$$H_4 = N_a^2\left[\frac{83}{35\pi} - \frac{3}{4} + \frac{N_a}{\pi}\left\{\frac{63,706}{4,725\pi} - \frac{15,157}{3,360}\right\} + \frac{128}{63\pi^2}N_a^2\right] \quad \text{(A-26d)}$$

It should be noted that the expansions for the square of the radius given in equations (A-2a) and (A-24b) are identical since the expansion variable is precisely the same quantity ($X = 2\sqrt{T}$). Thus, the first four coefficients H_0 to H_3 calculated from equations (A-3a) and the results for G_0 to G_3 from equations (A-6), (A-11), (A-14), and (A-17) exactly match the results in equations (A-25a) and (A-26). However, in view of the different transformations used in arriving at the transformed concentration fields $F(X, Z)$ and $L(X, W)$, the coefficient functions $F_j(Z)$ and $L_j(W)$ are not equivalent functions of their arguments.

NOTATION

- a = bubble radius
- a_0 = initial value of bubble radius
- c = concentration of solute in the liquid
- c_0 = initial concentration of solute
- c_i = concentration of solute at the interface
- $C(T, r)$ = scaled concentration field of the solute in the liquid; $C = (c - c_0)/(c_i - c_0)$
- $C_1(T, Y)$ = transformed concentration fields; defined in Eq. 6, for example
- D = diffusion coefficient of the dissolved gas in the liquid
- $F(X, Z)$ = transformed concentration field; $F(X, Z) = C_1(T, Y)$
- $F_j(Z)$ = expansion coefficients defined in Equation (13a)
- g = scaled bubble radius; $g = a/a_0$
- $G(X)$ = scaled bubble radius; $G(X) = g(T)$
- G_j = expansion coefficients defined in Eq. 13b
- $h(T)$ = square of the scaled bubble radius; $h(T) = g^2(T)$
- $H(X)$ = square of the scaled bubble radius; $H(X) = h(T)$
- H_j = expansion coefficients defined in Eq. A-2a
- $i^n erfc$ = repeated integral of the error function defined in Abramowitz and Stegun (1968)
- I_j = expansion coefficients defined in Eq. A-2b
- J_j = expansion coefficients defined in Eq. A-2c
- $L(X, W)$ = transformed concentration field defined in Eq. A-23c
- N_a = driving force parameter; $N_a = (c_i - c_\infty)/\rho$
- r = radial distance (from bubble center) scaled by the initial value of the bubble radius
- t = time
- T = scaled time; $T = Dt/a_0^2$
- W = new independent variable defined in Eq. A-23b
- X = new independent variable defined in Eq. 10a
- y = new independent variable defined in Eq. A-18a
- Y = new independent variable defined in Eq. 5
- Z = new independent variable defined in Eq. 10b

Greek Letters

- β = "growth" constant appearing in Eq. 17
- δ_{ij} = Kronecker delta function
- η = Scriven's similarity coordinate; related to Z by Eq. 17
- θ = scaled distance coordinate defined in Eq. 18
- ρ = gas density in the bubble
- σ = "stretching parameter" (Eq. 18)

LITERATURE CITED

Abramowitz, M., and I. Stegun, "Handbook of Mathematical Functions," pp. 299-300, Dover, New York (1968).

Bankoff, S. G., "Advances in Chemical Engineering," **6**, Academic Press, New York (1966).

Cable, M. and D. J. Evans, "Spherically Symmetric Diffusion—Controlled Growth or Dissolution of a Sphere," *J. Appl. Phys.*, **38** [7], 2899-2906 (1967).

Duda, J. L. and J. S. Vrentas, "Mathematical Analysis of Bubble Dissolution," *AIChE J.*, **15** [3], 351-56 (1969).

Duda, J. L. and J. S. Vrentas, "Heat and Mass-Transfer-Controlled Dissolution of an Isolated Sphere," *Int. J. Mass Heat Transfer*, **14** [3], 395-407 (1971).

Epstein, P. S. and M. S. Plesset, "On the Stability of Gas Bubbles in Liquid-Gas Solutions," *J. Chem. Phys.*, **18** [11], 1505-09 (1950).

Frischat, G. H. and H. J. Oel, "Diffusion of Neon in Glass Melt," *Phys. Chem. Glasses*, **8** [3], 92-95 (1967).

Pfeffer, W. F. and I. M. Krieger, "Bubble Solution Method for Diffusion Coefficient Measurements. An Experimental Evaluation," *J. Phys. Chem.*, **78**, 2516-21 (1974).

Readey, D. W. and A. R. Cooper, Jr., "Molecular Diffusion with a Moving Boundary and Spherical Symmetry," *Chem. Eng. Sci.*, **21** [10], 917-22 (1966).

Rosner, D. E. and M. Epstein, "Effects of Interface Kinetics, Capillarity, and Solute Diffusion on Bubble Growth Rates in Highly Supersaturated Liquids," *Chem. Eng. Sci.*, **27**, 69-88 (1972).

