Reduced-Order Adaptive Control

by

James H. Hutchinson, III

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

Hugh F. VanLandingham, Chairman

Stanoje Bingulac

William T. Baumann

June, 1990

Blacksburg, Virginia

Reduced-Order Adaptive Control

by

James H. Hutchinson, III

Committee Chairman: Hugh F. VanLandingham
Electrical Engineering

(ABSTRACT)

The method of Pseudo-Linear Identification (PLID) is developed for application in an adaptive control loop. The effects of noise are investigated for the case of full-order system identification, and the results are applied to the use of PLID as a reduced-order system estimator.

A self-tuning regulator (STR) is constructed using PLID and the effects of reducing the expected order of the system are demonstrated. A second adaptive control algorithm is presented wherein the STR controller is varied to achieve some degree of closeness to a given model (model-reference adaptive control).

Acknowledgements

I wish to thank my parents, Dr. and Mrs. James H. Hutchinson, Jr., and my sister Melinda for their unflagging support of my studies in every conceivable way, not the least being financial. The encouragement they offered during the last grueling weeks was especially appreciated. This work is dedicated to them.

To the members of my committee I offer my thanks for their suggestions to make this thesis more legible and coherent. I especially wish to thank them for service above and beyond the call of duty for coming in on Saturday for my defense.

Thanks also must be expressed to the guys in the control lab for helping to relieve the pressure when it got to be unbearable, for occasional helpful suggestions, and for generally making my experience here a pleasant one. Most special thanks to David Cooper, my classmate and friend who somehow put up with me for longer than I thought possible.

Table of Contents

1.0 Introduction

It is often the task of the engineer to design a system to satisfy certain specifications. While this is often sufficiently difficult to require years of training, there is an advantage to design work: the designer knows his system because he specifies it. A common case is the design of a controller to modify some characteristics of another system called the plant. The plant dynamics may be partially or wholly unknown. This is often the case with natural phenomena. Other times the plant may be known (perhaps having been designed and documented) but parameters have changed during operation. Since many controllers are designed from knowledge of the plant parameters, specifying a controller from inaccurate data would tend to degrade the performance of the plant.

In each of these cases it is of great help to the designer to be able to identify or estimate the plant involved. A number of techniques have been developed to estimate linear systems given a time history of input and output [1]. Another help to the designer is to the ability to estimate the states of a system when the actual states are for some reason unobtainable, since state feedback is a popular solution to many control problems. The Kalman filter is a prime example of such an estimator [2]. Some algorithms have been developed to simultaneously estimate the states and identify the system, such as the Extended Kalman Filter [3]. One of these is called Pseudo-Linear Identification (PLID) [4].

Having a running estimate of the system is often valuable to the designer. Some controllers that are desirable (optimal) in other ways are sensitive to changes in system parameters. These controllers may still be used in changing plants provided that an intervening "design" step is inserted into the controller. Taking plant changes into account in the computation of a control input is called adaptive control.

Previous work in this area includes Hopkins' dissertation [5], in which the stochastic, multi-input/multi-output version of PLID is thoroughly developed. Also of interest is the thesis of Kemp [6], in which PLID is used in an adaptive pole-placement scheme.

A case of interest is that of a plant of such complexity that it is difficult to design a controller to take into account all the dynamics of the system and keep within specified cost or time constraints. Additionally, a controller attempting to take all the dynamics of such a system into account quickly becomes very complex itself. It would be helpful to know if the plant could be adequately represented by a model of reduced complexity. If so, then perhaps the control law need not be very complex itself, resulting in considerable savings.

The idea of controller reduction has been put forth recently [7] as being preferable to plant reduction, the argument being that the later in the design process that approximations are made, the better the resulting controller is likely to be. However, in the case of simultaneous system identification and control calculation, the smaller the system being identified the less delay there is likely to be in applying the appropriate control. Some systems may be satisfactorily represented by a substantially reduced model.

In this work, the single-input/single-output (SISO) stochastic version of PLID is developed and its use in identifying various systems is demonstrated. It is then used to adaptively control various plants using an estimated reduced-order model. The control is chosen to minimize a given cost function, specifically a linear quadratic function of output error and control effort (the LQR). Finally, the weighting parameters of the LQR cost function are adjusted during operation, closing a second loop as the controller seeks to match a specified performance criterion.

2.0 SISO Stochastic PLID development

Consider the block diagram of a single-input/single-output (SISO) system shown in figure 2.0.1. This system may be assumed to be a discrete-time version of a strictly proper continuous-time system, sampled at intervals of T seconds and transformed using a zero-order hold [8]. Therefore the response of this system will be exactly the same as that of its continuous-time analog at time t = kT, where k is a non-negative integer. We may use the expression "time k" to mean "time t = kT."

The system may be represented mathematically with the following difference equations:

$$x_{k+1} = Ax_k + B(u_k + w_k) = Ax_k + B\mu_k \qquad (2.0.1a)$$

$$y_k = Cx_k \qquad (2.0.1b)$$

$$z_k = Cx_k + v_k = y_k + v_k \qquad (2.0.1c)$$

As is the custom in the literature, x_k represents the state at time k and y_k is the output of the system. Output or measurement noise is represented by v_k, and input or actuator noise by w_k. The actuator noise is assumed to be unmeasurable,

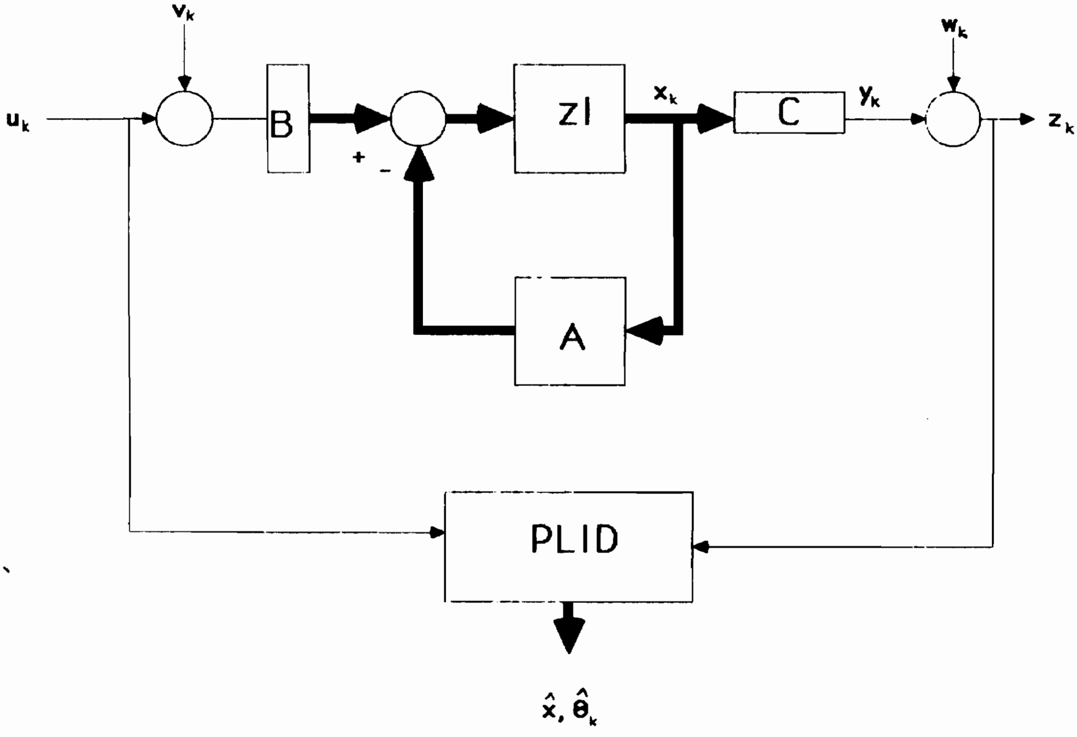


Figure 2.0.1. The plant assumed.

but of known statistics. We assume this because we are developing PLID for use with a controller, and we will therefore have knowledge of our commanded control input u_k but the actual input μ_k to the plant may be different.

We shall require our system to be completely state observable and completely state controllable. The first requirement is so that the system may be transformed into observable canonical form. The second is so that we may be certain of exciting all the modes in the system This is the "persistent excitation" requirement common to all identification methods.

2.1 Extendend State Model (ESM) representation

If we require that our system be completely state-observable, then we may by means of a linear transformation rewrite the system as

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}(u_k + w_k) = \tilde{A}x_k + \tilde{B}\mu_k \tag{2.1.1a}$$

$$z_k = \tilde{C}\tilde{x}_k + v_k \tag{2.1.1b}$$

where $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are in observable canonical form, as shown below. We will drop the tedious tilde ($\tilde{}$) notation, but the reader should bear in mind that the state now used is a linear transformation of the original state description.

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & a_1 \\ 1 & 0 & \cdots & 0 & a_2 \\ 0 & 1 & \cdots & 0 & a_3 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & a_n \end{bmatrix}, \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad \text{and} \quad x = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix}^T.$$

The reader will recall that the a_i and b_i parameters may be read off the transfer function description

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_n z^{-1} + b_{n-1} z^{-2} + \cdots + b_2 z^{-(n-1)} + b_1 z^{-n}}{1 - a_n z^{-1} - a_{n-1} z^{-2} - \cdots - a_2 z^{-(n-1)} - a_1 z^{-n}} \tag{2.1.2}$$

Looking at the transformed system in a row-wise manner, we may write the following system of equations.

$$\begin{aligned}
x_1(k+1) &= \qquad\qquad\quad a_1 x_n(k) + b_1[u_k + w_k] \\
x_2(k+1) &= x_1(k) + a_2 x_n(k) + b_2[u_k + w_k] \\
x_3(k+1) &= x_2(k) + a_3 x_n(k) + b_3[u_k + w_k] \\
&\vdots \\
x_n(k+1) &= x_{n-1}(k) + a_n x_n(k) + b_n[u_k + w_k]
\end{aligned} \tag{2.1.3}$$

Now note that the measurement of the transformed system is always the last component of the state plus the measurement noise: $z(k) = x_n(k) + v_k$. This may be rearranged to read $x_n(k) = z(k) - v_k$ and substituted into the preceding system of equations 2.1.3.

$$x_i(k+1) = x_{i-1}(k) + a_i[y(k) - v_k] + b_i[u_k + w_k] \tag{2.1.4}$$

Through some algebraic manipulation we may rework this system of equations into a well-known form. Recall that we are assuming to know the input and output at each time instant k, and that we are seeking to solve the equation for the state and transfer function parameters. Gather the noise components into a single vector and augment the state vector with vectors containing the numerator and denominator parameters of the transfer function (which are the matrix B and the last column of the matrix A). Then the new extended state representation (ESR) may be written as

$$s_{k+1} = Fs_k + G\eta_k \tag{2.1.5a}$$

$$z_k = Hs_k + w_k \tag{2.1.5b}$$

where

$$s_k = [\ x_k{}^T\ \Theta_A{}^T\ \Theta_B{}^T]^T,$$

$$\Theta_A = [a_1\ a_2\ \cdots\ a_n]^T, \quad \Theta_B = [b_1\ b_2\ \cdots\ b_n]^T,$$

$$F_k = \begin{bmatrix} J_n & y_k\ I_n & u_k\ I_n \\ 0 & I_n & 0 \\ 0 & 0 & I_n \end{bmatrix},$$

$$G_k = \begin{bmatrix} -\ \Theta_A & \Theta_B \\ 0 & 0 \end{bmatrix}, \quad H = [\ C \quad 0\ ], \quad \eta_k = \begin{bmatrix} v_k \\ w_k \end{bmatrix}$$

and J_n is an nxn zero matrix with ones on the first subdiagonal.

We will assume each component of the noise vector to be zero-mean and gaussian-distributed. Let the various covariances be known as follows.

$$E[w_k{}^2] = Q_1(k) \tag{2.1.6a}$$

$$E[v_k{}^2] = R(k) \tag{2.1.6b}$$

$$E[v_k w_k] = S_{vw}(k) \tag{2.1.6c}$$

Now the statistics of the noise vector may be constructed.

$$E[\eta_k] = 0 \tag{2.1.7a}$$

$$E[\eta_k \eta_k{}^T] = Q_k = \begin{bmatrix} R_k & S_{wv} \\ S_{vw} & Q_{1k} \end{bmatrix} \tag{2.1.7b}$$

The reader will note that although the original n^{th}-order system might have had all uncorrelated noise, when the extended state model is built and the composite noise vector constructed there *will* be some cross-covariance. At the very least, the output autocovariance term R_k is present.

$$E[\eta_k v_k] = S_k = \begin{bmatrix} R_k \\ S_{vw}(k) \end{bmatrix} \tag{2.1.8}$$

It has been shown by Hopkins that the update of the extended state vector under these noise assumtions is a first-order Markov process and is jointly gaussian. This implies that the observation equation 2.1.5b is jointly gaussian distributed, which in turn implies that the conditional linear minimum-mean-square error (MMSE) estimator is equivalent to the conditional mean estimator [9].

Calculate the linear conditional MMSE recursive prediction of the extended state vector by

$$\hat{s}_{k+1|k} = M_k \hat{s}_{k|k-1} + \overline{K}_k z_k \tag{2.1.9}$$

$\overline{K}$ is then chosen to minimize the conditional mean square error. If the prediction is to be unbiased, then we must have

$$E[s_{k+1}|s_k] = E[\hat{s}_{k+1}|s_k] \tag{2.1.10}$$

But, substituting equation 2.1.9 into equation 2.1.10,

$$E[s_{k+1}|s_k] = E[(M_k \hat{s}_{k|k-1} + \overline{K}_k z_k)|s_k] = F_k s_k \tag{2.1.9}$$

so that by including equation 2.1.5b

$$F_k s_k = M_k s_{k|k-1} + \overline{K}_k (H s_k) \tag{2.1.11}$$

Therefore

$$F_k = M_k + \overline{K}_k H \quad \text{or} \quad M_k = F_k - \overline{K}_k H$$

and so

$$\hat{s}_{k+1|k} = (F_k - \overline{K}_k H)\hat{s}_{k|k-1} + \overline{K}_k z_k$$

This may be rewritten as

$$\hat{s}_{k+1|k} = F_k \hat{s}_{k|k-1} + \overline{K}_k (z_k - H\hat{s}_{k|k-1}) \tag{2.1.12}$$

The preceding development has ignored the noise present in the model, so define the prediction error by

$$e_{k+1|k} \triangleq \hat{s}_{k+1|k} - s_{k+1|k} \tag{2.1.13}$$

$$= [(F_k - \overline{K}_k H)\hat{s}_{k|k-1} + \overline{K}_k z_k] - s_{k+1|k}$$

$$= [(F_k - \overline{K}_k H)\hat{s}_{k|k-1} + \overline{K}_k (Hs_k + v_k)] - s_{k+1|k}$$

$$= [(F_k - \overline{K}_k H)\hat{s}_{k|k-1} + \overline{K}_k (Hs_k + v_k)] - (F_k s_k + G_k \eta_k)$$

This expression may be rearranged to predict the next error given the current error:

$$e_{k+1|k} = (F_k - \overline{K}_k H)(\hat{s}_{k|k-1} - s_k) + \overline{K}_k w_k - G_k \eta_k$$

$$= (F_k - \overline{K}_k H)e_{k|k-1} + \overline{K}_k w_k - G_k \eta_k \tag{2.1.14}$$

Now we construct the error covariance matrix $P_{k+1|k}$ by

$$P_{k+1|k} = E[e_{k+1|k}e_{k+1|k}^{\top}] \tag{2.1.15}$$

This may be expanded by substituting equation 2.1.14 to find

$$\begin{aligned} P_{k+1|k} &= E[(F_k - \overline{K}_k H)e_{k|k-1}e_{k|k-1}^{\top}(F_k - \overline{K}_k H)^{\top} + \overline{K}_k v_k v_k^{\top}\overline{K}_k^{\top} \\ &\quad + G_k\eta_k\eta_k^{\top}G_k^{\top} - \overline{K}_k v_k\eta_k^{\top}G_k^{\top} - G_k\eta_k v_k^{\top}\overline{K}_k^{\top} \\ &\quad - (F_k - \overline{K}_k H)e_{k|k-1}v_k^{\top}\overline{K}_k^{\top} - \overline{K}_k v_k e_{k|k-1}^{\top}(F_k - \overline{K}_k H)^{\top} \\ &\quad - (F_k - \overline{K}_k H)e_{k|k-1}\eta_k^{\top}G_k^{\top} - G_k\eta_k e_{k|k-1}^{\top}(F_k - \overline{K}_k H)^{\top}] \end{aligned}$$

Using the linearity property of the expectation operator (and referring to Hopkins), this rather large expression reduces to

$$\begin{aligned} P_{k+1|k} &= (F_k - \overline{K}_k H)E[e_{k|k-1}e_{k|k-1}^{\top}](F_k - \overline{K}_k H)^{\top} + \overline{K}_k E[v_k v_k^{\top}]\overline{K}_k^{\top} \\ &\quad + G_k E[\eta_k\eta_k^{\top}]G_k^{\top} - \overline{K}_k E[v_k\eta_k^{\top}]G_k^{\top} - G_k E[\eta_k v_k^{\top}]\overline{K}_k^{\top} \end{aligned}$$

Note: $E[G_k\eta_k\eta_k^{\top}G_k^{\top}] = G_k E[\eta_k\eta_k^{\top}]G_k^{\top}$ since

$$E[G_k] = E\begin{bmatrix} -\hat{\theta}_A(k+1|k) & \hat{\theta}_B(k+1|k) & \\ & & I_{3n} \\ 0 & 0 & \end{bmatrix}$$

But

$$E[G_k] = \begin{bmatrix} -\hat{\Theta}_A(k+1|k) & \hat{\Theta}_B(k+1|k) \\ 0 & 0 \end{bmatrix} = G_k$$

because the parameters are assumed to remain constant and so

$$P_{k+1|k} = (F_k - \overline{K}_k H)P_{k|k-1}(F_k - \overline{K}_k H)^{\top} + \overline{K}_k R_k \overline{K}_k{}^{\top}$$

$$+ G_k Q_k G_k{}^{\top} - \overline{K}_k S_k{}^{\top} G_k{}^{\top} - G_k S_k \overline{K}_k{}^{\top}$$

To compute the mean-square-error prediction-minimizing gain $\overline{K}_k$, choose the functional trace($P_{k+1|k}$). To minimize $\overline{K}_k$, set

$$\frac{\partial}{\partial \overline{K}_k} \mathrm{tr}(P_{k+1|k}) = 0 \tag{2.1.16}$$

to find

$$2K_k(HP_{k|k-1}H^{\top} + R_k) - 2(F_k P_{k|k-1}H^{\top} + G_k S_k) = 0 \tag{2.1.17}$$

Solving,

$$K_{k|k} = [F_k P_{k|k-1}H^{\top} + G_k S_k](HP_{k|k-1}H^{\top} + R_k)^{-1} \tag{2.1.18}$$

This leads to

$$P_{k+1|k} = F_k P_{k|k-1} F_k^T + G_k Q_k G_k^T - K_{k|k}(H P_{k|k-1} H^T + R_k) K_{k|k}^T \quad (2.1.19)$$

and

$$\hat{s}_{k+1|k} = (F_k - K_{k|k} H)\hat{s}_{k|k-1} + K_{k|k} z_k \quad (2.1.20)$$

These last three equations define the PLID algorithm.

2.2 SISO Stochastic PLID algorithm and properties

It should be noted by the reader that due to the nature of G, only the state portion of the covariance update equation is affected by the middle term. This allows the state and parameter portions of the gain calculation and extended state update to be calculated separately. In practice, the extended state vector is calculated first. Then the parameter portion of it is used to update G_k, which is used in the gain and the covariance updates, as well as the state-only portion of the extended state update.

Presented below is the algorithm used in the following chapters:

$$\tilde{K}(k) = \begin{bmatrix} \tilde{K}_x(k) \\ \tilde{K}_\Theta(k) \end{bmatrix} = F_k P_{k|k-1} H^T [H P_{k|k-1} H^T + R_k]^{-1} \quad (2.2.1)$$

$$K_\Theta(k) = \tilde{K}_\Theta(k) \quad (2.2.2)$$

$$\tilde{s}(k+1|k) = \begin{bmatrix} \tilde{x}(k+1|k) \\ \tilde{\Theta}(k+1|k) \end{bmatrix} = F_{k-1} s_{k|k-1} \quad (2.2.3)$$

$$\Theta(k+1|k) = \tilde{\Theta}(k+1|k) + K_{\Theta}[z_k - Hs(k|k\text{-}1)] \quad (2.2.4)$$

$$G_k = [\ \Theta_A(k+1|k)\ \Theta_B(k+1|k)\ ] \quad (2.2.5)$$

$$K_x(k) = \tilde{K}_x(k) + G_k S_k[HP_{k|k-1}H^T + R_k]^{-1} \quad (2.2.6)$$

$$s(k+1|k) = \tilde{s}(k+1|k) + K_x[z_k - Hs(k|k\text{-}1)] \quad (2.2.7)$$

$$P(k+1|k) = F_k P(k|k\text{-}1)F_k^{\,T} + GQG^T - K(k)[HP_{k|k-1}H^T + R(k)]K^T(k) \quad (2.2.8)$$

The PLID algorithm has been shown, in the previous work, to be the optimal estimator in a least-squares sense. It has been shown to identify parameters without bias in the minimum time of 3n samples (deadbeat response) in the deterministic case. It has been observed to have a slight bias when noise is introduced to the model, and large amplitude noise delays convergence for a long time. Convergence has been proved for time-invariant parameters in the stochastic case by Hopkins, although disclaimers were added in the cases of too low a signal-to-noise ratio and numerical ill-conditioning. In the first case convergence may not be reached in a reasonably short period of time, while in the second case convergence may appear to have been reached, but to incorrect values.

