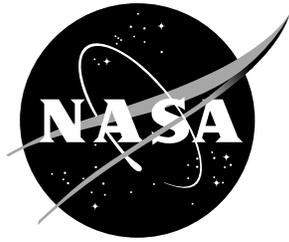


NASA Technical Memorandum 112862



# Deadbeat Predictive Controllers

Jer-Nan Juang  
*Langley Research Center, Hampton, Virginia*

Minh Phan  
*Princeton University, Princeton, New Jersey*

May 1997

National Aeronautics and  
Space Administration  
Langley Research Center  
Hampton, Virginia 23681-0001

# Deadbeat Predictive Controllers

**Jer-Nan Juang** \*

*NASA Langley Research Center  
Hampton, VA 23681*

**Minh Phan** †

*Princeton University  
Princeton, NJ 08544*

August 10, 1996

## Abstract

Several new computational algorithms are presented to compute the deadbeat predictive control law that brings the output response to rest after a finite number of time steps. The first algorithm makes use of a multi-step-ahead output prediction to compute the control law without explicitly calculating the controllability matrix. The system identification must be performed first and then the predictive control law is designed. The second algorithm uses the input and output data directly to compute the feedback law. It combines the system identification and the predictive control law into one formulation. The third algorithm uses an observable-canonical form realization to design the predictive controller. The relationship between all three algorithms is established through the use of the state-space representation. All algorithms are applicable to multi-input multi-output systems with disturbance inputs. In addition to the feedback terms, feedforward terms may also be added for disturbance

---

\*Principal Scientist, Structural Dynamics Branch.

†Assistant Professor, Department of Aerospace and Mechanical Engineering.

inputs if they are measurable. Although the feedforward terms do not influence the stability of the closed-loop feedback law, they enhance the performance of the controlled system.

## 1 Introduction

The traditional approach for active control of mechanical and aerospace systems involves four key steps including system modeling, system identification testing, controller design and verification tests. The procedure is very time consuming and costly. In many cases, such as the acoustic noise reduction for aircraft and vibration suppression for spacecraft, the approach cannot be quick enough to catch up with the system changes. On-line system identification and adaptive controller design become the only solution for the controlled system. Advanced algorithms must be developed for autonomous dynamic response and uncertainty characterization, and the controller design directly from input and output data.

There is a great amount of literature on the subject of adaptive control.<sup>1-10</sup> Most of them use a linear input-output model that describes the current output prediction as a linear combination of past input and output measurements. The finite difference model, which is commonly called the Auto-Regressive moving average model with eXogenous input (ARX), is the one used most often by researchers for the adaptive control design. For example, the Generalized Predictive Control (GPC)<sup>5</sup> starts with the ARX model with the absence of the direct transmission term and builds a multi-step ahead output predictor by solving the Diophantine equation recursively. The predictive control law is then computed using the Toeplitz matrix formed from the step response time history of the system in conjunction with a cost function with weighted input and output. There are three design parameters involved including the control weight, the prediction horizon and the control horizon. A proper combination of these parameters is required in order to guarantee stability of the

predictive control law. In contrast to the conventional approach, a novel approach has been introduced by the authors<sup>12</sup> integrating a state-space based modern control into its corresponding ARX model. It exploits the use of the relationship between the state-space model and the ARX model. The predictive controller thus derived has the same form as those derived from classical input-output models with the direct transmission term. Yet it may also be implemented as an observer-based full-state feedback controller. This provides flexibilities for control engineers to perform their job in a way that they prefer. Similar to GPC, the approach has one control design parameter and one identification parameter related to the order of the system. The control design parameter, which is similar to the GPC control horizon, gives the number of time steps for the system to become deadbeat (rest). For convenience, the approach described in Ref. [12] is referred to as the Deadbeat Predictive Control (DPC). The DPC guarantees closed-loop stability for a controllable system regardless of minimum or non-minimum phase. No special treatment is required when the system has a direct transmission term.

This paper develops several new deadbeat control algorithms to compute the deadbeat predictive control law. The feedback law is supposed to bring the output response to rest after a few specific time steps. The first algorithm makes use of a multi-step-ahead output prediction to compute the DPC without explicitly computing the controllability matrix as shown in Ref. [12]. Given the coefficient matrices of an ARX model, a recursive formulation for computing the multi-step-ahead output prediction is presented. The recursive formula is somewhat different from the one described in Ref. [5] for the Diophantine equation. The second algorithm uses the input and output data directly to compute the DPC without using the ARX coefficient matrices. It combines the system identification and the predictive

control law into one formulation. The third algorithm uses an observable-canonical form realization from an ARX model to derive the DPC. The approach is similar to that used in Ref. [12]. However, it has a different form of companion matrix for the state matrix. The relationship between all three algorithms is established through the use of the state-space representation. All three algorithms are applicable to multi-input multi-output systems with disturbance inputs. In addition to the feedback terms for DPC, feedforward terms may also be added for disturbance inputs if they are measurable. Although the feedforward terms do not influence the stability of the closed-loop feedback design, they enhance the performance of the controlled system. All good features for the method described in Ref. [12] remain true for the algorithms developed in this paper.

## 2 Multi-Step Output Prediction

The input output relationship of a linear system, even a nonlinear system, is commonly described by a finite difference model. Given a system with  $r$  inputs and  $m$  outputs, the finite difference equation for the  $r \times 1$  input  $u(k)$  and the  $m \times 1$  output  $y(k)$  at time  $k$  is

$$\begin{aligned}
 y(k) = & \alpha_1 y(k-1) + \alpha_2 y(k-2) + \cdots + \alpha_p y(k-p) \\
 & + \beta_0 u(k) + \beta_1 u(k-1) + \beta_2 u(k-2) + \cdots + \beta_p u(k-p)
 \end{aligned} \tag{1}$$

It simply means that the current output can be predicted by the past input and output time histories. The finite difference model is also often referred to as the ARX model where AR refers to the AutoRegressive part and X refers to the eXogeneous part. The coefficient matrices,  $\alpha_i$  ( $i = 1, 2, \dots, p$ ) of  $m \times m$  and  $\beta_i$  ( $i = 0, 1, \dots, p$ ) of  $m \times r$ , are commonly referred to as the observer Markov parameters (OMP) or ARX parameters. The matrix  $\beta_0$  is the direct transmission term.

By shifting a time step, one obtains

$$\begin{aligned}
y(k+1) &= \alpha_1 y(k) + \alpha_2 y(k-1) + \cdots + \alpha_p y(k-p+1) \\
&+ \beta_0 u(k+1) + \beta_1 u(k) + \beta_2 u(k-1) + \cdots + \beta_p u(k-p+1)
\end{aligned} \tag{2}$$

Define the following quantities

$$\begin{aligned}
\alpha_1^{(1)} &= \alpha_1 \alpha_1 + \alpha_2 & \beta_1^{(1)} &= \alpha_1 \beta_1 + \beta_2 \\
\alpha_2^{(1)} &= \alpha_1 \alpha_2 + \alpha_3 & \beta_2^{(1)} &= \alpha_1 \beta_2 + \beta_3 \\
\vdots & \quad \vdots & \vdots & \quad \vdots \\
\alpha_{p-1}^{(1)} &= \alpha_1 \alpha_{p-1} + \alpha_p & \beta_{p-1}^{(1)} &= \alpha_1 \beta_{p-1} + \beta_p \\
\alpha_p^{(1)} &= \alpha_1 \alpha_p & \beta_p^{(1)} &= \alpha_1 \beta_p
\end{aligned} \tag{3}$$

and

$$\beta_0^{(1)} = \alpha_1 \beta_0 + \beta_1 \tag{4}$$

Substituting  $y(k)$  from Eq. (1) into Eq. (2) yields

$$\begin{aligned}
y(k+1) &= \alpha_1^{(1)} y(k-1) + \alpha_2^{(1)} y(k-2) + \cdots + \alpha_p^{(1)} y(k-p) \\
&+ \beta_0 u(k+1) + \beta_0^{(1)} u(k) \\
&+ \beta_1^{(1)} u(k-1) + \beta_2^{(1)} u(k-2) + \cdots + \beta_p^{(1)} u(k-p)
\end{aligned} \tag{5}$$

The output measurement at time step  $k+1$  can be expressed as the sum of past input and output data with the absence of the output measurement at time step  $k$ . By induction, one may express the output measurement at the time step  $k+j$  by

$$\begin{aligned}
y(k+j) &= \alpha_1^{(j)} y(k-1) + \alpha_2^{(j)} y(k-2) + \cdots + \alpha_p^{(j)} y(k-p) \\
&+ \beta_0 u(k+j) + \beta_0^{(1)} u(k+j-1) + \cdots + \beta_0^{(j)} u(k) \\
&+ \beta_1^{(j)} u(k-1) + \beta_2^{(j)} u(k-2) + \cdots + \beta_p^{(j)} u(k-p)
\end{aligned} \tag{6}$$





Similarly,  $u_s(k)$  has  $s$  input data points starting from the time step  $k$  and  $u_p(k-p)$  has  $p$  input data points from  $k-p$ . The matrix  $\mathcal{T}$  is commonly called the Toeplitz matrix which is formed from the parameters,  $\beta_0, \beta_0^{(1)}, \dots$ , and  $\beta_0^{(s-1)}$  (the pulse response sequence). Indeed, assume that before time step  $k$ , the system is at rest, i.e.,  $u_p(k-p) = 0$  and  $y_p(k-p) = 0$ . At time step  $k$ , one applies to the system an unit pulse one at a time for each input, i.e.,  $u(k) = 1$  for a single input and  $u(k+1) = u(k+2) = \dots = 0$ . Equation (11) shows that  $y(k) = \beta_0$ ,  $y(k+1) = \beta_0^{(1)}$ ,  $\dots$ ,  $y(k+s-1) = \beta_0^{(s-1)}$ .