Ruckenstein, E., "A Generalized Penetration Theory for Unsteady Convective Mass Transfer," *Chem. Eng. Sci.*, **23**, 363-71 (1968).

Ruckenstein, E., and E. J. Davis, "Diffusion—Controlled Growth or Collapse of Moving and Stationary Fluid Spheres," *J. Colloid Interface Sci.*, **34** [1], 142-158 (1970).

Scriven, L. E., "On the Dynamics of Phase Growth," *Chem. Eng. Sci.*, **10**, 1-13 (1959).

Subramanian, R. S., and B. Chi, "Bubble Dissolution with Chemical Reaction," *Chem. Eng. Sci.*, **35**, 2185-94 (1980).

Subramanian, R. S., and M. C. Weinberg, "The Role of Convective Transport in the Dissolution or Growth of a Gas Bubble," *J. Chem. Phys.*, **72**, No. 12, 6811-6813 (1980).

Szekely, J., and S. D. Fang, "Non-Equilibrium Effects in the Growth of Spherical Gas Bubbles due to Solute Diffusion—II," *Chem. Eng. Sci.*, **28**, 2127-40 (1973).

Tao, L. N., "The General Solution and its Analyticity for Growth or Dissolution of a Gas Bubble," *J. Chem. Phys.*, **71**, 3455-61 (1979).

Weinberg, M. C., P. Onorato, and D. R. Uhlmann, "Behavior of Bubbles in Glass Melts: I Dissolution of a Stationary Bubble Containing a Single Gas," *J. Amer. Ceram. Soc.*, **63**, 175-80 (1980).

Manuscript received May 20, 1980; revision received October 31, and accepted November 21, 1980.

Long-Range Predictive Control

G. D. MARTIN

Setpoint, Inc.
Houston, TX 77079

Two process computer control algorithms which are based on long-range prediction are outlined. The methods are compared, similarities pointed out, and each is shown to become similar to dead-beat control under certain circumstances.

SCOPE

Two new process computer control techniques have been presented in the literature that have proven useful in industrial applications. Their success and the fact that both methods differ from the conventional state space or transfer function approaches have caused much interest in the petrochemical control community. Each of the new techniques is based on a non-minimal representation of the process, that is, the usual

0001-1541-81-4871-0748-$2.00.

parametric models are not employed. The process models are composed of parameters which can be obtained directly from input-output response curves.

Although each control technique has been discussed in the literature, a precise description of the control algorithms has not been presented. Also, since each method has evolved independently, one in Europe and the other in the United States, the accounts published to date have been quite dissimilar in philosophy, technical content and nomenclature. The similarities between the two approaches have thus been obscured.

CONCLUSIONS AND SIGNIFICANCE

This paper outlines the basic algorithms for each of two new process computer control algorithms. It presents for the first time a cohesive, step by step account of the calculation sequence for each technique. The presentation utilizes a consistent nomenclature which allows the functional similarities between the two algorithms to become more evident. As a point of reference, the more widely known dead-beat control algorithm is described. It is shown that under certain limiting conditions, each of the new methods becomes algorithmically similar to dead-beat control.

In summary, a cohesive account of two new and interesting process computer control techniques has been presented.

INTRODUCTION

Techniques for computer control of industrial processes have until recently resulted in on-line algorithms which take into account a very small controlled variable time history. Invariably, the prediction horizon over which the controlled variable is considered is short with respect to the process settling time.

Two relatively new methods have been presented in the literature (Richalet et al., 1977, 1978; Cutler and Ramaker, 1979; Prett and Gillette, 1979) which have demonstrated success in industrial multivariable control applications. Each differs from the more traditional approaches in that it is based on a long range prediction of the controlled variable and incorporates an internal model. Though similar in philosophy, the two methods differ considerably in formulation and algorithm detail. This paper discusses the control objective and basic algorithm for each of these new approaches. Similarities between the two approaches and between each and the more recognizable dead-beat control (Cadzow and Martens, 1970) algorithm will be discussed.

The IDentification and COMmand (IDCOM) Richalet et al., 1977, 1978) method is based on the process impulse response and utilizes a predictive heuristic scenario technique to calculate the manipulated variable. The Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1979; Prett and Gillette, 1979) method is based on the process step response and calculates manipulated variable moves via an inverse model. Both methods can be shown to be similar in function and each can be defined to reflect a dead-beat type control.

Although the methods discussed here are in general applicable to multivariable, constrained control problems, the following discussion will focus on the single-input, single-output unconstrained case. This will allow emphasis on fundamental algorithm similarities and simplify discussion.