It was also stated by Hopkins that PLID becomes suboptimal when the parameters begin to vary in a non-deterministic way due to the fact that the states will then have a non-gaussian probability denstiy function. It is this requirement that will be violated in the next chapter when reduced-order models will be estimated. When a system is assumed to be of a lower order than it actually is, the extra dynamics are unmodelled and appear as noise to the identification algorithm, but not noise with a gaussian distribution.

3.0 System Identification

In this chapter Pseudo-Linear Identification will be used to identify various systems. A special method for identifying highly oscillatory (modal) systems will be given. It will be demonstrated varying the value of the noise covariances supplied to PLID can be valuable in achieving convergence in the case of high noise.

Then PLID will be used to estimate reduced-order models of some systems. Systems with far-away poles and near pole-zero cancellations will be demonstrated, as well as systems not so given to easy order reduction. The value of changing the estimate of the noise covariances will be demonstrated in the case of model reduction.

3.1 Full-Order Simulations

In his thesis, Kemp demonstrated the ability of PLID to identify systems that were all combinations of stable, unstable, minimum phase and non-minimum phase. He showed that low-amplitude noise had the effect of extending the time it took for PLID to converge, but that convergence was still rapid. Higher-amplitude noise was demonstrated to significantly extend the time of convergence, and also the time it took to near convergence. In this section we shall investigate the effect of overestimating the noise covariance.

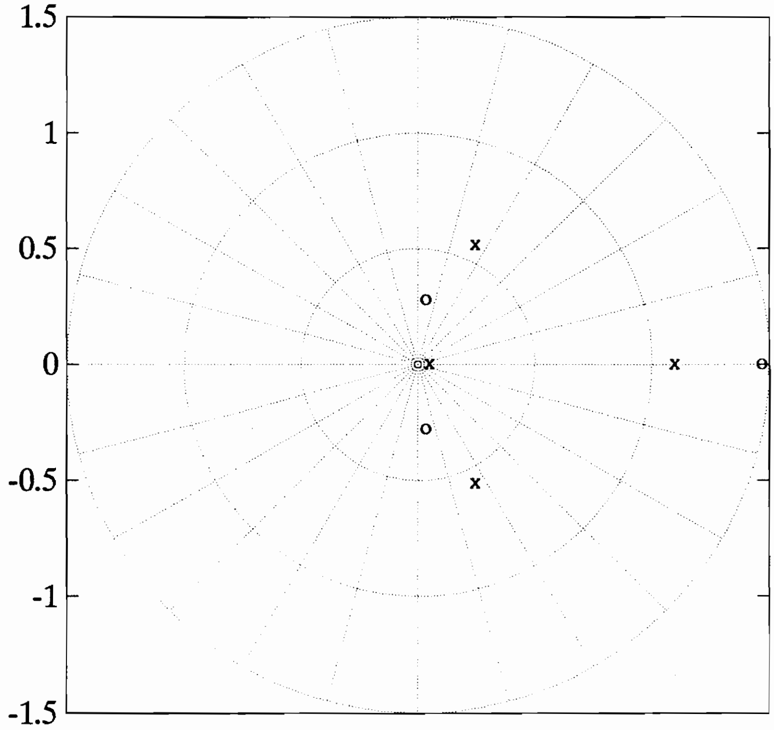
To demonstrate PLID's robustness, an unstable, non-minimum phase fourth-order plant (Plant A, Figure 3.1.1) was used for the following simulations. In order to satisfy the "persistent excitation" condition, a dithering signal of normally-distributed zero-mean noise was applied through the control input u_k (see Appendix for computer programs).

For the first case no noise was added to the input or output. This is the deterministic case, where the input and output of the plant are known exactly. It is seen (figure 3.1.2a) that the parameters converge to the correct values within the expected deadbeat time of 12 samples.

Next a pair of small-amplitude noise signals of known covariances were added to both the control input and the measurement. This is the stochastic case, the convergence of which was proved by Hopkins. For this case of a signal-to-noise ratio of approximately 60dB, we see (figure 3.1.2b) that convergence is rapid.

Larger-amplitude noise signals, ones only about 30dB below the input control signal, were then applied. It can be seen from figure 3.1.2c that convergence has been seriously delayed, and that some of the parameters are not even in the neighborhood of the correct values. It is of interest to note that parameter b_1 consistently demonstrates the most rapid and accurate convergence. This parameter corresponds to the open-loop gain of the plant.

An improvement in both the rate and accuracy of convergence of the parameters can be seen in figure 3.1.2d. In this case, the same larger-amplitude noise was applied to the plant, but the PLID algorithm was made to expect an even larger amplitude noise signal, one only 20dB below the control signal. It was



Poles and zeros of Plant A

Plant A $$G_A(z) = \frac{0.2325z^{-1} - 0.3583z^{-2} + 0.0428z^{-3} - 0.0270z^{-4}}{1 - 1.6434z^{-1} + 0.9493z^{-2} - 0.4012z^{-3} + 0.0179z^{-4}}$$

s-plane zeros at: 1.0, -4.0±4.0j

s-plane poles at: 0.25, -1.5±3.0j, -8.0

Figure 3.1.1. An unstable, non-minimum phase discrete-time system.

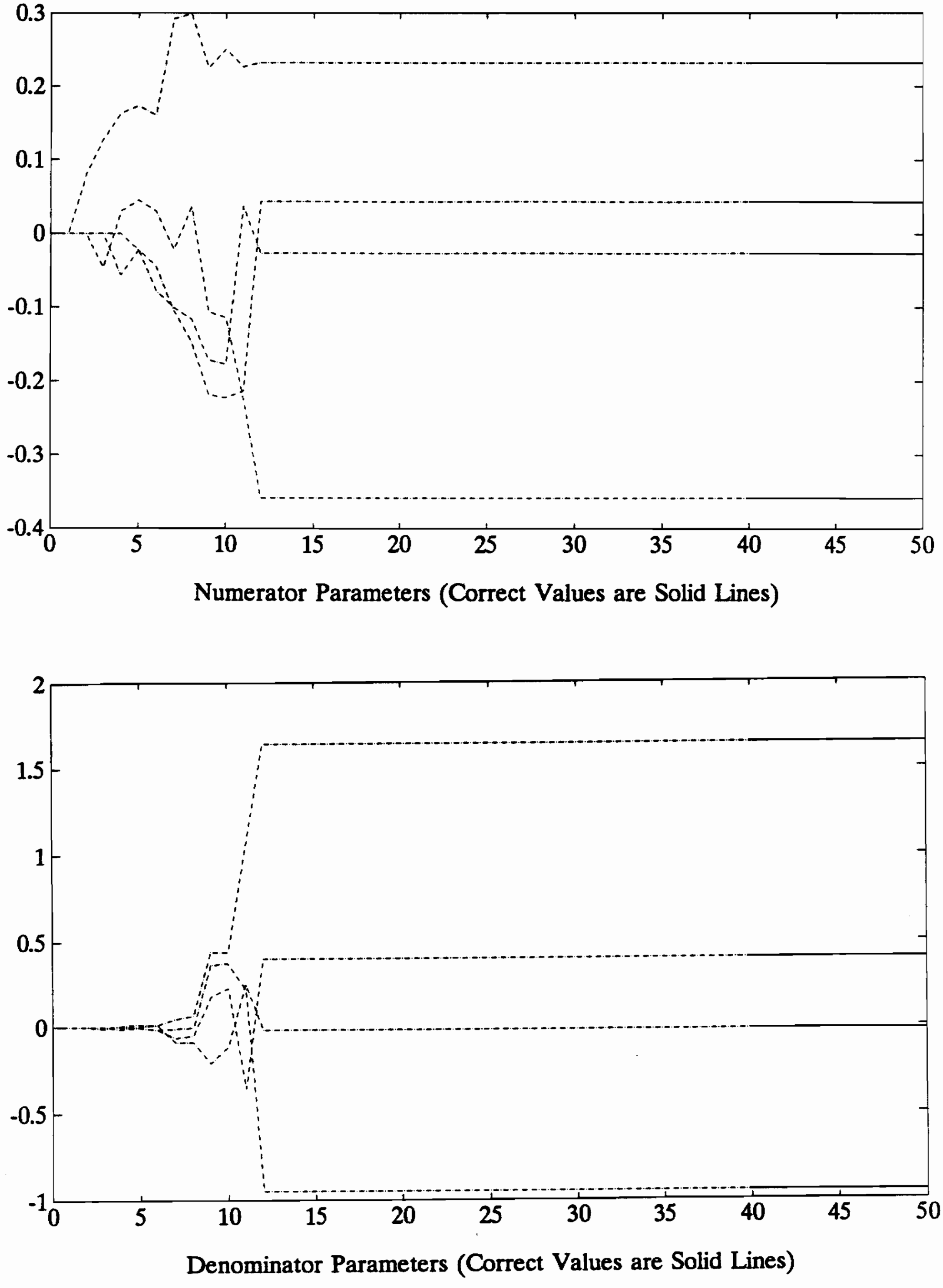


Figure 3.1.2a. Plant A identification with no noise.

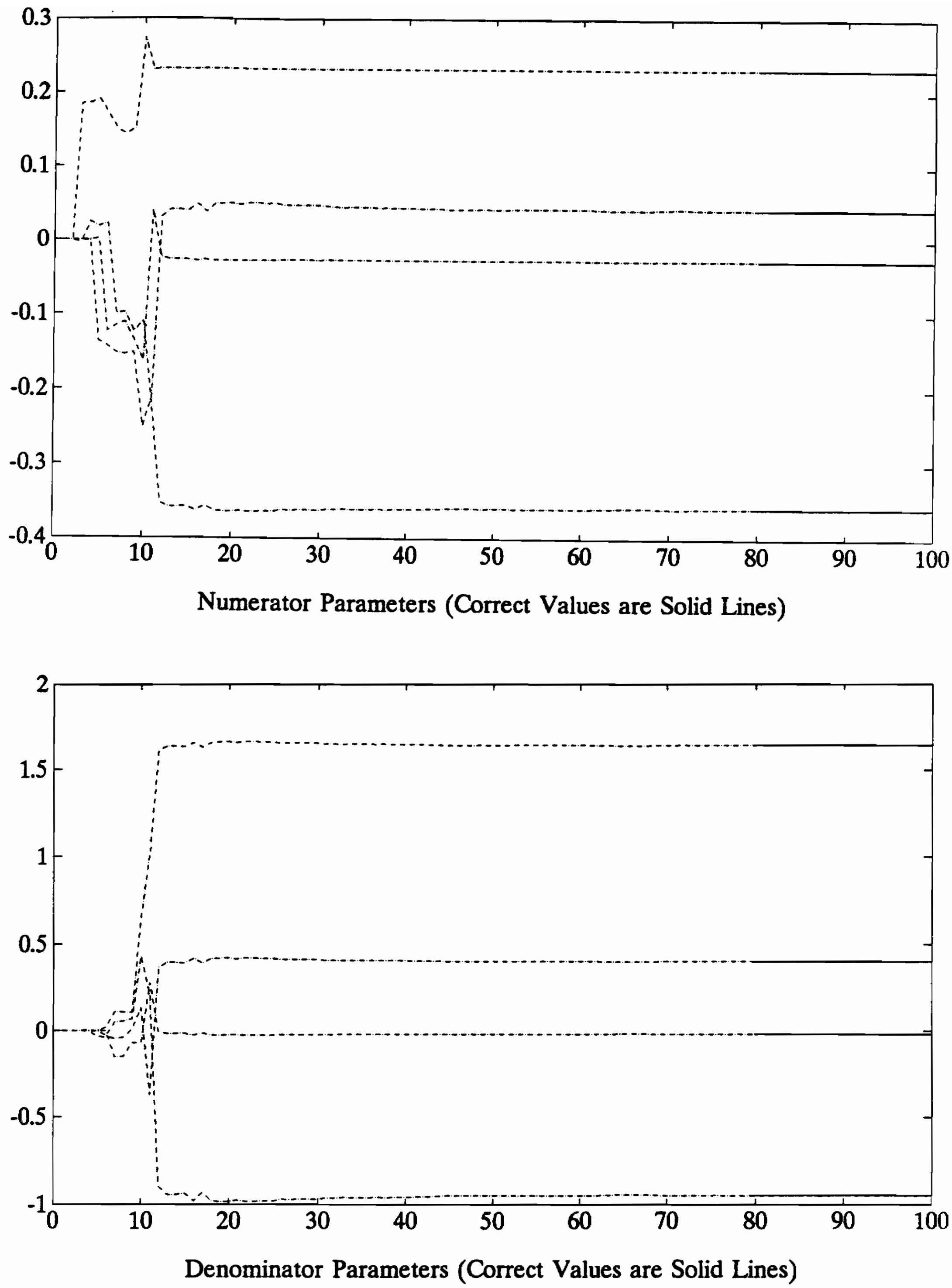


Figure 3.1.2b. Plant A identification with low noise.

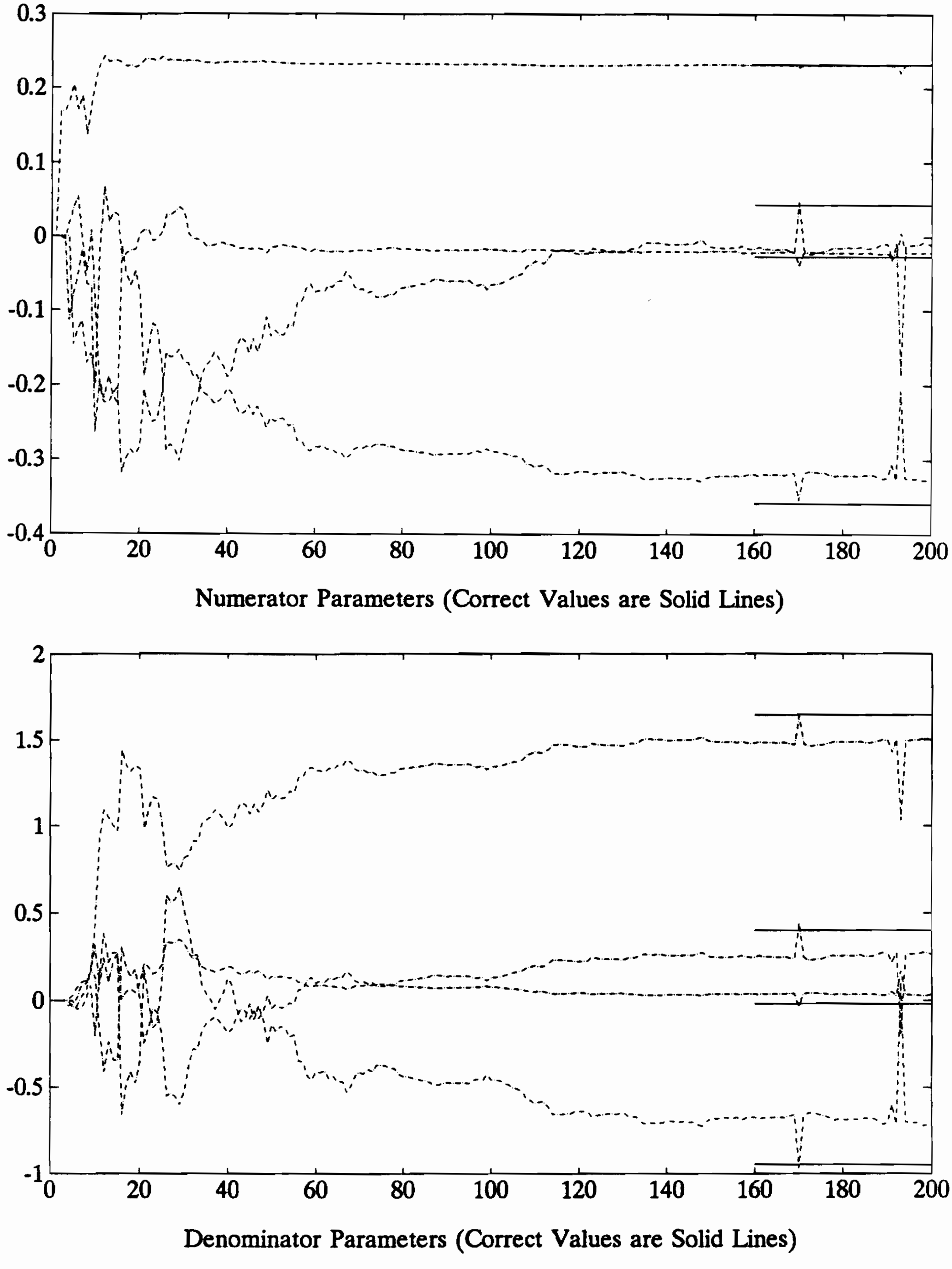


Figure 3.1.2c. Plant A identification with high noise.

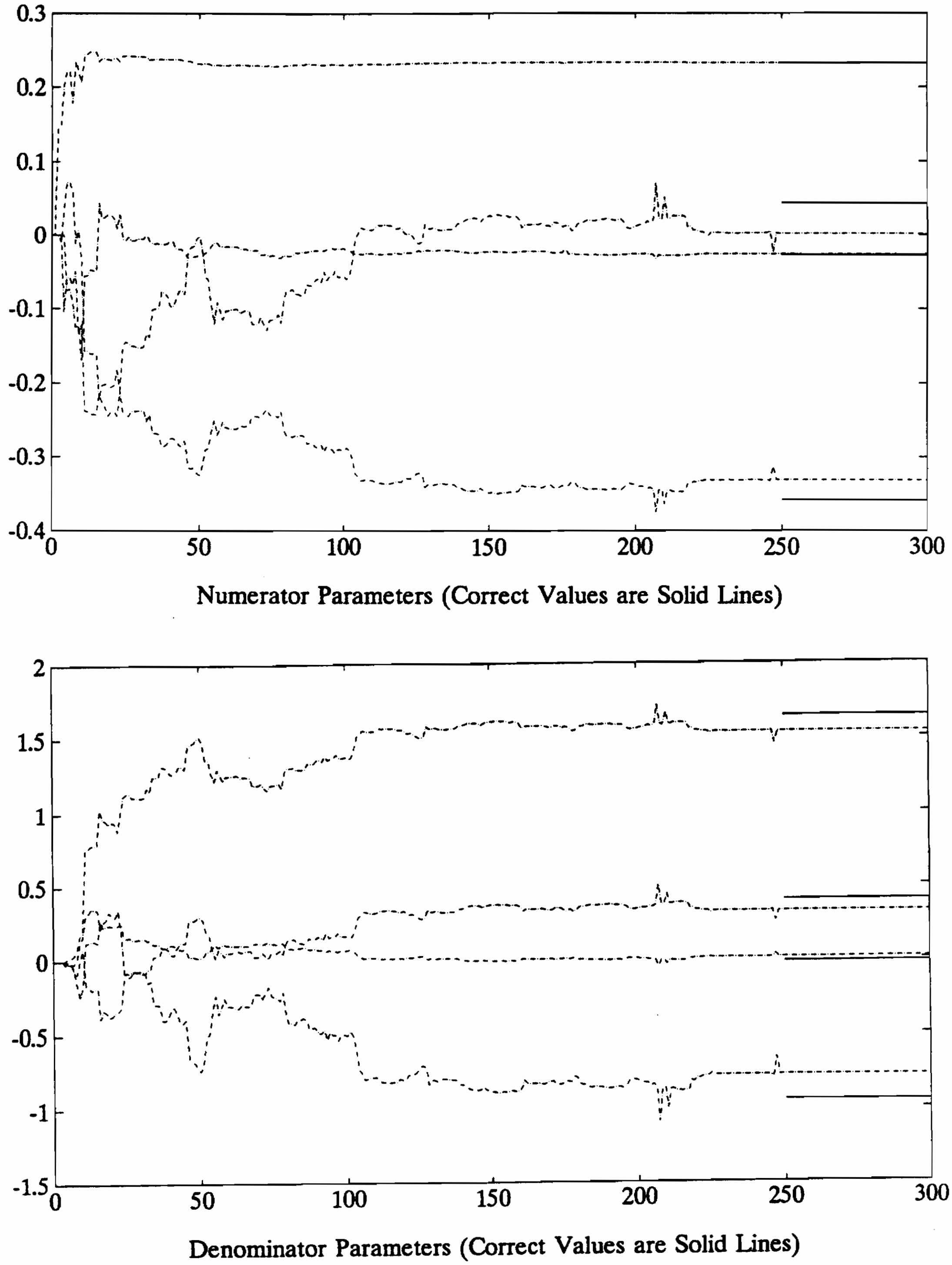
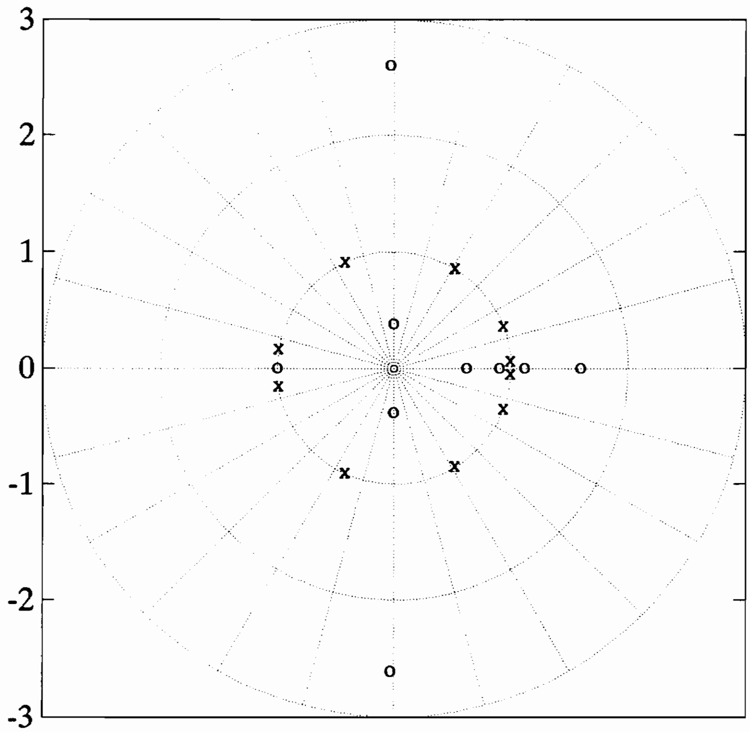


Figure 3.1.2d. Plant A identification with high noise overestimated.

observed over many trials that over-estimating the noise covariance helped in the convergence of the parameter estimates, as was demonstrated by this example. In particular, the spikes observed even after the parameters appear to have neared convergence in Figure 3.1.2c are seen to be considerably reduced in magnitude in Figure 3.1.2d. The parameter values are also seen to converge closer to the correct values when the noise covariance is over-estimated. This technique will be seen to help especially when reduced-order models are required.

