A State Space Formulation for Model Predictive Control

Model predictive control (MPC) schemes such as MOCCA, DMC, MAC, MPHC, and IMC use discrete step (or impulse) response data rather than a parametric model. They predict the future output trajectory of the process $\{\hat{y}(k+i), i=1, \ldots, P\}$, then the controller calculates the required control action $\{\Delta u(k+i), i=0, 1, \ldots, M-1\}$ so that the difference between the predicted trajectory and user-specified (setpoint) trajectory is minimized. This paper shows how the step (impulse) response model can be put into state space form thus reducing computation time and permitting the use of state space theorems and techniques with any of the above-mentioned MPC schemes. A series of experimental runs on a simple pilot plant shows that a Kalman filter based on the proposed state space model gives better performance that direct use of the step response data for prediction.

Sifu Li
Kian Y. Lim
D. Grant Fisher
Department of Chemical Engineering
University of Alberta
Edmonton, Canada T6G 2G6

Introduction

In this paper, the generic term "model predictive control" (MPC) is used to define the class of control techniques which include: MPHC (model predictive heuristic control) (Richalet et al., 1978); DMC (dynamic matrix control) (Cutler and Ramaker, 1980); MAC (model algorithmic control) (Rouhani and Mehra, 1982); IMC (internal model control) (Garcia and Morari, 1982); and MOCCA (multivariable, optimal, constrained control algorithm) (Sripada and Fisher, 1985). Each of these control schemes differs in detail but includes the following key features as illustrated in Figure 1:

1. The future outputs $Y_m^*(k) = \{y_m^*(k+i|k), i=0, 1, \ldots, P\}$ are predicted using a set of discrete step or impulse response coefficients rather than a typical state space or transfer function model.

2. A "correction term" for each element of the predicted output is usually calculated to account for the difference between the estimated value and the measured plant output. This correction can be calculated using known disturbance response data, by identifying parameters on-line to permit forecasting of future values or by estimation techniques such as a Kalman filter. The model based value, $Y_m^*(k)$, and the correction term, $R(k)$, are combined to produce the estimated trajectory $\hat{Y}(k) = \{\hat{y}(k+i|k), i=1, \ldots, P\}$.

3. A predictive control strategy is used to calculate the control action $\{\Delta u(k+i), i=0, 1, \ldots, M-1; M \leq P\}$ which minimizes a user-specified performance index, e.g., which minimizes the square of the difference between the desired trajectory $Y_d(k)$ and predicted trajectory $\hat{Y}(k)$ that would result if no further control action were taken.

4. The control calculation is usually formulated as an optimization problem: linear or nonlinear; weighted or unweighted; constrained or unconstrained. Depending on the characteristics of the optimization problem, the solution may require anything from simple off-line calculations to on-line, constrained, nonlinear optimization.

MPC is popular in industry and academia because it:

- Uses step response data which is relatively easy to obtain
- Is multivariable
- Handles time delays and mild nonlinearities
- Optimizes over a trajectory
- Can handle hard constraints
- Has demonstrated its effectiveness in industrial applications over the past ten years (Richalet et al., 1978; Cutler and Ramaker, 1980; Froisy and Richalet, 1986; Cutler and Hawkins, 1988; Kelly et al., 1988)

This paper is concerned primarily with the step of estimating the future process output trajectory, $\hat{Y}(k)$, based on the knowledge of the past control actions $\{\Delta u(k-i), i=1, \ldots, N\}$ and the measured process output, $y_p(k)$. It is shown that the process model and the problem of estimating $\hat{Y}(k)$ can be expressed in standard state space form. This permits the used of state space

S. Li is on leave from Department of Science and System Management, University of Science and Technology, Hefei, People's Republic of China.
Correspondence concerning this paper should be addressed to D. G. Fisher.

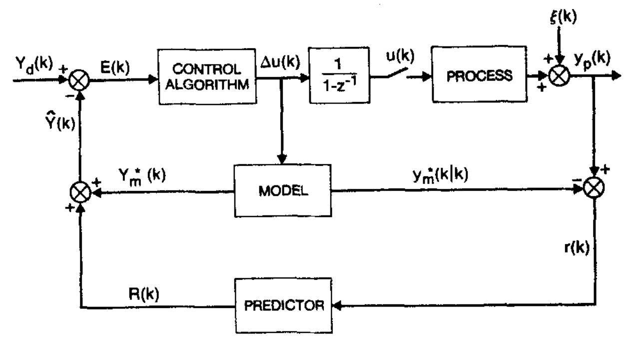

Figure 1. Typical model predictive control (MPC) system.

theory and can simplify the computation and/or improve the performance of the overall control system. Experimental results from a simple pilot-plant process are included to demonstrate the advantage of the proposed approach.

MPC with Nonparametric Process Model

A quantitative input/output relationship for a multivariable process can be developed based on discrete step response data. Assume that the process is initially at steady state ($y = u = 0$) and that a unit step change in the input, $u(t)$, is introduced at time k. The discrete unit step response of the output can be represented by:

$$\begin{aligned} y(k+i|k) &= a_i \quad i = 1, 2, \ldots, N \\ &= a_{ss} \quad i > N \end{aligned} \tag{1}$$

where $y(k+i|k)$ means that the outputs $y(k+i)$ at time k are based on all process inputs available up to and including time k-1. The integer N and the sampling time Δt are chosen by the user so that Eq. 1 is an adequate characterization of the process response and a_N is reasonably close to the final steady state value a_{ss}. Note that the output at any chosen time interval k is a function only of the changes in input, Δu, over the past N intervals *and of the input* $(N+1)$ *intervals ago*. For example, assuming superposition,

$$y(k|k) = \sum_{j=1}^{N} a_j \Delta u(k-j) + a_{ss} u(k-N-1) \tag{2}$$

where $y(\cdot)$ and $u(\cdot)$ are deviation variables such that at the initial steady state y and u are zero.