RESPONSE FORMULATION

To facilitate illustration, and at no loss in generality, consider the single-input, single-output dynamic system described by:

$$\begin{aligned} x(k+1) &= ax(k) + bu(k), \quad x(0) = 0 \\ y(k) &= cx(k) \end{aligned} \tag{1}$$

where $x(k)$ denotes the state variable at time k, $u(k)$ the manipulated variable (input), $y(k)$ the controlled variable (output) and a, b, c are real constants. Define the unit impulse sequence

$$\delta_o(k) = \begin{cases} 1 \text{ for } k = 0 \\ 0 \text{ for } k \neq 0 \end{cases} \tag{2}$$

and the unit step sequence

$$s_o(k) = \begin{cases} 0 \text{ for } k < 0 \\ 1 \text{ for } k \geq 0. \end{cases} \tag{3}$$

The unit impulse response sequence obtained by substituting Eq. 2 into Eq. 1 is given by:

$$h_k = ca^{k-1}b, \quad k = 1, 2, \ldots$$

where $\lim_{k\to\infty} h_k = 0$ *iff* $|a| < 1$ (stability). The unit step response sequence obtained by substituting Eq. 3 into Eq. 1 is:

$$s_k = c(a^{k-1} + a^{k-2} + \ldots)b, \quad k = 1, 2, \ldots$$

where $\lim_{k\to\infty} s_k = cb/(1-a)$ *iff* $|a| < 1$. Note that

$$s_k = \sum_{i=1}^{k} h_i, \quad k = 1, 2, \ldots$$

and

$$h_k = \nabla s_k$$

where ∇ denotes the backward shift operator,

$$(\nabla s_k = s_k - s_{k-1}).$$

LONG-RANGE PREDICTIVE CONTROL ALGORITHMS

IDCOM Algorithm

For the single-input, single-output system given by Eq. 1, consider the impulse response model:

$$y(k+N) = \sum_{i=1}^{N} h_{N-i+1}u(k+i-1) \tag{4}$$

where $N\Delta T >$ settling time and ΔT is the discrete control interval. Note that Eq. 4 relates the value of the controlled variable N steps into the future in terms of values of the manipulated variable between now and then.

IDCOM utilizes an impulse response internal model to calculate predicted output variable values, $y_{IM}(k)$, given known inputs, e.g.

$$y_{IM}(i_o + 1) = \mathbf{h}^T \mathbf{u}_{i_o}^{i_o - N+1}$$

where

$$\mathbf{h}^T \triangleq [h_N \; h_{N-1} \; \ldots \; h_1],$$

$$\mathbf{u}_{i_o}^{i_o-N+1} \triangleq \begin{bmatrix} u(i_o - N + 1) \\ u(i_o - N + 2) \\ \vdots \\ u(i_o) \end{bmatrix}$$

and i_o denotes the current time step so that $y(i_o)$ is the current measured value of the controlled variable.

The predicted $y_{IM}(k)$ values are compared with a desired reference trajectory defined as:

$$y_{MR}(i+1) = \beta y_{MR}(i) + (1-\beta)y_{SP}, \qquad y_{MR}(i_o) = y(i_o) \tag{5}$$

where

$$y_{SP} \overset{\Delta}{=} \text{the controlled variable setpoint}$$

and

$$\beta \overset{\Delta}{=} \text{a tuning factor.}$$

Here the reference trajectory is first order with time constant $\tau_{ref} = -\Delta T/\ln\beta$.

The IDCOM algorithm calculates by iteration a set of future manipulated variable values such that the simultaneously predicted controlled variable trajectory fits the reference trajectory $HP \leq N$ steps ahead in time. The time history of the calculation is illustrated in Figure 1. The IDCOM calculation at each step in time is as follows (after Richalet, et al., 1978):

1. Initialize the future (and the present) manipulated variable values, $u(i)$, $i = i_o, \ldots, i_o + HP - 1$. After IDCOM has been running, the $u(i)$, $i = i_o, \ldots, i_o + HP - 2$ can be initialized to the results from the last step and $u(i_o + HP - 1)$ could be extrapolated, e.g., $u(i_o + HP - 1) = 2u(i_o + HP - 2) - u(i_o + HP - 3)$. For the first initialization, $u(i) = u(i_o - 1)$, $i = i_o, \ldots, i_o + HP - 1$ would suffice.
2. Calculate the reference trajectory using Eq. 5.
3. Calculate the predicted controlled variable using the internal model:

$$y_{IM}(i+1) = y_{IM}(i) + \mathbf{h}^T(\mathbf{u}(i+1) - \mathbf{u}(i)),$$
$$i = i_o, \ldots, i_o + HP - 1$$
$$y_{IM}(i_o) = y(i_o)$$

where

$$\mathbf{u}(i) \overset{\Delta}{=} \mathbf{u}_{i-1}^{i-N},$$

4. Calculate the manipulated variable vector:

$$\mathbf{u}(i+1) = \mathbf{u}(i) - \left[\frac{\lambda}{\mathbf{h}^T\mathbf{h}}\right](y_{MR}(i+1) - y_{IM}(i+1))\mathbf{h},$$
$$i = i_o, \ldots, i_o + HP - 1$$

where λ is a relaxation factor.

5. Repeat steps 3 and 4 until the predicted manipulated variable values $u(i)$, $i = i_o, \ldots, i_o + HP - 1$ have converged.
6. Implement the move $\nabla u(i_o)$.