The vector  $y_s(k)$  in Equation (11) consists of three terms. The first term is the input vector  $u_s(k)$  including future inputs from time step  $k$  to  $k+s-1$ . Relative to the same time  $k$ , the second and third terms,  $u_p(k-p)$  and  $y_p(k-p)$ , are input and output vectors, respectively, with past known quantities from  $k-p$  to  $k-1$ . The future input vector  $u_s(k)$  is to be determined for feedback control.

### 3 Deadbeat Predictive Control Designs

There are two predictive control designs to be shown in this section. The first design is based on Eq. (11) with the assumption that the parameters  $\alpha_1, \alpha_2, \dots, \alpha_p$  and  $\beta_0, \beta_1, \dots, \beta_p$ , are given a priori. The second design uses the input and output data directly without explicitly involving the parameters  $\alpha_1, \alpha_2, \dots, \alpha_p$  and  $\beta_0, \beta_1, \dots, \beta_p$ .

#### 3.1 Indirect Algorithm

Consider the question: what should the future input signal  $u(k), u(k+1), \dots, u(k+q-1)$  be to make the future output sequence  $y(k+q), y(k+q+1), \dots, \infty$  equal to zero (deadbeat)? Here we have assumed that the control action starts at time step  $k$ . Before time  $k$ , the system is open-loop.

Let the control action be turned on at time step  $k$  and ended at  $k + q$ . In other words, the control action occurs only from  $u(k)$  to  $u(k + q - 1)$ , i.e.,  $u(k + q)$  and beyond the step  $k + q$  are all zero. Under this condition, Eq. (11) produces the following equation,

$$y_p(k + q) = \mathcal{T}'u_q(k) + \mathcal{B}'u_p(k - p) + \mathcal{A}'y_p(k - p) \quad (14)$$

where

$$y_p(k + q) = \begin{bmatrix} y(k + q) \\ y(k + q + 1) \\ \vdots \\ y(k + q + p - 1) \end{bmatrix}, \quad u_q(k) = \begin{bmatrix} u(k) \\ u(k + 1) \\ \vdots \\ u(k + q - 1) \end{bmatrix} \quad (15)$$

and

$$\begin{aligned} \mathcal{T}' = \mathcal{T}(qm + 1 : pm + qm, 1 : qr) &= \begin{bmatrix} \beta_0^{(q)} & \beta_0^{(q-1)} & \dots & \beta_0^{(1)} \\ \beta_0^{(q+1)} & \beta_0^{(q)} & \dots & \beta_0^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{(q+p-1)} & \beta_0^{(q+p-2)} & \dots & \beta_0^{(p)} \end{bmatrix} \\ \mathcal{B}' = \mathcal{B}(qm + 1 : pm + qm, :) &= \begin{bmatrix} \beta_p^{(q)} & \beta_{p-1}^{(q)} & \dots & \beta_1^{(q)} \\ \beta_p^{(q+1)} & \beta_{p-1}^{(q+1)} & \dots & \beta_1^{(q+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_p^{(q+p-1)} & \beta_{p-1}^{(q+p-1)} & \dots & \beta_1^{(q+p-1)} \end{bmatrix} \\ \mathcal{A}' = \mathcal{A}(qm + 1 : pm + qm, :) &= \begin{bmatrix} \alpha_p^{(q)} & \alpha_{p-1}^{(q)} & \dots & \alpha_1^{(q)} \\ \alpha_p^{(q+1)} & \alpha_{p-1}^{(q+1)} & \dots & \alpha_1^{(q+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_p^{(q+p-1)} & \alpha_{p-1}^{(q+p-1)} & \dots & \alpha_1^{(q+p-1)} \end{bmatrix} \end{aligned} \quad (16)$$

Equation (14) is a reduced version of Eq. (11) by cutting its first  $q$  equations and the equations beyond  $q + p - 1$ . The matrix  $\mathcal{T}'$  of dimension  $pm \times qr$  is formed from the pulse response (system Markov parameters). Note that  $m$  is the number of outputs,  $p$  is the order of the ARX model,  $r$  is the number of inputs, and  $q$  is the number of control steps. If one flips the columns in the left/right direction and preserves the rows of  $\mathcal{T}'$ , it becomes a Hankel matrix

of the pulse response, i.e.,

$$H = \begin{bmatrix} \beta_0^{(1)} & \beta_0^{(2)} & \cdots & \beta_0^{(q-1)} & \beta_0^{(q)} \\ \beta_0^{(2)} & \beta_0^{(3)} & \cdots & \beta_0^{(q)} & \beta_0^{(q+1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta_0^{(p)} & \beta_0^{(p+1)} & \cdots & \beta_0^{(q+p-2)} & \beta_0^{(q+p-1)} \end{bmatrix} \quad (17)$$

The Hankel matrix is known to have maximum rank of  $n$  which is the order of the system if  $pm \geq n$ . Choosing any number which is larger than  $pm$  does not increase the rank of  $\mathcal{T}'$ . That is why the number of rows for  $\mathcal{T}'$  is chosen to be  $pm$  even though any number greater than  $pm$  may be used to form Eq. (14). The integer  $q$  must also be chosen such that  $qr \geq n$  to make sure that the Hankel matrix has rank  $n$ .

The output vector  $y_p(k+q)$  in Eq. (14) includes the output sequence from the time step  $k+q$  to  $k+q+p-1$ . It depends on the input vector  $u_q(k)$  for the input sequence from the time step  $k$  to  $k+q-1$  which is one step behind the step  $k+q$  for the first output in  $y_p(k+q)$ . It also relies on  $u_p(k-p)$  and  $y_p(k-p)$  consisting of the input and output sequences from the time step  $k-p$  to  $k-1$ . The significance of Eq. (14) is that the input and output relation has been rewritten so that the output at time  $k+q$  and beyond can be computed from the input sequence from  $k-p$  to  $k+q-1$  and the output sequence from  $k-p$  to  $k-1$ . In other words, the output sequence from  $k$  to  $k+q-1$  is not required to be known for the prediction of the output at the time  $k+q$  and beyond. This prediction characteristic can be capitalized on for the feedback design shown below.

From Eq. (14), it is clear that the following equality

$$u_q(k) = -[\mathcal{T}']^\dagger [\mathcal{B}'u_p(k-p) + \mathcal{A}'y_p(k-p)] \quad (18)$$

will bring  $y_p(k+q)$  to rest, i.e.,

$$y_p(k+q) = \begin{bmatrix} y(k+q) \\ y(k+q+1) \\ \vdots \\ y(k+q+p-1) \end{bmatrix} = 0$$

The first  $r$  rows of Eq. (18) thus gives

$$\begin{aligned} u(k) &= -\text{first } r \text{ rows of } \{[\mathcal{T}']^\dagger\} [\mathcal{B}'u_p(k-p) + \mathcal{A}'y_p(k-p)] \\ &= \alpha_1^c y(k-1) + \alpha_2^c y(k-2) + \cdots + \alpha_p^c y(k-p) \\ &\quad + \beta_1^c u(k-1) + \beta_2^c u(k-2) + \cdots + \beta_p^c u(k-p) \end{aligned} \quad (19)$$

where the superscript  $c$  signifies the control parameters. The feedback control parameters  $\alpha_1^c, \dots, \alpha_p^c$  and  $\beta_1^c, \beta_2^c, \dots, \beta_p^c$  are to be used to compute the current control signal  $u(k)$  using the past  $p$  input and output measurements. The control action is supposed to bring the output to zero for all time steps larger than  $k+q$ . Along with the desired zero input  $u(k+q)$  and beyond, the system should be at rest, i.e., deadbeat, beyond time step  $k+q$ . That is in theory. In practice, when the system has input and output uncertainties, the control action can only bring the output down to the the level of uncertainties.

### 3.1.1 Computational Steps

The indirect method for predictive control design is summarized as follows.

- 1) Use any system identification (batch or recursive) technique to determine the open-loop observer Markov parameters (ARX) parameters  $\alpha_1, \dots, \alpha_p$ , and  $\beta_0, \beta_1, \dots, \beta_p$ , before the control action is turned on.
- 2) Compute the system Markov parameters (pulse response sequence) with the recursive formula, Eq. (8), and form the Toeplitz matrix  $\mathcal{T}'$  shown in Eq. (16). The integer  $q$

must be properly chosen such that the rank of  $\mathcal{T}'$  is  $n$  or  $pm$  whichever is the least where  $n$  is the order of the system and  $m$  is the number of outputs.