A system with complex s-plane poles very near the $j\omega$-axis will tend to exhibit a very oscillatory response. Such a system is illustrated in discrete time by Plant B in figure 3.1.3. This model represents a cantilevered beam, the input to which is a piezo-electric device that induces vibration and the output of which is proportional to the integral of strain along the length of the beam, as measured by an optical fiber adhering to the beam. It was found, through trial and error, that the best way to identify such a system using PLID is to apply a unit impulse and wait for the denominator coefficients a_i to converge. When they do, a dithering signal of small-amplitude noise should be applied and the covariance matrix reset to a large value. The a_i parameters will not be disturbed, but the b_i parameter estimates will diverge from the false values they had attained and then converge on the true parameters (figure 3.1.4). Identification of this system exhibited the same general behavior as the previous example, other than the characteristic of being very intolerant of noise during the initial phase as the poles are identified.



Poles and zeros of the modal system

Figure 3.1.3. Plant B, a modal system.

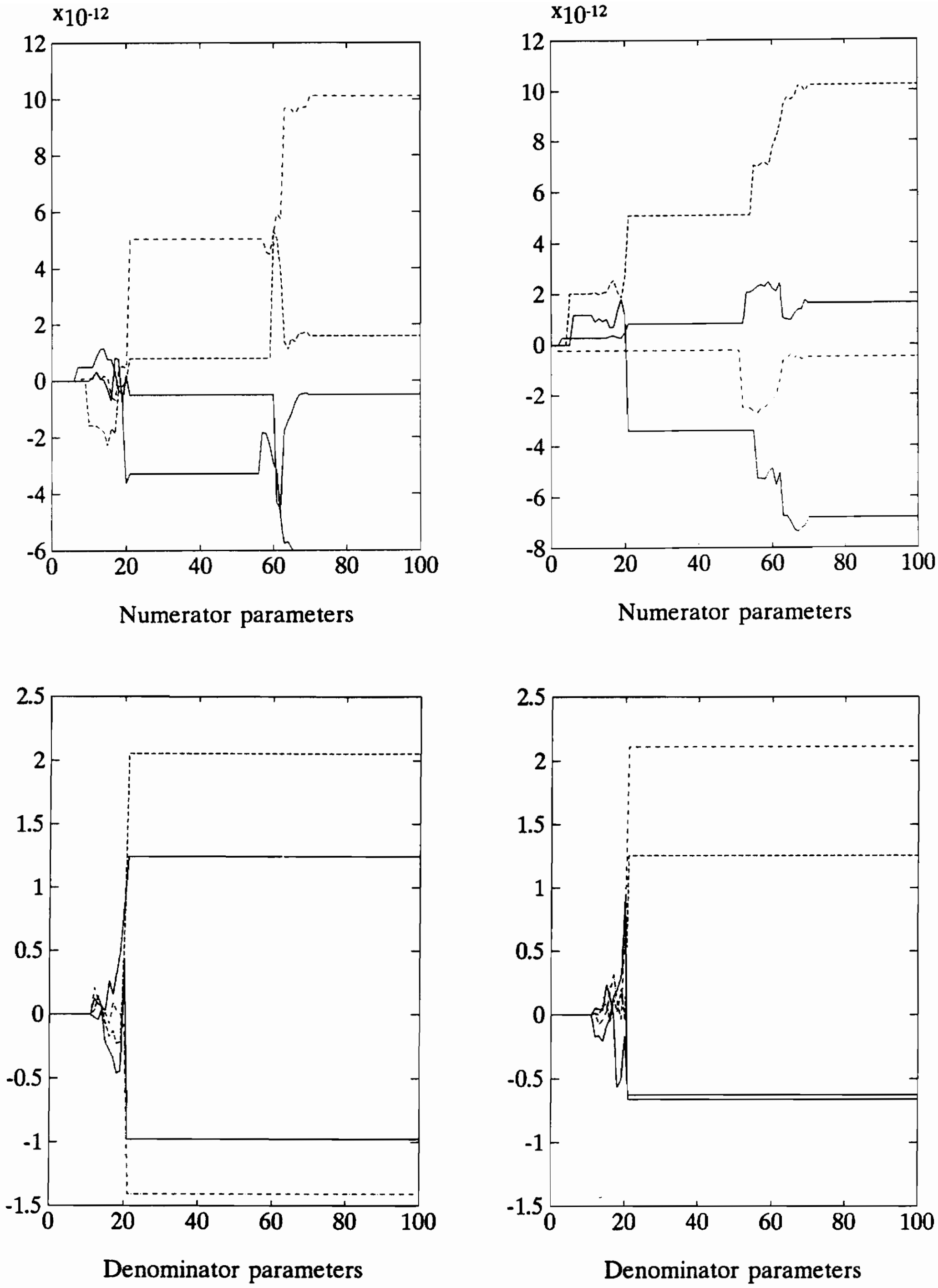


Figure 3.1.4. Identification of the Plant B parameters.

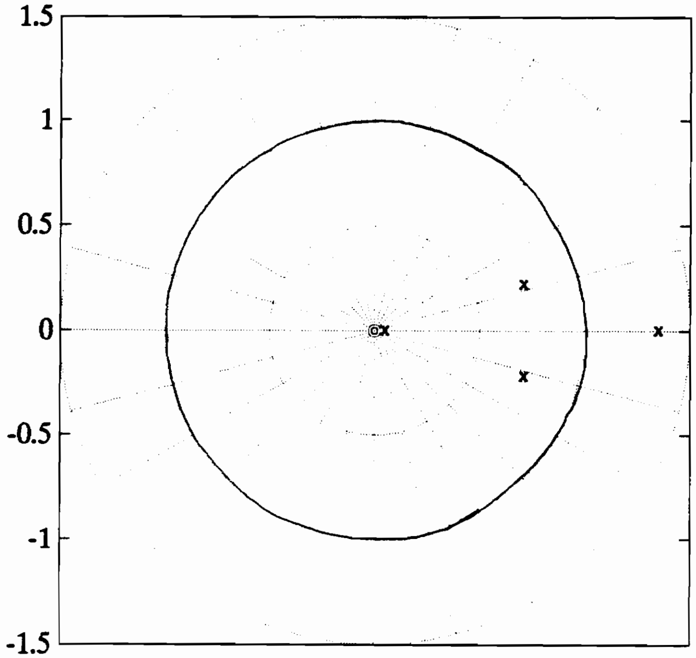
3.2 Reduced-Order Estimation

A feature of the PLID algorithm is that the order of the system to be identified is specified when the filter is constructed. This means that, given sufficient input, the algorithm will identify the "best-fitting" model, in a least-squares sense, of the given order.

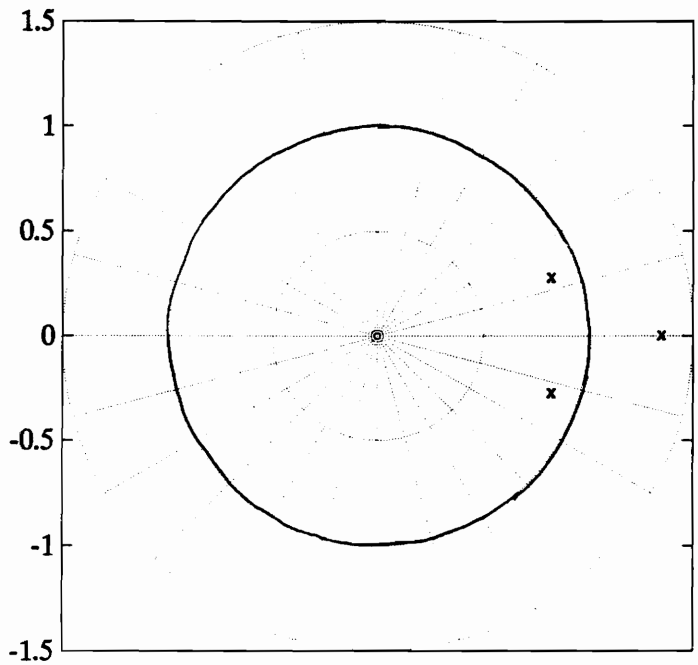
Some systems lend themselves to order reduction: those with a single, far-away pole, for example. One "rule of thumb" used is that a pole or zero may be neglected if it is more distant in the left half-plane than a factor of four of the real part of the next-farthest pole [10]. An example is provided in Plant C, Figure 3.2.1. Also, systems with a near pole/zero cancellation are easily reduced. The reader will recall that a system must be of minimal representation (must have no pole/zero cancellations) to be fully observable and fully controllable. The reduced system in this case will be a polynomial without the cancelling pole and zero. If the pole and zero are complex, the order will be reduced by two. Plant D in Figure 3.2.2 is an example of this type of easily-reduced system.

Other systems, however, are not so easily dealt with. While PLID gives the expected results in the preceding cases, it is in these systems that the estimation is the most valuable, when the engineer has little intuition about the expected system parameters.

Tenth-order plants will be used in the following section. They will be reduced to fifth-order and to third-order to demonstrate the possibility of reducing the complexity of a system to half or even to a third. Pole-zero plots, frequency and time responses are presented to demonstrate the closeness of fit of the estimated systems.

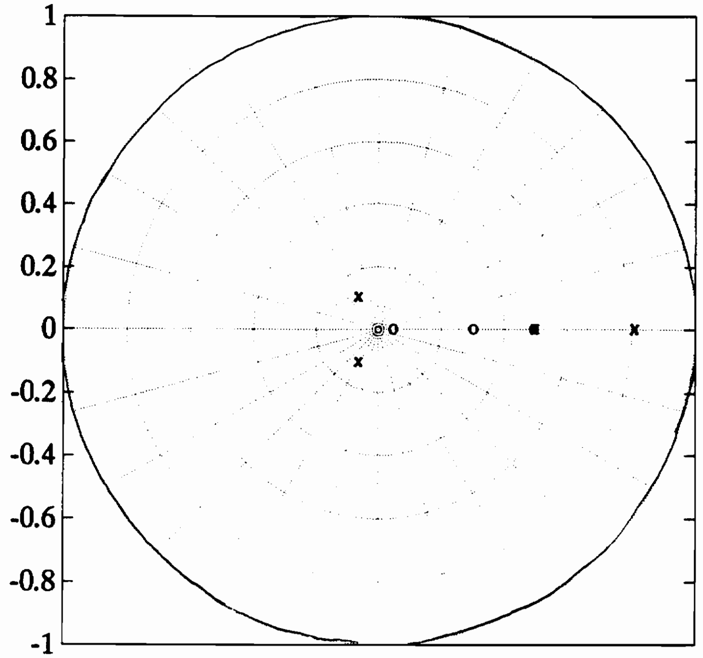


Actual 4th Order Poles

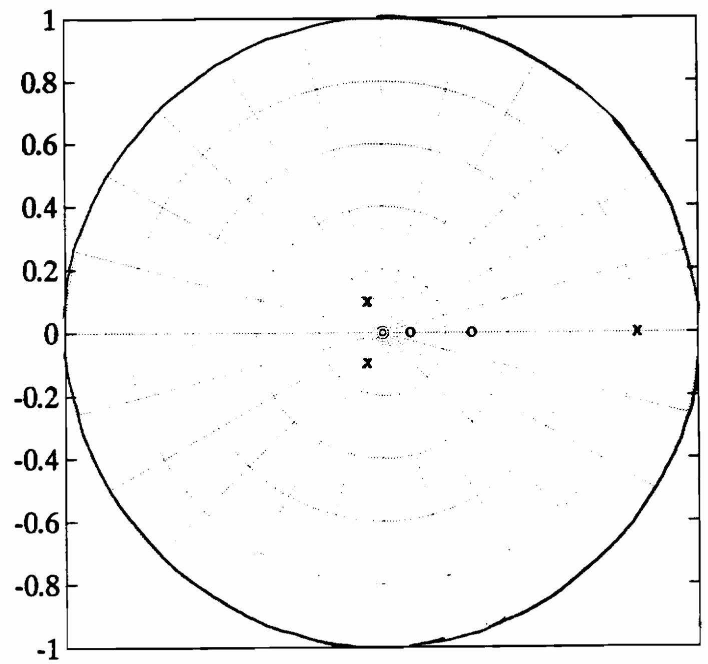


Identified 3rd Order Poles

Figure 3.2.1 Identification of Plant C, full and reduced order



Actual 4th Order System



Identified 3rd Order System

Figure 3.2.2 Identification of Plant D, full and reduced order

3.3 Reduced Order Simulations

Consider first plant E of figure 3.3.1a. In its continuous-time representation, it consists of a cluster of poles and zeros near the origin (maximum real value -0.8), then a cluster of three poles between -4 and -8, and another two at -40 and at -48. An input signal of sufficient amplitude to elicit a recognizable response from the far-away poles will tend to overwhelm the poles near the $j\omega$-axis. Such a system is not a good candidate for accurate identification, but is ideal for order reduction. Far-away poles such as this look to PLID like noise, making a system with multiple far-away poles not so readily reducible as those with only one. Additionally, the "noise" generated by these poles does not appear to be white noise and so is not well understood by PLID, since it violates the assumption of gaussian distribution of noise. This is a problem of unmodelled dynamics. It is likely in these simulations that the "random" sequence generated in the computer was not sufficiently rich to promote convergence, violating the assumption of persistent excitation, but the result is of greater interest. Though the full order model estimated by PLID does not match the parameters used to produce the input/output sequence, its time and frequency responses are very close (figure 3.3.1b). Similar results are found when the order of the estimated model is reduced to one half and to one third. Some iterations of this experiment produced stable representations, but many were unstable. There is a limit to how far a model may be reduced in order, but any reduction is likely to result in considerable savings in computer storage and computation time. Over-estimating the noise covariance, as was done in this case, is a good method of obtaining a closely-matching model. While the disturbance to PLID is not white noise,

