AIAA 81-4242

Output Predictive Algorithmic Control: Precision Tracking with Application to Terrain Following

J. Gary Reid*
Lear Siegler, Inc., Dayton, Ohio
David E. Chaffin†
Avionics Laboratory, Wright Patterson Air Force Base, Ohio
and
James T. Silverthorn‡
Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio

The output predictive digital control problem is formulated as an overdetermined linear least-squares problem with considerable design flexibility. An efficient numerical technique is derived for its solution, and some "rules of thumb" are given for the selection of the design parameters. The control switch time is seen to be an important factor affecting robustness, and a "robust" design strategy for selection of this time is presented. The greatest power of the control technique is seen in the tracking task. A hypothetical terrain-following example shows the digital, closed-loop, predictive controller to be extremely high in "bandwidth" and to have a very short "settling time."

Introduction

RAULT and Richalet[1,2] have recently introduced a new digital control concept termed "model predictive heuristic control," "scenario predictive control" or "model algorithmic control."[3]

This control technique is fundamentally and philosophically different than "feedback" controllers such as the linear quadratic regulator and its variants in which there is an explicit notion of "feedback" of the current "state" to derive a closed-loop control law. Instead, the closed-loop control is achieved by accomplishing, at every cycle of the digital control loop, three separate and very distinct functions:

1) *Prediction.* The system zero input response is predicted into the future a selected "horizon of prediction." It is shown in the later analysis and results that the duration of this prediction is a key parameter in the stability and robustness of the control scheme. This prediction calculation may be accomplished in any number of different ways, but it is essential that the prediction be closed-loop, based upon the available measurements of the actual output response. It is probably appropriate to use a Kalman filter or an observer to estimate the current "state," and then use the system mathematical model to predict the output response into the future.

2) *Desired Future Reference Trajectory.* The *desired* output response is calculated into the future for the same "horizon of prediction." This is what gives the control scheme its great power in the dynamic tracking task, for it is generally quite easy to specify how one wants the future output trajectory to behave. For example, in terrain following it is desired to have the future trajectory follow the future terrain profile. If the system is not currently on the desired path, then a simple exponential decay path to bring the system back to the desired path is constructed. Other schemes are certainly permissible in constructing the desired reference path, but a key requirement is to "keep it simple" for it must be reaccomplished at every control cycle.

3) *Future Control Calculation.* Calculation of the "best" set of future control inputs that will make the system follow the desired future path is accomplished. Again, this is done for the entire "horizon of prediction"; however, since all of the above calculations are repeated every control cycle, only the first of the computed future controls is utilized.

Obviously there is considerable design flexibility in this "predictive" control strategy, but this control flexibility is at the same time both a benefit and a detriment. For one thing, the computations can be "large" as compared to a simplistic feedback control scheme, if not carefully implemented. Also, there are inherently many design parameters to be chosen in the implementation, and these can present an enigma. Fortunately, some "rules of thumb" are provided which seem to give good performance over the systems tested. Finally, the control technique does not lend itself to simple stability or performance analysis. The control technique does not admit a simple closed-loop state variable model in which closed-loop stability may be determined on the basis of closed-loop eigenvalue locations.

Reid[4] showed that the same "output predictive" control structure as presented here could be formulated to achieve state deadbeat response; that is, the system "state" could be driven to zero in n or fewer steps, where n is the system state dimension, if the output predictive controller were properly formulated. In this case, the closed-loop eigenvalues are all at the origin. Robustness was analyzed on the basis of perturbation bounds to the inherent linear equation problem[5] and by use of the contraction mapping theorem.[6] Mehra[7] analyzed robustness of the one-step-ahead prediction controller based upon perturbations to the resulting closed-loop characteristic polynomial that could be calculated for this case. However, for the very complex predictive controller as presented herein, it is again felt that closed-loop stability and robustness analysis will have to be based on either closed-loop simulations (most likely) or linear equation perturbation bounds and the contraction mapping principle. This latter approach will be difficult and will provide, at best, worst-case bounds and perhaps some insight into how to best choose the design parameters. For example, the discretization sample time, T_D, may be analyzed to see how it influences the condition number of the inherent linear least-squares problem. This condition number, in turn gives a good indication of how "robust" the calculated solution will be to errors in model and computations.[5] This is the approach taken herein.

Received May 29, 1980; revision received March 9, 1981.

*Senior Engineering Specialist, Astronics Division. Member AIAA.
†Program Manager, Missile Fire Control.
‡Assistant Professor Aeronautical Engineering. Member AIAA.

The major contribution of this paper is to demonstrate how task 3 above—future control calculation—may be formulated and solved as a linear least-squares problem. This is accomplished by the introduction of "output smoothing points"; that is, the future output reference trajectory includes output sample points at a much finer sampling interval than the interval at which the controls are being updated. In the limit, one would like to control the entire continuous curve of the future output trajectory even while using the digital sample-and-hold type of inputs. This technique avoids the large oscillations and resulting instability that can be encountered when using only one-step-ahead prediction[8] by having an explicit weighting on output response *between* control switch times. By this technique it is seen that there is an implicit control of the internal energy or "state" of the system even though there is no explicit control of anything resembling "state"; large output trajectory oscillations *between* control switch times correspond to a situation of large internal energy (see Fig. 1). This internal energy may be controlled by insuring that the output trajectory remains close to the reference even between the control switch times.

The off-line calculations of this method are substantial, but the on-line calculations become as simple as taking a single vector inner product at each control cycle. Quite a few design parametres are involved in formulation of the controller and some "rules of thumb" are provided. Finally, closed-loop performance results are presented both for a regulator application in a very lightly damped system, and for a terrain-following tracking task using a simplified model of a nonminimum phase aircraft. The nonminimum phase aspect of the dynamics are seen to present little problem for the controller if properly implemented, and the reasons for this are discussed.

