IN-39
044201

NASA Technical Memorandum 112863

Recursive Deadbeat Controller Design

Jer-Nan Juang
Langley Research Center, Hampton, Virginia

Minh Q. Phan
Princeton University, Princeton, New Jersey

May 1997

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

Recursive Deadbeat Controller Design

Jer-Nan Juang [*]
NASA Langley Research Center
Hampton, VA 23681
and
Minh Q. Phan [†]
Princeton University
Princeton, NJ 08544

Abstract

This paper presents a recursive algorithm for the deadbeat predictive control that brings the output response to rest after a few time steps. The main idea is to put together the system identification process and the deadbeat control design into a unified step. It starts with reformulating the conventional multi-step output-prediction equation to explicitly include the coefficient matrices to weight past input and output time histories for computation of feedback control force. The formulation thus derived satisfies simultaneously the system identification and the deadbeat control requirements. As soon as the coefficient matrices are identified satisfying the output prediction equation, no further work is required to design a deadbeat controller. The method can be implemented recursively just as any typical recursive system identification techniques.

[*] Principal Scientist, Structural Dynamics Branch. Fellow AIAA

[†] Assistant Professor, Department of Aerospace and Mechanical Engineering.

1 Introduction

There are many interesting technical problems in the area of controlled aerospace structures that NASA researchers are trying to solve. These problems, for example, include acoustic noise reduction, flow control, ride quality control, flexible spacecraft attitude control and vibration suppression. Active or passive control for a dynamic system is not a new subject. Many control techniques[1-5] are available today and ready to be used for application to these interesting problems. Some of the techniques are the quadratic optimization technique[1-2], the pole placement technique[3], the virtual passive technique[4], the energy dissipation technique[5], and the adaptive control technique.[6-18] Some researchers prefer to work in the frequency domain using the frequency response functions (FRF) while others use the state-space model (SSM) in the time-domain to design controllers. The model-based techniques need a mathematical model (FRF of SSM) within a certain level of accuracy to design a controller. Except for a few simple cases, system identification must be involved in the design process to verify the open-loop model and the closed-loop design as well. As a result, it may take considerable time to iterate the design process until performance requirements are met. For the systems with minimum uncertainties, the iteration procedure would not bother the control engineers, as long as a satisfactory control design can be found.

For systems with unknown disturbances and considerable uncertainties, the controller must be able to adapt the unknown changes in real time. Adaptive control techniques are developed for this purpose. The approach is to adjust the control gains to reflect the system changes so as to continuously check and meet the performance requirements. Most adaptive control techniques require the controlled system to be minimum phase[19-25] in the sense that all the system transmission zeros are stable. The minimum-phase system in the continuous-time domain does not guarantee its minimum phase in the discrete-time domain. In practice, only a few structural systems in the discrete-time domain are minimum phase. Predictive controller designs[11-18] were developed to particularly address the non-minimum-phase problems with the hope that they can be implemented in real time. Two indirect techniques and one direct technique were derived in Ref. 18 using the concept of deadbeat predictive control law that brought the output response to rest after a few finite time steps. The indirect techniques require identification of coefficient matrices of a finite-difference model representing the controlled system. The deadbeat predictive controllers are then computed using the identified coefficient matrices. Note that the identified matrices minimize the output error between the estimated and real outputs. The direct technique computes the deadbeat predictive controller directly from input and output data without explicitly identifying the system parameters. However, it requires to minimize the output error first and then perform a Hankel-like[18] matrix to calculate the control gains for the past input and past output signal. Since it takes time to invert a matrix, both direct and indirect algorithms have a drawback for real time application. Nevertheless, the direct algorithm did provide the fundamental framework for further development of a recursive technique for real time implementation.

A new recursive technique is presented in this paper for the design of a deadbeat predictive controller. It uses the approach derived in Ref. 18 for the direct algorithm. The technique computes the gain matrices recursively and directly from input and output data in every sampling period. In addition, the recursive formula satisfies both system identification and deadbeat predictive control equations simultaneously. As a result, the design process is completed in such a way that there is no time delay between the identification step and the control gain computation step.