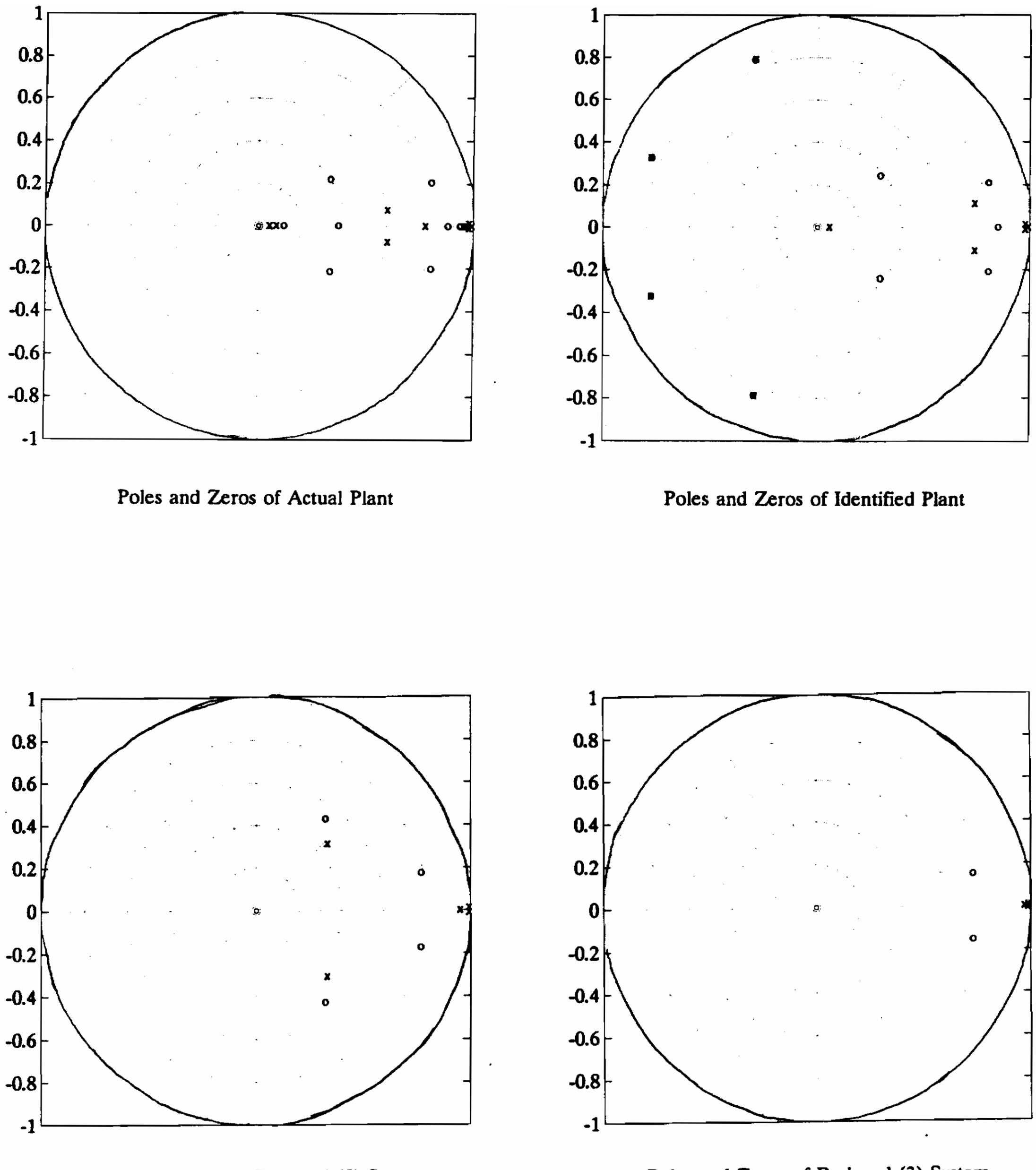


Figure 3.3.1a Plant E pole-zero plots

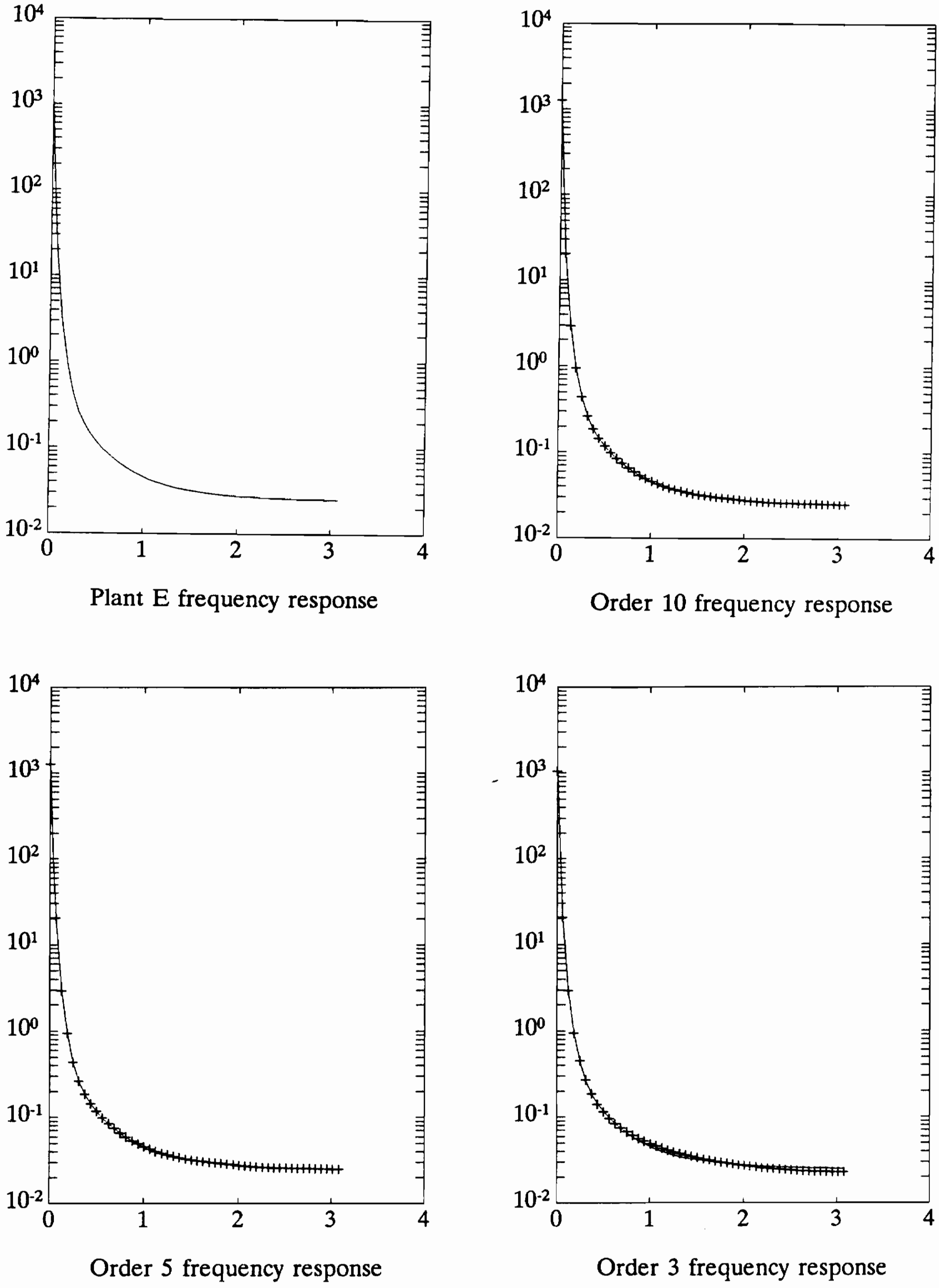


Figure 3.3.1b Plant E frequency responses

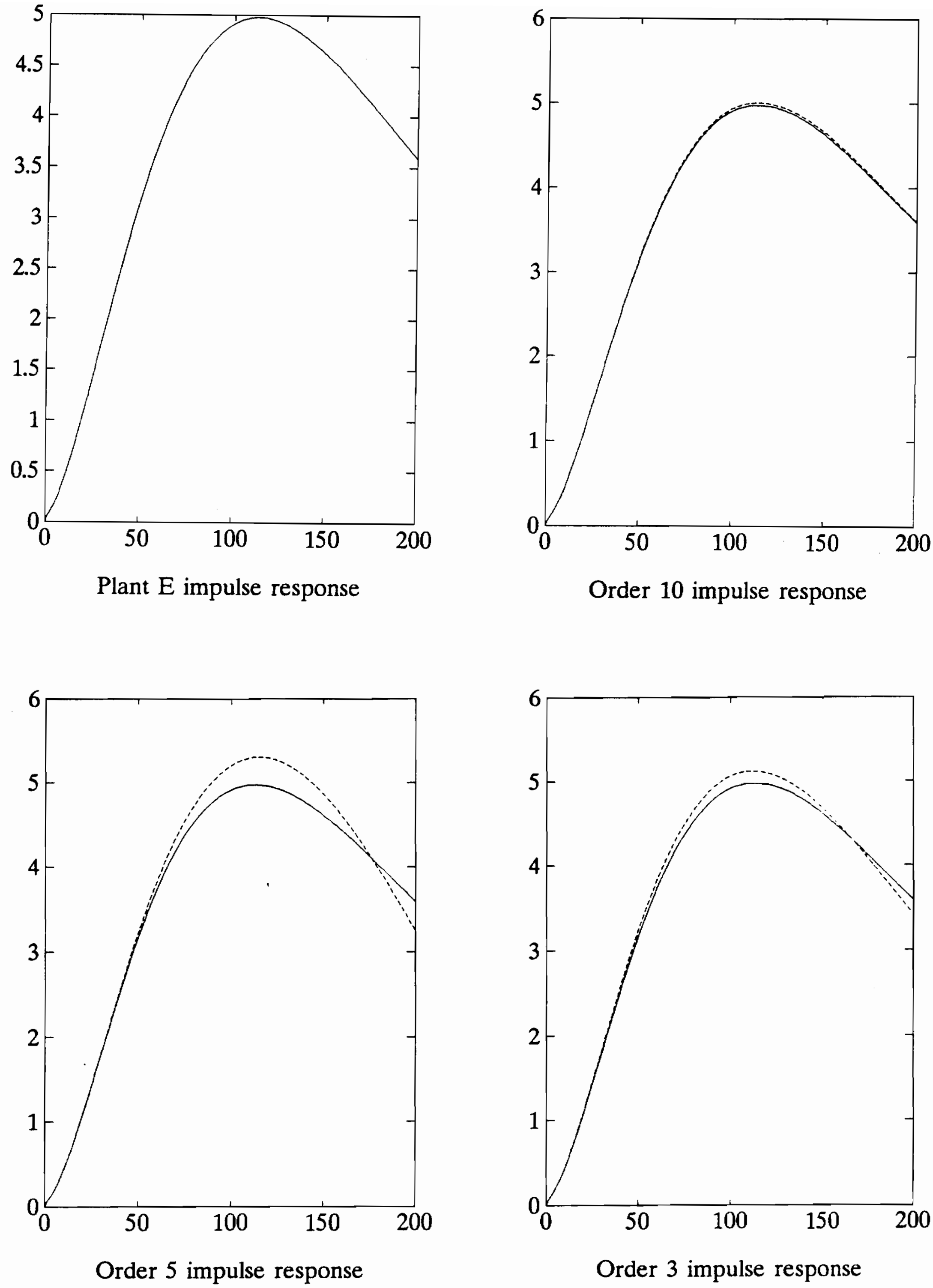
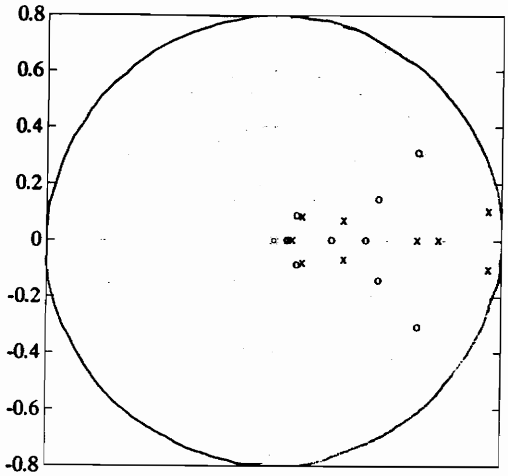


Figure 3.3.1c Plant E time responses

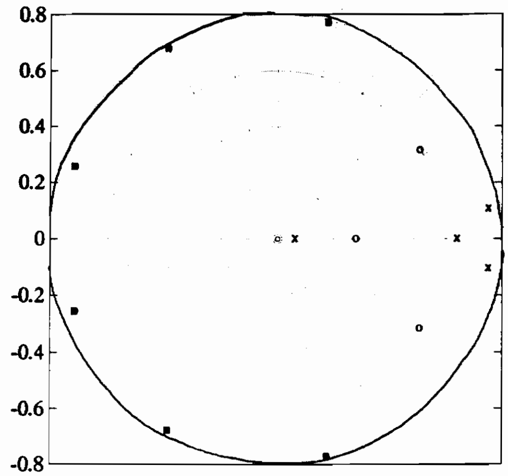
assuming it is and increasing the estimate of its covariance very often gives a servicable model. In fact, it may be noted that PLID produces a full-order estimate with four near pole-zero cancellations, providing a hint that the system may easily be reduced at least to order six. Then the further reduction to fifth-order removes the far-away pole (near the origin in the discrete-time complex frequency plane).

Plant F, in Figure 3.3.2a, displays a number of near pole-zero cancellations, but none as close as in the example of Plant C. As in the preceding case, PLID fails to reproduce the original plant, but instead provides one of nearly identical time and frequency reponse. In this example PLID produces six near pole-zero cancellations in its estimate of the full-order system, indicating that the system may be well-represented by a fourth-order model. When a fifth-order model was specified, PLID produced a model with a single near pole-zero cancellation, effectively a fourth-order model. This example was reduced yet again, to third-order, averaging the real poles to preserve their dynamics.

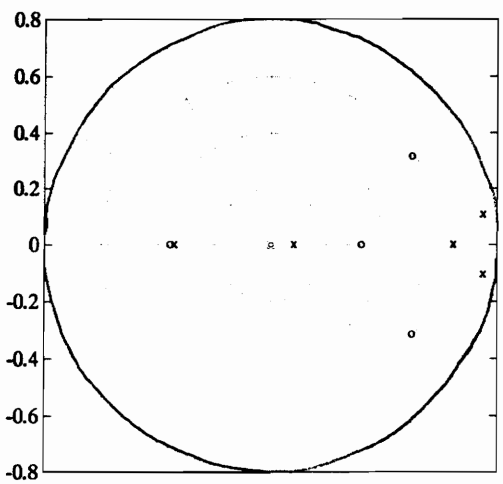
Reduction of Plant G, Figure 3.3.3a, is more of a challenge to the designer. There are three dominant poles (nearest the unit circle) and two dominant zeros. The remaining poles and zeros are clustered near the origin, toward the positive real side. PLID again provides an excellent model with respect to the frequency- and time- domain responses, indicating by means of five near pole-zero cancellations that the model is reducible at least that far. When that is done, an additional pair of near pole-zero cancellations is produced, indicating that reduction to third-order is feasible, leaving the expected dominant poles and zeros and an excellent model.



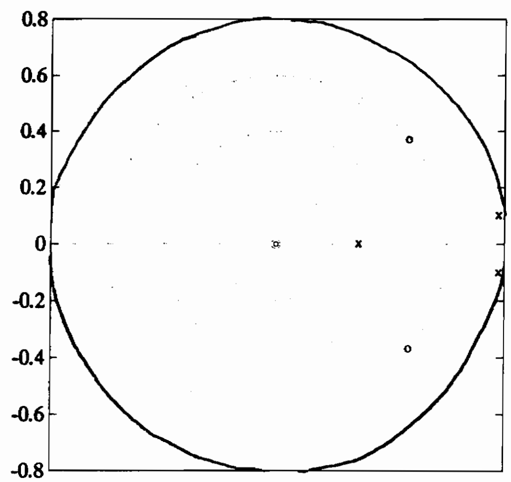
Poles and Zeros of Actual Plant



Poles and Zeros of Identified Plant



Poles and Zeros of Reduced (5) System



Poles and Zeros of Reduced (3) System

Figure 3.3.2a Plant F pole-zero plots

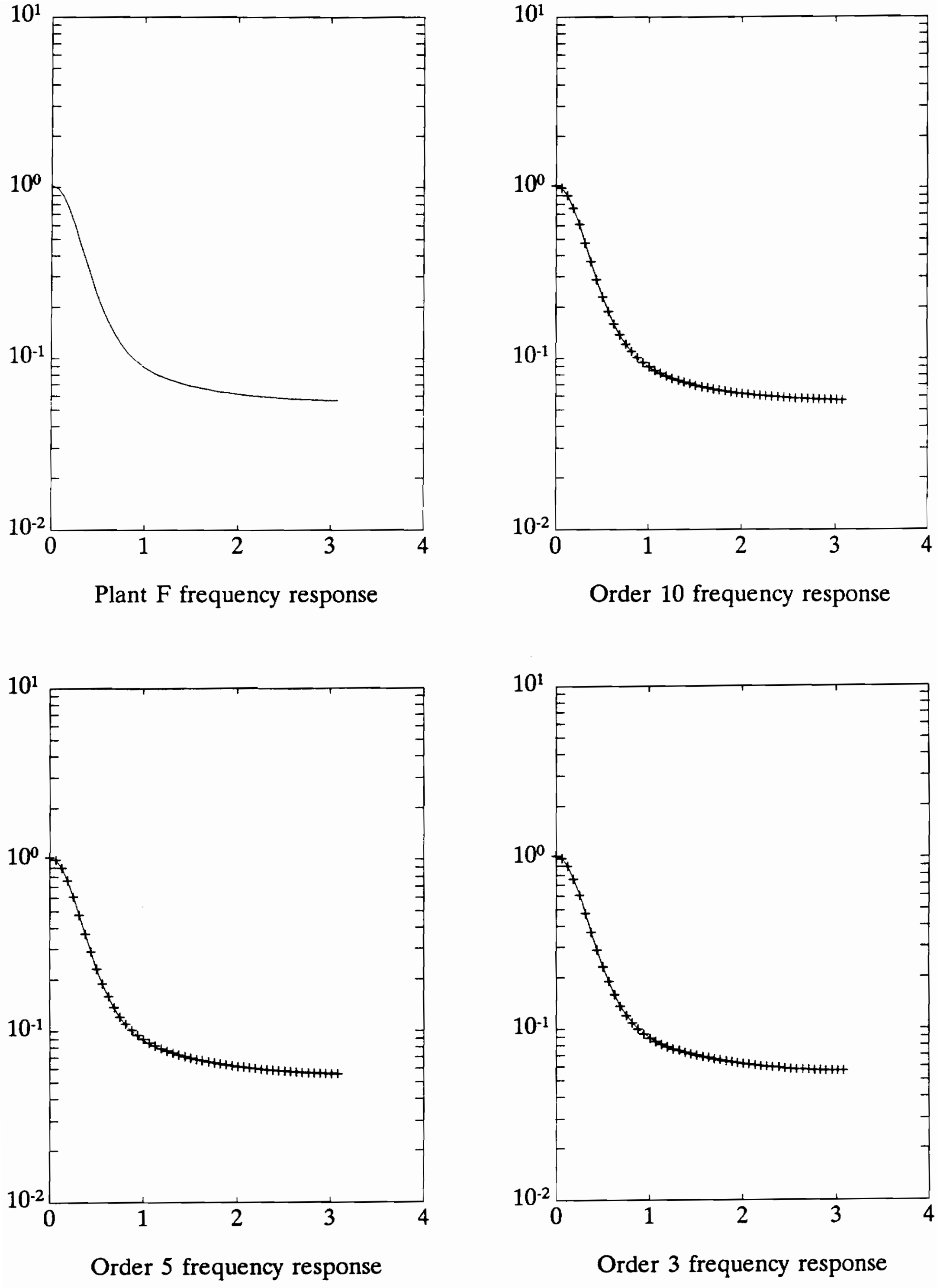
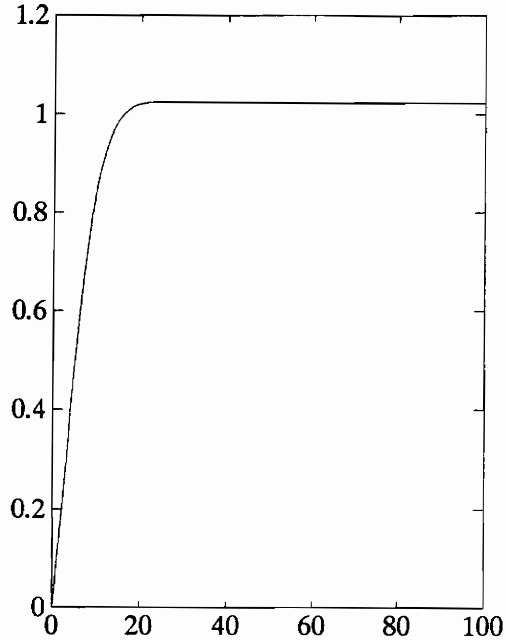
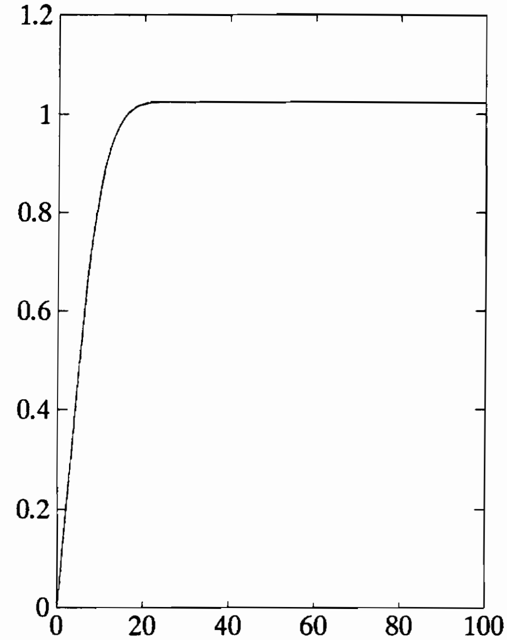


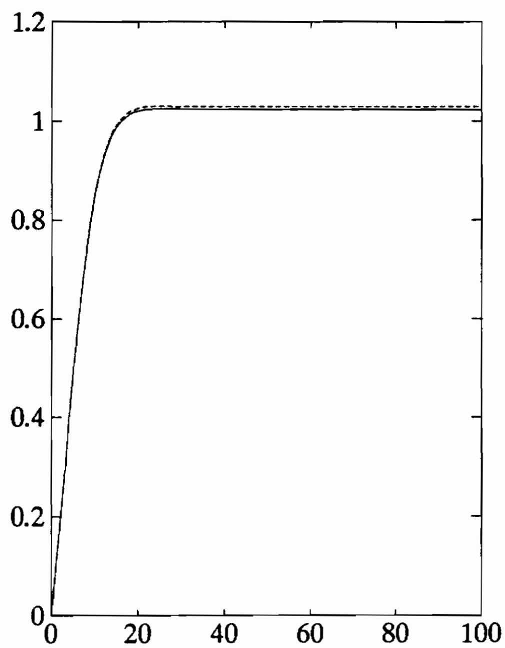
Figure 3.3.2b Plant F frequency responses