Predicted output trajectory

The future output trajectory $\{y_m(k+i), i = 1, \ldots, P; P \leq N\}$ can be predicted using the relationship:

$$y_m(k+i|k+i) = \sum_{j=1}^{N} a_j \Delta u(k+i-j) + a_{ss} u(k+i-N-1) \quad i = 1, \ldots, P \tag{3}$$

For control applications it is convenient to break Eq. 3 into two parts involving: 1) present and future (unknown) inputs and 2) past (known) inputs. thus, the effect of future (unknown) inputs, $\{\Delta u(k+i), i = 0, 1, \ldots, M-1\}$, on the process output trajectory is:

$$\begin{bmatrix} y_m(k+1|k+1) \\ y_m(k+2|k+2) \\ \vdots \\ y_m(k+P|k+P) \end{bmatrix} = \begin{bmatrix} y_m^*(k+1|k) \\ y_m^*(k+2|k) \\ \vdots \\ y_m^*(k+P|k) \end{bmatrix} + A_2 \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+M-1) \end{bmatrix} \tag{4}$$

where the matrix A_2 is given by:

$$A_2 = \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_M & a_{M-1} & \cdot & \cdot & a_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_P & a_{P-1} & \cdots & \cdots & a_{P-M+1} \end{bmatrix}_{PxM}$$

The contribution to the future output trajectory due to past input changes (i.e., the predicted output trajectory if $\{\Delta u(k+i), i = 0, 1, \ldots, M-1\}$, are zero) is given by:

$$\begin{bmatrix} y_m^*(k+1|k) \\ y_m^*(k+2|k) \\ \vdots \\ y_m^*(k+P|k) \end{bmatrix} = a_{ss} \begin{bmatrix} u(k-N) \\ u(k-N+1) \\ \vdots \\ u(k-N+P-1) \end{bmatrix} + A_1 \begin{bmatrix} \Delta u(k-N+1) \\ \Delta u(k-N+2) \\ \vdots \\ \Delta u(k-1) \end{bmatrix} \tag{5}$$

where the matrix A_1 is given by:

$$A_1 = \begin{bmatrix} a_N & a_{N-1} & \cdots & \cdots & \cdots & a_2 \\ 0 & a_N & a_{N-1} & \cdots & \cdots & a_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & a_N & \cdots & a_{P+1} \end{bmatrix}_{Px(N-1)}$$

or in a more compact notation

$$Y_m = A_2 \Delta U_f + Y_m^* \tag{4a}$$

$$Y_m^* = A_1 \Delta U_p + a_{ss} U_p \tag{5a}$$

where the subscripts p and f on the inputs, U, specify past and future values respectively. The equations for multiinput (e.g., disturbances) single output (MISO) systems follow directly from Eqs. 4 and 5. Multivariable systems are discussed later.

Impulse response model

If impulse response data are used in place of the step response data, Eqs. 1 through 5 are still valid if:

1. a_i is replaced by h_i where $h_i = a_i - a_{i-1}$ are the impulse reponse coefficients and $h_1 = a_1$
2. $\Delta u(k+i)$ is replaced by $u(k+i)$
3. a_{ss} is replaced by h_{ss} where $h_{ss} = a_{ss} - a_N$

Control action

The task of the controller is to calculate the present and future control action, $\Delta U_f(k) = \{\Delta u(k+i), i = 0, 1, \ldots, M-1; M \leq P \leq N\}$ such that the predicted trajectory satisfies a user-specified performance index, e.g., follows the desired trajectory. Several different performance indices can be specified. One of the simplest and most familiar is the weighted least squares performance index:

$$J = \frac{1}{2}\{[Y_d(k) - Y_m(k)]^T\Gamma(k)[Y_d(k) - Y_m(k)] + [\Delta U_F(k)]^T\Gamma_m(k)[\Delta U_F(k)]\} \quad (6)$$

where Γ and Γ_m are the weighting matrices on the output deviations and the changes in the input variable, respectively. Other linear or nonlinear performance measures can also be used. Given Eq. 6, the next step is to use Eq. 4a to express Y_m as the sum of two terms: Y_m^* due to past inputs and $A_2\Delta U_f$ due to $\Delta u(k)$ plus future control action. If the error trajectory at time k is defined as (cf. Figure 1):

$$E(k) = [Y_d(k) - \hat{Y}(k)] \quad (7)$$

then the performance index, Eq. 6, becomes

$$J = \frac{1}{2}\{[Y_d(k) - Y_m(k)]^T\Gamma(k)[Y_d(k) - Y_m(k)] + [\Delta U_F(k)]^T\Gamma_m(k)[\Delta U_F(k)]\} \quad (6)$$

Calculation of ΔU_f to minimize J is a standard weighted least squares problem and the result is

$$\Delta U_f = A^*E(k) \quad (9)$$

$$A^* = [A_2^T\Gamma A_2 + \Gamma_m]^{-1}A_2^T\Gamma \quad (10)$$

where A^* (the inverse or pseudoinverse of A_2) is a constant matrix calculated off-line. If $\Delta u(k)$ is calculated and implemented at each interval, only the first row of A^* is needed for these calculations. (Note that M values of Δu are included in the minimization but most users choose to implement only $\Delta u(k)$ and repeat the entire optimization at each control interval.) Constraints can be handled by converting the control calculation to an on-line, linear or nonlinear, optimization problem, e.g., linear or quadratic programming (Prett and Gillette, 1980; Morshedi et al., 1985; Ricker, 1985; Morshedi, 1986; Campo and Morari, 1986).