This paper begins with a brief introduction of multi-step output prediction.[18] The basic formulation for a deadbeat controller design is then derived giving the mathematical foundation of the recursive method. A recursive formula with computational steps is also included for real-time implementation. With a slight modification, the formula is extended to compute the feedforward gain for a measurable or predictable disturbance input. Finally, several numerical examples are given for illustration of the method.

2 Multi-Step Output Prediction[18]

The input/output relationship of a linear 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

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

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, α_i $(i = 1, 2, \cdots, p)$ of $m \times m$ and β_i $(i = 1, 2, \cdots, p)$ of $m \times r$ are commonly referred to as the observer Markov parameters (OMP) or ARX parameters. The matrix β_0 is the direct transmission term.

Equation (1) produces the following multi-step output prediction[18]

$$y_p(k+q) = \mathcal{T}' u_{p+q}(k) + \mathcal{B}' u_p(k-p) + \mathcal{A}' y_p(k-p) \tag{2}$$

where

$$
\begin{aligned}
y_p(k) &= \begin{bmatrix} y(k+q) \\ y(k+q+1) \\ \vdots \\ y(k+q+p-1) \end{bmatrix}, \quad u_{q+p}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+q+p-1) \end{bmatrix}, \\
y_p(k-p) &= \begin{bmatrix} y(k-p) \\ y(k-p+1) \\ \vdots \\ y(k-1) \end{bmatrix}, \quad u_p(k-p) = \begin{bmatrix} u(k-p) \\ u(k-p+1) \\ \vdots \\ u(k-1) \end{bmatrix}
\end{aligned}
\tag{3}
$$

and

$$
\begin{aligned}
\mathcal{T}' &= \begin{bmatrix}
\beta_0^{(q+1)} & \beta_0^{(q)} & \cdots & \beta_0^{(1)} & \beta_0 & 0 & \cdots & 0 \\
\beta_0^{(q+2)} & \beta_0^{(q+1)} & \cdots & \beta_0^{(2)} & \beta_0^{(1)} & \beta_0 & \cdots & 0 \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\beta_0^{(q+p-1)} & \beta_0^{(q+p-2)} & \cdots & \beta_0^{(p)} & \beta_0^{(p-1)} & \beta_0^{(p-2)} & \cdots & \beta_0
\end{bmatrix} \\
\mathcal{B}' &= \begin{bmatrix}
\beta_p^{(q)} & \beta_{p-1}^{(q)} & \cdots & \beta_1^{(q)} \\
\beta_p^{(q+1)} & \beta_{p-1}^{(q+1)} & \cdots & \beta_1^{(q+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\beta_p^{(q+p-1)} & \beta_{p-1}^{(q+p-1)} & \cdots & \beta_1^{(q+p-1)}
\end{bmatrix} \\
\mathcal{A}' &= \begin{bmatrix}
\alpha_p^{(q)} & \alpha_{p-1}^{(q)} & \cdots & \alpha_1^{(q)} \\
\alpha_p^{(q+1)} & \alpha_{p-1}^{(q+1)} & \cdots & \alpha_1^{(q+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_p^{(q+p-1)} & \alpha_{p-1}^{(q+p-1)} & \cdots & \alpha_1^{(q+p-1)}
\end{bmatrix}
\end{aligned}
\tag{4}
$$

The quantity $y_p(k+q)$ represents the output vector with a total of p data points for each sensor from the time step $k+q$ to $k+q+p-1$, whereas $y_p(k-p)$ includes the p data from $k-p$ to $k-1$. Similarly, $u_{q+p}(k)$ has $q+p$ 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 formed from the parameters, $\beta_0, \beta_0^{(1)}, \ldots,$ and $\beta_0^{(q+p-1)}$ (the pulse response sequence).

The vector $y_p(k+q)$ in Eq. (2) consists of three terms. The first term is the input vector $u_{q+p}(k)$ including inputs from time step k to $k+q+p-1$. 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 from time step $k-p$ to $k-1$, respectively. The future input vector $u_s(k)$ is to be determined for feedback control. The matrices $\mathcal{B}'$ and $\mathcal{A}'$ may be computed from observer Markov parameters α_i $(i = 1, 2, \cdots, p)$ and β_i $(i = 1, 2, \cdots, p)$, or directly from input and output data.