- 3) Form matrices  $\mathcal{A}'$  and  $\mathcal{B}'$  shown in Eq. (16) with their elements computed using the recursive formula, Eq. (7).
- 4) Use Eq. (19) to compute the feedback control parameters  $\alpha_1^c, \dots, \alpha_p^c$  and  $\beta_1^c, \beta_2^c, \dots, \beta_p^c$ .

### 3.2 Direct Algorithm

One may be interested in computing the feedback control parameters shown in Eq. (19) directly from input and output data. That is to bypass the first three steps of the indirect method for a predictive control design. To achieve the goal, first start with Eqs. (12) and (13), and form the following input and output matrices.

$$\begin{aligned}
 Y_s(k) &= [y_s(k) \quad y_s(k+1) \quad \cdots \quad y_s(k+N-1)] \\
 &= \begin{bmatrix} y(k) & y(k+1) & \cdots & y(k+N-1) \\ y(k+1) & y(k+2) & \cdots & y(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(k+s-1) & y(k+s) & \cdots & y(k+s+N-2) \end{bmatrix} \\
 U_s(k) &= [u_s(k) \quad u_s(k+1) \quad \cdots \quad u_s(k+N-1)] \\
 &= \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+N-1) \\ u(k+1) & u(k+2) & \cdots & u(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ u(k+s-1) & u(k+s) & \cdots & u(k+s+N-2) \end{bmatrix}
 \end{aligned} \tag{20}$$

and

$$\begin{aligned}
Y_p(k-p) &= [y_p(k-p) \quad y_p(k-p+1) \quad \cdots \quad y_s(k-p+N-1)] \\
&= \begin{bmatrix} y(k-p) & y(k-p+1) & \cdots & y(k-p+N-1) \\ y(k-p+1) & y(k-p+2) & \cdots & y(k-p+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(k-1) & y(k) & \cdots & y(k+N-2) \end{bmatrix}
\end{aligned} \tag{21}$$

$$\begin{aligned}
U_p(k-p) &= [u_p(k-p) \quad u_p(k-p+1) \quad \cdots \quad u_p(k-p+N-1)] \\
&= \begin{bmatrix} u(k-p) & u(k-p+1) & \cdots & u(k-p+N-1) \\ u(k-p+1) & u(k-p+2) & \cdots & u(k-p+N) \\ \vdots & \vdots & \ddots & \vdots \\ u(k-1) & u(k) & \cdots & u(k+N-2) \end{bmatrix}
\end{aligned}$$

where  $N$  is an integer. The data matrices  $U_s(k)$  and  $Y_s(k)$  include the input and output data information up to the data point  $k+s+N-2$ , whereas  $U_p(k-p)$  and  $Y_p(k-p)$  have data up to  $k+N-2$ .

Application of Eq. (11) yields

$$Y_s(k) = \mathcal{T}U_s(k) + \mathcal{B}U_p(k-p) + \mathcal{A}Y_p(k-p) \tag{22}$$

or

$$Y_s(k) = \begin{bmatrix} \mathcal{T} & \mathcal{B} & \mathcal{A} \end{bmatrix} \begin{bmatrix} U_s(k) \\ U_p(k-p) \\ Y_p(k-p) \end{bmatrix} \tag{23}$$

Let the integers  $s$  and  $N$  be chosen large enough in the sense that the matrix  $U_s(k)$  of dimension  $sr \times N$  with  $sr \leq N$  has rank  $sr$ , the matrix  $U_p(k-p)$  of dimension  $pr \times N$  with  $pr \leq N$  has rank  $pr$ , and the matrix  $Y_p(k-p)$  of dimension  $pm \times N$  with  $pm \leq N$  has rank  $pr$ . Again,  $r$  means the number of inputs and  $m$  represents the number of outputs. Equation (22) produces the following least-squares solution

$$\begin{bmatrix} \mathcal{T} & \mathcal{B} & \mathcal{A} \end{bmatrix} = Y_s(k) \begin{bmatrix} U_s(k) \\ U_p(k-p) \\ Y_p(k-p) \end{bmatrix}^\dagger \tag{24}$$

where  $\dagger$  means the pseudo-inverse. From the triple  $[\mathcal{T}, \mathcal{B}, \mathcal{A}]$ , it is easy to extract the triple  $[\mathcal{T}', \mathcal{B}', \mathcal{A}']$  defined in Eq. (16) for computing the control parameters  $\alpha_1^c, \dots, \alpha_p^c$  and  $\beta_1^c, \beta_2^c, \dots, \beta_p^c$  using Eq. (19).

Equation (22) has some redundant equations which may be eliminated to directly compute the triple  $[\mathcal{T}', \mathcal{B}', \mathcal{A}']$  without computing  $[\mathcal{T}, \mathcal{B}, \mathcal{A}]$ . Indeed, let us set

$$s = q + p$$

and delete the first  $qm$  rows of Eq. (24). Equation (24) reduces to

$$[\mathcal{T}'' \quad \mathcal{B}' \quad \mathcal{A}'] = Y_p(k+q) \begin{bmatrix} U_{q+p}(k) \\ U_p(k-p) \\ Y_p(k-p) \end{bmatrix}^\dagger \quad (25)$$

where  $\mathcal{T}''$ ,  $\mathcal{B}'$ , and  $\mathcal{A}'$  are obtained by deleting the first  $qm$  rows of  $\mathcal{T}$  and  $\mathcal{B}$ , and  $\mathcal{A}$  respectively. The matrices  $\mathcal{B}'$  and  $\mathcal{A}'$  are identical to those defined in Eq. (16). The matrix  $\mathcal{T}''$  has more columns than  $\mathcal{T}'$  defined in Eq. (16), i.e.,

$$\mathcal{T}' = \mathcal{T}''(:, 1 : qr) \quad (26)$$

Now, the data matrices become

$$\begin{aligned} Y_p(k+q) &= [y_p(k+q) \quad y_p(k+q+1) \quad \cdots \quad y_p(k+q+N-1)] \\ &= \begin{bmatrix} y(k+q) & y(k+q+1) & \cdots & y(k+q+N-1) \\ y(k+q+1) & y(k+q+2) & \cdots & y(k+q+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(k+q+p-1) & y(k+q+p) & \cdots & y(k+q+p+N-2) \end{bmatrix} \end{aligned} \quad (27)$$

$$\begin{aligned} U_{q+p}(k) &= [u_{q+p}(k) \quad u_{q+p}(k+1) \quad \cdots \quad u_{q+p}(k+N-1)] \\ &= \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+N-1) \\ u(k+1) & u(k+2) & \cdots & u(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ u(k+q+p-1) & u(k+s) & \cdots & u(k+q+p+N-2) \end{bmatrix} \end{aligned}$$

At this moment, all input and output data are measured from the open-loop system, before any control action begins.

From the triple  $[\mathcal{T}'', \mathcal{B}', \mathcal{A}']$ , the control law from Eq. (19) can be applied to compute the control gain parameters,

$$\begin{aligned}
u(k) &= \text{--first } r \text{ rows of } \{[\mathcal{T}''(:, 1 : qr)]^\dagger\} [\mathcal{B}'u_p(k-p) + \mathcal{A}'y_p(k-p)] \\
&= \alpha_1^c y(k-1) + \alpha_2^c y(k-2) + \cdots + \alpha_p^c y(k-p) \\
&\quad + \beta_1^c u(k-1) + \beta_2^c u(k-2) + \cdots + \beta_p^c u(k-p)
\end{aligned} \tag{28}$$

### 3.2.1 Computational Steps

The computation steps involved in the direct method for predictive control design are:

- 1) Form the data matrices  $Y_p(k+q)$  and  $U_{q+p}(k)$  defined in Eq. (27), and  $Y_p(k-p)$  and  $U_p(k-p)$  defined in Eq. (21). The integer  $p$  must be chosen such that  $pm \geq n$  where  $m$  is the number of outputs and  $n$  is the anticipated system order. The integer  $q \geq p$  is chosen such that the Hankel matrix defined in Eq. (17) has rank  $n$ .
- 2) Compute the least-squares solution, Eq. (25), to determine  $\mathcal{T}'', \mathcal{B}'$ , and  $\mathcal{A}'$ .
- 3) Use Eq. (28) to compute the feedback control parameters  $\alpha_1^c, \dots, \alpha_p^c$  and  $\beta_1^c, \beta_2^c, \dots, \beta_p^c$ .

The direct method seems simpler in computation. Nevertheless, it by no means implies that the direct method will save time in computation compared to the indirect method which includes the computation of the observer Markov (ARX) parameters. The reason is that the direct method involves a larger matrix manipulation in computing  $\mathcal{T}'', \mathcal{B}'$ , and  $\mathcal{A}'$  from Eq. (25). In addition, there is no theoretical proof that the direct method is more robust than the indirect method with the presence of system uncertainties.

## 4 Observable-Canonical Form Representation

Some researchers may be interested in knowing the corresponding state-space representation for the techniques described earlier. There are cases where a state-space model is very useful in conducting controller designs particularly for those engineers who have strong background in modern control theory. It also provides them with flexibilities for real-time implementation.

