatic systems. *Autom. Remote Control* **24**, 837–844.
Psiaki, M. L. and Park, K., 1990, Trajectory optimization for real-time guidance: Part 1, time-varying LQR on a parallel processor, in *Proceedings of the* 1990 *American Control Conference*, pp. 248–253, May 1990.
Richalet, J., Rault, A., Testud, J. L. and Papon, J., 1978, Model predictive heuristic control: applications to industrial processes. *Automatica* **14**, 413–428.
Ricker, N. L., 1985, Use of quadratic programming for constrained internal model control. *Ind. Engng Chem. Process Des. Dev.* **24**, 925–936.
Schmid, C. and Biegler, L. T., 1990, Application of multistep Newton-type controllers to fluid catalytic cracking, in *Proceedings of the* 1990 *American Control Conference*, pp. 581–586, May 1990.
Thomas, Y. A., 1975, Linear quadratic optimal estimation and control with receding horizon. *Electron. Lett.* **11**, 19–21.