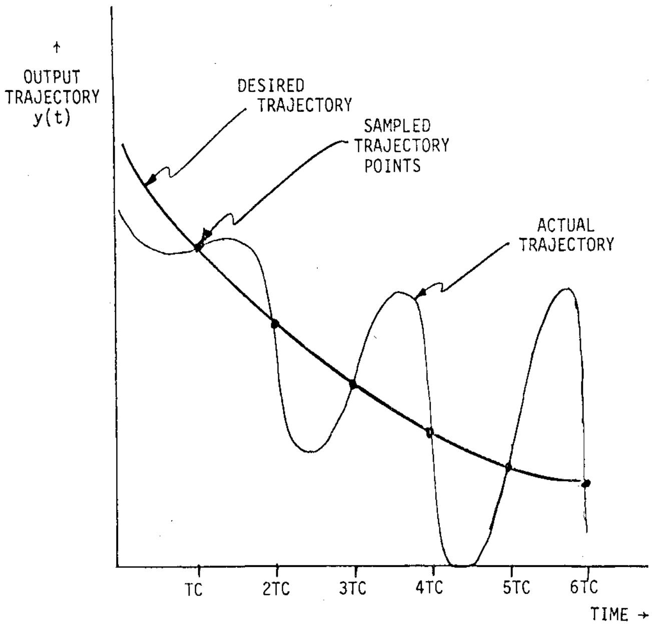

Fig. 1 Example of how sampled output can exactly match desired path while closed-loop system is actually unstable.

Mathematical Formulation

It is desired to control the continuous, nth order, single-input/single-output linear system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

$$y(t) = Cx(t) \tag{2}$$

with discrete time model

$$x(k+1) = Fx(k) + Gu_d(k) \tag{3}$$

$$y(k\cdot T_C) = Cx(k) \tag{4}$$

where

$$F \equiv e^{A\cdot T_C} \qquad G = \left(\int_0^{T_C} e^{A\tau}\mathrm{d}\tau\right)B \tag{5}$$

The system is assumed to be single-input/single-output (SISO) for ease of presentation, but the subsequent results are easily generalized to the multi-input/multi-ouput case. The control input is assumed to be piecewise constant, that is,

$$u(t) = u_d(k) \qquad t\in(k\cdot T_C, (k+1)\cdot T_C] \tag{6}$$

for the control switch time, T_C.

Letting zero be the current time, future output is related to present state $x(0)$, and future inputs $\{u_d(0), u_d(1), \dots\}$ via

$$y(k\cdot T_C) = y_{zi}(k\cdot T_C) + \sum_{i=0}^{k-1} h_d(k-i)u_d(i) \tag{7}$$

The $y_{zi}(k\cdot T_C)$ is the zero input response which may be computed by

$$y_{zi}(k\cdot T_C) = CF^k x(0) \tag{8}$$

and $h_d(k)$ is the discrete impulse response function

$$h_d(k) = CF^{k-1}G \tag{9}$$

The control problem is formulated by setting up a linear equation problem to force the predicted output in Eq. (7) to match the desired future path, $y_d(k\cdot T_C)$, $k=1, 2, 3, \dots$. If the control task is a regulator, then the desired path brings the system down to rest; if the control task is tracking, then the desired path is made to follow the desired tracking trajectory.

The control problem is set up as an overdetermined linear least-squares problem. This is accomplished by examination of the future output trajectory *intermediate* to the control switch interval, T_C. By this approach, the undesired output oscillations between control switch times, which were encountered in Ref. 8 using one-step-ahead prediction, can be avoided.

To this end let L be the number of discrete controls to be predicted into the future and let NSM be the number of output "smoothing points" to be calculated between each control switch time. Then it may be shown that the linear equation problem of "output predictive algorithmic control" can be formulated as

$$\tilde{H}U = Y_D - Y_{zi} \tag{10}$$

where

$$U \equiv \begin{bmatrix} u_d(0) \\ \Delta u(1) \\ \Delta u(2) \\ \cdot \\ \cdot \\ \Delta u(L-1) \end{bmatrix}_{L\times 1} \tag{11}$$

$$Y_D \equiv \begin{bmatrix} y_D(1) \\ y_D(2) \\ \cdot \\ \cdot \\ y_D(\mathrm{NSM}\cdot L) \end{bmatrix}_{(\mathrm{NSM}\cdot L)\times 1} \tag{12}$$

$$Y_{zi} \equiv \begin{bmatrix} y_{zi}(1) \\ y_{zi}(2) \\ \cdot \\ \cdot \\ \cdot \\ y_{zi}(\mathrm{NSM}\cdot L) \end{bmatrix}_{(\mathrm{NSM}\cdot L)\times 1} \tag{13}$$

$$\bar{H} \equiv \begin{bmatrix} h_1 & 0 & \dots & 0 \\ h_2 & h_1 & \dots & 0 \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ h_L & h_2 & \dots & h_1 \end{bmatrix}_{(\mathrm{NSM}\cdot L)\times L} \tag{14}$$

$$h_k \equiv \begin{bmatrix} \sum_{i=1}^{\mathrm{NSM}\cdot(k-1)} h_d(i) \\ \cdot \\ \cdot \\ \cdot \\ \sum_{i=1}^{\mathrm{NSM}\cdot k-1} h_d(i) \end{bmatrix}_{\mathrm{NSM}\times 1} \tag{15}$$

and now, because of the shorter sample time, $T_D = T_C/\mathrm{NSM}$, the appropriate $h_d(i)$ are found from Eqs. (3-5) and (9) using T_D in Eq. (5) rather than T_C. Note from Eq. (11) that the actual future control at time $t\in(k\cdot T_C,\ (k+1)\cdot T_C]$ is given by

$$u(t) = u_d(k) = u_d(0) + \Delta u(1) + \Delta u(2) + \cdots + \Delta u(k) \tag{16}$$

that is, the $\Delta u(i)$ are the *changes* to the previous control applied. This turns out to be physically meaningful in that it is often more important to know, and to minimize, the amount of *control change* from sample to sample rather than the absolute control level itself. The control linear equation problem (10) may be formulated either way, and it simply means that the $\bar{H}$ in Eq. (14) and the least-squares weighting would have to be modified appropriately. Also note that, while Eq. (10) is formulated and "solved" with all L future controls as in Eq. (11), only the first $u_d(0)$ is actually applied to the system; in a closed-loop operation the problem will be reformulated and solved at each control sample time. Thus it is really only necessary, on-line, to calculate $u_d(0)$. It is necessary, however, to formulate Eq. (10) with all L controls, as the "horizon of prediction, L" has a very strong effect upon closed-loop performance.