DMC Algorithm

For the system given by Eq. 1, consider the step response model:

$$y(k+j) = s_j \nabla u(k), \quad j = 1, \ldots, N. \tag{6}$$

Future values of the controlled variable can be expressed in terms of the present manipulated variable move.

The DMC algorithm calculates the present manipulated variable move to minimize the square-summed distance between the predicted controlled variable (y_p) trajectory and the setpoint. That is, $\nabla u(i_o)$ is determined such that:

$$\sum_{j=1}^{N} (y_{SP} - y_p(i_o + j))^2$$

is minimized, or

$$\nabla u^*(i_o) = \mathbf{g}^L \mathbf{E}_{i_o+N+1}^{i_o+1} \tag{7}$$

where

$$\mathbf{g} \overset{\Delta}{=} \begin{bmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ s_N \end{bmatrix},$$

$$\mathbf{E}_{i_o+N+1}^{i_o+1} = y_{SP}col[1] - \mathbf{y}_{p}{}_{i_o+N+1}^{i_o+1},$$

the superscript "*" denotes solution and the superscript "L" denotes left inverse, i.e.,

$$\mathbf{g}^L = (\mathbf{g}^T\mathbf{g})^{-1}\mathbf{g}^T$$

and, since $\mathbf{g}$ is Nx1 here,

$$\mathbf{g}^L = \frac{\mathbf{g}^T}{\sum_{i=1}^{N} s_i^2}.$$

DMC utilizes an internal model to update $\mathbf{E}$ directly, dealing in terms of predicted errors. The DMC calculation at each step in time is described below (after Cutler and Ramaker):

1. Calculate the predicted error response to the last manipulated variable move made.

$$\mathbf{E}_{p}{}_{i_o+N}^{i_o} = \mathbf{g}\nabla u(i_o - 1),$$

2. Update the predicted error by subtracting the contribution due to the last move in the manipulated variable.

$$\mathbf{E}_{i_o+N}^{i_o} = (\mathbf{E}_c - \mathbf{E}_p)_{i_o+N}^{i_o},$$

3. Calculate the present actual error.

$$E_a = y_{SP} - y(i_o),$$

4. Translate the error vector so that $E(i_o) = E_a$

$$\mathbf{E}_{i_o+N}^{i_o} = \mathbf{E}_{i_o+N}^{i_o} + (E_a - E(i_o))col[1],$$

5. Shift/extrapolate the error vector as follows:

$$\mathbf{E}_{c}{}_{i_o+N+1}^{i_o+1} = \mathscr{S}(\mathbf{E}_{i_o+N}^{i_o})$$

where

$$\mathscr{S}\left(\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_N \end{bmatrix}\right) = \begin{bmatrix} x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_N \\ 2x_N - x_{N-1} \end{bmatrix},$$

6. Calculate the new manipulated variable move

$$\nabla u^*(i_o) = \left[\frac{\mathbf{g}}{W}\right]^L \left[\frac{\mathbf{E}_{c}{}_{i_o+N+1}^{i_o+1}}{0}\right] \tag{8}$$

and implement it. Note that an additional equation has been appended to the control equation, cf. Eqs. 7 and 8. The equation is:

$$0 = W\nabla u(i_o),$$

where W is the move size weighting parameter. Increasing W causes $\nabla u(i_o)$ to decrease, and $\nabla u(i_o)$ is determined to minimize:

$$\sum_{j=1}^{N} (y_{SP} - y_p(i_o + j))^2 + (W\nabla u(i_o))^2.$$

Dead-Beat Control Algorithm

To construct the dead-beat control algorithm, it is useful to consider first a simpler form of control algorithm called minimum norm control (Cadzow, 1965; Cadzow and Martens, 1980). By repeated application of Eq. 1, the following expression