3 Deadbeat Predictive Control Designs

Several deadbeat control algorithms have been developed[18] using the multi-step output prediction, Eq.(2). Among these algorithms, the direct algorithm uses the input and output data directly without using α_i $(i = 1, 2, \cdots, p)$ and β_i $(i = 1, 2, \cdots, p)$ to first compute $\mathcal{B}'$ and $\mathcal{A}'$ and then design a deadbeat predictive controller. The goal was to make the direct algorithm suitable for real-time implementation in the sense that the deadbeat controller may be updated at every sampling interval. Unfortunately, it involves a matrix inverse that is difficult, if not impossible, to compute it recursively. To overcome the computational difficulty, an alternative algorithm is developed in this section.

Let $\mathcal{T}'$ be partitioned into two parts such that Eq.(2) becomes

$$y_p(k+q) = \mathcal{T}_o u_p(k+q) + \mathcal{T}_c u_q(k) + \mathcal{B}' u_p(k-p) + \mathcal{A}' y_p(k-p) \tag{5}$$

where

$$u_p(k+q) = \begin{bmatrix} u(k+q) \\ u(k+q+1) \\ \vdots \\ u(k+q+p-1) \end{bmatrix}, \quad u_q(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+q-1) \end{bmatrix} \tag{6}$$

and

$$\begin{aligned} \mathcal{T}_o &= \begin{bmatrix} \beta_0 & 0 & \cdots & 0 \\ \beta_0^{(1)} & \beta_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{(p-1)} & \beta_0^{(p-2)} & \cdots & \beta_0 \end{bmatrix} \\ \mathcal{T}_c &= \begin{bmatrix} \beta_0^{(q)} & \beta_0^{(q-1)} & \cdots & \beta_0^{(1)} \\ \beta_0^{(q+1)} & \beta_0^{(q)} & \cdots & \beta_0^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{(q+p-1)} & \beta_0^{(q+p-2)} & \cdots & \beta_0^{(p)} \end{bmatrix} \end{aligned} \tag{7}$$

Both $\mathcal{T}_o$ of $pm \times pr$ and $\mathcal{T}_c$ of $pm \times qr$ are formed from system 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 an integer. Given any input and output sequence $u(k)$ and $y(k)$, Eq. (5) must be satisfied and can be used for identifying coefficient matrices $\mathcal{T}_o$, $\mathcal{T}_c$, $\mathcal{A}'$, and $\mathcal{B}'$.

Note that the matrix $\mathcal{T}_c$ of $pm \times qr$ is a Hankel-like matrix[18] which has rank n where n is the order of the system. For the case where $qr \geq pm \geq n$, there are qr elements in $u_q(k)$ of $qr \times 1$ with only n independent equations in Eq. (5). As a result, Eq. (5) provide multiple solutions for $u_q(k)$ with the minimum-norm solution expressed by

$$u_q(k) = \mathcal{T}_c^{\dagger} y_p(k+q) - \mathcal{T}_c^{\dagger}\mathcal{T}_o u_p(k+q) - \mathcal{T}_c^{\dagger}\mathcal{B}' u_p(k-p) - \mathcal{T}_c^{\dagger}\mathcal{A}' y_p(k-p) \tag{8}$$

or in matrix form

$$u_q(k) = \begin{bmatrix} -\mathcal{T}_c^{\dagger}\mathcal{A}' & -\mathcal{T}_c^{\dagger}\mathcal{B}' & \mathcal{T}_c^{\dagger} & -\mathcal{T}_c^{\dagger}\mathcal{T}_o \end{bmatrix} \begin{bmatrix} y_p(k-p) \\ u_p(k-p) \\ y_p(k+q) \\ u_p(k+q) \end{bmatrix} \tag{9}$$