MPC with State Space Model

A recursive relationship for estimating current and future values of the process output $\{y(k+i), i = 0, 1, \ldots, N\}$ can be developed as follows. Assume that at some arbitrary point in time $(k-1)$ that the $N+1$ elements of the state vector $X(k-1)$ are known. [Note that the $X(\cdot)$ vector is defined as the current process output plus estimates of the next N values.] Thus, $X(k-1)$ could represent an initial steady state or could be calculated using Eq. 4 and is represented by the top line in Figure 2. Now assume that a change in the input variable $\Delta u(k-1)$ is made and that it is desired to estimate the new output trajectory $X(k)$. From Figure 2 and the definition of the step response data, it follows that

$$\begin{aligned} x_1(k) &= x_2(k-1) + a_1\Delta u(k-1) \\ x_2(k) &= x_3(k-1) + a_2\Delta u(k-1) \\ \cdots \quad & \quad \cdots \quad \quad \cdots \end{aligned} \quad (11)$$

Extending Eq. 11 to the entire trajectory, $i = 1, \ldots, N+1$ gives

$$x_i(k) = x_{i+1}(k-1) + a_i\Delta u(k-1) \quad (12)$$

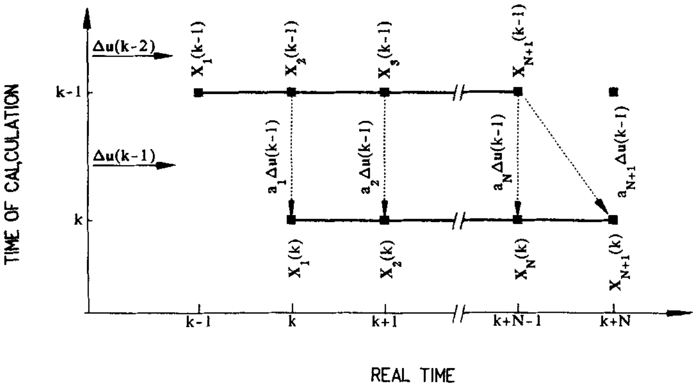

Figure 2. Calculation of the output trajectory vector $X(K) = x_i(k) = y(k+i-1), i = 1, \ldots, N+1$ based on the known trajectory, $X(k-1)$, and step response coefficients a_i.

[Expressing Eq. 12 in terms of the output values a time k gives

$$y_m^*(k+i-1|k) = y_m^*(k+i-1|k-1) + a_i \Delta u(k-1) \quad (12b)$$

where the first argument of y_m^* is the "real time" from Figure 2 and the second argument is the time of the calculation.] However, by definition of the step response (Eq. 1), the effect of $\Delta u(k-i)$ on the states $x_i(k)$ for $i \geq N$ is zero. Therefore, $a_i = a_{ss}$ for $i \geq N$ and

$$x_{N+2}(k-1) = x_{N+1}(k-1) \quad (13)$$

Equations 12 and 13 can be put in the more compact vector/matrix notation:

$$X(k) = \phi X(k-1) + \theta \Delta u(k-1) \quad (14)$$

where

$$X(k) = [y_m^*(k|k) \quad y_m^*(k+1|k) \ldots y_m^*(k+N|k)]^T_{(N+1)\times 1}$$

$$X(k-1) = [y_m^*(k-1|k-1) \quad y_m^*(k|k-1) \ldots y_m^*(k+N-1|k-1)]^T_{(N+1)\times 1}$$

$$\phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}_{(N+1)\times(N+1)}$$

$$\theta = [a_1 \quad a_2 \ldots a_N \quad a_{ss}]^T_{(N+1)\times 1}$$

The output equation, $Y(k) = y_m(k|k)$ (i.e., the current estimate) can be obtained by using:

$$Y(k) = HX(k) \quad (15)$$

where

$$H = [1 \quad 0 \quad 0 \ldots 0]_{1\times(N+1)}$$

Note that Eqs. 14 and 15 have the same form as a state space model and that ϕ, θ, and H have very special (sparse) forms.

Reduced-Order state space form of process model

The dimension of the state vector in Eq. 14 can be reduced from $N+1$ to $P+1$ as shown in Appendix A. Equation 14 is still valid if the vectors and matrices are defined as:

$$X(k) = [y_m^*(k|k) \quad y_m^*(k+1|k) \ldots y_m^*(k+P|k)]^T_{(P+1)\times 1} \quad (16)$$

$$\Delta U(k-1) = [\Delta u(k-N+P-1) \ldots \Delta u(k-1)]^T_{(N-P+1)\times 1} \quad (17)$$

$$\phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}_{(P+1)\times(P+1)} \quad (18)$$

$$\theta = \begin{bmatrix} 0 & \cdots & \cdots & 0 & a_1 \\ 0 & \cdots & \cdots & 0 & a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{ss} & h_N & \cdots & h_{P+2} & a_{P+1} \end{bmatrix}_{(P+1)\times(N-P+1)} \quad (19)$$

Note that the input term in Eq. 14 becomes a matrix/vector product rather than a vector/scalar product and the output equation remains the same. The lower dimension of the state vector simplifies the implementation of state space systems, e.g., Kalman filters. Note also that ΔU rather than Δu is used in the following discussion.