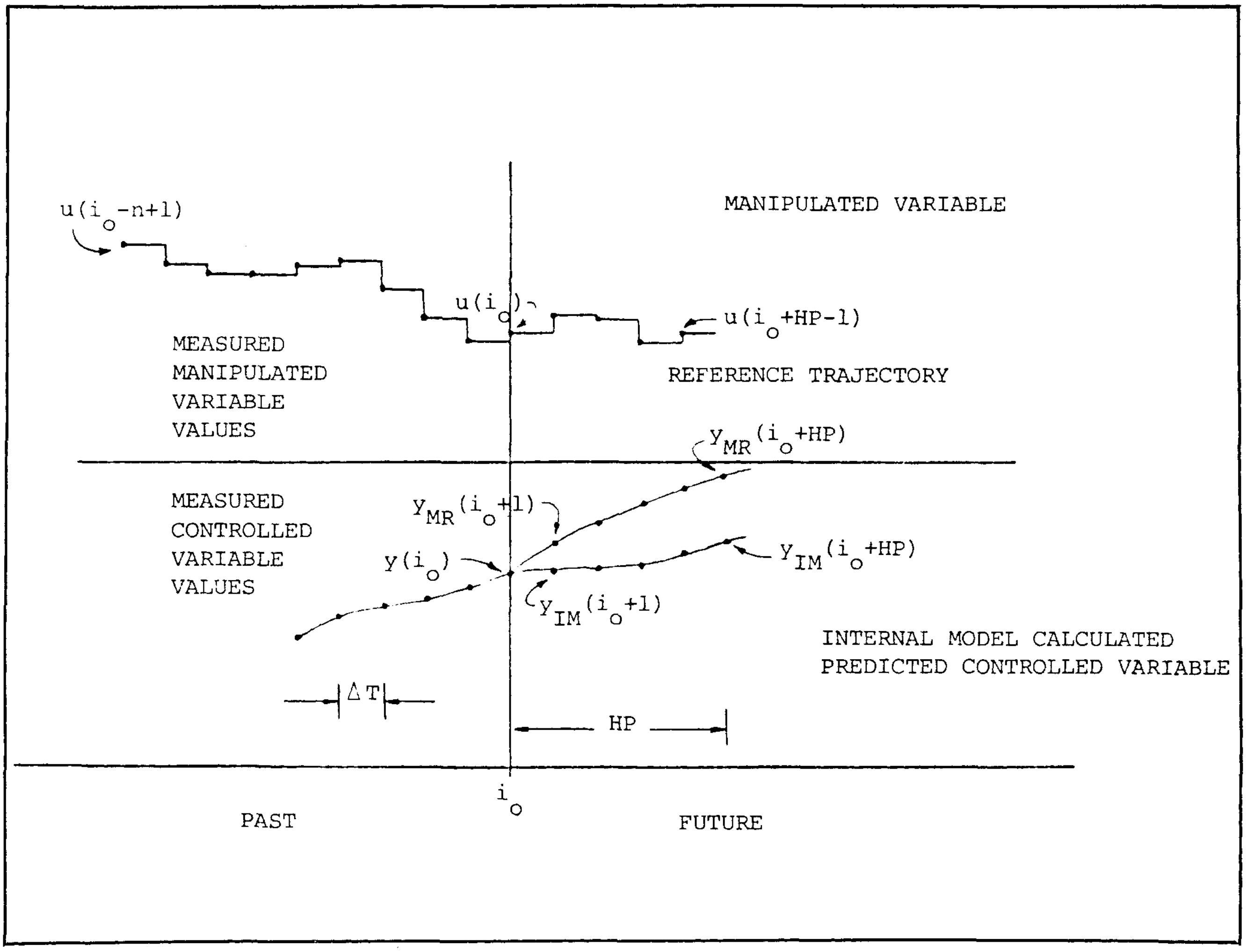

Figure 1. IDCOM time history.

can be obtained:

$$y(N) - ca^N x(o) = \sum_{i=1}^{N} h_{N-i+1} u(i-1) \tag{9}$$

where $x(0)$ represents the value of the state at time 0. Equation 9 can be written as:

$$y(i_o + N) = a^N y(i_o) + \mathbf{h}^T \mathbf{u}^{io}_{i_o+N-1}. \tag{10}$$

The value of the controlled variable N steps into the future is expressed in terms of its present value and the manipulated variable values between now and then.

For minimum norm control the objective is to drive the output to zero in N steps, hence the desired $y(i_o + N)$ equals zero. Note that for this simple example (Eq. 10) can be solved exactly for any N. (See Cadzow (1965) for a discussion of minimum norm control when N is less than the dimension of the state.) The resulting minimum norm control also minimizes

$$\sum_{j=1}^{N} (u(i_o + j - 1))^2$$

and is given by:

$$\mathbf{u}^{io}_{i_o+N-1} = -[\mathbf{h}^T]^R a^N y(i_o)$$

where the superscript "R" denotes right inverse, i.e.:

$$[\mathbf{h}^T]^R = \mathbf{h}(\mathbf{h}^T\mathbf{h})^{-1}$$

Although minimum norm control will drive the output to zero in N steps, at the sampling instants, there is no guarantee that it remains zero between the sampling instants. For dead-beat control the objective is to drive the output to zero in N steps and cause it to remain there, even between sampling instants. The dead-beat controller is thus designed such that the derivative of the output is zero after N steps.