For the case where $qr = pm$, Eq. (9) is unique. To simplify Eq. (9), define the following notations

$$F_c = \begin{bmatrix} -\mathcal{T}_c^{\dagger}\mathcal{A}' & -\mathcal{T}_c^{\dagger}\mathcal{B}' \end{bmatrix} \text{ and } F_o = \begin{bmatrix} \mathcal{T}_c^{\dagger} & -\mathcal{T}_c^{\dagger}\mathcal{T}_o \end{bmatrix} \tag{10}$$

and

$$v_p(k-p) = \begin{bmatrix} y_p(k-p) \\ u_p(k-p) \end{bmatrix} \text{ and } v_p(k+q) = \begin{bmatrix} y_p(k+q) \\ u_p(k+q) \end{bmatrix} \tag{11}$$

where both F_c and F_o are $qr \times p(m+r)$ matrices, and both $v_p(k-p)$ and $v_p(k+q)$ are $(pm+pr) \times 1$ column vectors. Equation (9) thus becomes

$$u_q(k) = \begin{bmatrix} F_c & F_o \end{bmatrix} \begin{bmatrix} v_p(k-p) \\ v_p(k+q) \end{bmatrix} \tag{12}$$

Equation (12) is another form of finite difference model for system identification. For any given input and output data, there exists a set of F_c and F_o satisfying Eq. (12). Using Eq. (12) to develop a deadbeat controller is shown in the following.

Let us assume that the input vector $u_q(k)$ is chosen such that

$$u_q(k) = F_c v_p(k-p) \tag{13}$$

In order to satisfy Eq. (12), the following equation must hold

$$F_o v_p(k+q) = \begin{bmatrix} \mathcal{T}_c^{\dagger} & -\mathcal{T}_c^{\dagger}\mathcal{T}_o \end{bmatrix} \begin{bmatrix} y_p(k+q) \\ u_p(k+q) \end{bmatrix} = 0 \tag{14}$$

If $\mathcal{T}_c^{\dagger}$ of $qr \times pm$ is full rank pm with $qr \geq pm$, and $u_p(k+q)$ is set to zero, then $y_p(k+q)$ becomes zero. As a result, the control action $u_q(k)$ computed from Eq. (13) is a deadbeat controller which makes $y_p(k+q)$ to zero after q time steps. The integer q is thus called the deadbeat control horizon, whereas the integer p is commonly referred to as the system identification (or observer) horizon. The condition, $qr \geq pm$, means that the control action should not be faster than the state observation. That makes the physical sense.

In view of Eq. (6), the first r rows of $u_q(k)$ is the input vector $u(k)$ at time k. Define F_{c1} and F_{o1} as the first r rows of F_c and F_o, respectively. The control action at $u(k)$ should be

$$\begin{aligned} u(k) &= [\text{the first } r \text{ rows of } F_c]\, v_p(k-p) \\ &= F_{c1} v_p(k-p) \end{aligned} \tag{15}$$

where F_{c1} is the control gain matrix to be determined.

The first r rows of Eq. (12) produces

$$\begin{aligned} u(k) &= \left(\text{the first } r \text{ rows of } \begin{bmatrix} F_c & F_o \end{bmatrix}\right) \begin{bmatrix} v_p(k-p) \\ v_p(k+q) \end{bmatrix} \\ &= \begin{bmatrix} F_{c1} & F_{o1} \end{bmatrix} \begin{bmatrix} v_p(k-p) \\ v_p(k+q) \end{bmatrix} \end{aligned} \tag{16}$$

Equation (16) indicates that the input $u(k)$ is related to the past input sequence $u(k-p)$ to $u(k-1)$ and output sequence $y(k-p)$ to $y(k-1)$, and future input sequence $u(k+q)$ to $u(k+q+p-1)$ and output sequence $y(k+q)$ to $y(k+q+p-1)$. There is a total of q time steps gap from k to $k+q-1$. A different integer q produces a different set of coefficient matrices F_{c1} and F_{o1} that satisfies Eq. (16).