Now, Eq. (10) is an overdetermined linear equation problem for which a least-squares, "approximate" solution is appropriate. To provide for control design flexibility, a weighted least-squares criterion

$$J(U^*) \equiv \sum_{i=1}^{\mathrm{NSM}\cdot L} q_i[\bar{H}_i U - (y_D(i) - y_{zi}(i))]^2 \tag{17}$$

is utilized, where $\bar{H}_i$ is the ith row vector of H. Letting $Q = \mathrm{diag}\ [q_1, q_2, \ldots q_{\mathrm{NSM}\cdot L}]$ be the matrix of weighting parameters, the "normal equation" solution to Eq. (10), which minimizes Eq. (17), is given by

$$U^* = (\bar{H}^T Q \bar{H})^{-1} \bar{H}^T Q (Y_D - Y_{zi}) \tag{18}$$

The solution to Eq. (18) turns out to be efficient with regard to memory storage because the large dimension $\bar{H}$ in Eq. (14) never has to be actually created and put into memory. Rather, it may be shown (see Ref. 9 for details) that the much lower dimension, symmetric, $L\times L$ matrix $(\bar{H}^T Q \bar{H})$ may be created from

$$[(\bar{H}^T Q \bar{H})_{i,j}] = \Big[\sum_{k=\max(i,j)}^{L} h^T_{k-i+1}\ Q_k h_{k-j+1}\Big] \tag{19}$$

where Q_i is the NSM$\times$NSM diagonal submatrix of Q such that $Q = \mathrm{diag}\ [Q_1, Q_2, \ldots, Q_L]$. Likewise the $L\times 1$ vector $\bar{H}^T Q (Y_D - Y_{zi})$ may be created from the scaler elements, $i = 1, 2, \ldots, L$,

$$\sum_{k=1}^{i} h^T_{i-k+1} Q_k (Y_D - Y_{zi})_k \tag{20}$$

where $(Y_D - Y_{zi})_k$ are the $L\times 1$ dimension partitions of the NSM$\cdot L$ dimension $(Y_D - Y_{zi})$. Therefore, by Eqs. (19) and (20) it is seen that only the L vectors $\{h_1, h_2, \ldots, h_L\}$ need to be created rather than the entire $\bar{H}$ matrix in Eq. (14). This reduces memory storage requirements necessary to compute Eq. (18) by a considerable amount. It is remarked, though, that if memory is not a problem it would definitely be preferrable for reasons of numerical conditioning to use the singular value decomposition to solve Eq. (17) (see, eg., Stewart[10]).

Once the least-squares solution of Eq. (18) is obtained, the first element of U^*, the $u_d^*(0)$, is applied to the system over the next T_C units of time. Then the problem is reformulated at the next sample time, and the next $u_d^*(0)$ is calculated and applied. In this fashion the control law is made "closed loop."

The only remaining variables in Eq. (18) which are not fully defined are the predicted output, Y_{zi}, and the desired trajectory, Y_D. The former would normally involve an estimator (e.g., a Kalman filter) to insure that prediction is made "closed loop," and detailed consideration of this is beyond the scope of the present paper. The future desired trajectory is control objective dependent. The regulator and tracking cases are discussed in the next two setions.

Regulator Case

The solution of the "output predictive algorithmic controller," Eq. (18), involves the Y_D vector of Eq. (12) of the future desired trajectory $y_D(k)$, $k = 1, 2, \ldots, \mathrm{NSM}\cdot L$. If the control objective is to reach some constant value y_{set}, then an exponential decay path to reach this y_{set} is

$$y_D(k) = y_{\mathrm{set}} - \beta^k [y_{\mathrm{set}} - y(0)] \tag{21}$$

where

$$\beta = e^{-T_C/(\mathrm{NSM}\cdot\tau)} \tag{22}$$

T_C = control switch time
NSM = number of output samples per control
τ = desired time constant
$y(0)$ = output value at current time

For the regulator application, $y_{\mathrm{set}} = 0$; therefore, the exponential decay path of Eq. (21) decays to the origin with absolute time constant τ. If a very rapid response is desired, then τ is made very small with accompanying large values of control necessary. If a slower response is acceptable, then τ may be made larger with smaller inputs required. The influence on the control problem, according to the least-squares weighting [Eq. (17)], is that the system is forced to follow this desired exponential decay trajectory "as close as it can."

The desired system time constant is thus selectable by the system designer according to the desired control levels available and the desired performance criterion. With proper control levels available almost any speed of response could be achieved with virtually no overshoot.[9] However, the speed of response demanded has been shown to be strongly linked to closed loop robustness.[7] This fact is also verified by our studies. Demanding a very fast response will result in poor robustness.