Consider the continuous state equations

$$\dot{x} = \tilde{a}x + \tilde{b}u$$

$$y = \tilde{c}x$$

For $\dot{x}(N\Delta T) = 0$, it follows that

$$0 = \tilde{a}x(N\Delta T) + \tilde{b}u(N\Delta T)$$

or

$$0 = \tilde{a}[a^{N-1}bu(0) + a^{N-2}bu(1) + \ldots + bu(N-1)] + \tilde{b}u(N). \tag{11}$$

Combining Eqs. 10 and 11 yields:

$$\begin{bmatrix} y(i_o + N) - a^N y(i_o) \\ 0 \end{bmatrix} = \begin{bmatrix} h_N & h_{N-1} & \ldots & h_1 & 0 \\ \tilde{a}a^{N-1} & \tilde{a}a^{N-2}b & \ldots & \tilde{a}b & \tilde{b} \end{bmatrix} \cdot [\mathbf{u}^{io}_{i_o+N}]. \tag{12}$$

For this particular problem Eq. 12 can be solved exactly for any N. The resulting dead-beat control also minimizes

$$\sum_{j=1}^{N} (u(i_o + j))^2$$

and is given by:

$$\mathbf{u}^{io}_{i_o+N} = \mathbf{D}^R \begin{bmatrix} y(i_o + N) - a^N y(i_o) \\ 0 \end{bmatrix}$$

where

$$\mathbf{D} \triangleq \begin{bmatrix} h_N & h_{N-1} & \dots & h_1 & 0 \\ \tilde{a}a^{N-1}b & \tilde{a}a^{N-2}b & \dots & \tilde{a}b & \tilde{b} \end{bmatrix}.$$

ALGORITHM SIMILARITIES

The IDCOM and DMC algorithms are similar in many respects, moreover, each can be shown to behave like dead-beat control under certain circumstances.

Similarities Between IDCOM and DMC

The IDCOM and DMC algorithms are similar in that each:

- Utilizes a non-minimal model representation
- Considers predicted values of the controlled variable
- Incorporates an internal model
- Uses actual measurements to update the prediction (feedback)
- Has a tuning parameter which dampens control action

For IDCOM and for DMC, increasing the tuning parameter decreases speed of response. The primary tuning parameter for IDCOM is the reference trajectory time constant, τ_{ref}. A large τ_{ref} results in sluggish, robust control with small moves in the manipulated variable. Decreasing τ_{ref} has the combined effect of speeding up closed loop response, decreasing robustness and producing larger manipulated variable moves.

DMC uses no reference trajectory to smooth the control action. With $W = 0$, the control is similar to IDCOM with $\tau_{ref} = 0$. The effects of increasing W are discussed in the next subsection.

Extended DMC Algorithm

The previous DMC discussion was limited to the DMC calculation of the next move in the manipulated variable. It is possible to extend the manipulated variable calculation horizon to account for several future moves. Note, however, that always only the first move is implemented at each control step. As one might expect, the greater the number of moves considered in the manipulated variable the more the controlled response becomes like dead-beat control.

Consider again the DMC control objective, except now additional manipulated variable moves are to be calculated. The control objective becomes: find $\nabla u(k), \nabla u(k+1), \dots, \nabla u(k + n_u - 1)$ such that future errors are minimized. With $n_u < N$, the solution utilizes the left inverse as follows. (If $n_u = N$, the solution would utilize the normal inverse; n_u is normally much smaller than N.)

$$\nabla \mathbf{u}^{io}_{i_o+n_u-1} = -\begin{bmatrix} \mathbf{G} \\ \hline \mathbf{W} \end{bmatrix}^L \begin{bmatrix} \mathbf{E}^{io+1}_{i_o+N+1} \\ \hline \mathbf{0} \end{bmatrix}$$

where now $\mathbf{W} = \text{diag}\,[W]$ and

$$\mathbf{G} \triangleq \begin{bmatrix} s_1 & & & \mathbf{0} \\ s_2 & s_1 & & \\ \vdots & \vdots & \ddots & s_1 \\ \vdots & \vdots & & \vdots \\ s_N & s_{N-1} & \dots & s_{N-n_u+1} \end{bmatrix}$$

The approach to dead-beat type control with increasing n_u can be demonstrated by examination of the control equation in terms of y, with $W = 0$

$$\mathbf{y}^{io+1}_{i_o+N+1} = \mathbf{G}\nabla \mathbf{u}^{io}_{i_o+n_u-1}.$$

and the equivalent control equation in terms of the manipulated variable, u, rather than moves, ∇u,

$$\mathbf{y}^{io+1}_{i_o+N+1} = \mathbf{G}'\mathbf{u}^{io}_{i_o+n_u-1}. \qquad (13)$$

In the limiting case, when $n_u = N$, taking note that we are not interested in calculating $u(i_o - 1)$, actually a past move, Eq. 13 becomes

$$\begin{bmatrix} y(i_o+1) \\ y(i_o+2) \\ \vdots \\ y(i_o+N+1) \end{bmatrix} = \begin{bmatrix} h_1 & & & \mathbf{0} \\ h_2 & h_1 & & \\ \vdots & \vdots & \ddots & \\ \vdots & \vdots & & h_1 \\ h_N & h_{N-1} & \dots\ h_3 & s_2 \end{bmatrix} \begin{bmatrix} u(i_o) \\ u(i_o+1) \\ \vdots \\ u(i_o+N-1) \end{bmatrix}.$$

Note that the $(N-1)$th row is:

$$y(i_o + N) = [h_{N-1}\ h_{N-2} \dots h_1]\,[\mathbf{u}^{io}_{i_o+N-1}]$$

and compare with Eq. 12. Obviously, the manipulated variable moves calculated based on Eq. 13 are very similar to those calculated based on Eq. 12. Experience has shown that large n_u with W small gives rise to a dead-beat type response with quick settling of the controlled variable as a result of violent ringing of the manipulated variable. Smaller, more reasonable manipulated variable moves result from larger W (or smaller n_u). The advantage of larger n_u (quicker response) is diminished for reasonable values of W and n_u can probably be reduced to three control intervals or less with no substantial loss in performance.