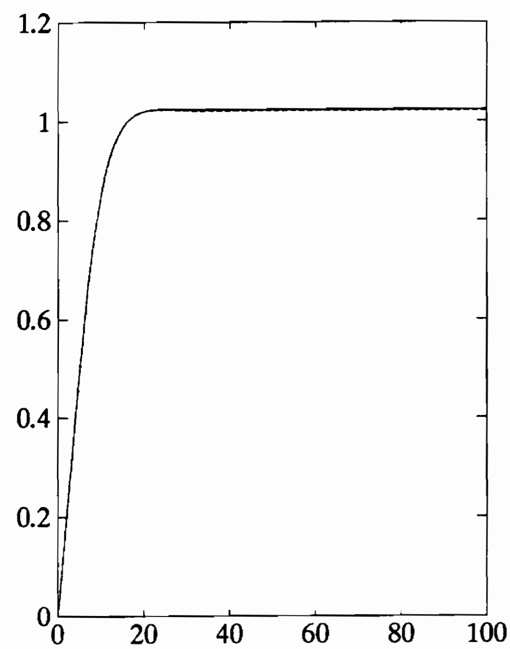
Plant F step response



Order 10 step response



Order 5 step response



Order 3 step response

Figure 3.3.2c Plant F time responses

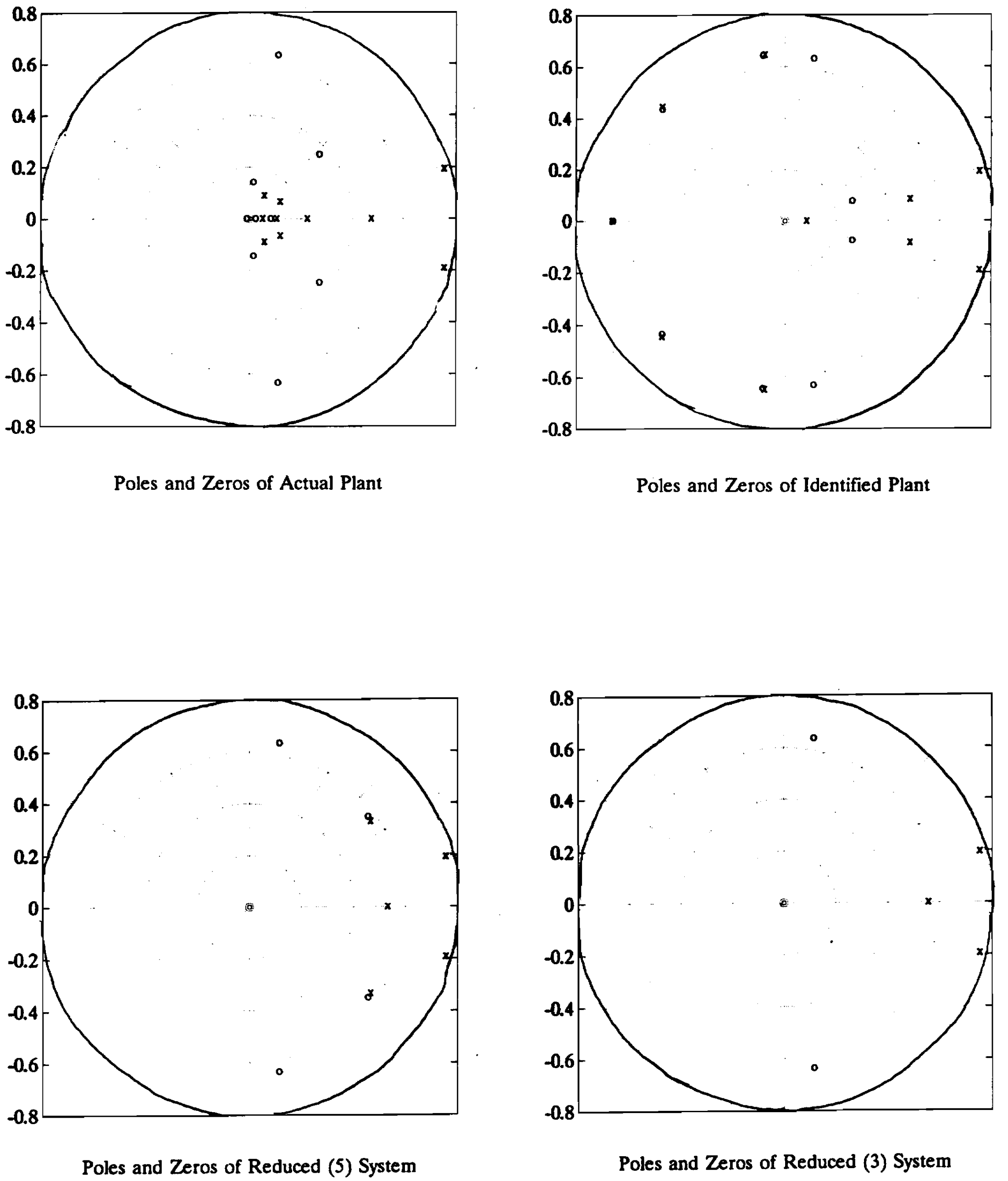


Figure 3.3.3a Plant G pole-zero plots

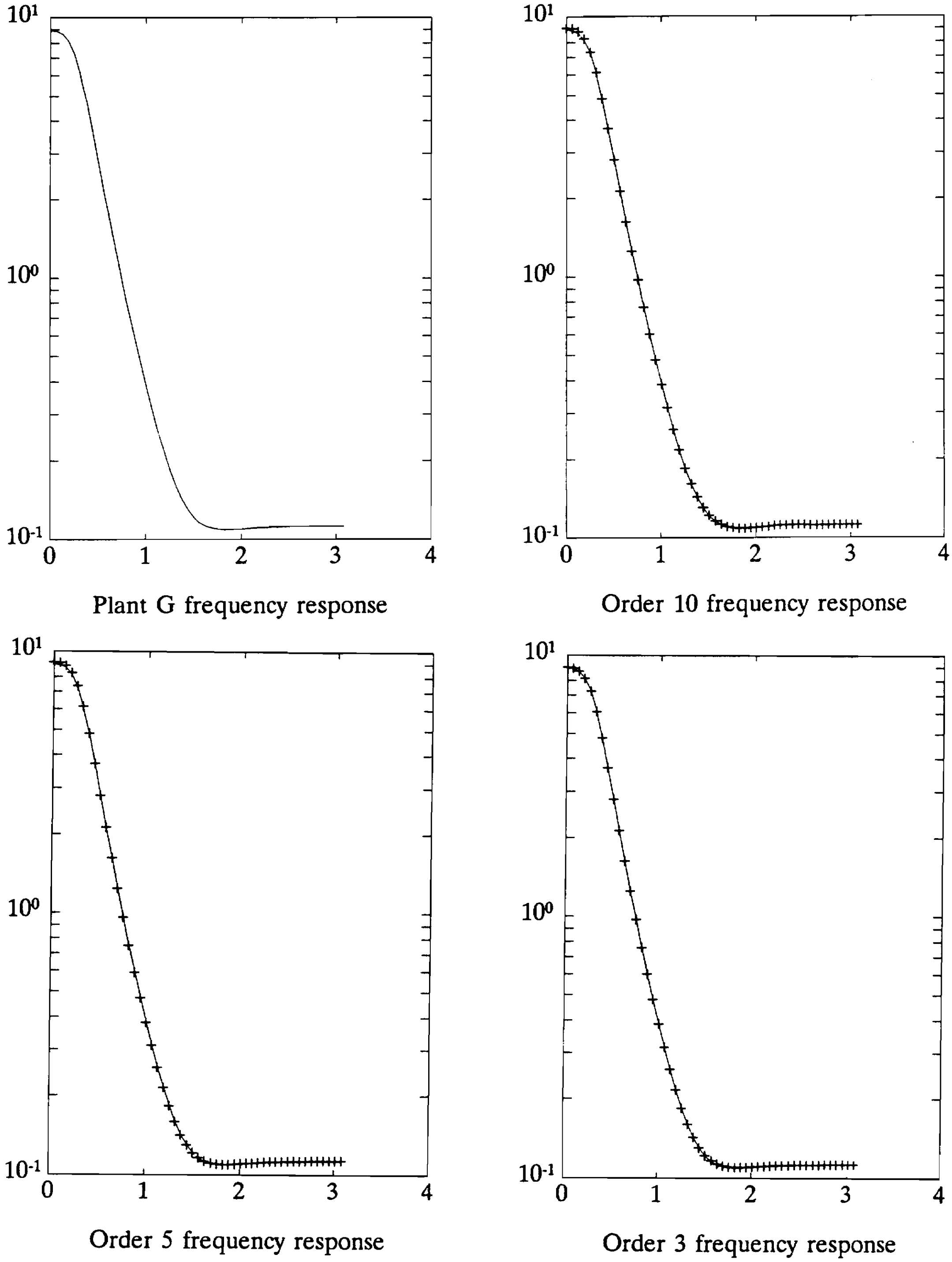
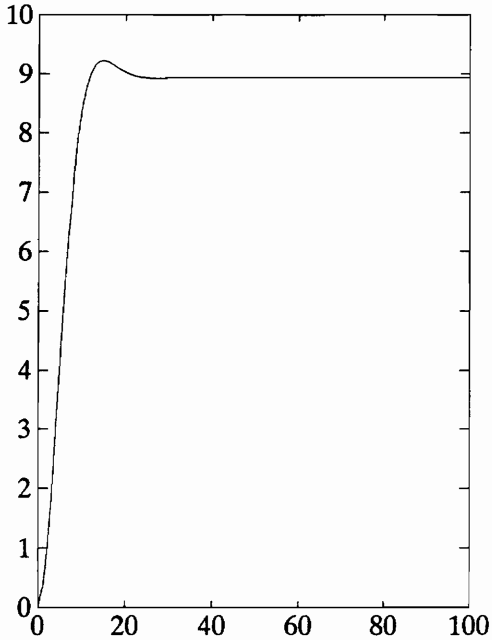
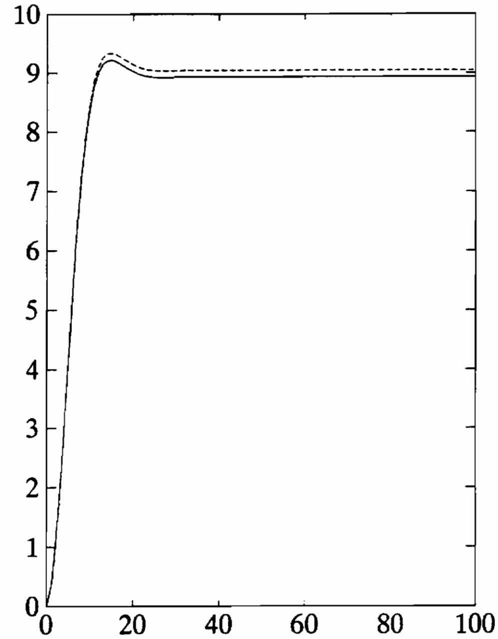


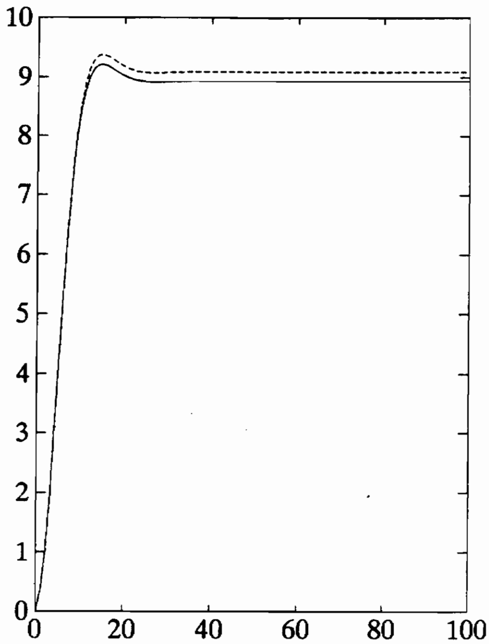
Figure 3.3.3b Plant G frequency responses



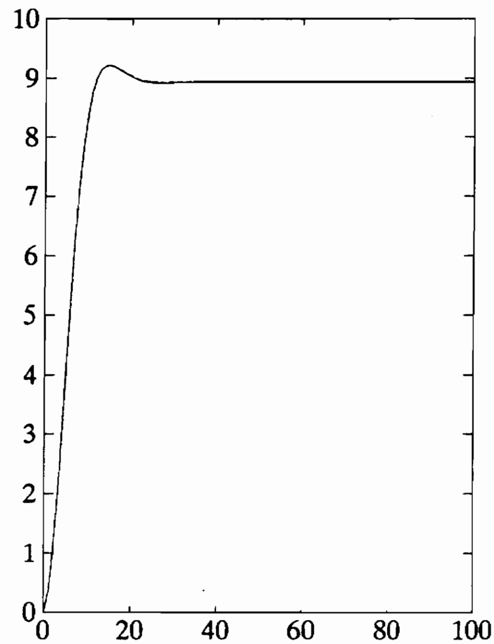
Plant G step response



Order 10 step response



Order 5 step response



Order 3 step response

Figure 3.3.3c Plant G time responses

3.4 Conclusions

We have seen that Pseudo-Linear Identification may be used to identify both full- and reduced-order models of existing plants. Although when system order is large PLID does not exactly reproduce the plant (due to violations of the assumptions under which PLID was derived), the time- and frequency-domain responses of the models produced are in most cases nearly identical. This is all that is required for many design techniques. That models as simple as one-third the actual order of the plant fit as closely as was demonstrated means that complexity in every stage of controller design may be reduced. The reader should beware, though, that no model is likely to be infinitely reducible. Overestimating the noise covariance used as input to PLID may produce a better model, but the time- and frequency-domain responses shoud be checked to insure that the estimated model is close enough for the purpose.

A peculiar feature of PLID is that when attempting to identify a large-order model that is for some reason difficult to identify, it identifies a number of pole-zero cancellations, effectively performing an determination of the model order, as well.

In the following chapters adaptive controllers are implemented. Since these controllers involve computations that grow geometrically in number with system order, any reduction in system order will result in considerable time and storage savings.

4.0 An Adaptive Control Application

Perhaps the simplest control systems are open loop. Launching a projectile toward a target is a readily understood example. Aiming the projectile is the only control involved. After the projectile is on its way, there is no controlling it. If the target moves, or the aim is not true, the projectile misses.

"Closing the loop" is a major improvment. The state of the plant (perhaps the position and velocity of the projectile) can be checked and the control law can adjust the control input accordingly. A rocket is a more advanced example of a projectile. Perhaps a cross-wind pushes the rocket off-course. If the input is the direction of the thrust, the rocket can be steered toward its target.

A similar improvment is made by the introduction of adaptive control. With this method, plant changes can be provided for. Extending the rocket example, suppose the control law for the direction of thrust was designed with the rocket assumed to remain at constant mass. During flight the rocket burns fuel, ejecting the mass of the burned fuel behind it to provide propulsion. Thus the mass changes during the trajectory. This might lead to erroneous control input if, say, the direction of the thrust is in some way proportional to mass. Then the control applied would become too great as mass decreased, leading to oscillations about the path as increasingly greater corrections are made. If the mass can be measured, and the control law designed with mass as a parameter, as the mass changes so can the control law.

Kemp examined the use of PLID in tracking time-varying parameters while adaptively placing the closed-loop system poles. In this chapter PLID will be applied to the plants identified in chapter 3, both full and reduced order estimations, in order to control them with the LQR technique.

4.1 The Self-Tuning Regulator

Among the different algorithms for the implementation of adaptive control schemes is the Self-Tuning Regulator (STR) [11-13]. In this method, a type of controller is decided upon, one that can be designed from knowledge of the plant. An initial guess about the plant parameters is made, and the system is allowed to run. As time goes on and more information about the plant dynamics is available, the estimate of the plant parameters is improved and the controller is adjusted accordingly.

Pseudo-Linear Identification seems custom-made for this type of application. Not only does PLID provide the system parameters with (usually) ever-increasing accuracy, it also provides estimates of the states of the plant. Since many controllers require access to the states for state feedback, PLID is ideal.

For our controller we will use state feedback

$$u^* = -Kx \qquad (4.1.1)$$

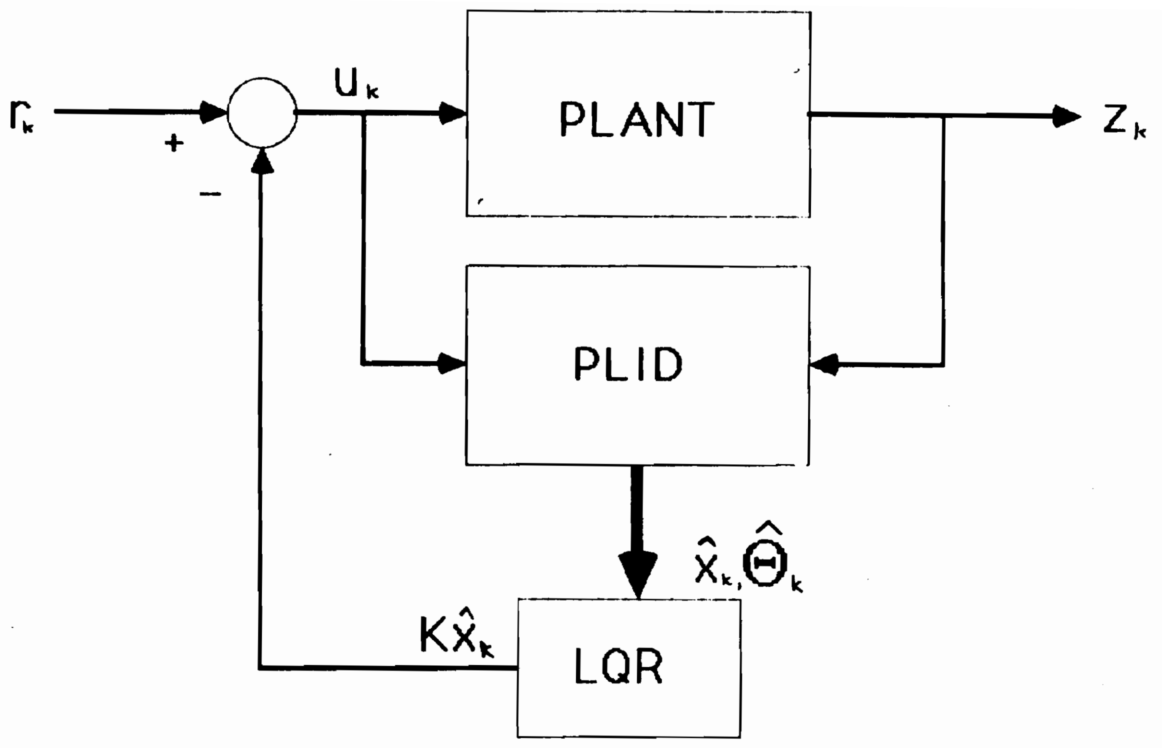


Figure 4.1.1. Self-tuning regulator block diagram.

with the gain matrix K calculated according to the method of the Linear Quadratic Regulator [14].

The gain matrix is calculated to minimize a cost function

$$J(x,u) = \int x^T Qx + u^2 R \ \ dt \tag{4.1.2}$$

where Q is the penalty matrix for the state and R is the penalty on control effort. Since the states are those of observable canonical form and are thus linear combinations of the states that may have been defined by the description of the system dynamics, we will make the substitution of y = Cx to drive the output to zero, rather than the states. This has the additional benefit of reducing Q to a constant, which we will make use of in chapter 5. Thus the final cost fuction is

$$J(x,u) = \int x^T C^T QCx + u^2 R \ \ dt \tag{4.1.3}$$

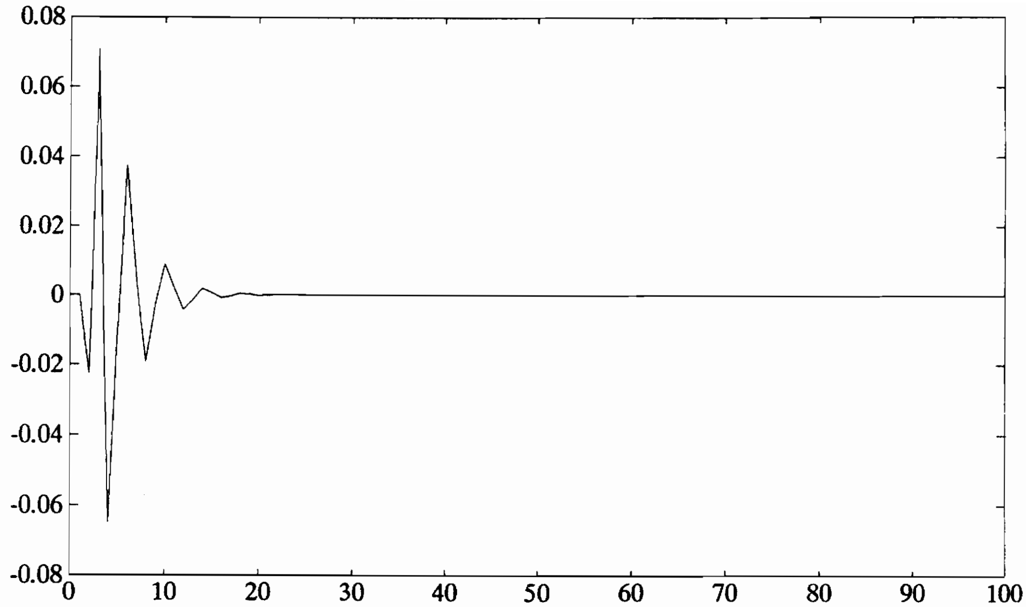
4.2 Simulation Results

Results of simulations are presented in the following figures. Plants E, F, and G are used again. In each case Q = 10 and R = 1 were used so as to place relatively more emphasis on the regulation of the output to zero than conservation of control effort. A simulation of full state feedback is presented as a comparison in each case. Since the algorithm for the calculation of the optimal gain matrix K involves the inversion of a matrix that contains the parameters a_i, the plants were dithered with a random noise input for a short time to obtain an initial guess of the parameters so that they are not all zero. Noises of covariance on the order of 10^{-8} were added to the input and output, but overestimated to be on the order of 10^{-6}.

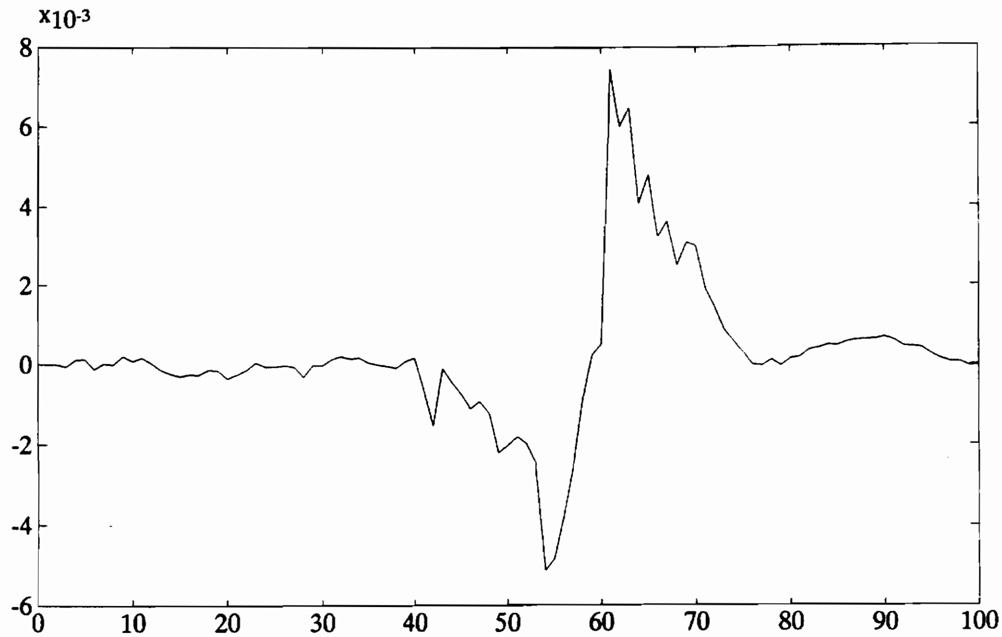
Again plant E, with its far-away poles simulating noise, proves the most difficult to accurately reproduce (figure 4.2.1a). Delayed convergence with large-amplitude "hunting" is displayed in the full-order estimated model's response. This is a result of the parameter estimates not converging. This is not as apparent in the reduced-order cases, due to the characteristics of Plant E discussed earlier. Even though the control is not smooth, note that the output amplitude is much smaller than that of the full state feedback case. This could be further reduced with increased emphasis on output tracking (a larger value of Q), but Q and R were left constant here to show the characteristics of the reduced-order models.

As the model order is reduced, the responses improve, both in amplitude of response and in smoothness (figures 4.2.1b-c). It can be seen in the plots of the time histories of the gains that the parameters are identified more quickly as the order is reduced. Once the parameters converge, the gains become constant.

Plant F displays much cleaner results (figures 4.2.2 a-c), even though it is apparent that the parameters were not identified as quickly in the cases of the tenth and fifth order estimations. This is perhaps to be expected, since having a zero nearly cancel a pole diminishes its contribution to the system response much more than moving a pole toward minus infinity (in the s-plane), making it in effect a smaller-order system to begin with. The controlled output does, however, more nearly match the ideal case of full state feedback with respect to curve shape and amplitude.

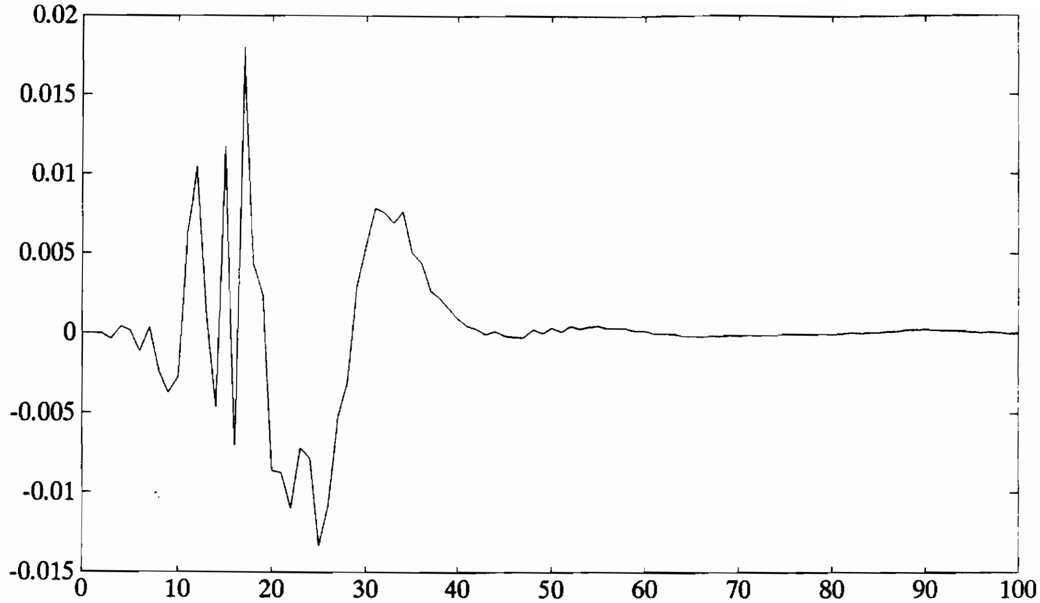


Output vs. time for order 10 plant

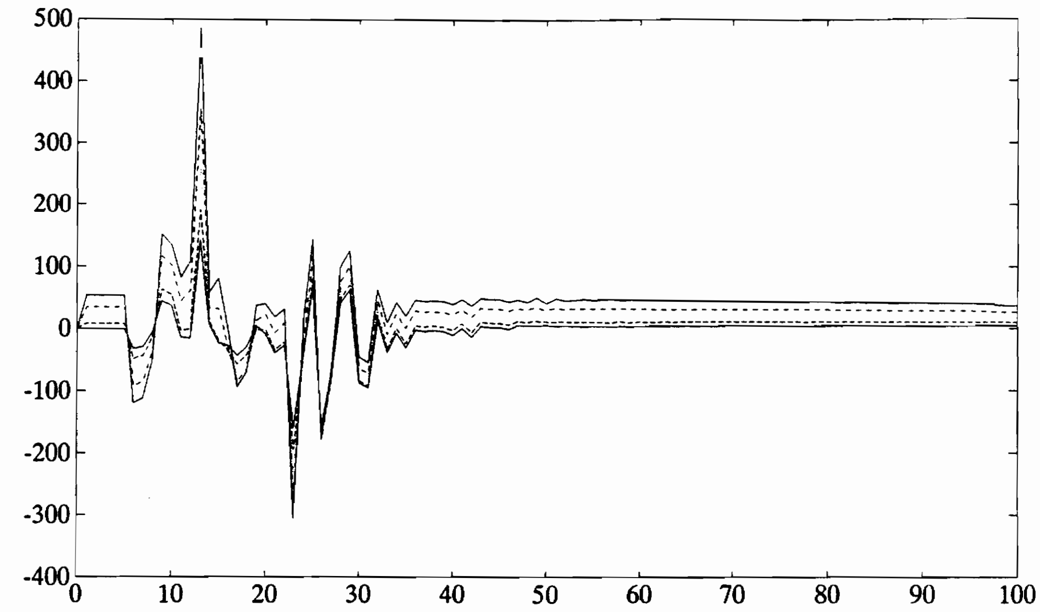


Output vs. time for order 10 estimation

Figure 4.2.1a. Plant E full order full state feedback and full order estimate.