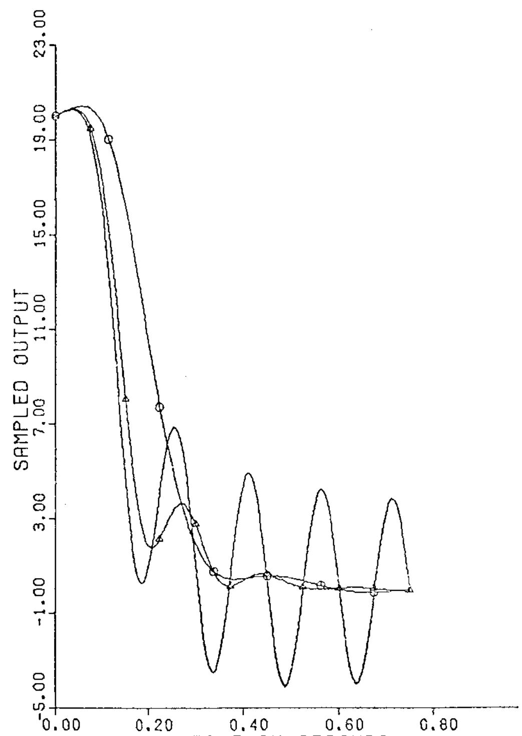

Fig. 2 Closed-loop response for $L=3$ (no marks), $L=4$ (△), and $L=7$ (○).

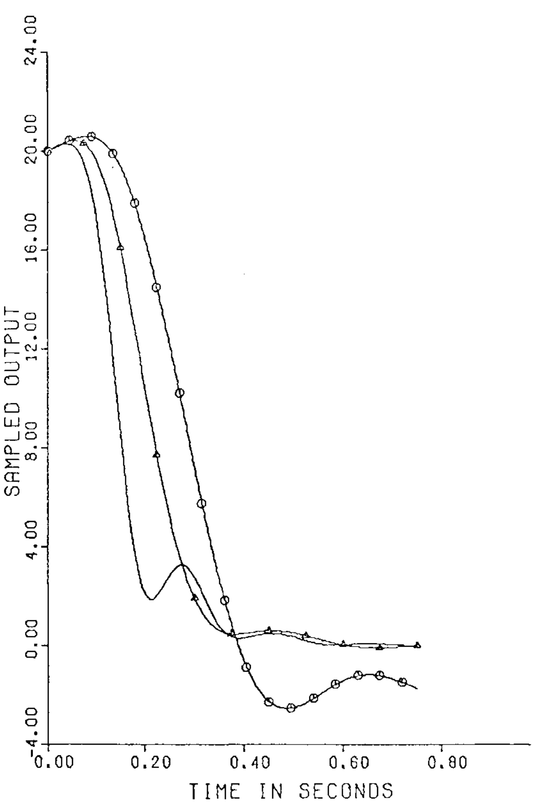

Fig. 3 Closed-loop response for NSM = 2 (no marks), NSM = 4 (△), and NSM = 10 (○).

NSM and L Selection

The number of "smoothing points" on the output between control changes, NSM, and the number of prediction steps into the future, L, turned out to be key parameters affecting closed-loop performance and stability—particularly L, the number of control predictions into the future. A lightly

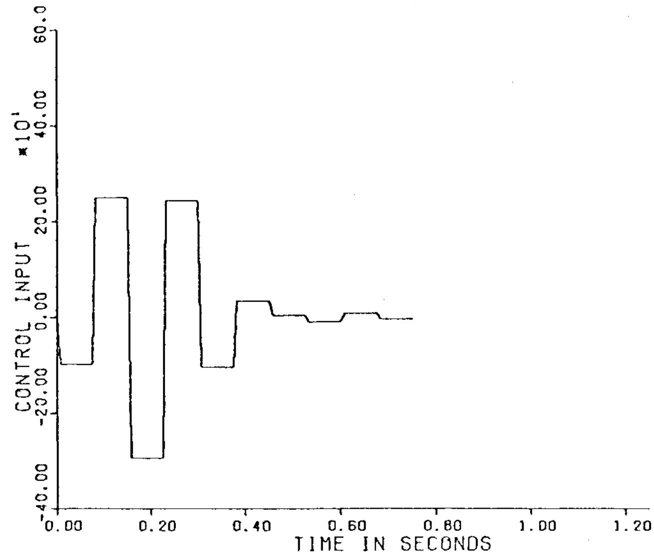

Fig. 4 Controls applied for NSM = 2.

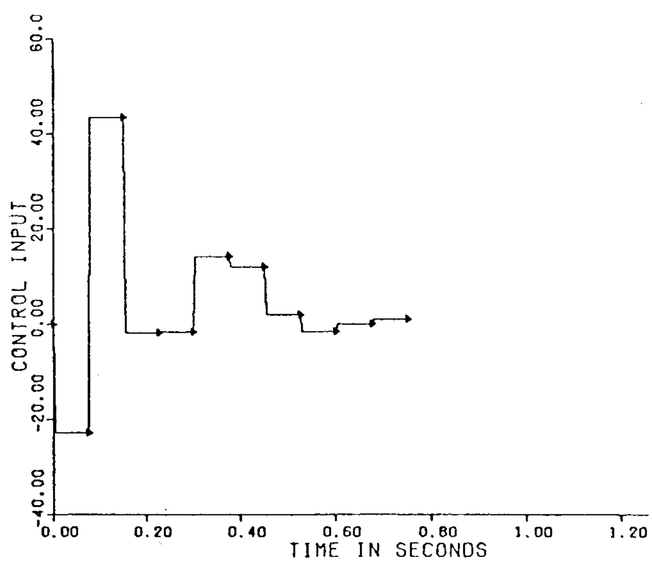

Fig. 5 Controls applied for NSM = 4.

damped, all-pole system with eigenvalues at $-0.25 \pm 15.4\,j$ and $-0.55 \pm 6.0\,j$ was used for simulation analysis. These modes correspond roughly to the dominant flutter modes of the B-52 wing,[9] but, for our purposes, they merely represent a very lightly damped system.