Alternate IDCOM Algorithm

The heuristic iterative convergence approach outlined in the IDCOM discussion does not lend itself to direct comparison with dead-beat control. An alternate formulation can be defined which utilizes an inverse model and should in general reflect IDCOM behavior.

Consider again the impulse response model given by Eq. 4. The set of simultaneous equations defining future values of the controlled variable in terms of past and future manipulated variable values is as follows.

$$\begin{bmatrix} y_{IM}(i_o+1) \\ \vdots \\ y_{IM}(i_o+n_y) \end{bmatrix} = \begin{bmatrix} h_N & h_{N-1} & \dots & h_1 & & \mathbf{0} \\ & h_N & \dots & h_2 & h_1 & \\ & & \ddots & \vdots & & \ddots \\ \mathbf{0} & & h_N \dots & h_{(n_y)} & h_{(n_y-1)} \dots & h_1 \end{bmatrix} \cdot \begin{bmatrix} u(i_o - N + 1) \\ \vdots \\ u(i_o - 1) \\ u(i_o) \\ \vdots \\ u(i_o + n_u - 1) \end{bmatrix} \qquad (14)$$

where n_y denotes the number of predicted controlled variable values ($n_y < N$) and n_u denotes the number of predicted manipulated variable values. Note that for IDCOM, $n_y = n_u = HP$. Equation 14 can be written as:

$$\left(\begin{bmatrix} y_{IM}(i_o+1) \\ \vdots \\ y_{IM}(i_o+n_y) \end{bmatrix} - \begin{bmatrix} h_N & h_{N-1} & \dots & h_2 \\ & h_N & \dots & h_3 \\ \mathbf{0} & & \ddots & \vdots \\ & & h_N \dots & h_{(n_y+1)} \end{bmatrix} \begin{bmatrix} u(i_o - N + 1) \\ \vdots \\ u(i_o - 1) \end{bmatrix} \right)$$

$$= \begin{bmatrix} h_1 & & \\ \vdots & \ddots & 0 \\ h_{n_y} & \dots & h_1 \end{bmatrix} \begin{bmatrix} u(i_o) \\ \vdots \\ u(i_o + n_u - 1) \end{bmatrix}, \quad (15)$$

where the left hand side is known since (i) it involves past values of the manipulated variable and (ii) the future values of the controlled variable should match the reference trajectory over n_y, $y_{IM}(i) = y_{MR}(i)$, $i = i_o + 1, \dots, i_o + n_y$. The solution of Eq. 15 then is:

$$\mathbf{u}^{*i_o}_{i_o+n_u-1} = \mathbf{H}^{-1}\mathbf{E}^{i_o+1}_{i_o+n_y}$$

where

$$\mathbf{H} \triangleq \begin{bmatrix} h_1 & & \\ \vdots & \ddots & 0 \\ h_{n_y} & \dots & h_1 \end{bmatrix}$$

and the predicted controlled variable error

$$\mathbf{E}^{i_o+1}_{i_o+n_y} = \left(\begin{bmatrix} y_{MR}(i_o + 1) \\ \vdots \\ y_{MR}(i_o + n_y) \end{bmatrix} - \begin{bmatrix} h_N & h_{N-1} & \dots & h_2 \\ & h_N & \dots & h_3 \\ 0 & & \ddots & \vdots \\ & & h_N \dots & h_{(n_y+1)} \end{bmatrix} \cdot \begin{bmatrix} u(i_o - N + 1) \\ \vdots \\ u(i_o - 1) \end{bmatrix} \right).$$

It follows that since $n_u = n_y$, $\mathbf{H}$ is square and the predicted values of the manipulated variable are such that the reference trajectory is followed exactly.

The approach to dead-beat type control with increasing n_y (and n_u) is most apparent when $n_y = n_u = N$. Then, the control equation becomes:

$$\begin{bmatrix} E(i_o + 1) \\ \vdots \\ E(i_o + N) \end{bmatrix} = \begin{bmatrix} h_1 & & \\ \vdots & \ddots & 0 \\ h_N & \dots & h_1 \end{bmatrix} \begin{bmatrix} u(i_o) \\ \vdots \\ u(i_o + N - 1) \end{bmatrix}.$$

Note that the N-th now is:

$$E(i_o + N) = [h_N \dots h_1]\,[\mathbf{u}^{i_o}_{i_o+N-1}]$$

and compare with Eq. 12. Again, the similarity to dead-beat type control is obvious.