State space form using impulse response data

Equation 14 is valid if the a_i are replaced by h_i, a_{ss} by h_{ss} and $\Delta u(\cdot)$ by $u(\cdot)$. In the reduced order model, a similar model using impulse response can be obtained using the procedure in Appendix A.

Multiinput/Multioutput (MIMO) model

The multivariable state space model with r output and s input variables can be constructed by simply redefining the vectors and matrices as block vectors and matrices:

$$X(k) = [X^1(k), \ldots, X^r(k)]^T \quad (20)$$

$$\phi = \begin{bmatrix} \phi_{11} & 0 & \cdots & \cdots & 0 \\ 0 & \phi_{22} & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdots & \cdots & 0 & \phi_{rr} \end{bmatrix} \quad (21)$$

$$\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \cdots & \theta_{1s} \\ \theta_{21} & \theta_{22} & \cdots & \cdots & \theta_{2s} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \theta_{r1} & \theta_{r2} & \cdots & \cdots & \theta_{rs} \end{bmatrix} \quad (22)$$

$$H = \begin{bmatrix} H_{11} & 0 & \cdots & \cdots & 0 \\ 0 & H_{22} & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdots & \cdots & 0 & H_{rr} \end{bmatrix} \quad (23)$$

The contents of the block matrices depend on the type of model used, i.e., full or reduced order. The MIMO input vector for the full and reduced order model is:

$$\Delta U(k-1) = [\Delta u^1(k-1), \ldots, \Delta u^s(k-1)]^T \quad (24)$$

and

$$\Delta U(k-1) = [\Delta U^1(k-1), \ldots, \Delta U^s(k-1)]^T \quad (25)$$

respectively. The process model defined by Eqs. 14 and 15 or Eqs. 16 to 19 is a nonminimal state space model. Thus, many of the theorems and results from the state space area can be applied to interpret and improve model predictive control schemes such as MOCCA (Sripada and Fisher, 1985), MAC (Rouhani and Mehra, 1982), and DMC (Cutler and Ramaker, 1980). In the following sections we will look at computational efficiency, observability, controllability, and the Kalman filter.

Computational efficiency of prediction

Control schemes such as the first version of MOCCA (Sripada and Fisher, 1985) use Eq. 5 to predict the effect of past inputs on the future output trajectory. The recursive equations represented by Eqs. 14 and 16 to 19 offer a much more efficient alternative. For a SISO system with $N = 40$ and $P = M = 10$, Eq. 5 requires 396 multiplications and 385 additions. The recursive relationship in Eq. 14 requires 41 multiplications and 41 additions. The reduced-order recursive relationship in Eqs. 16 to 19 requires 41 multiplications and 51 additions. Since the prediction must be made at every control interval, the computational advantages of Eqs. 14 and 16 to 19 over Eq. 5 is significant.

Observability and controllability

The observability matrix for the SISO system represented by Eqs. 14 and 15 is

$$[H, H\phi, \ldots, H\phi^{N+1}]$$

It can be shown that the rank of the observability matrix is $N + 1$. Thus, the system is completely observable and the state (cf. predicted output trajectory) can be reconstructed from the measured output $y_p(k)$ and the known input.

It can also be shown that the controllability matrix for the SISO system

$$[\Gamma, \Gamma\phi, \ldots, \Gamma^{N+1}\phi]$$

is nonsingular, hence the system is controllable.

Open-loop predictor

The future outputs of the process cannot be measured directly but the state vector $X(k) = \{y_m^*(k+i|k), i = 0, 1, \ldots, P\}$ can be reconstructed by means of a simple open-loop predictor defined by:

$$\overline{X}(k) = \phi\overline{X}(k-1) + \theta\Delta U(k-1) \quad (26)$$

$$\hat{X}(k) = \{\overline{X}(k) + K[y_p(k) - y_m^*(k|k)]\} \quad (27)$$

$$\hat{Y}(k) = H_P\hat{X}(k) \quad (28)$$

where

$$y_m^*(k|k) = S\overline{X}(k).$$

$$H_P = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix}_{P\times(P+1)}$$

It is used to extract $\{y_m^*(k+i|k), i = 1, \ldots, P\}$ from the vector $\hat{X}(k)$.

$$K = [k_1, \ldots, k_P, k_{P+1}]^T_{(P+1)\times 1}$$

$$S = [1, 0, \ldots, 0]_{1\times(P+1)}$$

is used to extract $y_m^*(k|k)$ from $\overline{X}(k)$.

$\hat{Y}(k)$ is the corrected, model-based, predicted trajectory of the process.

The predictor corresponding to Eqs. 26 to 28 (with $k_i = 1$ for simplicity) is shown in Figure 3. This simple form of linear predictor is used in MPC schemes such as the simpler formulations of MOCCA (Sripada and Fisher, 1985) and DMC (Cutler and Ramaker, 1980) typically with $k_i = 1$.