Figure 2 shows the influence of a number of different values of L when the time constant τ is chosen as 0.1. For prediction lengths of $L<3$, the system could not be made closed-loop stable. For prediction lengths $L>7$, there was little improvement in performance. Note that $L=7$ corresponds to the prediction length $(2n-1)$ required for the deadbeat control linear equation problem.[4] Thus, as a rule of thumb, it seems that L should be selected between the bounds $n \le L \le (2n-1)$, where n is the underlying state dimension.

The number of smoothing points, NSM, also had a very key—indeed surprising—effect upon closed-loop performance. It was known that NSM ≥ 2 was required or else the linear equation problem, Eq. (10), would be reduced to one-step-ahead prediction regardless of the value L selected.[7,8] It also seemed logical that the greater the value of NSM the tighter the closed-loop trajectory should stay to the desired exponential decay trajectory. However, as seen in Fig. 3, increasing NSM beyond 4 (note $n=4$) produced more oscillations in the closed-loop response. Control levels required for NSM = 2 and NSM = 4 also dropped by a factor of 10 (see Figs. 4 and 5), and the controls for NSM = 10 (not

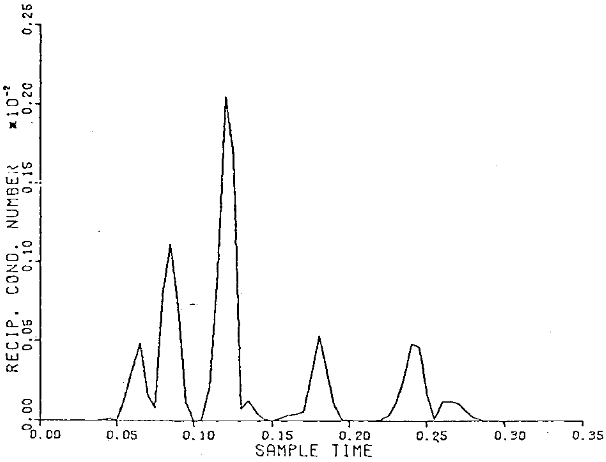

Fig. 6 Reciprocal condition number of $[\bar{H}'Q\bar{H}]$.

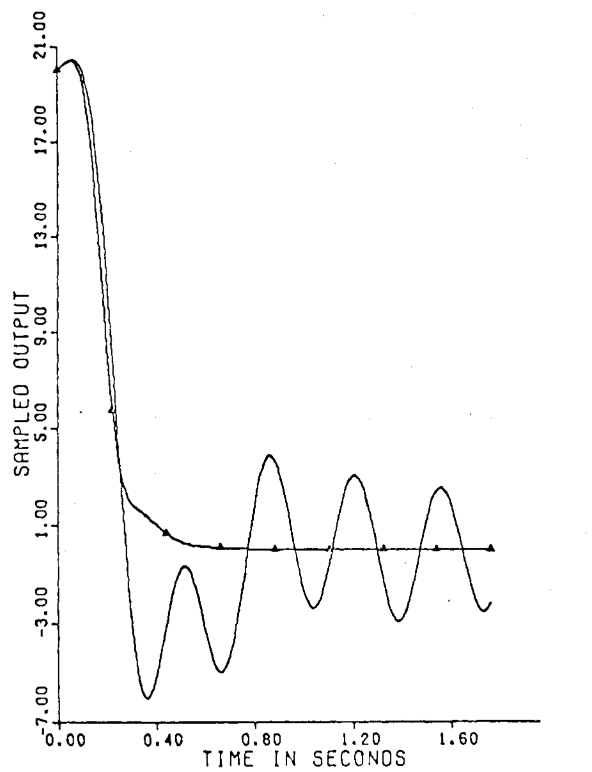

Fig. 7 Closed-loop response for truth model (△) and 30% model perturbation (no marks) using $T_C = 0.088$.

shown) were an order of magnitude less than that for NSM = 4. Thus, as a rule of thumb, NSM = n seems to be a reasonable value. However, the actual closed-loop performance is also strongly influenced by two other factors as well—the sample time T_C and the least-squares cost function Q.

Closed-Loop Robustness and Sample Time, T_C

Unlike the deadbeat control law of Ref. 4, the absolute speed of response is controlled here more by the desired time constant τ than it is by contol switch time T_C. However, just like the deadbeat case of Ref. 4, T_C can have a strong influence upon the conditioning of the inherent linear equation problem (10) and hence upon closed-loop robustness. However, since a weighted least-squares solution is used with the normal equations as in Eq. (18), it is now appropriate to examine the condition number of $(\bar{H}^TQ\bar{H})$ vs T_C.

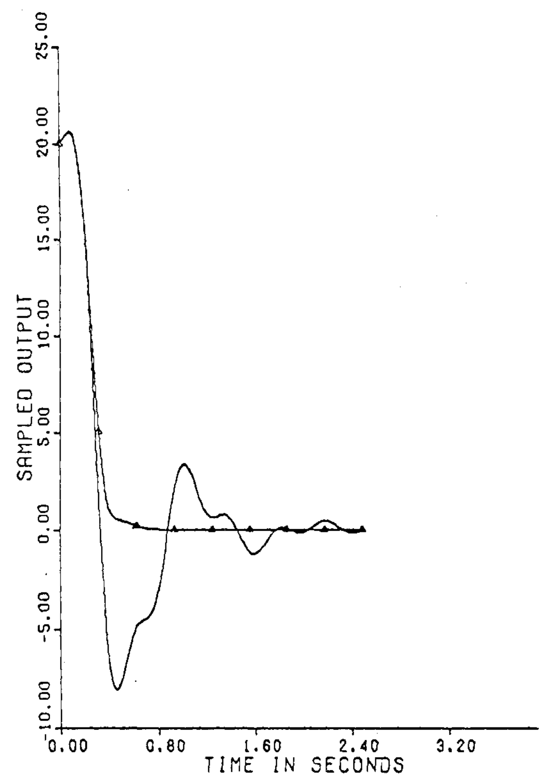

Fig. 8 Closed-loop response for truth model (△) and 30% model perturbation (no marks) using $T_C = 0.124$.