Given Eq. (1) or equivalently Eq. (6), there is a direct way of determining the system matrices for a state-space representation. Let us choose the state variables as

$$\begin{aligned}
 x_1(k) &= y(k) - \beta_0 u(k) \\
 x_2(k) &= y(k+1) - \beta_0 u(k+1) - \beta_0^{(1)} u(k) \\
 x_3(k) &= y(k+2) - \beta_0 u(k+2) - \beta_0^{(1)} u(k+1) - \beta_0^{(2)} u(k) \\
 &\vdots \quad \quad \quad \vdots \\
 x_p(k) &= y(k+p-1) \\
 &\quad - \beta_0 u(k+p-1) - \beta_0^{(1)} u(k+p-2) - \dots - \beta_0^{(p-1)} u(k)
 \end{aligned} \tag{29}$$

where each vector  $x_i(k)$ ,  $i = 1, 2, \dots, p$ , has length  $m$ , which is the number of outputs.

The set of equations in Eq. (29) yields

$$\begin{aligned}
 x_1(k+1) &= x_2(k) + \beta_0^{(1)} u(k) \\
 x_2(k+1) &= x_3(k) + \beta_0^{(2)} u(k) \\
 x_3(k+1) &= x_4(k) + \beta_0^{(3)} u(k) \\
 &\vdots \quad \quad \quad \vdots \\
 x_p(k+1) &= y(k+p) \\
 &\quad - \beta_0 u(k+p) - \beta_0^{(1)} u(k+p-1) - \dots - \beta_0^{(p-1)} u(k+1) \\
 &= \alpha_1 x_p(k) + \alpha_2 x_{p-1}(k) + \dots + \alpha_p x_1(k) + \beta_0^{(p)} u(k)
 \end{aligned} \tag{30}$$

where the last equation is obtained by using Eqs. (1) and (9). The above equations can be

arranged in matrix form as

$$x(k+1) = Ax(k) + Bu(k) \quad (31)$$

$$y(k) = Cx(k) + Du(k) \quad (32)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{p-1}(k) \\ x_p(k) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & I \\ \alpha_p & \alpha_{p-1} & \alpha_{p-2} & \cdots & \alpha_2 & \alpha_1 \end{bmatrix},$$

$$B = \begin{bmatrix} \beta_0^{(1)} \\ \beta_0^{(2)} \\ \vdots \\ \beta_0^{(p-1)} \\ \beta_0^{(p)} \end{bmatrix}, \quad C = [I \ 0 \ \cdots \ 0 \ 0], \quad (33)$$

$$D = \beta_0$$

Recall that  $p$  is the number of available observer Markov parameters,  $m$  the number of outputs and  $r$  the number of inputs. The state vector  $x$  becomes an  $mp \times 1$  vector, the state matrix  $A$  an  $mp \times mp$  matrix, the input matrix  $B$  an  $mp \times r$  matrix, and the output matrix  $C$  an  $m \times mp$  matrix. A state-space model in the form of Eq. (33) is said to be in the *canonical-form*.

The observability matrix of the canonical-form realization is

$$\mathcal{Q} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix} \quad (34)$$

The matrix  $\mathcal{Q}$  is an identity matrix which is obviously nonsingular. It implies that the observability matrix  $\mathcal{Q}$  has a rank of  $mp$  and thus all states in the state vector  $x$  are observable.

Are they controllable as well? First, form the controllability matrix

$$H = [B \ AB \ A^2B \ \cdots \ A^{s-1}B]$$

$$= \begin{bmatrix} \beta_0^{(1)} & \beta_0^{(2)} & \beta_0^{(3)} & \dots & \beta_0^{(s)} \\ \beta_0^{(2)} & \beta_0^{(3)} & \beta_0^{(4)} & \dots & \beta_0^{(s+1)} \\ \beta_0^{(3)} & \beta_0^{(4)} & \beta_0^{(5)} & \dots & \beta_0^{(s+2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_0^{(p)} & \beta_0^{(p+1)} & \beta_0^{(p+2)} & \dots & \beta_0^{(s+p-1)} \end{bmatrix} \quad (35)$$

where Eq. (9) has been used to form this matrix. The controllability matrix  $H$  is a  $pm \times rs$  Hankel matrix formed from system Markov parameters (pulse response sequence). The maximum rank of  $H$  is  $n$  which is the order of the system. Assume that the integer  $s$  is chosen large enough, i.e.,  $rs \geq pm$ . If  $pm = n$ , the rank of  $H$  is identical to that of  $\mathcal{Q}$ . As a result, the state-space representation, Eq. (33), is a minimum realization from given observer Markov parameters  $\alpha_1, \alpha_2, \dots, \alpha_p, \beta_0, \beta_1, \dots, \beta_p$ . A state-space representation is a minimum realization if and only if it is controllable and observable, i.e., the state matrix is the minimum order.

The maximum order of the model, Eq. (33), is  $mp$  which is the dimension of the realized state matrix  $A$ . If the number  $p$  is chosen such that  $mp$  is larger than the order of the system, then the triplet  $[A, B, C]$  is not a minimum realization. This is because the canonical-form, Eq. (33), is observable (the rank of  $\mathcal{Q}$  is  $pm$ ), but not controllable (the rank of  $H$  is less than  $pm$ ). In this case, some of the states in the state vector  $x$  are not controllable. In general, the order of a system under test is not known *a priori*. The number  $mp$  tends to be chosen significantly larger than the “effective” order of the system to accommodate the measurement noise and system uncertainties. “Effective” here means the part of the model that can be excited by the inputs and measured by the outputs. A state-space model in the form of Eq. (33) is thus named to be in the *observable canonical-form*.

One may be interested in knowing the observer which makes the state matrix become

deadbeat in certain number of time steps. First, recall the matrices  $\alpha_1^{(0)}, \alpha_1^{(1)}, \alpha_1^{(2)}, \dots, \alpha_1^{(p-1)}$  defined in Eq. (10). The following observer gain matrix

$$G = \begin{bmatrix} \alpha_1^{(0)} \\ \alpha_1^{(1)} \\ \vdots \\ \alpha_1^{(p-2)} \\ \alpha_1^{(p-1)} \end{bmatrix} \quad (36)$$

will result in

$$(A + GC)^p = \begin{bmatrix} \alpha_1^{(0)} & I & 0 & \cdots & 0 & 0 \\ \alpha_1^{(1)} & 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{(p-2)} & 0 & 0 & \cdots & 0 & I \\ \alpha_1^{(p-1)} + \alpha_p & \alpha_{p-1} & \alpha_{p-2} & \cdots & \alpha_2 & \alpha_1 \end{bmatrix}^p = 0 \quad (37)$$

In other words, the observer gain  $G$  will bring the observer state matrix  $A + GC$  to zero in  $p$  steps. The matrix  $G$  may be used to estimate the state vector  $x$  for full state feedback control designs. For a system with significant uncertainties, the deadbeat observer will converge to the steady state Kalman filter under certain conditions regarding the data length and the choice of  $p$ .

Careful examination of the definition for the state vector, Eq. (29), and the predictive output equation, Eq. (11), reveals that

$$x(k) = \mathcal{B}^o u_p(k-p) + \mathcal{A}^o y_p(k-p) \quad (38)$$

where  $\mathcal{B}^o$  is a  $pm \times pr$  matrix and  $\mathcal{A}^o$  is a  $pm \times pm$  matrix,

$$\mathcal{B}^o = \mathcal{B}(1 : pm, :) = \begin{bmatrix} \beta_p & \beta_{p-1} & \cdots & \beta_1 \\ \beta_p^{(1)} & \beta_{p-1}^{(1)} & \cdots & \beta_1^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_p^{(p-1)} & \beta_{p-1}^{(p-1)} & \cdots & \beta_1^{(p-1)} \end{bmatrix} \quad (39)$$

$$\mathcal{A}^o = \mathcal{A}(1 : pm, :) = \begin{bmatrix} \alpha_p & \alpha_{p-1} & \cdots & \alpha_1 \\ \alpha_p^{(1)} & \alpha_{p-1}^{(1)} & \cdots & \alpha_1^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_p^{(p-1)} & \alpha_{p-1}^{(p-1)} & \cdots & \alpha_1^{(p-1)} \end{bmatrix}$$

Note that the state vector has size  $pm \times 1$ . Equation (38) signifies the relationship between the state vector and the input and output data. It implies that the state at time step  $k$  can be estimated from the past  $p$  input and output data. This provides the basis for predictive control designs for a system represented by a state-space model.

## 4.1 Deadbeat Predictive Control Gain

Given a state-space representation, there are many ways to design a feedback law to control the system. Common methods include optimal control design, pole placement technique, virtual passive technique, etc. Here, a deadbeat feedback design similar to that discussed earlier will be introduced.