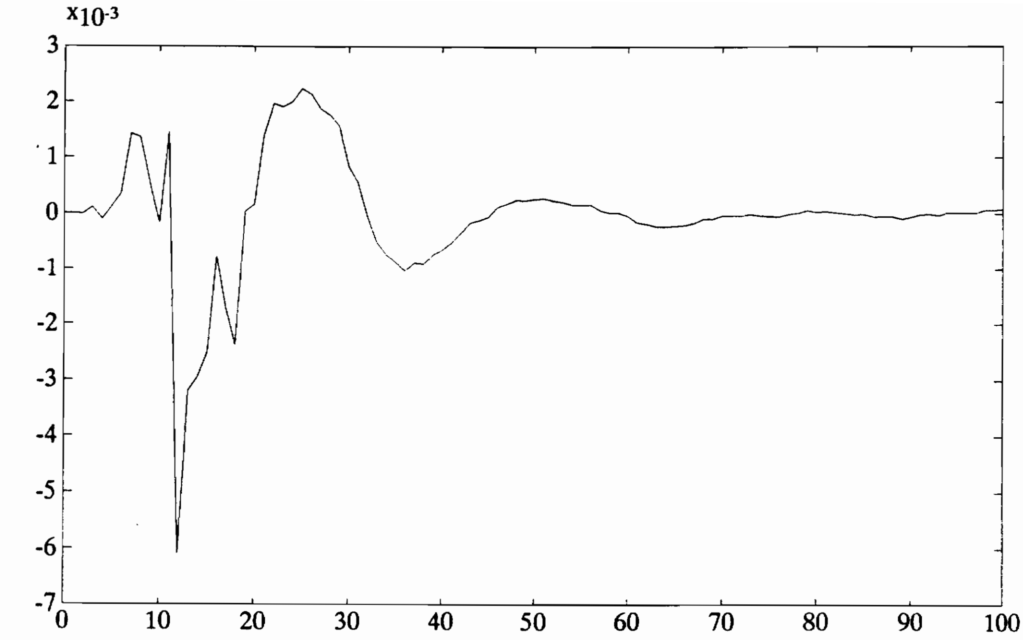


Output vs. time for order 5 estimation

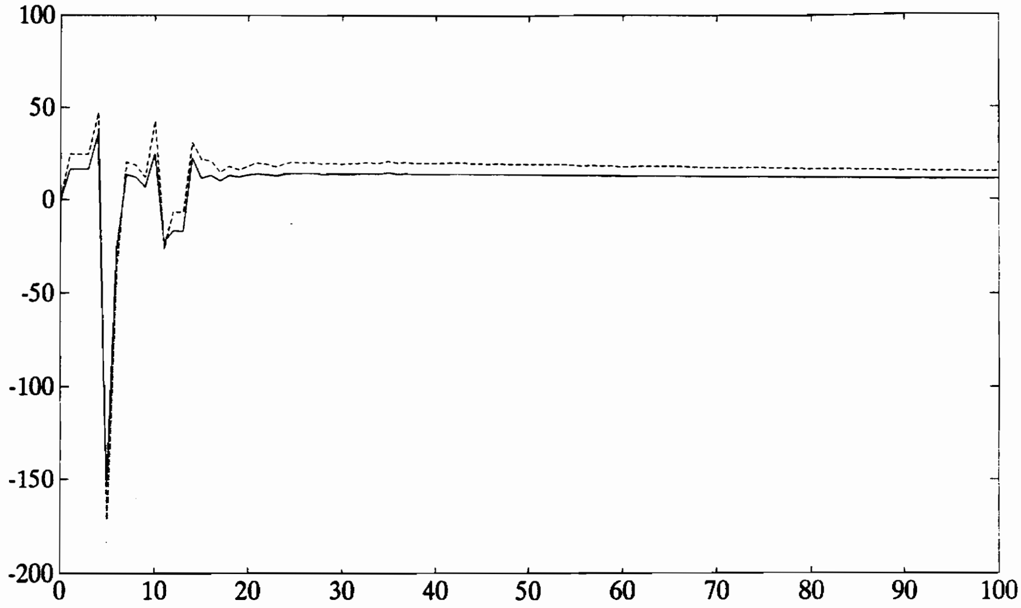


Feedback gains vs. time for order 5 estimation

Figure 4.2.1b. Plant E fifth-order estimate and gain history.

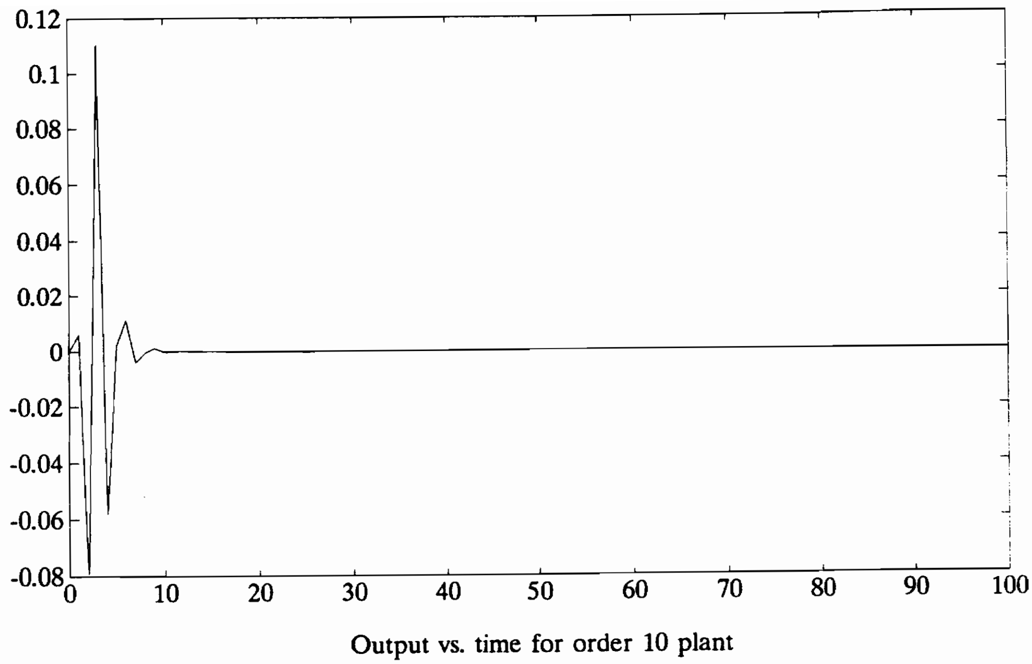


Output vs. time for order 3 estimation



Feedback gains vs. time for order 3 estimation

Figure 4.2.1c. Plant E third-order estimate and gain history.



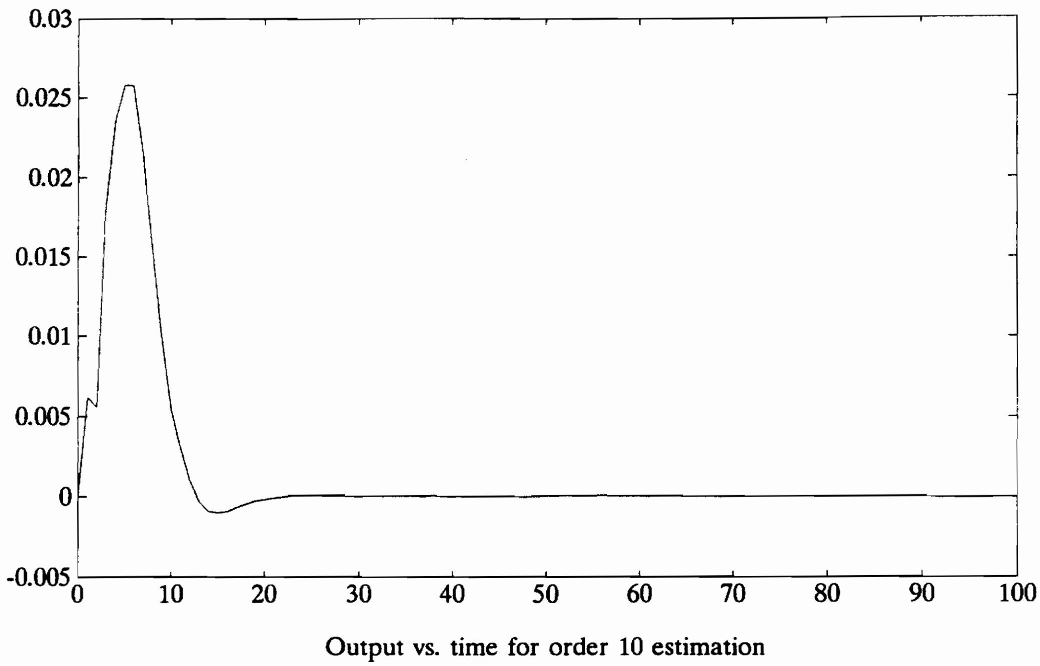
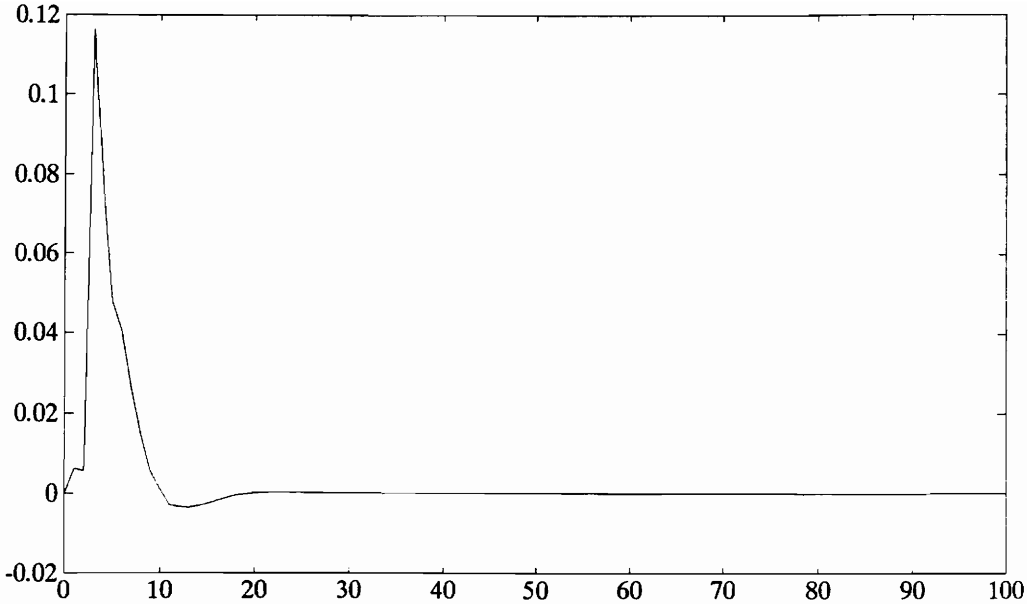
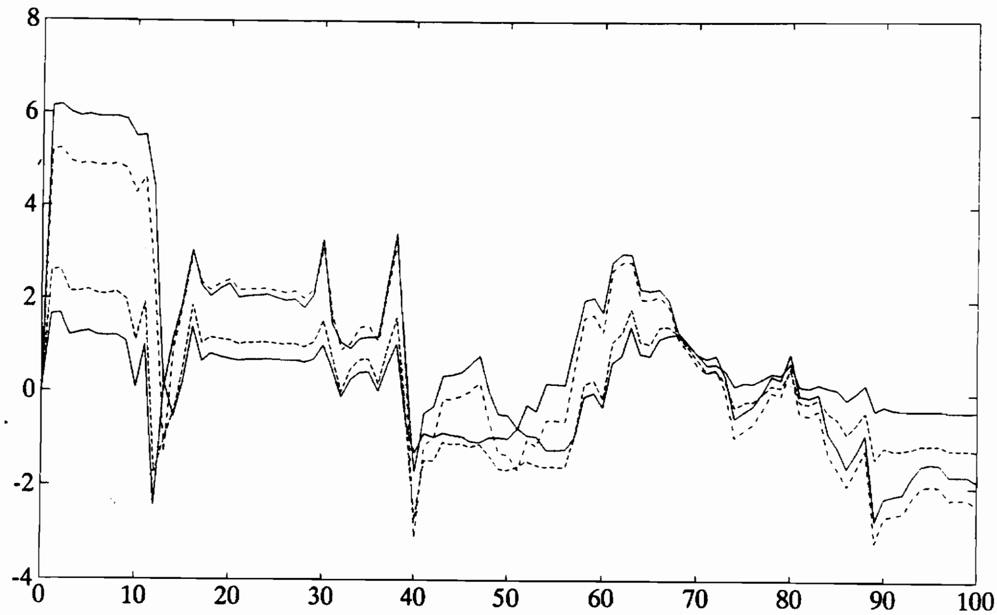


Figure 4.2.2a. Plant F full order full state feedback and full order estimate.

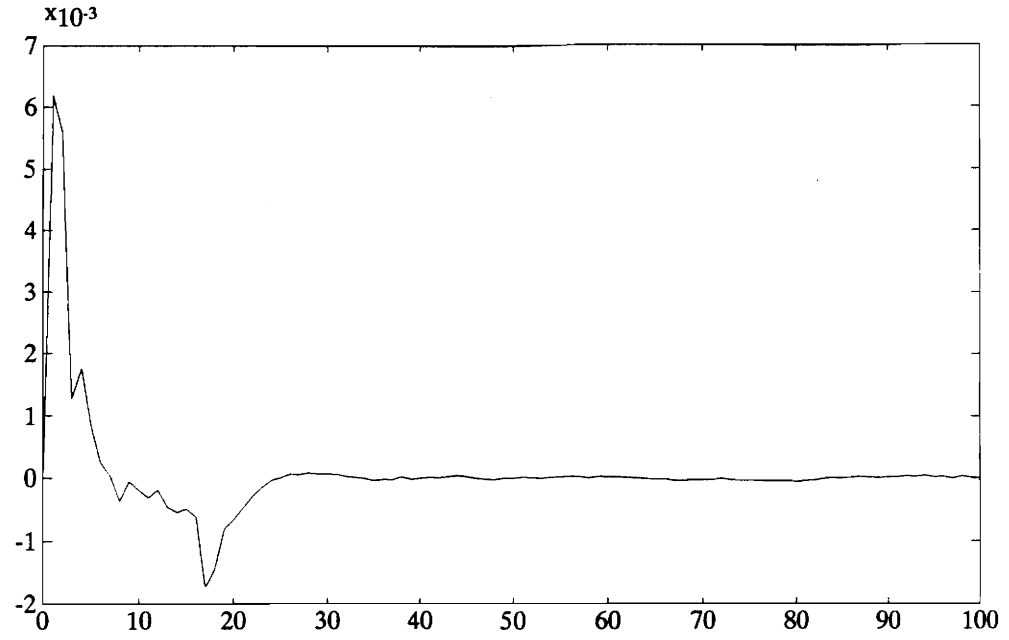


Output vs. time for order 5 estimation

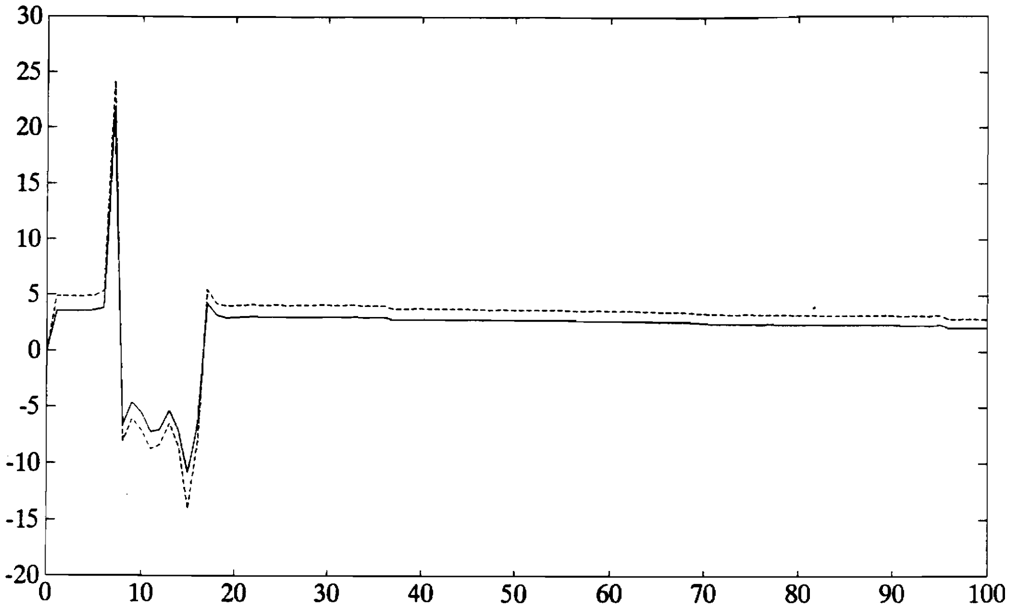


Feedback gains vs. time for order 5 estimation

Figure 4.2.2b. Plant F fifth-order estimate and gain history.

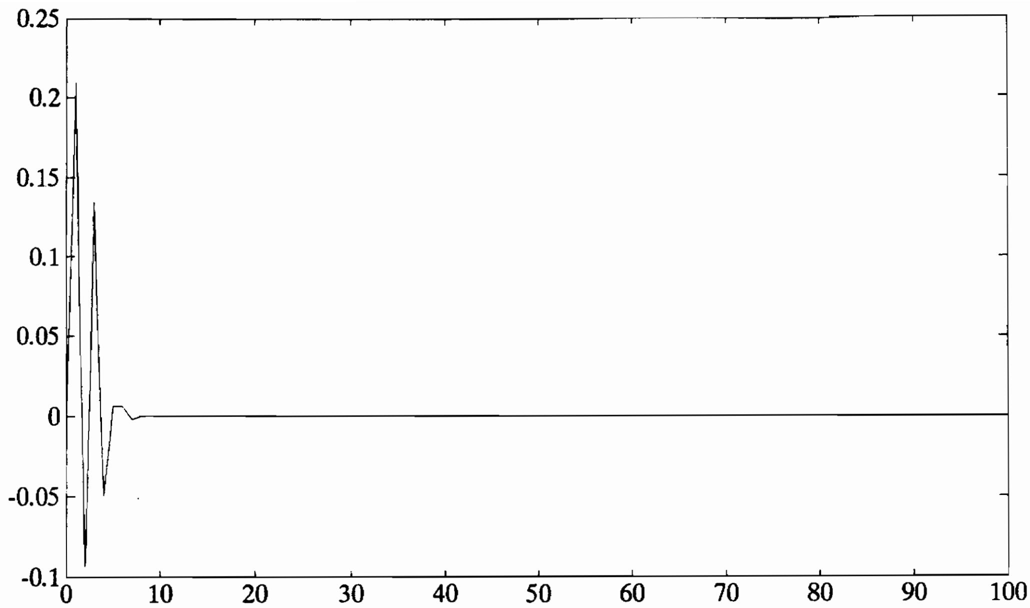


Output vs. time for order 3 estimation

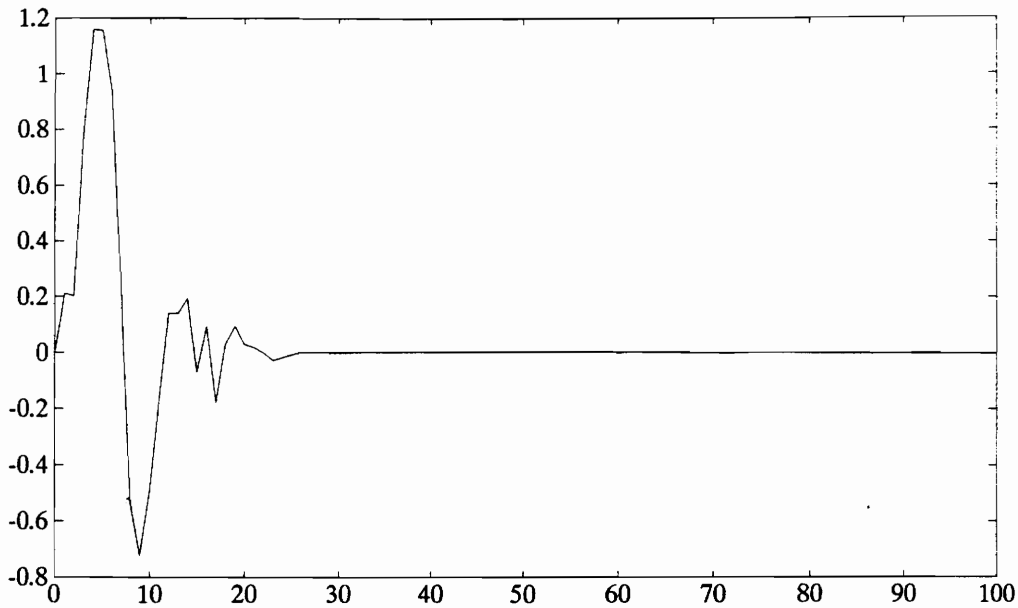


Feedback gains vs. time for order 3 estimation

Figure 4.2.2c. Plant F third-order estimate and gain history.