The reciprocal condition number of $(\bar{H}^TQ\bar{H})$ is shown in Fig. 6 for the same fourth-order system using $L = 7$, NSM = 4, and $Q = \mathrm{diag}[1,1,\ldots,1, \vdots\ 10^6, 10^6,\ldots,10^6]$, where the break in Q occurs at the midpoint of the $(\mathrm{NSM}\cdot L) \times (\mathrm{NSM}\cdot L)$ diagonal Q matrix. With this selection of the Q weighting matrix and with the time constant for exponential decay selected at the very fast $\tau = 0.1$, the system is forced to reset in essentially four steps of the piecewise constant sample-and-hold input. This is a "dead-beat" response for this four-state system.[4]

This very fast response ($\tau = 0.1$) tends to accentuate the robustness characteristics. With this fast response the "best" value of T_C appeared to be $T_C = 0.128$, the value of T_C corresponding to the peak of the reciprocal condition number of $(\bar{H}^TQ\bar{H})$ (see Fig. 6). This assessment was based on experimental studies.[9] The value $T_C = 0.088$, corresponding to the second lower peak (see Fig. 6), also gave acceptable "robust" performance but not quite as good as $T_C = 0.128$. The two controllers' closed-loop response starting from an arbitrary initial state is shown in Fig. 7 ($T_C = 0.088$) and Fig. 8 ($T_C = 0.124$). Under conditions of no model mismatch between the controller and the actual system, both controllers give highly acceptable "dead-beat" performance. However, under conditions of a 30% model mismatch between the assumed eigenvalues and the true eigenvalues, the value $T_C = 0.124$ is seen to give preferred closed-loop performance owing to its smaller condition number of $(\bar{H}^TQ\bar{H})$. For values of T_C selected near the valleys of the reciprocal condition number plot (Fig. 6), the system is closed-loop unstable even for the slightest of model mismatch.

As mentioned, the robustness issues are no doubt accentuated here by demanding the very fast time constant $\tau = 0.1$. With a slower time constant of exponential decay demanded of the system, the robustness properties of the closed loop system would likely be improved. However, it should be mentioned that the reciprocal condition number plot (Fig. 6) is *independent* of the time constant τ. The time constant τ dictates the exponential decay fo the *desired* response and has nothing to do with the matrix $(\bar{H}^TQ\bar{H})$.

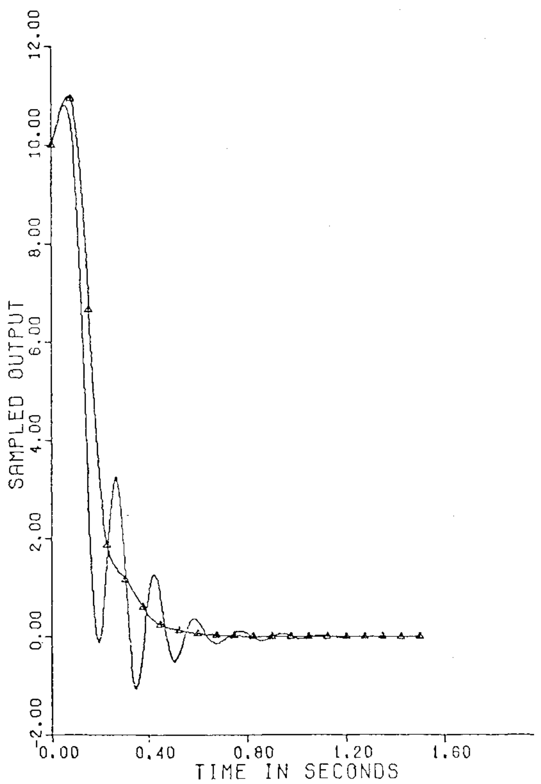

Fig. 9 Closed-loop response using $Q=$ Eq. (23) (△) and $Q=I$ (no marks).

Therefore, it is hypothesized that picking T_C to minimize the condition number (maximize the reciprocal condition number) will be a robust design strategy regardless of the basic speed of response demanded of the system. This and many more issues regarding closed-loop robustness are left to be explored.

Least-Squares Weighting Matrix, Q

The previous control design parameters mentioned (τ, L, NSM, and T_C) offer considerable design flexibility in themselves, but certainly the greatest flexibility exists in the symmetric, positive definite, weighting matrix Q. For computational reasons it is best to keep this diagonal or at least block diagonal. But even diagonal it has NSM$\cdot L$ free parameters. A number of possibilities were explored in Ref. 9 for selection of Q, but as a general rule it can be said that Q, which tended to weight the latter portion of the predicted trajectory did much better than equal weighting or weighting heavily the early portion of the trajectory. Indeed, for the examples looked at, the best weighting matrix, Q, of all turned out to be

$$Q=\mathrm{diag}[1,1,\ldots 1 \;\vdots\; 10^6,10^6,\ldots,10^6] \tag{23}$$

which makes the linear least-squares problem of Eq. (17) approximate the deadbeat control linear equation problem of Ref. 4. Figure 9 shows the system response for this Q, Eq. (23), and for a Q equal to the identity matrix. In fact, based upon these studies, it was concluded that the "output predictive algorithmic controller" offers little performance advantage over the deadbeat controller for the regulator problem. Therefore, attention turned to the tracking problem for which the very flexible control scheme here seemed ideally suited.