To solve Eq. (16), let us first form the following matrices

$$\begin{aligned}
U(k) &= \begin{bmatrix} u(k) & u(k+1) & \cdots & u(N-p-q+1) \end{bmatrix} \\
V_p(k-p) &= \begin{bmatrix} y(k-p) & y(k-p+1) & \cdots & y(N-2p-q+1) \\ u(k-p) & u(k-p+1) & \cdots & u(N-2p-q+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(k-1) & y(k) & \cdots & y(N-p-q) \\ u(k-1) & u(k) & \cdots & u(N-p-q) \end{bmatrix} \qquad (17) \\
V_p(k+q) &= \begin{bmatrix} y(k+q) & y(k+q+1) & \cdots & y(N-p+1) \\ u(k+q) & u(k+q+1) & \cdots & u(N-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(k+q+p-1) & y(k+q+p) & \cdots & y(N) \\ u(k+q+p-1) & u(k+q+p) & \cdots & u(N) \end{bmatrix}
\end{aligned}$$

where N is the data length used for estimation of coefficient matrices F_{c1} and F_{o1}. Application of Eq. (16) thus yields

$$U(k) = \begin{bmatrix} F_{c1} & F_{o1} \end{bmatrix} \begin{bmatrix} V_p(k-p) \\ V_p(k+q) \end{bmatrix} \qquad (18)$$

Let the integer N be chosen large enough such that the matrix $U(k)$ of $r \times (N-p-q-k+2)$ has rank r, $V_p(k-p)$ of $p(r+m) \times (N-p-q-k+2)$ and $V_p(k+q)$ of $p(r+m) \times (N-p-q-k+2)$ have rank $pr+n$ where n is the order of the system. Again, r means the number of inputs and m represents the number of outputs. Equation (18) produces the following least-squares solution

$$\begin{bmatrix} F_{c1} & F_{o1} \end{bmatrix} = U(k) \begin{bmatrix} V_p(k-p) \\ V_p(k+q) \end{bmatrix}^{\dagger} \qquad (19)$$

where † means the pseudo-inverse.

4 Recursive Least-Squares Algorithm

There are many recursive algorithms[2] available to solve the least-squares problem. The classical least-squares method is the most straightforward approach and is also the basis for the others. The classical recursive method is briefly described here.

Equation (16) can be written in a compact matrix form

$$\begin{aligned} u(k) &= \begin{bmatrix} F_{c1} & F_{o1} \end{bmatrix} \begin{bmatrix} v_p(k-p) \\ v_p(k+q) \end{bmatrix} \\ &= F\bar{v}_p(k-1) \end{aligned} \tag{20}$$

where

$$F = \begin{bmatrix} F_{c1} & F_{o1} \end{bmatrix},\ v(k) = \begin{bmatrix} y(k) \\ u(k) \end{bmatrix},\ \text{and}\ \ \bar{v}_p(k-1) = \begin{bmatrix} v(k-p) \\ \vdots \\ v(k-1) \\ v(k+q) \\ \vdots \\ v(k+q+p-1) \end{bmatrix} \tag{21}$$

First, define the following quantities

$$G_p(k) = \frac{\bar{v}_p^T(k)P_p(k-1)}{1+\bar{v}_p^T(k)P_p(k-1)\bar{v}_p(k)} \tag{22}$$

$$\hat{u}(k+1) = \widehat{F}(k)\bar{v}_p(k) \tag{23}$$

Next, compute the following quantities

$$P_p(k) = P_p(k-1)[I-\bar{v}_p(k)G_p(k)] \tag{24}$$

$$\widehat{F}(k+1) = \widehat{F}(k)+[u(k+1)-\hat{u}(k+1)]G_p(k) \tag{25}$$

Equations (22) (25) constitute the fundamental Recursive Least-Squares (RLS) formulation for identifying the gain matrix F including F_{c1} for the deadbeat controller design and F_{o1} for the need of system identification. The initial values of $P_p(0)$ and $\bar{Y}_p(1)$ can be either assigned or obtained by performing a small batch least-squares after gathering a sufficient number of data. If initial values are to be chosen, $P_p(0)$ and $\widehat{F}(1)$ can be assigned as $dI_{2p(r+m)}$ and $0_{r\times 2p(r+m)}$, respectively, where d is a large positive number. The positive constant d is the only parameter required for the initialization. The proper choice of d is based on practical experience. The initialization introduces a *bias* into the parameter estimate $\widehat{F}(k)$ produced by the recursive least-squares method. For large data lengths, the exact value of the initialization constant is not important.

4.1 Computational Steps after initialization

The computational steps for the recursive deadbeat control method are summarized in the following.