Kalman filter

When process noise, $w(k)$ and measurement noise, $v(k)$ are present, then a Kalman filter will generally give better performance than a deterministic predictor. The state space form of the process model, Eq. 14 with the noise terms included is given by:

$$X(k) = \phi X(k-1) + \theta\Delta u(k-1) + Dw(k-1) \quad (29)$$

$$Y(k) = HX(k) + v(k) \quad (30)$$

If $w(k)$ and $v(k)$ are independent white noise sequences, the standard Kalman Filter design procedure (Åström, 1970;

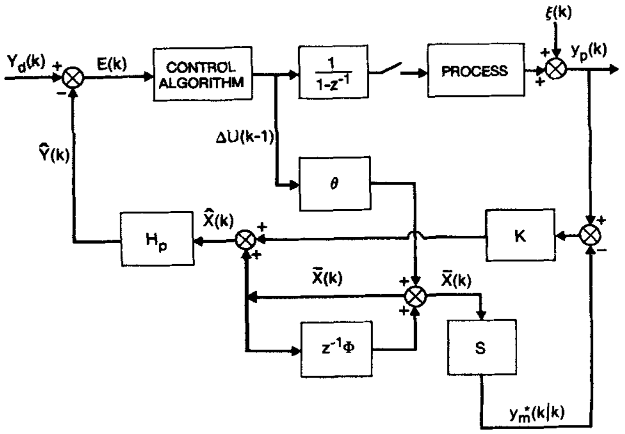

Figure 3. Model predictive control with open-loop prediction of the output trajectory (equivalent to Figure 1 and DMC).

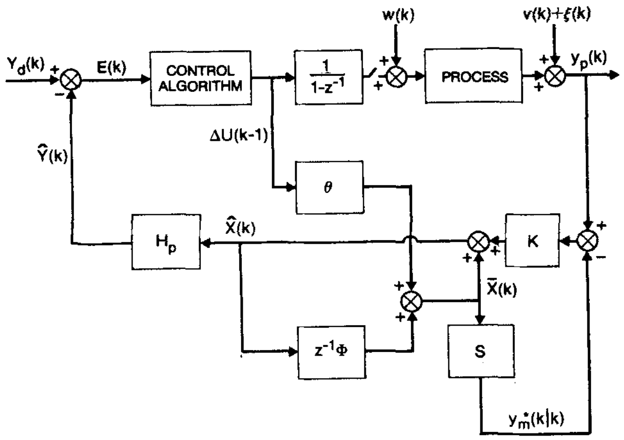

Figure 4. Model predictive control using a Kalman filter for prediction of the future output trajectory, $\hat{Y}(k)$, of a system with process and measurement noise.

Åström and Wittenmark, 1984; Goodwin and Sin, 1984) leads to

Time Update of the State

$$\bar{X}(k) = \phi\hat{X}(k-1) + \theta\Delta U(k-1) \tag{31}$$

Time Update of Covariance

$$M(k) = \phi P(k-1)\phi^T + DR_wD^T \tag{32}$$

Gain Calculation

$$K(k) = M(k)H^T[HM(k)H^T + R_v]^{-1} \tag{33}$$

Measurement Update of the State and Prediction

$$\hat{X}(k) = \bar{X}(k) + K(k)[y_p(k) - y_m^*(k|k)] \tag{34}$$

$$\hat{Y}(k) = H_p\hat{X}(k) \tag{35}$$

Measurement Update of Covariances

$$P(k) = M(k) - K(k)HM(k) \tag{36}$$

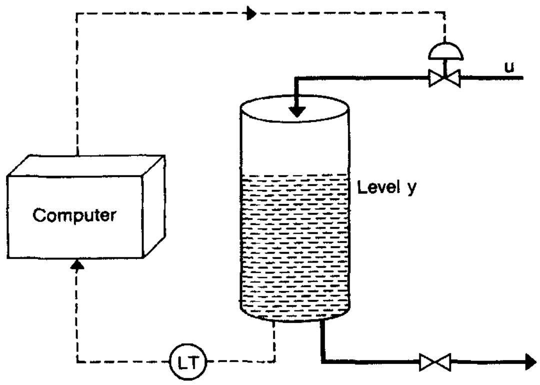

Figure 5. Simplified diagram of process system used for experimental evaluation of MPC.

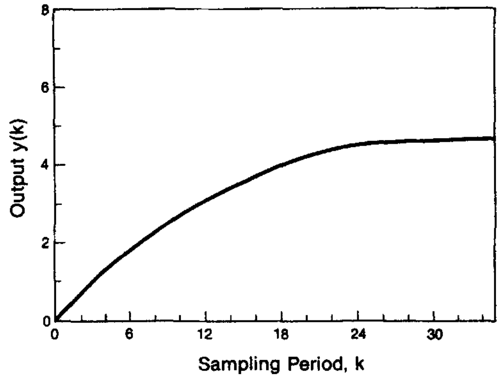

Figure 6. Smoothed step response of the process shown in Figure 5.

where

R_w = covariance of $w(t)$

R_v = covariance of $v(t)$

$M(k)$ = a priori error covariance

$P(k)$ = a posteriori error covariance

$K(k)$ = Kalman gain

Note that the matrix D is often unknown. However, the only place it appears is in Eq. 32. In practice, this is compensated by "tuning" using R_w. In most applications, a steady state Kalman Filter will give satisfactory performance. In this case $K(k)$ is constant and can be calculated *off-line* by integration of Eqs. 32, 33 and 36. The state estimation is then reduced to Eqs. 31, 34 and 35 as illustrated by Figure 4.