Tracking

The tracking task implementation of the control law, Eq. (18), requires only one modification from the regulator case. This one modification involves formation of the future desired trajectory in Y_D, Eq. (12). Let $y_{tr}(t)$, $t \geq 0$, where 0 is

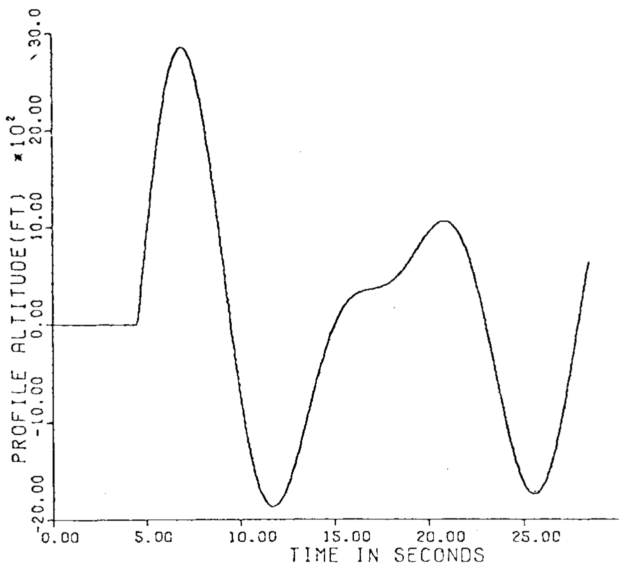

Fig. 10 Desired tracking path.

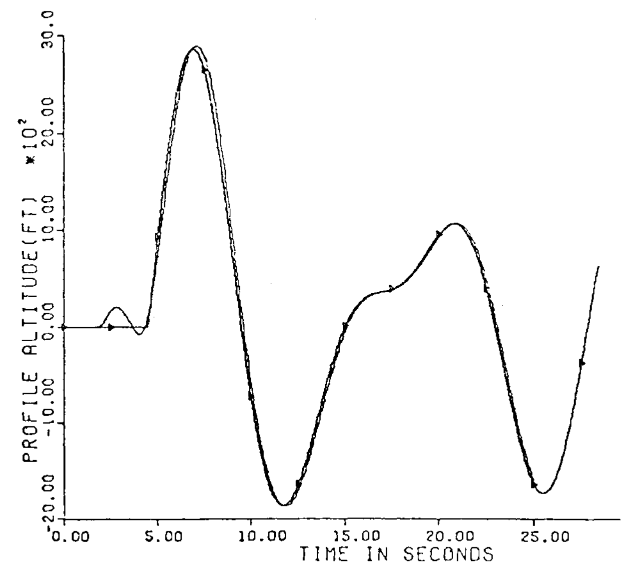

Fig. 11 Closed-loop tracking response vs desired tracking path (△).

the current time, denote the "true" desired future output path. Then we let

$$y_D(1)=y_{tr}(T_D)+\beta\cdot(y(0)-y_{tr}(0)) \tag{24}$$

$$y_D(2)=y_{tr}(2\cdot T_D)+\beta^2\cdot(y(0)-y_{tr}(0)) \tag{25}$$

$$\vdots$$

$$y_D(L\cdot \mathrm{NSM})=y_{tr}(L\cdot \mathrm{NSM}\cdot T_D)+\beta^{L\cdot \mathrm{NSM}}\cdot(y(0)-y_{tr}(0)) \tag{26}$$

$$T_D \equiv T_C/\mathrm{NSM} \tag{27}$$

is the sample time of the output points. Thus, the discrete $y_D(k)$ approaches the true continuous $y_{tr}(t)$ exponentially as k increases. With Y_D defined as such, the tracking implementation of the "output predictive algorithmic controller" is again defined by Eq. (18).

To test the scheme, a hypothetical terrain-following example was formulated in which the $y_{tr}(t)$ desired future output path became the future hypothetical terrain profile (see Fig. 10). No physical restraints were put upon the problem as this was formulated purely as a test of the control tracking performance.

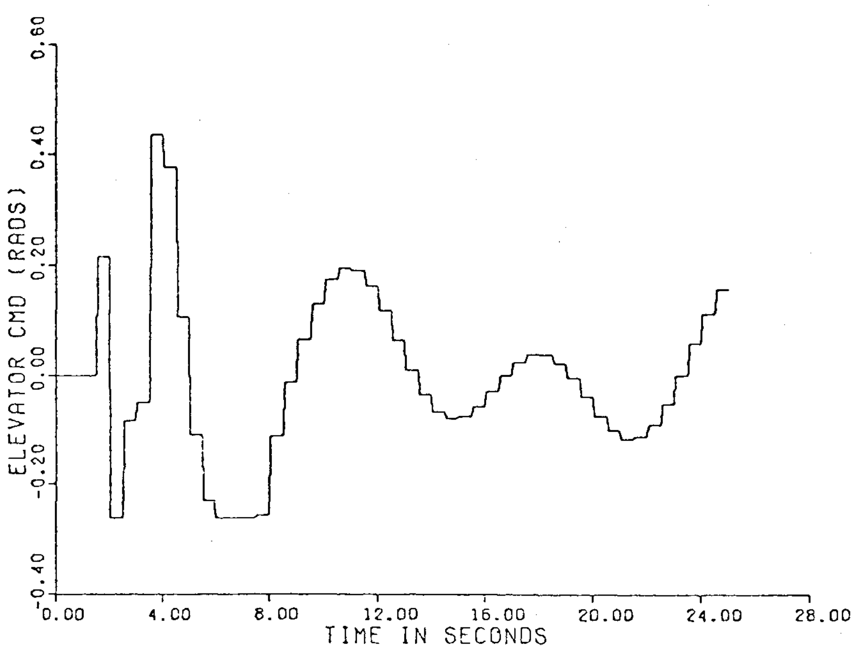

Fig. 12 Control inputs for Fig. 11 response.