With some algebraic manipulations, Eq. (31) produces

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ x(k+2) &= A^2x(k) + [AB \quad B] \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \\ &\vdots \quad \quad \quad \vdots \\ x(k+q) &= A^qx(k) + \mathcal{T}'u_q(k) \end{aligned} \quad (40)$$

where

$$u_q(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+q-1) \end{bmatrix} \quad (41)$$

and

$$\begin{aligned} \mathcal{T}' &= [A^{q-1}B \quad A^{q-2}B \quad \dots \quad B] \\ &= \begin{bmatrix} \beta_0^{(q)} & \beta_0^{(q-1)} & \dots & \beta_0^{(1)} \\ \beta_0^{(q+1)} & \beta_0^{(q)} & \dots & \beta_0^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{(q+p-1)} & \beta_0^{(q+p-2)} & \dots & \beta_0^{(p)} \end{bmatrix} \end{aligned} \quad (42)$$

The matrix  $\mathcal{T}'$  is an  $n \times qr$  controllability matrix with  $n$  being the order of the system and  $r$  the number of inputs. The integer  $q$  must be chosen such that  $qr \geq n$  to assure that the matrix  $\mathcal{T}'$  has rank of  $n$ . Note that  $\mathcal{T}'$  shown in both Eqs. (16) and (42) are identical.

Equation (40) shows that the state  $x(k+q)$  at time  $k+q$  becomes zero when the input series  $u(k), u(k+1), \dots, u(k+q-1)$  is given by

$$u_q(k) = -[\mathcal{T}']^\dagger A^q x(k) \quad \implies \quad x(k+q) = 0 \quad (43)$$

which clearly implies that the input  $u(k)$  at time  $k$  is

$$\begin{aligned} u(k) &= -G_c x(k) \\ &= -\{\text{first } r \text{ rows of } [\mathcal{T}']^\dagger\} A^q x(k) \end{aligned} \quad (44)$$

Equation (44) gives a state-feedback controller that drives the state  $x(k)$  at time step  $k$  to zero after  $q$  time steps. One straightforward method of computing the gain matrix  $G_c$  is first to identify the set of system matrices  $A$ ,  $B$ ,  $C$ , and  $D$  from input and output data, and then compute the gain matrix from Eq. (44).

Substituting Eq. (38) for  $x(k)$  into Eq. (44) yields

$$\begin{aligned}
u(k) &= -\{\text{first } r \text{ rows of } [T']^\dagger\} A^q \{\mathcal{B}^o u_p(k-p) + \mathcal{A}^o y_p(k-p)\} \\
&= \alpha_1^c y(k-1) + \alpha_2^c y(k-2) + \cdots + \alpha_p^c y(k-p) \\
&\quad + \beta_1^c u(k-1) + \beta_2^c u(k-2) + \cdots + \beta_p^c u(k-p)
\end{aligned} \tag{45}$$

The control laws obtained from Eq. (19) and Eq. (45) should be identical. This implies

$$\mathcal{B}' = A^q \mathcal{B}^o \quad \text{and} \quad \mathcal{A}' = A^q \mathcal{A}^o$$

or

$$\begin{aligned}
\mathcal{B}(qm+1 : qm+pm, :) &= A^q \mathcal{B}(1 : pm, :) \\
\mathcal{A}(qm+1 : qm+pm, :) &= A^q \mathcal{A}(1 : pm, :)
\end{aligned} \tag{46}$$

This result provides an interesting connection between the state matrix  $A$  and the submatrices of  $\mathcal{A}$  and  $\mathcal{B}$  defined in Eq. (13). It should be not surprised because they are all computed from the observer Markov parameters  $\alpha_1, \alpha_2, \dots, \alpha_p, \beta_0, \beta_1, \dots, \beta_p$ .

#### 4.1.1 Computational Steps

The observable-canonical form representation for the predictive control design is summarized in the following

- 1) Use any system identification (batch or recursive) technique to determine the open-loop observer Markov parameters (ARX) parameters  $\alpha_1, \dots, \alpha_p$ , and  $\beta_0, \beta_1, \dots, \beta_p$ , before the control action is turned on.
- 2) Form the state-space model shown in Eqs. (31) and (32) with its system matrices  $A, B, C$ , and  $D$  defined in Eq. (33), and the corresponding observer gain matrix defined in Eq. (36).

- 3) Compute matrices  $\mathcal{A}^o$  and  $\mathcal{B}^o$  shown in Eq. (39) with their elements computed using the recursive formula, Eq. (7).
- 4) Calculate the control gain matrix  $G_c$  defined in Eq. (44) using the controllability matrix shown in Eq. (42) with a given integer  $q$ . The integer  $q$  must be large enough so that  $pr \geq n$  where  $r$  is the number of inputs and  $n$  is the order of the system.
- 5) Use Eq. (45) to compute the feedback control parameters  $\alpha_1^c, \dots, \alpha_p^c$  and  $\beta_1^c, \beta_2^c, \dots, \beta_p^c$ .

Some researchers may prefer to use the state-space representation described by the system matrices  $A, B, C, D$ , the observer gain matrix  $G$ , and the control gain matrix  $G_c$  for real-time implementation. The control gain  $G_c$  can be computed using any other existing methods such as the pole placement techniques, optimal control methods, etc.

## 5 Feedback and Feedforward for Disturbance Input

In addition to the control input, there may be other disturbance inputs applied to the system. Some type of disturbances comes from the known sources that can be measured. This section addresses the predictive feedback designs including feedforward from the disturbance inputs that are measurable.

With the disturbance input involved, the finite difference model shown in Eq. (1) becomes

$$\begin{aligned}
y(k) = & \alpha_1 y(k-1) + \alpha_2 y(k-2) + \dots + \alpha_p y(k-p) \\
& + \beta_{c0} u_c(k) + \beta_{c1} u_c(k-1) + \beta_{c2} u_c(k-2) + \dots + \beta_{cp} u_c(k-p) \\
& + \beta_{d0} u_d(k) + \beta_{d1} u_d(k-1) + \beta_{d2} u_d(k-2) + \dots + \beta_{dp} u_d(k-p) \quad (47)
\end{aligned}$$

where the subscripts  $c$  and  $d$  are used to signify the corresponding quantities associated with the control input and the disturbance input, respectively. Accordingly, Eq. (11) can be rewritten as

$$y_s(k) = \mathcal{T}_c u_{cs}(k) + \mathcal{T}_d u_{ds}(k) + \mathcal{B}_c u_{cp}(k-p) + \mathcal{B}_d u_{dp}(k-p) + \mathcal{A}y_p(k-p) \quad (48)$$

where

$$\begin{aligned} u_{cs}(k) &= \begin{bmatrix} u_c(k) \\ u_c(k+1) \\ \vdots \\ u_c(k+s-1) \end{bmatrix}, \quad u_{ds}(k) = \begin{bmatrix} u_d(k) \\ u_d(k+1) \\ \vdots \\ u_d(k+s-1) \end{bmatrix}, \\ u_{cp}(k-p) &= \begin{bmatrix} u_c(k-p) \\ u_c(k-p+1) \\ \vdots \\ u_c(k-1) \end{bmatrix}, \quad u_{dp}(k-p) = \begin{bmatrix} u_d(k-p) \\ u_d(k-p+1) \\ \vdots \\ u_d(k-1) \end{bmatrix} \end{aligned} \quad (49)$$

and

$$\begin{aligned} \mathcal{T}_c &= \begin{bmatrix} \beta_{c0} & & & \\ \beta_{c0}^{(1)} & \beta_{c0} & & \\ \vdots & \vdots & \ddots & \\ \beta_{c0}^{(s-1)} & \beta_{c0}^{(s-2)} & \cdots & \beta_{c0} \end{bmatrix}, \quad \mathcal{T}_d = \begin{bmatrix} \beta_{d0} & & & \\ \beta_{d0}^{(1)} & \beta_{d0} & & \\ \vdots & \vdots & \ddots & \\ \beta_{d0}^{(s-1)} & \beta_{d0}^{(s-2)} & \cdots & \beta_{d0} \end{bmatrix} \\ \mathcal{B}_c &= \begin{bmatrix} \beta_{cp} & \beta_{c(p-1)} & \cdots & \beta_{c1} \\ \beta_{cp}^{(1)} & \beta_{c(p-1)}^{(1)} & \cdots & \beta_{c1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{cp}^{(s-1)} & \beta_{c(p-1)}^{(s-1)} & \cdots & \beta_{c1}^{(s-1)} \end{bmatrix}, \quad \mathcal{B}_d = \begin{bmatrix} \beta_{dp} & \beta_{d(p-1)} & \cdots & \beta_{d1} \\ \beta_{dp}^{(1)} & \beta_{d(p-1)}^{(1)} & \cdots & \beta_{d1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{dp}^{(s-1)} & \beta_{d(p-1)}^{(s-1)} & \cdots & \beta_{d1}^{(s-1)} \end{bmatrix} \end{aligned} \quad (50)$$

In Eq. (48), the subscripts  $c$  and  $d$  mean the quantities resulting from the control input and the disturbance input, respectively. Equations (49) and (50) show that the mathematical forms for the control input and the disturbance input are identical. However, the control input is changeable but the disturbance input is not.