Experimental Results

The difference between simple open-loop prediction (Figures 1 and 3) and techniques such as the Kalman filter (Figure 4) which are feedback plus filtering are demonstrated by a series of experimental runs on a single input/single output level control process which includes standard industrial instrumentation and a minicomputer system designed for industrial real time control applications. The experimental system shown in Figure 5 consists of a tank in which the level is controlled by manipulation of the inlet flow rate. The first-order level system is mildly non-

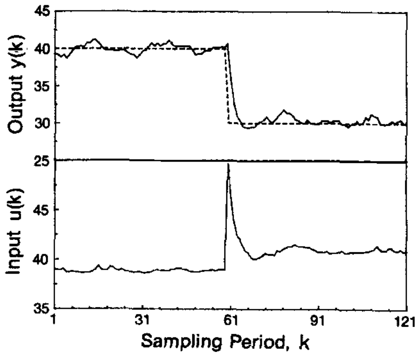

Figure 7. Servo control using MPC with the Kalman filter predictor in Figure 4.

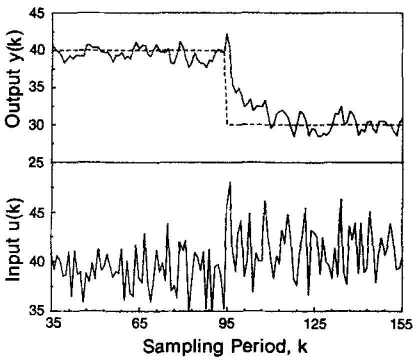

Figure 8. Servo control using MPC and the simple open loop predictor in Figure 1 or 3 (cf. Figure 7).

linear, and both the process gain and time constant change by a factor of about two as the level is changed from low to high. The step response of the level to a unit change in inlet flow is shown in Figure 6. N was chosen as 40.

In all the runs in this paper, the reference trajectory was a simple step from the current value to the desired final setpoint value. The weighted least squares objective function in Eq. 6 was used with weighting of $\Gamma = 1.0I$ on the output variable and $\Gamma_m = 0.5I$ on the manipulated variable where I is the identity matrix. The controller design was based on $N = 40$, $P = 40$, $M = 5$ with $\Delta t = 8$ seconds.

Kalman filter vs. open-loop predictor

To demonstrate MPC performance in the presence of noise, the system input $u(k)$ and the measurement of the system output $y_p(k)$ were modified by adding zero mean white noise with covariances of $R_w = 0.25$ and $R_v = 0.40$, respectively. Figure 7 shows a very fast and smooth response to a 25% decrease in setpoint using a Kalman filter.

For comparison, the response using an open-loop predictor (Figures 1 and 3) is plotted in Figure 8. Note that the simplest DMC or MOCCA formulation with $\{\hat{r}(k+i|k) = \hat{r}(k), i = 1, \ldots, P\}$ can be interpreted as an open-loop predictor of the type shown in Figure 3. The response is much slower and significantly noisier, particularly the manipulated variable.

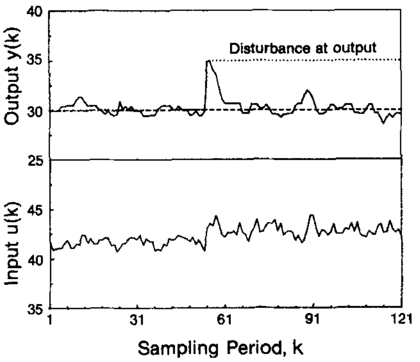

Figure 9. Regulatory control using MPC with a Kalman filter predictor.

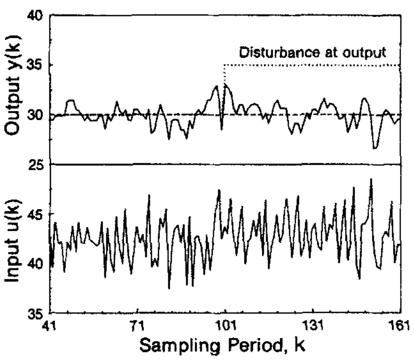

Figure 10. Regulatory control using MPC with a simple open-loop predictor (cf. Figure 9).

The performance using the Kalman filter and the open-loop predictor in the presence of a 15% step change in the disturbance, $\xi(k)$, is shown in Figures 9 and 10, respectively. The performance of the system using the *KF* is again clearly superior. The performance of the standard MPC in Figures 8 and 10 could be improved by careful tuning of filters added to the process input and output measurements. However, the Kalman filter is optimal under the assumed conditions.

These experimental results demonstrate that, even for a very simple process, state space technique such a Kalman Filter can offer significantly improved performance. However, MPC is normally used on multivariable, constrained processes in industry. Fortunately, state space techniques are easily extended to multivariable processes, and numerous publications over the past two decades have demonstrated that they provide an excellent basis for problem formulation and implementation.

Conclusions

- The step or impulse response models used in MPC can be rearranged into a standard, linear state space form.
- The state space model provides a link between the somewhat *ad hoc* beginnings of MPC and modern state space techniques. For example, the state space model can be used in the prediction step and/or for the design of a Kalman filter to generate a feedback trajectory $\hat{Y}(k)$ in Figure 1.
- Mathematical analysis plus experimental results from a small pilot plant show that there are significant computational savings and improved performance associated with the proposed state space approach.