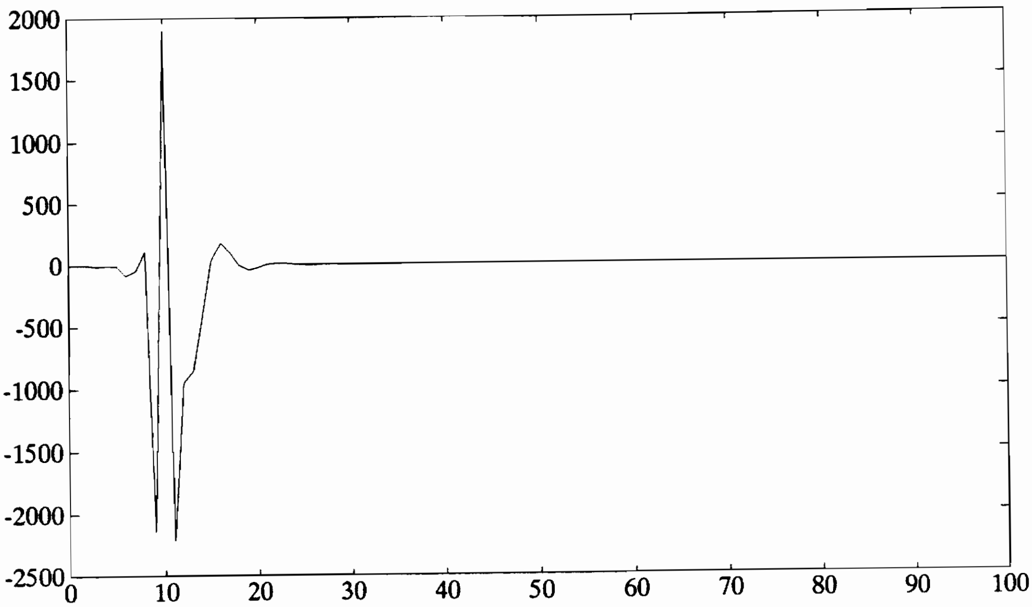


Output vs. time for order 10 plant

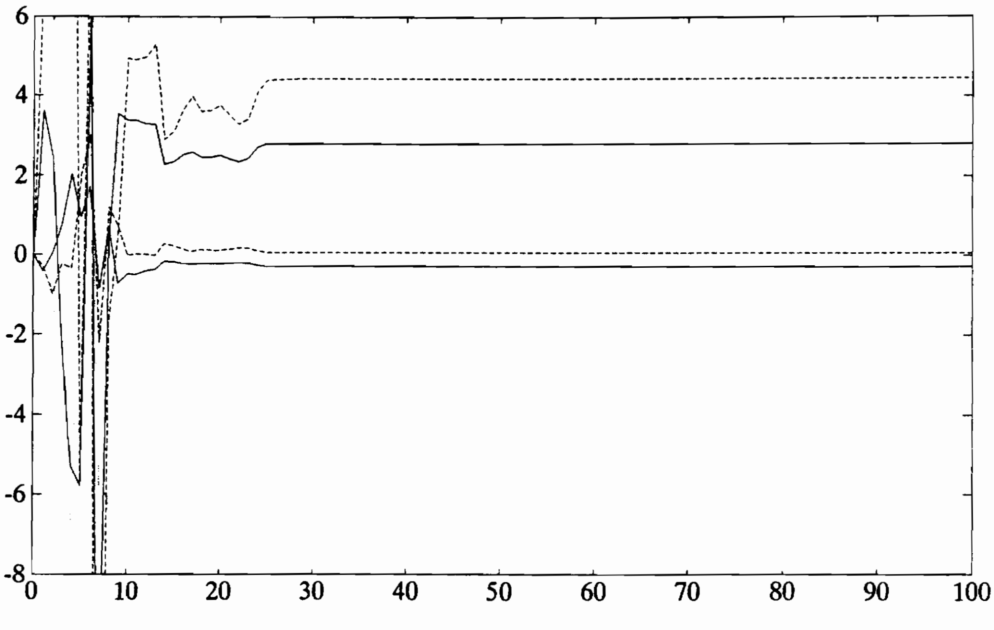


Output vs. time for order 10 estimation

Figure 4.2.3a. Plant G full order full state feedback and full order estimate.

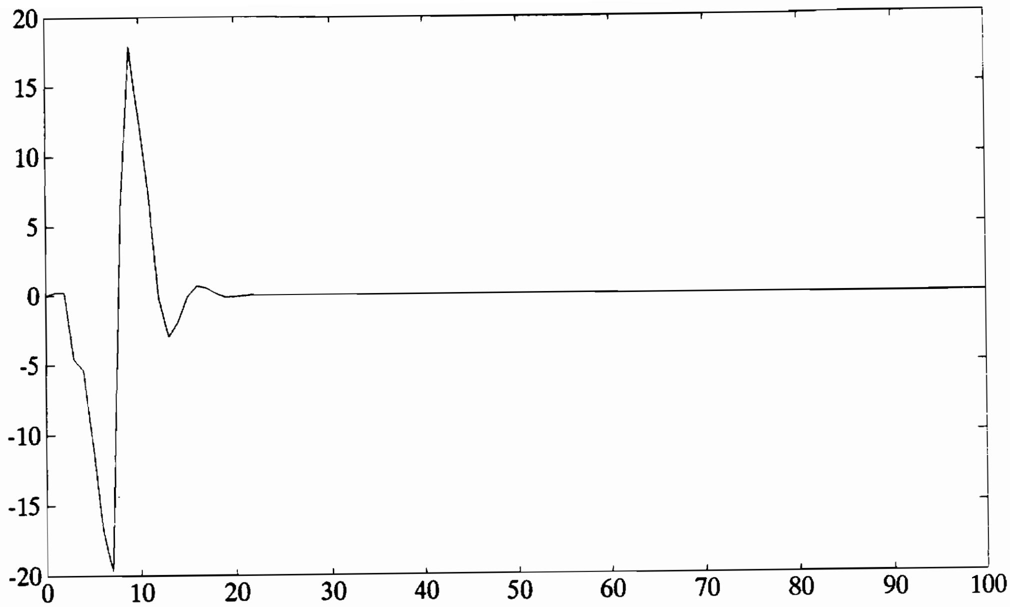


Output vs. time for order 5 estimation

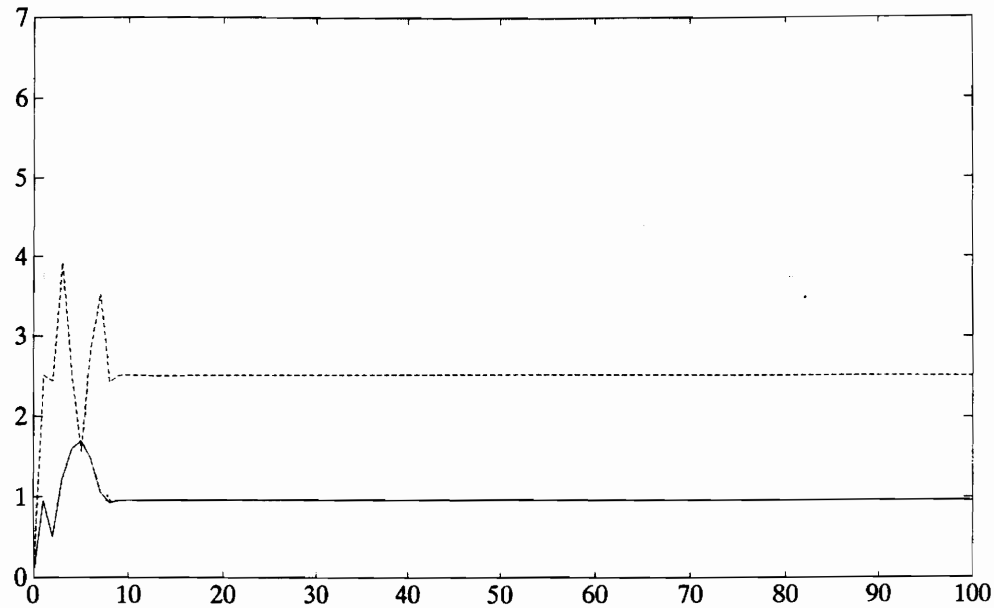


Feedback gains vs. time for order 5 estimation

Figure 4.2.3b. Plant G fifth-order estimate and gain history.



Output vs. time for order 3 estimation



Feedback gains vs. time for order 3 estimation

Figure 4.2.3c. Plant G third-order estimate and gain history.

Plant G is apparently not much more difficult to control (figures 4.2.3 a-c). Its output histories display a much closer resemblence to the classic LQR plot. A system that is harder to intuitively reduce may be easier for PLID to handle, providing a richer set of input and output data from its more obvious dynamics. The plots look like the classic LQR, but with the output regulation delayed while identification is taking place.

4.3 Conclusions

Though the commanded input may not provide a persistently exciting signal to the plant, the adaptive LQR using PLID is able to control the plants fairly well using reduced-order models. Even when the controlled output looked jagged in comparison to the ideal case of full state feedback, the magnitude of the oscillations was not as great. Variation of the state (or output) and control effort penalties Q and R can change the characteristics of the response. When using PLID in a control scheme, it is helpful to provide the algorithm with an initial guess of the parameter values to prevent a possibly large and incorrect control input from being applied.

5.0 A Further Adaptation

It was seen in the last chapter that some characteristics of the reduced-order models were not very similar to those of the ideal case, full state feedback. It was stated that performance could be improved by changing the values of Q and R. In this chapter it will be demonstrated that this is the case.

A reference signal r_k describing a square wave is input to plants E, F, and G, which then are to track this signal in their output. As the parameters converge, Q is increased in relation to R to improve performance by a specified measure.

5.1 Model-Reference Adaptive Control

The self-tuning regulator design presupposes that the desired control signal will be a function of the plant parameters. For this reason the STR is called direct adaptive control. Another approach to adaptive control is called model-reference adaptive control. This method involves matching the response of a plant chosen as ideal for some reason. This ideal plant is run in parallel with the actual plant and the controller is adjusted via a compensator so as to minimize the difference between the output of the actual plant and the output of the ideal

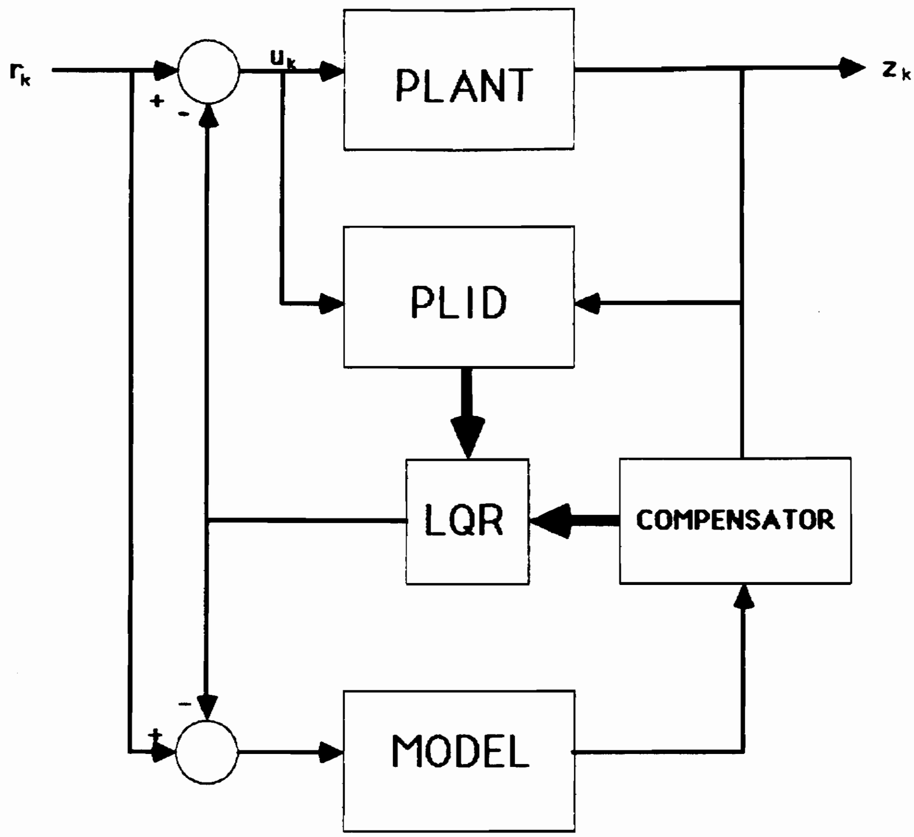


Figure 5.1.1. Model-reference adaptive control block diagram.

plant. Since the plant parameters are not directly used in calculating the controller parameters, this type of adaptive control is called indirect.

As used in this chapter, model-reference adaptive control involves closing a second loop, providing another way for the controller to adapt. For the ideal response, a second-order plant with continuous-time poles at $-3\pm3j$ was chosen for its low overshoot and quick settling time. Much is known about second-order plants, and they are easy to design. If a large-order plant can be made to follow closely the response of such a small plant, high-frequency oscillations would likely be reduced. This is a goal in many designs, since high-frequency responses are often bad for actuators, sensors, and plants.

5.2 Simulation Results

It is desired for the estimated system's response to match as closely as possible an ideal response in some way. To simulate this, a reference signal was applied to each of the three tenth-order plants, which were converted to Type I response [15] by placing a gain in series with the plant to force the overall gain to be unity.

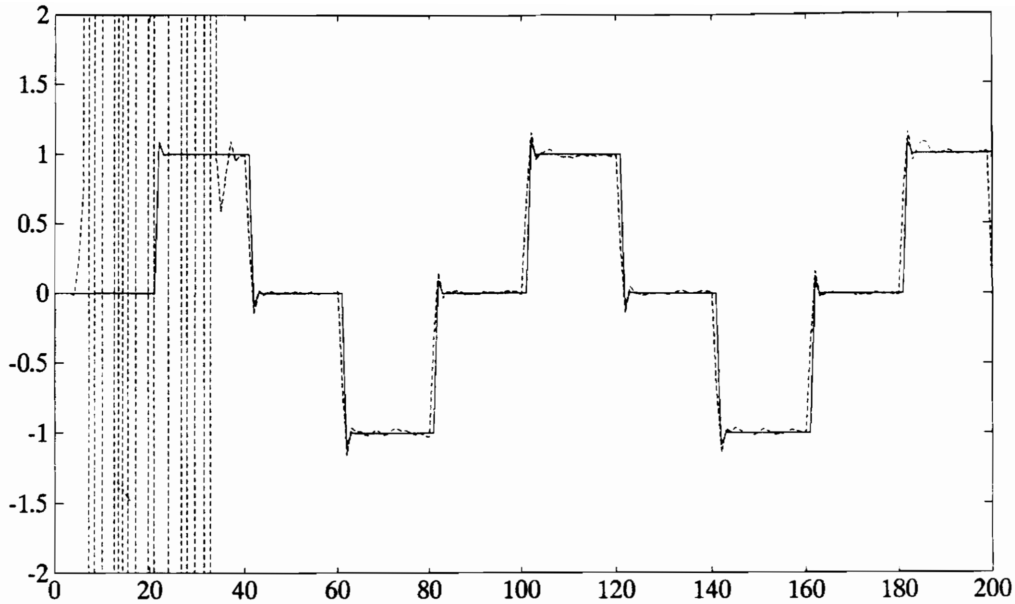
Noise was again added to both the input and output of the plant, and the covariance of the noise overestimated.

The reference signal chosen was a square wave with a period of 80 sampling units, since the commanding input would then be likely to be sharp and so be rich in frequency content. Within each period the signal is zero for 20 units, positive unity for 20 units, zero again for 20 units, and finally negative unity for 20 units.

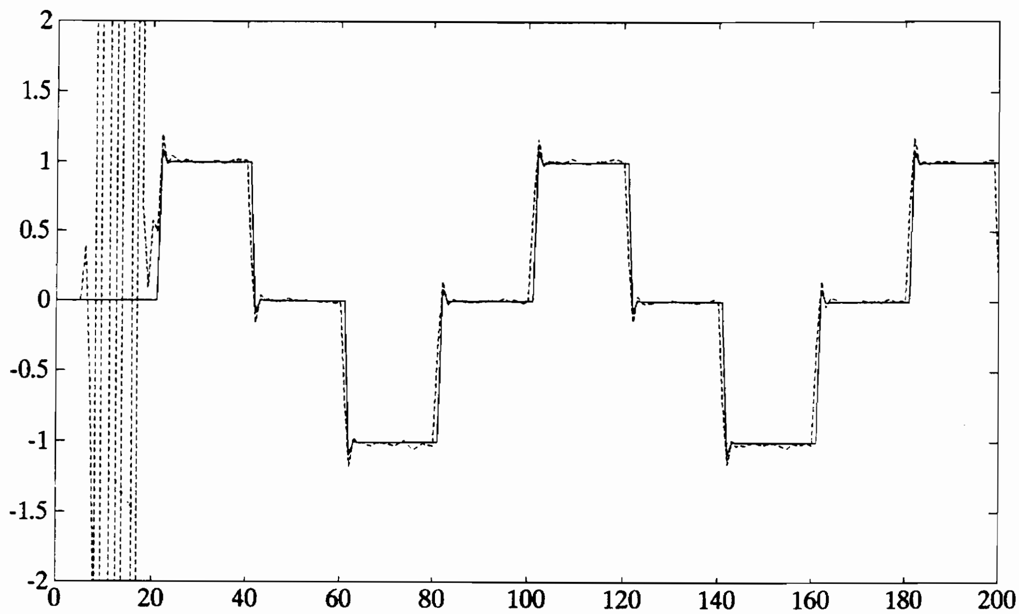
Each plant was reduced in order to one-half and one-third (to fifth and third order, respectively). An attempt to minimize the difference between the ideal response and the response of the estimated systems was made. A minimum-mean-square criterion was chosen for the following plots. If the mean of the squared error for the current 20 iterations is greater than for the previous 20, the value of Q is doubled. If the mean-square error has decreased, then the value of Q is replaced with its square root. To begin, $Q = R = 1$.

As can be seen from the plots, model following is excellent after the transient period when the PLID algorithm is converging on a reduced-order model. The transients are introduced when incorrect control signals are calculated based upon erroneous estimates of the (reduced-order) plant. Since these transients are rather large (several orders of magnitude larger than the signal to be tracked), a form of "block invariance" of the type suggested by Shimkin and Feuer [16] might be introduced in practice. Then the controller parameters, though continually estimated, would only be updated after several periods of calculation. Beginning with an initial estimate of the states would be helpful, also. These simulations were run with a close but incorrect guess of the parameters, as done previously, but with the state portion of the extended state vector set to zero.

Note that the lower the order, the shorter the transient period. After examining the reduction of Plant E to one half and one third of its actual order (figure 5.2.1), and noticing that the difference in responses was nearly negligible, a full-order approximation was run for Plants F and G (figures 5.2.2 and 5.2.3). Other than extending the transient period significantly, no great difference was noted between a tenth-order system and its third-order approximation. The extra dynamics, recall, were estimated by PLID to be near pole-zero cancellations, so this response is as should be expected.