Using the same concept as discussed earlier, let us assume that the control action starts at the time step  $k$  and hopefully ends at  $k+q$ . This assumption is possible in theory for the noise-free case without disturbances. It is impossible for the case with random uncorrelated

disturbances. However, it is used here to obtain a stable feedback design. Equation (48) thus becomes

$$y_p(k+q) = \mathcal{T}'_c u_{cq}(k) + \mathcal{T}'_d u_{dq}(k) + \mathcal{B}'_c u_{cp}(k-p) + \mathcal{B}'_d u_{dp}(k-p) + \mathcal{A}' y_p(k-p) \quad (51)$$

where

$$\begin{aligned} \mathcal{T}'_c &= \mathcal{T}(qm+1 : qm+pm, 1 : qr_c) \\ \mathcal{T}'_d &= \mathcal{T}(qm+1 : qm+pm, 1 : qr_d) \\ \mathcal{B}'_c &= \mathcal{B}_c(qm+1 : qm+pm, :) \\ \mathcal{B}'_d &= \mathcal{B}_d(qm+1 : qm+pm, :) \\ \mathcal{A}' &= \mathcal{A}(qm+1 : qm+pm, :) \end{aligned} \quad (52)$$

Note that  $\mathcal{T}'_c$  is a  $pm \times qr_c$  matrix where  $r_c$  is the number of control inputs and  $\mathcal{T}'_d$  is a  $pm \times qr_d$  matrix where  $r_d$  is the number of disturbance inputs. It is unrealistic to predict any future disturbance signal beyond time step  $k+q$ , assume that the disturbance signal is predictable. This statement can be clearly justified by using the state-space representation approach similar to that shown in Eqs. (40) to (44).

For simplicity, assume that the goal of the control action is to minimize the output due to the disturbance. From Eq. (51), the control input starting from time  $k$ , which satisfies the following equation

$$u_{cq}(k) = -[\mathcal{T}'_c]^\dagger \{ \mathcal{T}'_d u_{dq}(k) + \mathcal{B}'_c u_{cp}(k-p) + \mathcal{B}'_d u_{dp}(k-p) + \mathcal{A}' y_p(k-p) \} \quad (53)$$

will ideally bring the output to zero after  $q$  time steps. This control law requires knowledge of the future disturbance beyond the current time  $k$ , i.e.,  $u_{dq}(k)$  defined in Eq. (49) with  $s = q+p$ . If the disturbance is uncorrelated, it is impossible to make any prediction. Thus, the control action should be

$$u_{cq}(k) = -[\mathcal{T}'_c]^\dagger \{ \mathcal{B}'_c u_{cp}(k-p) + \mathcal{B}'_d u_{dp}(k-p) + \mathcal{A}' y_p(k-p) \} \quad (54)$$

which will not bring the output response to zero after the  $q$ -step control action but minimize it. That is the best one can do for a unknown disturbance sequence.

The first  $r$  rows of Eq. (54) provides the control law for the input at any the time  $k$

$$\begin{aligned}
u_c(k) &= -\{\text{first } r \text{ rows of } [\mathcal{T}'_c]^\dagger\} \{\mathcal{B}'_c u_{cp}(k-p) + \mathcal{B}'_d u_{dp}(k-p) + \mathcal{A}' y_p(k-p)\} \\
&= \alpha_1^c y(k-1) + \alpha_2^c y(k-2) + \cdots + \alpha_p^c y(k-p) \\
&\quad + \beta_{c1}^c u_c(k-1) + \beta_{c2}^c u_c(k-2) + \cdots + \beta_{cp}^c u_c(k-p) \\
&\quad + \beta_{d1}^c u_d(k-1) + \beta_{d2}^c u_d(k-2) + \cdots + \beta_{dp}^c u_d(k-p)
\end{aligned} \tag{55}$$

In addition to the control feedback, the control law shown in Eq. (55) includes the feedforward due to the past disturbance time history. Equation (55) may be called the finite-difference model for the feedback and feedforward predictive controller. Although the control law developed in this section is for the purpose of damping out the output response, it can be easily enhanced to follow a desired output response.

## 6 Computational Steps

The indirect method for the predictive control design with feedback and feedforward is summarized as follows.

- 1) Use any system identification (batch or recursive) technique to determine the open-loop observer Markov parameters (ARX parameters),  $\alpha_1, \dots, \alpha_p$ ,  $\beta_{c0}, \beta_{c1}, \dots, \beta_{cp}$ , and  $\beta_{d0}, \beta_{d1}, \dots, \beta_{dp}$ , before the control action is turned on.
- 2) Compute the system Markov parameters (pulse response sequence) for the map from the control input to the system output with the recursive formula, Eq. (8), and form

the Toeplitz matrix  $\mathcal{T}'_c$  shown in Eq. (52). The integer  $q$  must be properly chosen such that the rank of  $\mathcal{T}'_c$  is  $n$  or  $pm$  whichever is the least where  $n$  is the order of the system and  $m$  is the number of outputs.

- 3) Form matrices  $\mathcal{A}'$ ,  $\mathcal{B}'_c$  and  $\mathcal{B}'_d$  shown in Eq. (52) with their elements computed using the recursive formula, Eq. (7). One may first compute the combined  $\mathcal{B}'$  from Eq. (7) which include the control input and the disturbance input, and then separate them into two pieces, i.e.,  $\mathcal{B}'_c$  and  $\mathcal{B}'_d$ .
- 4) Use Eq. (55) to compute the feedback control parameters  $\alpha_1^c, \dots, \alpha_p^c$  and  $\beta_{c1}^c, \beta_{c2}^c, \dots, \beta_{cp}^c$ , and the feedforward parameters  $\beta_{d1}^c, \beta_{d2}^c, \dots, \beta_{dp}^c$ .

Although this section only describes the indirect method for computing feedback and feedforward parameters for the system with both control and disturbance inputs. The same approach is applicable for the other methods presented in this paper.

## 6.1 Closed-Loop Representation

In order to characterize the closed-loop response, the closed-loop frequency response function or state-space representation is commonly needed. The first step is to integrate the two finite difference models for the open-loop system and its predictive controller together.

Equation (47) together with Eq. (55) forms the closed-loop finite difference model,

$$\begin{aligned}
 \begin{bmatrix} I & -\beta_{c0} \\ 0 & I \end{bmatrix} \begin{bmatrix} y(k) \\ u_c(k) \end{bmatrix} &= \begin{bmatrix} \alpha_1 & \beta_{c1} \\ \alpha_1^c & \beta_{c1}^c \end{bmatrix} \begin{bmatrix} y(k-1) \\ u_c(k-1) \end{bmatrix} + \dots + \begin{bmatrix} \alpha_p & \beta_{cp} \\ \alpha_p^c & \beta_{cp}^c \end{bmatrix} \begin{bmatrix} y(k-p) \\ u_c(k-p) \end{bmatrix} \\
 &+ \begin{bmatrix} \beta_{d0} \\ 0 \end{bmatrix} u_d(k) + \begin{bmatrix} \beta_{d1} \\ \beta_{d1}^c \end{bmatrix} u_d(k-1) + \dots + \begin{bmatrix} \beta_{dp} \\ \beta_{dp}^c \end{bmatrix} u_d(k-p)
 \end{aligned} \tag{56}$$

or equivalently,

$$\begin{aligned}
v(k) &= \bar{\alpha}_1 v(k-1) + \bar{\alpha}_2 v(k-2) + \cdots + \bar{\alpha}_p v(k-p) \\
&+ \bar{\beta}_0 u_d(k) + \bar{\beta}_1 u_d(k-1) + \bar{\beta}_2 u_d(k-2) + \cdots + \bar{\beta}_p u_d(k-p)
\end{aligned} \tag{57}$$

where

$$\begin{aligned}
v(k) &= \begin{bmatrix} y(k) \\ u_c(k) \end{bmatrix}, \quad \bar{\alpha}_i = \begin{bmatrix} I & \beta_{c0} \\ 0 & I \end{bmatrix} \begin{bmatrix} \alpha_i & \beta_{ci} \\ \alpha_i^c & \beta_{ci}^c \end{bmatrix}, \\
\bar{\beta}_0 &= \begin{bmatrix} I & \beta_{c0} \\ 0 & I \end{bmatrix} \begin{bmatrix} \beta_{d0} \\ 0 \end{bmatrix}, \quad \bar{\beta}_i = \begin{bmatrix} I & \beta_{c0} \\ 0 & I \end{bmatrix} \begin{bmatrix} \beta_{di} \\ \beta_{di}^c \end{bmatrix}
\end{aligned} \tag{58}$$

for  $k = 1, 2, \dots, \infty$ , and  $i = 1, 2, \dots, p$ . The vector  $v(k)$  has the length of  $m + r_c$  where  $m$  is the number of outputs and  $r_c$  is the number of control inputs. Each matrix  $\bar{\alpha}_i$  has the size of  $(m + r_c) \times (m + r_c)$  and  $\bar{\beta}_i$  is  $(m + r_d) \times r_d$  where  $r_d$  is the number of disturbance inputs. A state-space representation or its corresponding frequency response function can be directly derived from Eq. (57) for closed-loop analysis by examining its closed-loop poles and zeros.

## 7 Numerical Example

A simple spring-mass-damper system is used to illustrate various controllers. Several different cases will be discussed ranging from single-input/single-output to multi-input/multi-output. First, the noise-free case is shown and then the case with additive measurement noise is discussed.