Notation

A^* = controller gain matrix
a_i = stop response coefficients
$E(k)$ = error trajectory
h_i = impulse response coefficients
$r(k)$ = calculated current residual
$R(k)$ = predicted trajectory of future residuals
$\hat{Y}$ = corrected future output trajectory
Y_d = desired output trajectory
Y_m = future output prediction based on past and future inputs
$y_m^*(k|k)$ = current output prediction based on past inputs
Y_m^* = future output prediction based on past inputs
$\hat{X}$ = estimate of future output trajectory

Greek letters

Γ – output variable weighting matrix
Γ_m – change in input variable weighting matrix
Δu – change in input variable
ΔU_f – current and future change in input variables
ΔU_p – past changes in input variable

Literature Cited

Åström, K. J., *Introduction to Stochastic Control Theory,* Academic Press, New York (1970).

Åström, K. J., and B. Wittenmark, *Computer Controlled Systems: Theory and Design,* Prentice-Hall, Englewood Cliffs, NJ (1984).

Campo, P. J., and M. Morari, "∞-Norm Formulation of Model Predictive Control Problems," *Proceeding American Control Conference,* 339, Seattle (1986).

Cutler, C. R., and B. L. Ramaker, "Dynamic Matrix Control—A Computer Control Algorithm," *Proc. Automatic Control Conf.,* San Francisco, Paper WP5-B (1980); *AIChE Mtg.,* Paper No. 51B, Houston (1979).

Cutler, C. R., and R. B. Hawkins, "Application of a Large Predictive Multivariable Controller to a Hydrocracker Second Stage Reactor," *Proc. Amer. Control Conf.,* Atlanta, 284 (1988).

Froisy, J. B., and J. Richalet, "Industrial Applications of IDCOM," *Proc. Int. Conf. on Chemical Process Control (CPC III),* M. Morari and T. McAvoy, eds., CACHE and Elsevier, Amsterdam (1986).

Garcia, C. E., and M. Morari, "Internal Model Control: 1. A Unifying Review and Some New Results," *Industrial and Engineering Chemistry: Process Design and Development,* **21**(2), 308 (1982).

Goodwin, G. C., and K. S. Sin, *Adaptive Filtering Prediction and Control,* Prentice-Hall (1984).

Kelly, S. J., M. D. Rogers, and D. W. Hoffman, "Quadratic Dynamic Matrix Control of Hydrocracking Reactors," *Proc. Amer. Control Conf.,* Atlanta, 295 (1988).

Morshedi, A. M., C. R. Cutler, and T. A. Skrovanek, "Optimal Solution of Dynamic Matrix Control with Linear Programming (LDMC)," *Proc. Amer. Control Conf.,* Boston, 199 (1985).

Morshedi, A. M., "Universal Dynamic Matrix Control," *Proc. Int. Conf. on Chemical Process Control (CPC III),* M. Morari and T. McAvoy, eds., CACHE and Elsevier, Amsterdam (1986).

Prett, D. M., and R. D. Gillett, "Optimization and Constrained Multivariable Control of a Catalytic Cracking Unit," *Proc. Automatic Control Conf.,* San Francisco, Paper WP5-C (1980).

Richalet, J., A. Rault, J. L. Testud, and J. Papon, "Model Predictive Heuristic Control: Application to Industrial Processes," *Automatica,* **14,** 413 (1978); Original version published in *IFAC Symp. on Identification and System Parameter Estimation,* Tbilisi, Georgian Republic, USSR (1976).

Ricker, N. L., "Use of Quadratic Programming for Constrained Internal Model Control," *Industrial and Engineering Chemistry: Process Design and Development,* **24**(4), 925 (1985).

Rouhani, R., and R. K. Mehra, "Model Algorithmic Control (MAC): Basic Theoretical Properties," *Automatica,* **18,** 401 (1982).

Sripada, N. R., and D. Fisher, "Multivariable Optimal Constrained Control Algorithm (MOCCA): Part 1. Formulation and Application," *Proc. Int. Conf. on Industrial Process Modeling and Control,* **1,** Hangzhou, China (June, 1985).

Appendix: Reduced-Order State Space Model

Assume that the process can be represented by Figure A.1, where the process input/output relationship, G_p, is defined by the known step response data $\{a_1, a_2, \ldots, a_N, a_{ss}\}$ or the equivalent impulse response data $\{h_1, h_2, \ldots, h_N, h_{ss}\}$ where $h_1 = a_1$ and $h_i = a_i - a_{i-1}$. Similarly, let the relationship between the external disturbance, $d(k-1)$, and the output be defined by the known step response data $\{b_1, b_2, \ldots, b_N, b_{ss}\}$ or the equivalent impulse response data $\{g_1, g_2, \ldots, g_N, g_{ss}\}$.

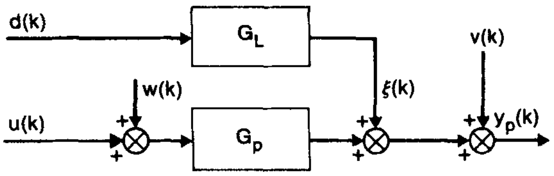

Figure A1. Schematic diagram of process.

From the definition of the step response data (plus the assumption of superposition and linear scaling), it is possible to develop the following equations that estimate future values of the process output. The notation $y_m^*(k|k)$ means the estimated value of the model output at time k is calculated using input values up to and including time $k-1$.