The plant model chosen was a simplified, ideal, four-state model of the pitch dynamics of a modern fighter aircraft:

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} -3.854 & -12.240 & 0 & 0 \\ 1 & -2.834 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1077 & 1077 & 0 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} 34.170 \\ 0.331 \\ 0 \\ 0 \end{bmatrix} \epsilon \tag{28}$$

where q is the pitch rate, α angle of attack, θ pitch angle, h altitude, and ϵ elevator input. The output variable is then altitude. The system is nonminimum phase with transfer function

$$\frac{h(s)}{\epsilon(s)} = \frac{-356.81\,(s-14.93)(s+18.78)}{s^2\,(s+3.34 \pm j\,3.46)} \tag{29}$$

The following were the design parameters chosen: $\tau = 0.3$, $T_C = 0.5$, NSM $= 4$, $L = 7$, and

$$Q = \text{diag}[1,2,4,8,16,\ldots]_{28\times 28} \tag{30}$$

These design parameters were chosen largely on the previous rules of thumb and with a small amount of testing to pick a suitable Q matrix. The result in tracking the reference path shown in Fig. 10 (which is really the sum of three sinusoids of frequencies 0.5, 0.6, and 0.8 Hz started at $t = 5$) is shown in Fig. 11. Again it is emphasized that this is not representative of an actual terrain-following task but it demonstrates the real power of this controller. These results (and other test cases presented in Ref. 9) indicate that the output predictive algorithmic controller can be designed with an extremely high bandwidth and a very short settling time. The control inputs for this response are shown in Fig. 12. Notice that the output has achieved "steady-state" tracking after the discontinuity at $t = 5$ long before the discrete inputs settle into "steady state."

Nonminimum Phase

As noted, the dynamics of the aircraft chosen are nonminimum phase (they have a zero in the right-half plane). A zero in the right-half plane is typically known to give controller designs a great deal of difficulty. They give this controller little difficulty because of the predictive nature of the controller. The underlying difficulty in control of a nonminimum phase system is the fact that the initial system response is opposite in direction to the eventual steady-state response. Imagine trying to steer an automobile whose initial response to turning the wheel to the right, is to head left. This effect becomes more pronounced the closer the right-half plane zero moves to the imaginary axis. (See Ref. 11 for more examples and discussion.) Thus, by predicting the system response far enough ahead into the future, and by putting little or no weighting upon the initial, immediate predicted output response, the controller can be designed to essentially ignore this initial nonminimum phase characteristic. The controller then works with relatively little difficulty. Mehra[12] draws some of these same insights regarding the use of such a predictive controller scheme for the control of nonminimum phase systems. He also suggestes that the internal predictive model might be modified to further compensate for the undesirable nonminimum phase behavior.

Conclusions

This paper has developed a least-squares implementation of "output predictive algorithmic control" and shown a computational implementation to reduce memory storage requirements. The influence of the various design parameters has been shown along with some useful "rules of thumb" for their selection. The control switch time is seen to be very important to closed-loop robustness, and a method is suggested to find the "best" switch time by picking the value which maximizes the reciprocal condition number of the normal equations. Finally, the very favorable tracking performance has been shown for a hypothetical terrain-following tracking task.

References

[1]Richalet, J., Rault, A., Testud, J.L., and Papon J., "Model Predictive Heuristic Control: Application to Industrial Process Control," *Automatica,* Vol. 14, 1976, pp. 413-419.

[2]Richalet, J., "General Principles of Scenario Predictive Control Techniques," *Proceedings, 1980 Joint Automatic Control Conference,* San Francisco, Calif., Aug. 1980, FA9-A.

[3]Mehra, R.K., Kessel, W.C., Rault, A., Richalet, J., and Papon, J., "Model Algorithmic Control Using IDCOM for the F-100 Jet Engine Multivariable Control Design Problem," in *Alternatives for Linear Multivariable Control with Turbofan Engine Theme Problem,* edited by Sain, Peczkowki, and Melsa, National Engineering Consortium, Inc., Chicago, 1978, pp. 317-350.

[4]Reid, J.G., Mehra, R.K., and Kirkwood, W., "Robustness Properties of Output Predictive Deadbeat Control: SISO Case," *Proceedings, 1979 IEEE Conference on Decision and Control,* Clearwater, Fla.; Dec. 1979, pp. 307-314.

[5]Steward, G.W., "On the Perturbation of Pseudo-Inverses, Projections, and Linear Least Squares Problems," *SIAM Review,* Vol. 4, 1977, pp. 634-662.

[6]Holtzman, J., *Nonlinear System Theory: A Functional Analysis Approach,* Prentice-Hall, Englewood Cliffs, N.J., 1970, pp. 23-60.

[7]Mehra, R.K., Rouhani, R., Rault, A., and Reid, J.G., "Model Algorithmic Control: Theoretical Results on Robustness," *Proceedings, 1979 Joint Automatic Control Conference,* Denver, Colo., June 1979, pp. 387-392.

[8]Colson, H.J., "Application of Model Algorithmic Control to a Lightly Damped Single Input Single Output System," AFIT M.S. Thesis, Air Force Institute of Technology, Wright Patterson AFB, Ohio, Dec. 1978.

[9]Chaffin, D.E., "The Application of Output Predictive Digital Control to Wing Flutter Suppression and Terrain Following Problems," AFIT M.S. Thesis, Air Force Institute of Technology, Wright Patterson AFB, Ohio, Dec. 1979.

[10]Stewart, G.W., *Introduction to Matrix Computations,* Academic Press, N.Y., 1974, pp. 317-326.

[11]Reid, J.G., *Linear System Fundamentals: Continuous and Discrete, Classic and Modern,* McGeraw Hill, N.Y., (in press).

[12]Mehra, R.K. and Rouhani, R., "Theoretical Considerations on Model Algorithmic Control for Nonminimum Phase Systems," *Proceedings, 1980 Joint Automatic Control Conference,* San Francisco, Calif., Aug. 1980, TA8-B.