Consider a three-degree-of-freedom spring-mass-damper system

$$M\ddot{w} + \Xi\dot{w} + Kw = u$$

where

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad \Xi = \begin{bmatrix} \zeta_1 + \zeta_2 & -\zeta_2 & 0 \\ -\zeta_2 & \zeta_2 + \zeta_3 & -\zeta_3 \\ 0 & -\zeta_3 & \zeta_3 \end{bmatrix},$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

where  $m_i, k_i, \zeta_i, i = 1, 2, 3$  are the mass, spring stiffness, and damping coefficients, respectively. For this system, the order of the equivalent state-state representation is 6 ( $n = 6$ ). The control force applied to each mass is denoted by  $u_i, i = 1, 2, 3$ . The variables  $w_i, i = 1, 2, 3$  are the positions of the three masses measured from their equilibrium positions. In the simulation,  $m_1 = m_2 = m_3 = 1Kg, k_1 = k_2 = k_3 = 1,000N/m, \zeta_1 = \zeta_2 = \zeta_3 = 0.1N - sec/m$ . The system is sampled at  $50Hz$  ( $\Delta t = 0.02sec$ ). Let the measurements  $y_i$  be the accelerations of the three masses,  $y_i = \ddot{w}_i, i = 1, 2, 3$ .

Let us consider a single-control-input, single-disturbance-input and single-output case where the control input to the system is the force on the first mass (i.e.,  $u_c = u_1$ ), the disturbance input is at the second mass (i.e.,  $u_d = u_2$ ), and the output is the acceleration of the third mass (i.e.,  $y = \ddot{w}_3$ ) (non-collocated actuator-sensor). Therefore, the smallest order of the ARX model  $p$  is 6 corresponding to a deadbeat observer, and the smallest value for  $q$  is also 6 corresponding to a deadbeat controller which will bring the entire system to rest in exactly 6 time steps if no disturbance input is present. Note that this is a non-minimum phase system. The choice of the minimum  $p = q = 6$  is to make the Hankel matrix formed from the system pulse response (system Markov parameters) to have rank of 6 which is the order of the system. This deadbeat controller is not practical because it needs excessive control. Instead, consider the case where the controller is computed with  $q = 50$ . The

controller computed using the indirect algorithm for this system has the form

$$\begin{aligned}
 u_c(k) = & -0.080u_c(k-1) - 0.020u_c(k-2) - 0.023u_c(k-3) \\
 & + 0.059u_c(k-4) + 0.094u_c(k-5) + 0.010u_c(k-6) \\
 & + 1.048y(k-1) - 3.819y(k-2) + 6.404y(k-3) \\
 & - 6.785y(k-4) + 4.173y(k-5) - 1.603y(k-6) \\
 & - 0.058u_d(k-1) - 0.278u_d(k-2) + 0.254u_d(k-3) \\
 & - 0.016u_d(k-4) - 0.192y(k-5) + 0.288u_d(k-6)
 \end{aligned}$$

In Fig. 1, the open-loop and closed-loop frequency response functions from the disturbance input to the output are shown. The solid curve is the open-loop response and the dashed curve is the closed-loop response. The peaks in Fig. 1 of the open-loop response are considerably reduced ( $> 10$  dB).

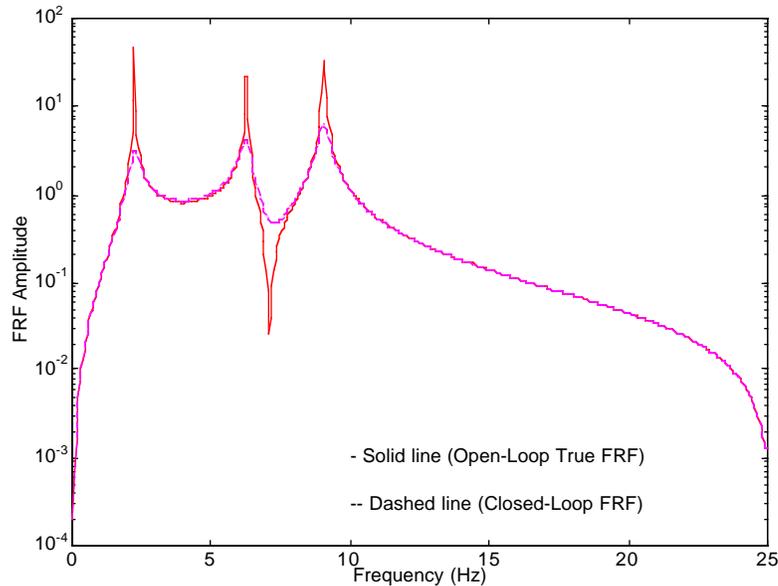


Figure 1: Open-loop and closed-loop frequency response functions (FRF) from the disturbance input to the output

Next, we consider the case where there is an additional measurement available for feedback control (unequal number of inputs and outputs). In addition to the acceleration of the third mass, acceleration measurement of the second mass is also available. The direct

transmission term in this case is non-zero. The minimum order of the ARX model is  $p = 3$ . For comparison purpose, the control parameter is kept at  $q = 50$ . The controller in this case is

$$\begin{aligned}
u_c(k) = & -0.080u_c(k-1) - 0.294u_c(k-2) + 0.412u_c(k-3) \\
& + \begin{bmatrix} -0.456 & 4.805 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} \\
& + \begin{bmatrix} 0.746 & 0.233 \end{bmatrix} \begin{bmatrix} y_1(k-2) \\ y_2(k-2) \end{bmatrix} \\
& + \begin{bmatrix} -0.4557 & -5.461 \end{bmatrix} \begin{bmatrix} y_1(k-3) \\ y_2(k-3) \end{bmatrix} \\
& -4.818u_d(k-1) + 8.752u_d(k-2) - 3.8985u_d(k-3)
\end{aligned}$$

Note that with the additional measurements, fewer time steps (and fewer controller gains) are required. This is a reflection of the fact that complete state estimation can now be achieved faster with the additional sensors. All three algorithms produce identical controllers for the noise-free cases. The frequency response functions are not shown here because they are similar to the one shown in Fig. 1.

Let the output be added with some measurement noise so that the signal to noise ratio is 4.5. The noise is random normally distributed. For the indirect algorithm, set the values of  $p$  and  $q$  to  $p = 10$  and  $q = 30$ . Although the minimum order of the the ARX model is  $p = 3$ , the larger value is given to accommodate the measurement noise. The open-loop and closed-loop frequency response functions from the disturbance input to the first and second outputs are shown in Figs. 2 and 3. Again, all the peaks of the open-loop response function are considerably reduced. Figure 3 shows the effect of the direct transmission term at frequencies near the Nyquist frequency.

For the direct algorithm, set the values of  $p$  and  $q$  as  $p = 7$  and  $q = 30$ . The open-loop and closed-loop frequency response functions from the disturbance input to the first and

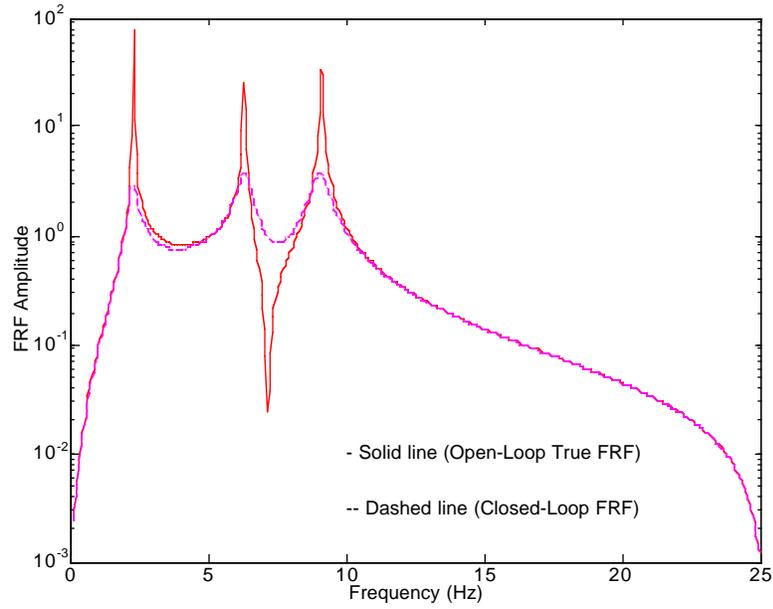


Figure 2: Open-loop and closed-loop frequency response functions (FRF) from the disturbance input to the first output for the indirect algorithm