Now assume that at time k, an estimate of the future output trajectory $\{y_m^*(k+i|k-1),\ i = 0, 1, \ldots, P\}$ is known. Also assume that at time k, the input variables $\Delta u(k-1)$ and $\Delta d(k-1)$ are known. [Note that $\Delta d(k)$ may also be known but it usually has no effect on $y_m^*(k|k)$]. Then by definition, the prediction of the output trajectory at time k using $\Delta u(k-1)$ is:

$$\begin{aligned} y_m^*(k|k) &= y_m^*(k|k-1) + a_1\Delta u(k-1) + b_1\Delta d(k-1) \\ y_m^*(k+1|k) &= y_m^*(k+1|k-1) + a_2\Delta u(k-1) + b_2\Delta d(k-1) \\ \cdots \quad & \quad \cdots \quad \cdots \quad \cdots \\ y_m^*(k+P|k) &= y_m^*(k+P|k-1) + a_{P+1}\Delta u(k-1) \\ &\quad + b_{P+1}\Delta d(k-1) \end{aligned} \tag{A1}$$

Using the past contributions of Eq. 5, the following may be written for $y_m^*(k+P|k-1)$ and $y_m^*(k+P-1|k-1)$:

$$\begin{aligned} y_m^*(k+P|k-1) = a_{P+2}&\Delta u(k-2) \\ &+ a_{P+3}\Delta u(k-3) + \cdots \\ &+ a_N\Delta u(k-N+P) \\ &+ a_{ss}u(k-N+P-1) \\ &+ b_{P+2}\Delta d(k-2) \\ &+ b_{P+3}\Delta d(k-3) + \cdots \\ &+ b_N\Delta d(k-N+P) \\ &+ b_{ss}d(k-N+P-1) \end{aligned} \tag{A2}$$

$$\begin{aligned} y_m^*(k+P-1|k-1) = a_{P+1}&\Delta u(k-2) \\ &+ a_{P+2}\Delta u(k-3) + \cdots \\ &+ a_N\Delta u(k-N+P-1) \\ &+ a_{ss}u(k-N+P-2) \\ &+ b_{P+1}\Delta d(k-2) \\ &+ b_{P+2}\Delta d(k-3) + \cdots \\ &+ b_N\Delta d(k-N+P-1) \\ &+ b_{ss}d(k-N+P-2) \end{aligned} \tag{A3}$$

Subtracting Eq. A3 from Eq. A2, and collecting all the similar terms yields:

$$\begin{aligned} y_m^*(k+P|k-1) = y_m^*(k+P-1|k-1) \\ + (a_{P+2} - a_{P+1})\Delta u(k-2) \\ + (a_{P+3} - a_{P+2})\Delta u(k-3) + \cdots \\ + (a_N - a_{N-1})\Delta u(k-N+P-1) \\ + a_N \Delta u(k-N+P-1) \\ + a_{ss}\Delta u(k-N+P-1) \\ + (b_{P+2} - b_{P+1})\Delta d(k-2) \\ + (b_{P+3} - b_{P+2})\Delta d(k-3) + \cdots \\ + (b_N - b_{N-1})\Delta d(k-N+P) \\ + b_N \Delta d(k-N+P-1) \\ + b_{ss}\Delta d(k-N+P-1) \end{aligned} \quad (A4)$$

Now, the difference between step response coefficients can be replaced by impulse response coefficients and the last two terms can also be combined, to give:

$$\begin{aligned} y_m^*(k+P|k-1) = y_m^*(k+P-1|k-1) \\ + h_{P+2}\Delta u(k-2) \\ + h_{P+3}\Delta u(k-3) + \cdots \\ + h_N \Delta u(k-N+P) \\ + h_{ss}\Delta u(k-N+P-1) \\ + g_{P+2}\Delta d(k-2) \\ + g_{P+3}\Delta d(k-3) + \cdots \\ + g_N \Delta d(k-N+P) \\ + g_{ss}\Delta d(k-N+P-1) \end{aligned} \quad (A5)$$

Now substituting Eq. A5 for $y_m^*(k+P|K-1)$ in the last equation of Eq. A1 and rewriting Eq. A1 in vector/matrix form:

$$\begin{bmatrix} y_m^*(k|k) \\ y_m^*(k+1|k) \\ \cdots \\ \cdots \\ y_m^*(k+P|k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} y_m^*(k-1|k-1) \\ y_m^*(k|k-1) \\ \cdots \\ \cdots \\ y_m^*(k+P-1|k-1) \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & \cdots & \cdots & \cdots & a_1 \\ 0 & \cdots & \cdots & \cdots & a_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & 0 & a_P \\ h_{ss} & h_N & \cdots & h_{P+2} & a_{P+1} \end{bmatrix} \begin{bmatrix} \Delta u(k-N+P-1) \\ \Delta u(k-N+P) \\ \cdots \\ \Delta u(k-2) \\ \Delta u(k-1) \end{bmatrix} + \begin{bmatrix} 0 & \cdots & \cdots & \cdots & b_1 \\ 0 & \cdots & \cdots & \cdots & b_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & 0 & b_P \\ g_{ss} & g_N & \cdots & g_{P+2} & b_{P+1} \end{bmatrix} \begin{bmatrix} \Delta d(k-N+P-1) \\ \Delta d(k-N+P) \\ \cdots \\ \Delta d(k-2) \\ \Delta d(k-1) \end{bmatrix} \quad (A6)$$

Note that $d(\cdot)$ is any measured or known process signal. If d is nonzero, the controller compensates for this input by using feedforward control based on the coefficient matrix in Eq. A6. Equation A6 is equivalent to Eq. 14 if $P = N$.

Manuscript received on Mar. 9, 1987, and revision received Aug. 8, 1988.