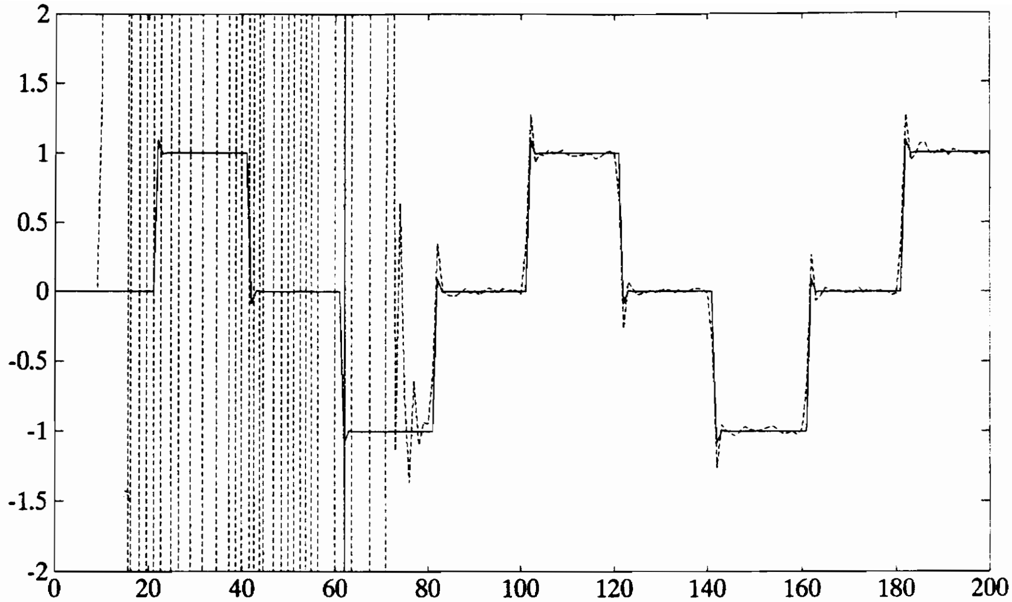


Estimated order 5 system and ideal responses

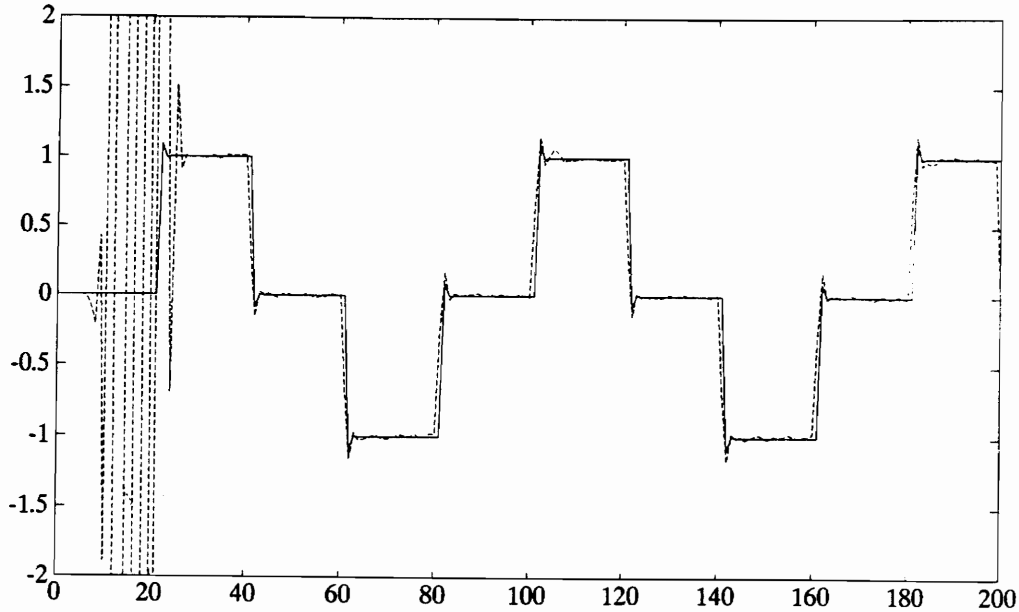


Estimated order 3 system and ideal responses

Figure 5.2.1 Plant E fifth and third-order vs. ideal responses.

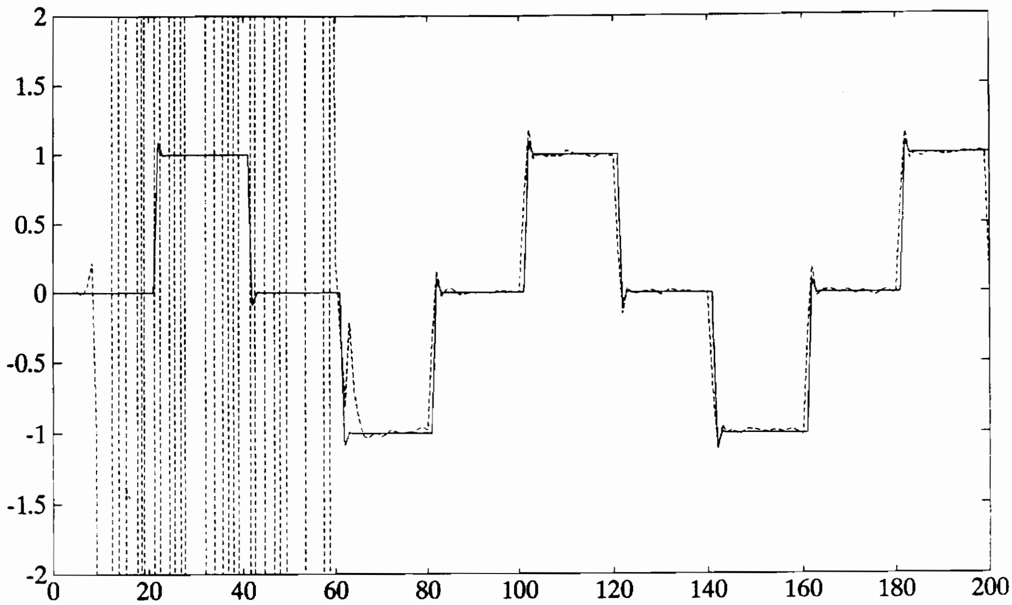


Estimated order 10 system and ideal responses

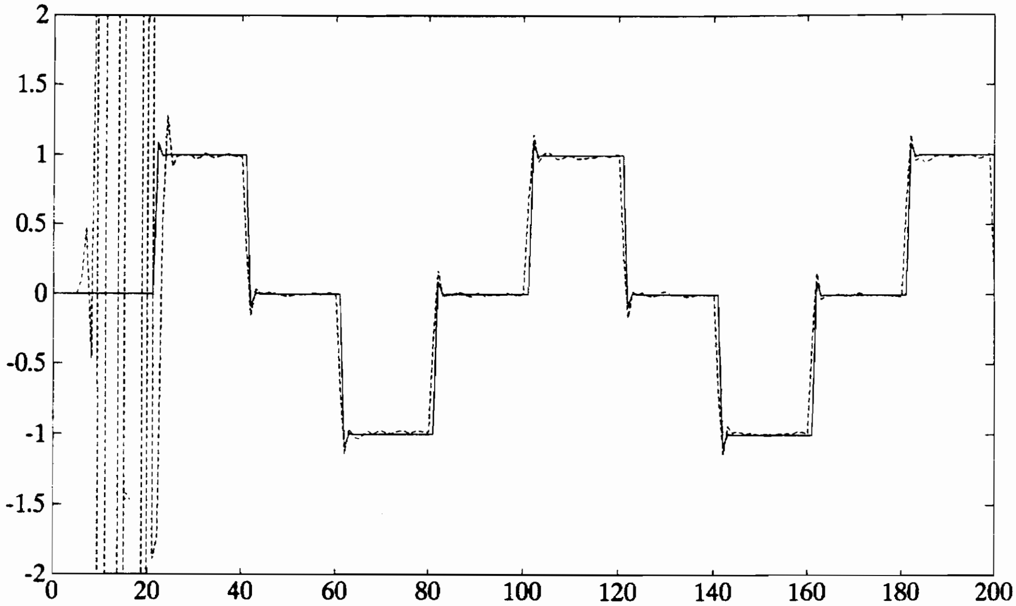


Estimated order 3 and ideal responses

Figure 5.2.2 Plant F full and third-order vs. ideal responses.



Estimated order 10 system and ideal responses



Estimated order 3 system and ideal responses

Figure 5.2.3 Plant G full and third-order vs. ideal responses.

5.3 Conclusions

No significant difference in model following was noted among the different plants used here. This would lead one to believe that such a double adaptive controller might be quite robust. Its chief disadvantage is the large amount of time needed for all the calculations. The lack of difference among the different order approximations of the same plant would also suggest that if a low-order model produces a satisfactory (or perhaps superior) response, then using a higher order would be a waste of resources. This is especially so since the low-order parameters converge much sooner, giving essentially the same performance sooner, due to the far fewer calculations necessary.

6.0 Conclusions

It has been demonstrated that Pseudo-Linear Identificaton may be used in an adaptive control scheme with success. That PLID provides simultaneous parameter and state estimates makes it a good choice for control algorithms requiring state feedback based on knowledge of the plant. Though it is not without its limitations, good results can be had.

Since PLID is rather intolerant of large-amplitude noise, overestimating the noise covariance has been shown to be a valuable tool in achieving convergence, especially when reducing the order of the model. A practical application of this feature would be to increase the estimate of the noise covariances until no improvement is seen in the rate of convergence, or until a suitable model is produced. Future work in this area would include a proof of stability in the case of model reduction, which is really the case of unmodelled parameters.

In this work the adaptation involved was simply in identifying the plant as the controller came online. No attempt was made to change the plant parameters while in operation. In future work, Kemp's tracking PLID algorithm might be implemented, and the parameters varied while model reduction feature is being used. It is expected that the algorithm would be less sensitive to small changes in parameters when used as a model order reducer, since it would in effect be throwing away the higher- frequency dynamics.

The limitations of PLID include the standard system identification limitations. Not every system is a good candidate for reduction, and some models may not be accurate when reduced beyond a certain point. The experimenter should check the reduced-order model against the full-order model with respect to similarity between their time and frequency responses.

Persistent excitation is assumed in the derivation of PLID and is necessary for convergence. This is a particularly disappointing limitation when used in an adaptive control scheme, since good control often means poor excitation. PLID displays the characteristic of holding its estimates when not provided with persistently exciting input, which might be mistaken for convergence. Though this estimate is the best one possible in the least-squares sense, it may not yet be very good.

The large magnitude transients in the output displayed in the plots in chapter five would likely be unwanted in a real system. To prevent this, PLID could be run off-line until convergence is neared, then put into operation in conjunction with the controller.

List of References

1. Sinha, N. K., and Kuszta, B., Modeling and Identification of Dynamic Systems, Van Nostrand Reinhold, 1983

2. Stengel, R. F., Stochastic Optimal Control Theory and Application, John Wiley & Sons, 1986

3. Gelb, A., Applied Optimal Estimation, Cambridge, MA: MIT Press, 1974

4. Hopkins, M. A., and VanLandingham, H. F., "Joint State and Parameter Identification Using Pseudo-Linear Identification", under review for *IEE Proceedings-D: Control Theory and Applications*

5. Hopkins, M. A., Pseudo-Linear Identification: Optimal Joint Parameter and State Estimation of Linear Stochastic MIMO Systems, Ph. D. Dissertation, Virginia Polytechnic Institute and State Univerisity, March 1988

6. Kemp, R. S., Pseudo-Linear Identification and Its Application to Adaptive Control, M. S. Thesis, Virginia Polytechnic Institute and State Univerisity, March 1987

7. Anderson, B. D. O., and Liu, Y., "Controller Reduction: Concepts and Approaches", *I.E.E.E. Transactions on Automatic Control*, Vol. 34, #8, pp. 802-5

8. VanLandingham, H. F., Introduction to Digital Control Systems, New York, NY: Macmillan, 1985

9. Anderson, B. D. O., and Moore, J. B., Optimal Filtering, Englewood Cliffs, NJ: Prentice-Hall, 1979

10. Franklin, G. F., Powell, J. D., and Emami-Naeini, A., Feedback Control of Dynamic Systems, Reading, MA: Addison-Wesley, 1986

11. Sastry, S., and Bodson, M., Adaptive Control: Stability, Convergence, and Robustness, Engelwood Cliffs, NJ: Prentice-Hall, 1989

12. Astrom, K. J., and Wittenmark, B., Adaptive Control, Reading MA: Addison-Wesley, 1989

13. Goodwin, G. C., and Sin, K. S., Adaptive Filtering, Prediction, and Control, Englewood Cliffs, NJ: Prentice-Hall, 1984

14. Kirk, D. E., Optimal Control Theory, Englewood Cliffs, NJ: Prentice-Hall, 1970

15. Van de Vegte, J., Feedback Control Systems, Englewood Cliffs, NJ: Prentice-Hall, 1986

16. Shimkin, N., and Feuer, A., "On the Necessity of 'Block Invariance' for the Convergence of Adaptive Pole-Placement Algorithm with Persistently Exciting Input", *I.E.E.E. Transactions on Automatic Control*, Vol. 33, #8, pp. 775-780

17. MathWorks Inc, The, PC-MATLAB™ User's Guide, South Natick, MA: The MathWorks, Inc, 1989

Appendix

The software package PC-MATLAB^{TM} [17] was used to perform the simuations and to produce the plots in this thesis. The following programs for use with this package were written by the author.

```
function [EG,F,H,J,P,Q] = initiate(order,Q1,R,z)

%   Initialize the matrices used by PLID for the extended state representation

%   J. H. Hutchinson, III
%   16 June 1990

C = [zeros(1,order-1) 1];
J = [zeros(1,order); eye(order-1) zeros(order-1,1)];
Q = zeros(2);
Q(1,1) = R;
Q(2,2) = Q1;
P = 200*eye(3*order);
H = [C zeros(1,2*order)];
F = [J z(2)*eye(order) z(1)*eye(order) ; zeros(2*order,order) eye(2*order)];
EG = zeros(3*order,2);
```

```
function [s,P] = rsplid(s,EG,F,H,J,P,Q,R,z)

%    Pseudo-Linear IDentification method to estimate state and
%    parameters simultaneously.

%    J. H. Hutchinson, III
%    16 June 1990

%    Find order of system

[r,c] = size(s);
order = c/3;

%    Update the matrices with new observations

F(1:order,:) = [J z(1,2)*eye(order) z(1,1)*eye(order)];
M = inv(H*P*H' + R);
error = z(1,2) - H*s';

%    Calculate the new ESR estimate and gain

Kt = F*P*H'*M;
temp = F*s' + Kt*error;

%    Separate the state and parameter vectors

ss = temp(1:order,1);
sp = temp(order+1:3*order,1);

%    Update E[G] with the new estimates

EG(1:order,1) = -sp(1:order,1);
EG(1:order,2) = sp(order+1:2*order,1);
```

```
EGs = EG(1:order,:);

%   Correct the state vector and state portion of the gain

Ks = Kt(1:order,1) + EGs*Q(:,1)*M;
ss = ss + EGs*Q(:,1)*M*error;

K = [Ks' Kt(order+1:3*order,1)']';

%   Update the covariance matrix

P = F*P*F' + EG*Q*EG' - K*(H*P*H' + R)*K';
Pa = P(order+1:2*order,order+1:2*order);
Pb = P(2*order+1:3*order,2*order+1:3*order);
P(1:order,1:order) = P(1:order,1:order) + Q(1,1)*Pa;
P(1:order,1:order) = P(1:order,1:order) + Q(2,2)*Pb;

%   Rebuild the ESR vector

s = [ss' sp'];


function [num,den] = recover(s)

%   Recovers the numerator and denominator coefficients from the ESR
%   trajecotries

%   J. H. Hutchinson, III
%   16 June 1990

[r,c] = size(s);
order = c/3;
```

```
%   Initialize the vectors

num = zeros(1,order+1);
den = zeros(1,order+1);
den(1,1) = 1;

%   Average the last ten numerator parameter estimates

for i = 1:order
   for j = 1:10
      num(i+1) = num(i+1) + s(r+1-j,c+1-i);
   end
   num(i+1) = num(i+1) / 10;
end

%   Average the last ten denominator estimates

for i = 1:order
   for j = 1:10
      den(i+1) = den(i+1) - s(r,c+1-order-i);
   end
   den(i+1) = den(i+1) / 10;
end


function [num,den] = convert(z,p)

%   Convert vectors containing s-plane pole and zero locations into z-plane
%   numerator and denominator coefficient vectors

%   Calculate s-plane numerator and denominator coefficient vectors

num = poly(z);
```

```
den = poly(p);

%   Make the vectors the same size by adding zeros to the numerator vector

[r,c] = size(den);
[rr,cc] = size(num);

if cc < c,
  num = [zeros(1,c-cc) num];
end

%   Calculate the state-space representation

[A,B,C,D] = tf2ss(num,den);

%   Calculate the sampling period (just above the Nyquist frequency)

T = (pi / (max(abs(eig(A))))) / 1.05;

%   Calculate the discrete-time state-space representation

[Ad,Bd] = c2d(A,B,T);

%   Extract the transfer function for the z-plane coefficient vectors

[num,den] = ss2tf(Ad,Bd,C,D,1);



function [s,u,y] = irun(order,s0,num,den,Q1,R,steps,mag)

%   Apply input to the system and record the output and ESR trajectories
```

```
%   J. H. Hutchinson, III
%   16 June 1990

%   Calculate the state-space representation

[A,B,C,D] = tf2ss(num,den);
[n,m] = size(B);

%   Build a dither signal of normally-distributed random noise

rand('normal');
u = zeros(1,steps+1) + rand(1,steps+1)*sqrt(mag);
u(1) = u(1) + 1;

%   Initialize the input and output vectors

x = zeros(n,steps+1);
y = zeros(1,steps+1);
s = zeros(steps+1,3*order);
s(1,:) = s0;

%   Initialize the PLID recursion matrices

[EG,F,H,J,P,Q] = initiate(order,Q1,R,[u(1) y(1)]);

%   Build noise vectors for input and output

noiseu = rand(1,steps+1) * sqrt(Q1);
noisey = rand(1,steps+1) * sqrt(R);

%   Run the system for the specified number of samples
```

```
for i = 1:steps
   x(:,i+1) = A*x(:,i) + B*(u(i)+noiseu(i));
   y(i+1) = C*x(:,i);
   z = [u(i) y(i+1)+noisey(i+1)];
   [s(i+1,:),P] = rsplid(s(i,:),EG,F,H,J,P,Q,R,z);
end


function [K,s,u,y] = lqadapt(order,s0,num,den,SP,CE,Q1,R,steps)

%   Simulate adaptive LQ control of an estimated model of a given plant

%   J. H. Hutchinson, III
%   16 June 1990

%   Calculate the state-space representation of the given plant

[A,B,C,D] = tf2ss(num,den);
[n,m] = size(B);

%   Initialize input, output, gain, and state vectors

u = zeros(1,steps+1);
y = u;
x = ones(n,steps+1);
s = zeros(steps+1,3*order);
s(1,:) = s0;
K = zeros(steps+1,order);

%   Initialize the estimated state-space model

Aest = [zeros(1,order); eye(order-1) zeros(order-1,1)];
Best = zeros(order,1);
```

```
Cest = [zeros(1,order-1) 1];

%   Initialize the PLID recursion matrices

[EG,F,H,J,P,Q] = initiate(order,Q1,R,[u(1) y(1)]);

%   Build the noise vectors

rand('normal');
noiseu = rand(1,steps+1) * sqrt(Q1);
noisey = rand(1,steps+1) * sqrt(R);

%   Run the system for the specified number of samples

for i = 1:steps
  x(:,i+1) = A*x(:,i) + B*(u(i)+noiseu(i));
  y(i+1) = C*x(:,i);
  z = [u(i) y(i+1)+noisey(i)];
  [s(i+1,:),P] = rsplid(s(i,:),EG,F,H,J,P,Q,R,z);
  Aest(:,order) = s(i+1,order+1:2*order)';
  Best = s(i+1,2*order+1:3*order)';
  K(i+1,:) = dlqr(Aest,Best,Cest'*SP*Cest,CE);
  u(i+1) = -K(i+1,:)*s(i+1,1:order)';
end


function [u,y,uest,yest] = adatrak(order,s0,num,den,nd,dd,Q1,R,steps)

%   Simulate the operation of a model-reference adaptive controller outside a self
%   tuning regulator

%   J. H. Hutchinson, III
%   16 June 1990
```

```
%   Calculate the state-space representation of the specified plant

[A,B,C,D] = tf2ss(nd,dd);
[n,m] = size(B);

%   Calculate the gain necessary for a type I system

for i = 1:n
  if nd(n+1-i) ~= 0,
    gain = nd(n+1-i);
  end
end

%   Initialize the input and output vectors

u = zeros(1,steps+1);
y = u;
uest = u;
yest = y;
ref = u;

%   Build  the noise vectors

rand('normal');
noiseu = rand(1,steps+1) * sqrt(Q1);
noisey = rand(1,steps+1) * sqrt(R);

%   Build the square-wave reference signal

level = -1;
for i = 1:steps
  if rem(i,20) == 1,
    if level == 0,
```

```
      level = -lastlev;
    else
      lastlev = level;
      level = 0;
    end
  end
  ref(i) = ref(i) + level;
end

%   Initialize the PLID recursion matrices

[EG,F,H,J,P,Q] = initiate(order,Q1,R,[u(1) y(1)]);

%   Build the state-space model of specified order

Aest = [zeros(1,order); eye(order-1) zeros(order-1,1)];
Best = zeros(order,1);
Cest = [zeros(1,order-1) 1];

%   Initialize the variables

SP = 1;
CE = 1;
laste = 0;
mse = 0;
gain = 1;
x = ones(n,1);
s = s0;

%   Run the simulation for the specified number of samples

for i = 1:steps
```

```
%   Calculate the mean-square error

  if rem(i,20) == 0,
    mse = mse * 0.05;
    if mse > laste,
      SP = SP*2;
    else
      SP = sqrt(SP);
    end
  else
    if rem(i,20) == 1,
      laste = mse;
      mse = 0;
      if i > 1,
        gain = 1 / abs(yest(i-5));
      end
    end
  end

  yest(i) = Cest*s(1:order)';
  y(i) = C*x;
  mse = mse + (y(i) - yest(i))^2;
  x = A*x + B*(uest(i)+noiseu(i)+ref(i));
  z = [uest(i)+ref(i) y(i)+noisey(i)];

  [s,P] = rsplid(s,EG,F,H,J,P,Q,R,z);

  Aest(:,order) = s(1,order+1:2*order)';
  Best = s(1,2*order+1:3*order)';
  K = dlqr(Aest,Best,Cest'*SP*Cest,CE);
  uest(i+1) = -K*s(1,1:order)';

end
```

Vita

James H. Hutchinson, III was born January 29, 1965 in Opelika Alabama. He grew up in the vicinity of Murfreesboro, Tennessee where he graduated from Oakland High School in June, 1983. He attended college on a dual-degree program, whereby he spent three years at Middle Tennessee State University studying mathematics and two more years at the Georgia Institute of Technology studying electrical engineering. He received his B.S. from MTSU in May, 1988 and his B.E.E. from Georgia Tech in June, 1988.

He began graduate school at the Virginia Polytechnic Institute and State University in August, 1988 and finished in June, 1990, receiving the Master of Science in Electrical Engineering. He is employed at the Manned Flight Simulator of the Naval Air Test Center in Patuxent River, Maryland.

James H. Hutchinson, III