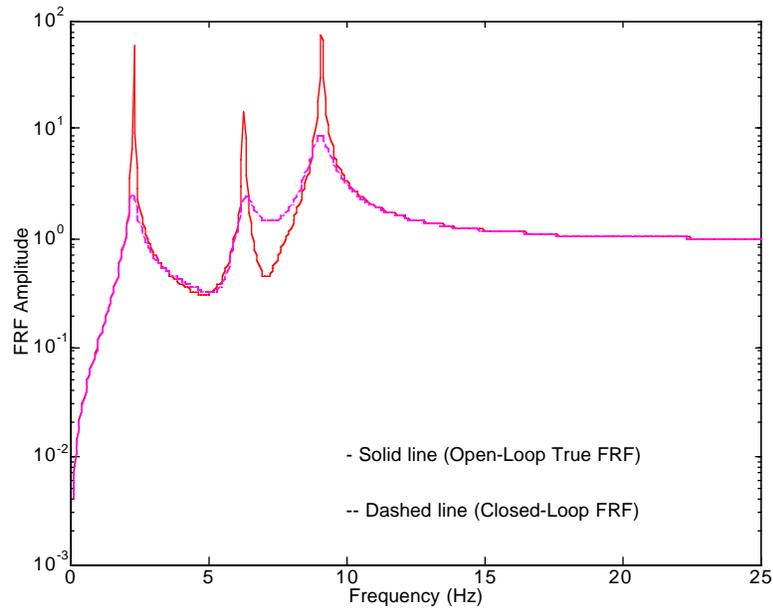


Figure 3: Open-loop and closed-loop frequency response functions (FRF) from the disturbance input to the second output for the indirect algorithm

second outputs are shown in Figs. 4 and 5. Some differences can be seen from Figs. 2 and 3 for the indirect algorithm, and Figs. 4 and 5 for the direct algorithm. Nevertheless, they are very similar although their input and output gain matrices (not shown) are quite different. The direct algorithm takes somewhat a less value of  $p$  to achieve the same control effect. This does not mean that the direct algorithm is computationally more efficient than the indirect algorithm.

## 8 Concluding Remarks

Three novel algorithms were developed for deadbeat predictive control designs. These algorithms are simple and easy to compute and so they are good candidates to be used for real-time implementation in a micro-processor. The first algorithm (indirect method) uses the multi-step-ahead output prediction to compute the control law. All computations are performed recursively. The most time consuming task is the computation of the matrix pseudo-inverse of a Hankel matrix formed by the system pulse response time history. The Hankel matrix plays the major role of establishing the rule of selecting the identification parameter and the control design parameter (i.e., control horizon). It also provides the basis to establish the uniqueness of the deadbeat predictive control law. Using the multi-step-ahead output prediction, the second algorithm (direct method) was developed combining system identification and control law into one formulation. It computes the Hankel matrix and other quantities directly from input and output data. This by no means implies that the second algorithm is more robust or computationally efficient than the first one. Nevertheless, it provides a clear insight into the fundamental structure of the deadbeat predictive control law. The third algorithm provides the state-space representation of the deadbeat predictive control law. It computes the deadbeat gain for observer-based full-state feedback that may

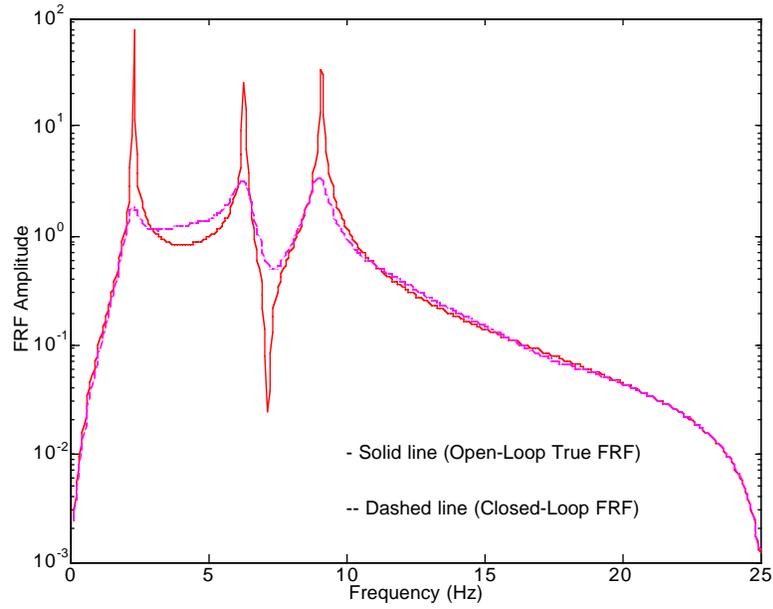


Figure 4: Open-loop and closed-loop frequency response functions (FRF) from the disturbance input to the first output for the direct algorithm

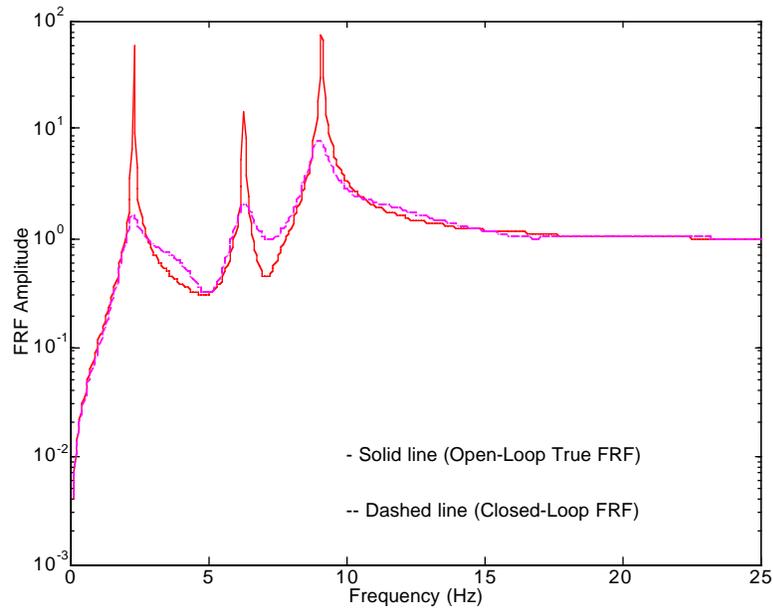


Figure 5: Open-loop and closed-loop frequency response functions (FRF) from the disturbance input to the second output for the direct algorithm

then be converted into the input and output gain used in the classical predictive control designs. The connection between the classical state-space control law and the predictive control law is clearly identified. Since the control gains are designed from the input-output models, they may be adaptively tuned from on-line input and output measurements. As a result, these controllers should be able to handle the systems with slowly time-varying dynamics, provided that input and output data are sufficiently rich to allow reasonable system identification. The system dynamics may be large and complex such as open-loop unstable, underdamped poles, etc.

## 9 References

1. Richalet, J., Rault, A., Testud, J.L., Papon, J., "Model Predictive Heuristic Control: Applications to Industrial Processes," *Automatica*, Vol. 14, No. 5, pp. 413-428, 1978.
2. De Keyser, R.M.C. and Van Cauwenberghe, A.R., "A Self-Tuning Multi-Step Predictor Application," *Automatica*, Vol. 17, No. 1, pp. 167-174, 1979.
3. Peterka, V., "Predictor-Based Self-Tuning Control," *Automatica*, Vol. 20, No. 1, pp. 39-50, 1984.
4. Ydstie, B.E., "Extended Horizon Adaptive Control," *Proceedings of the 9th IFAC World Congress*, Vol. VII, pp. 133-138, 1984.
5. Clarke, D.W., Mohtadi, C., and Tuffs, P.S., "Generalized Predictive Control - Part I. The Basic Algorithm," *Automatica*, Vol. 23, No. 2, pp. 137-148, 1987.
6. Clarke, D.W., Mohtadi, C., and Tuffs, P.S., "Generalized Predictive Control - Part II. Extensions and Interpretations," *Automatica*, Vol. 23, No. 2, pp. 149-160, 1987.

7. Goodwin, G.C. and Sin, K.S., Adaptive Filtering, Prediction, and Control, Prentice-Hall, 1984.
8. Astrom, K.J, and Wittenmark, B.,Adaptive Control, Addison-Wesley Publishing Company, 1989.
9. Soeterboek, R., Predictive Control: A Unified Approach, Prentice-Hall, 1992.
10. Mosca, E., Optimal, Predictive, and Adaptive Control, Prentice-Hall, 1995.
11. Bialasiewicz, J.T., Horta, L.G., and Phan, M., “Identified Predictive Control,” *Proceedings of the American Control Conference*, Baltimore, Maryland, June 1994.
12. Phan, M. G., and Juang, J.-N., “Predictive Feedback Controllers for Stabilization of Linear Multivariable Systems,” *Journal of Guidance, Control, and Dynamics*, To appear.
13. Chen, C.-W., Huang, J.-K., Phan, M. and Juang, J.-N., “Integrated System Identification and Modal State Estimation for Control of Large Flexible Space Structures,” *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, pp. 88-95, Jan.-Feb. 1992.
14. Phan, M., Horta, L.G., Juang, J.-N., and Longman, R.W., “Linear System Identification Via An Asymptotically Stable Observer,” *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, pp. 59-86, October 1993.
15. Juang, J.-N., Phan, M., Horta, L.G., and Longman, R.W., “Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments,” *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, March-April 1993, pp. 320-329.

16. Juang, J.-N., Applied System Identification, Prentice-Hall, 1994.
17. Phan, M., Horta, L.G., Juang, J.-N., and Longman, R.W., "Improvement of Observer/Kalman Filter Identification (OKID) by Residual Whitening," *Journal of Vibrations and Acoustics*, Vol. 117, April 1995, pp. 232-239.