At this point the similarity between the alternate IDCOM formulation presented here and the extended DMC formulation can be carried to completion. The manipulated variable moves resulting from the above are likely to be satisfactory only when the reference trajectory time constant is large (say, greater than the dominant process time constant). For quick responding closed loop control, however, the above may lead to prohibitively large manipulated variable moves. This situation can be corrected by reducing n_u, ($n_u \le n_y$). (It should be noted that reducing n_u also improves the conditioning of the potentially ill conditioned control equation.) The calculation then becomes:

$$\mathbf{u}^{*i_o}_{i_o+n_u-1} = \mathbf{H}^L\mathbf{E}^{i_o+1}_{i_o+n_y}$$

where now

$$\mathbf{H} \triangleq \begin{bmatrix} h_1 & & 0 \\ \vdots & \ddots & h_1 \\ \vdots & & \vdots \\ h_{n_y} & \dots & h_{(n_y-n_u+1)} \end{bmatrix}$$

and the solution to Eq. 15 is such that

$$\|\mathbf{E}^{i_o+1}_{i_o+n_y} \quad \mathbf{H}\mathbf{u}^{*i_o}_{i_o+n_u-1}\|_2 \text{ is minimized.}$$

NOTATION

- a, b, c = discrete model parameters
- $\tilde{a}, \tilde{b}, \tilde{c}$ = continuous model parameters
- $\mathbf{D}$ = dead-beat control matrix
- $\mathbf{E}^i_j$ = DMC predicted error column vector from i to j
- E_a = DMC present actual error
- $\mathbf{E}_c$ = DMC error vector to inverse model
- $\mathbf{E}_p$ = DMC model predicted update to error vector
- $\mathbf{G}$ = DMC step response matrix
- $\mathbf{G}'$ = augmented DMC step response matrix
- $\mathbf{g}$ = step response matrix
- $\mathbf{g}^L$ = left inverse of $\mathbf{g} = (\mathbf{g}^T\mathbf{g})^{-1}\mathbf{g}^T$
- $\mathbf{H}$ = alternate IDCOM impulse response matrix
- HP = IDCOM controlled variable prediction horizon length
- $\mathbf{h}$ = impulse response matrix
- $[\mathbf{h}^T]^R$ = right inverse of $\mathbf{h}^T = \mathbf{h}(\mathbf{h}^T\mathbf{h})^{-1}$
- h_k = impulse response sequence element k
- i_o = current time step
- N = number of control intervals in impulse response data
- n_u = number of DMC manipulated variable moves calculated
- n_y = number of alternate IDCOM controlled variable prediction steps
- s_k = step response sequence element k
- $\mathscr{S}$ = shift/extrapolate operator
- $\mathfrak{s}_o(k)$ = unit step response at time k
- $\mathbf{u}^{*i}_j$ = manipulated variable solution column vector from i to j
- $\mathbf{u}^i_j$ = manipulated variable input column vector from i to j
- $u(k)$ = manipulated variable input at time k
- $\mathbf{W}$ = DMC move size weighting matrix = diag[W]
- W = DMC move size weighting parameter
- $x(k)$ = state variable at time k
- $\mathbf{y}^i_j$ = controlled variable column vector from i to j
- $y(k)$ = controlled variable output at time k
- $y_{IM}(k)$ = IDCOM internal model output at time k
- $y_{MR}(k)$ = IDCOM model reference output at time k
- $y_p(k)$ = DMC predicted controlled variable at time k
- $y_{SP}(k)$ = controlled variable setpoint
- β = IDCOM model reference trajectory parameter
- ∇ = backward shift operator ($\nabla x(k) = x(k) - x(k-1)$)
- ΔT = discrete control interval
- $\delta_o(k)$ = unit impulse response at time k
- λ = IDCOM control algorithm relaxation factor
- τ_{ref} = IDCOM reference trajectory time constant = $-\Delta T/\ln\beta$

LITERATURE CITED

Cutler, C. R. and B. L. Ramaker, "Dynamic Matrix Control—A Computer Control Algorithm," 86th AIChE National Meeting (April 4, 1979).

Cadzow, J. A., "A Study of Minimum Norm Control for Sampled-Data Systems," Proceedings of Joint Automatic Control Conference, Session XIX, Paper 2 (1965).

Cadzow, J. A. and H. R. Martens, *Discrete-Time and Computer Control Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1970).

Prett, D. M. and R. D. Gillette, "Optimization and Constrained Multivariable Control of a Catalytic Cracking Unit," 86th AIChE National Meeting (April 4, 1979).

Richalet, J., A. Rault, J. L. Testud, and J. Papon, "Model Algorithmic Control of Industrial Processes," *Digital Computer Applications to Process Control*, V. N. Lempe, ed., IFAC and North-Holland Publishing Co. (1977).

Richalet, J., A. Rault, J. L. Testud, and J. Papon, "Model Predictive Heuristic Control: Applications to Industrial Processes," *Automatica*, **14**, pp. 413-428 (1978).

Manuscript received August 29, 1980; revision received November 3, and accepted November 21, 1980.