1. Form the vector $\bar{v}_p(k)$ as shown in Eq. (21) with the new input $v(k+q+p-1)$ as the last $r+m$ rows.

2. Compute the gain vector $G_p(k)$ by inserting $P_p(k-1)$ and $\bar{v}_p(k)$ in Eq. (22). In this step, one should compute $\bar{v}_p^T(k)P_p(k-1)$ first and then use the result to calculate $[\bar{v}_p^T(k)P_p(k-1)]\bar{v}_p(k)$.

3. Compute the estimated output $\hat{u}(k+1)$ by substituting $\widehat{F}(k)$ and $\bar{v}_p(k)$ into Eq. (23).

4. Update $P_p(k-1)$ to obtain $P_p(k)$ with $\bar{v}_p(k)$ formed from the first step and $G_p(k)$ computed from the second step.

5. Update $\widehat{F}(k)$ to obtain $\widehat{F}(k+1)$ from Eq. (25) with the input signal $u(k+1)$, the estimated input $\hat{u}(k+1)$, and the computed gain $G_p(k)$.

No matrix inverse is involved in these computational steps. Updating $P_p(k)$ and $G_p(k)$ takes more time than computing other quantities. The recursive procedure derived for updating the least-squares solution $\widehat{F}(k)$ is very general in the sense that it is valid for any linear equation such as Eq. (19).

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. Another type of disturbances is not known but its correlation is known. This section addresses the predictive feedback designs including feedforward from the disturbance inputs that are measurable or predictable..

With the disturbance input involved, the multi-step output prediction equation becomes

$$y_p(k+q) = \mathcal{T}_c u_{cq}(k) + \mathcal{T}_d' u_{d(q+p)}(k) + \mathcal{T}_o u_{cp}(k+q) + \mathcal{B}_c' u_{cp}(k-p) + \mathcal{B}_d' u_{dp}(k-p) + \mathcal{A}' y_p(k-p) \tag{26}$$

where

$$\begin{aligned} y_p(k) &= \begin{bmatrix} y(k+q) \\ y(k+q+1) \\ \vdots \\ y(k+q+p-1) \end{bmatrix}, \quad u_{d(q+p)}(k) = \begin{bmatrix} u_d(k) \\ u_d(k+1) \\ \vdots \\ u_d(k+q+p-1) \end{bmatrix} \\ u_{cp}(k+q) &= \begin{bmatrix} u_c(k+q) \\ u_c(k+q+1) \\ \vdots \\ u_c(k+q+p-1) \end{bmatrix}, \quad u_{cq}(k) = \begin{bmatrix} u_c(k) \\ u_c(k+1) \\ \vdots \\ u_c(k+q-1) \end{bmatrix} \end{aligned} \tag{27}$$

The subscripts c and d signify the quantities related to the control force and the disturbance force, respectively. The form of the matrix $\mathcal{T}_d'$ associated with the disturbances u_d is similar to $\mathcal{T}'$ defined in Eq. 4. The matrix $\mathcal{T}_c'$ 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. The forms of $\mathcal{B}_c'$ and $\mathcal{B}_d'$ are also similar but corresponding to different type of forces. Note that $\mathcal{T}_c, \mathcal{T}_o$,and $\mathcal{B}_c'$ are quantities associated with the control force $u_c(k)$.

A similar equation to Eq. (8) can thus be derived as

$$\begin{aligned} u_{cq}(k) &= \mathcal{T}_c^\dagger y_p(k+q) - \mathcal{T}_c^\dagger \mathcal{T}_o u_{cp}(k+q) - \mathcal{T}_c^\dagger \mathcal{T}_d' u_{d(q+p)}(k) \\ &\quad -\mathcal{T}_c^\dagger \mathcal{A}' y_p(k-p) - \mathcal{T}_c^\dagger \mathcal{B}_c' u_{cp}(k-p) - \mathcal{T}_c^\dagger \mathcal{B}_d' u_{dp}(k-p) \end{aligned} \tag{28}$$

or in matrix form

$$u_{cq}(k) = \begin{bmatrix} -\mathcal{T}_c^\dagger \mathcal{A}' & -\mathcal{T}_c^\dagger \mathcal{B}_c' & -\mathcal{T}_c^\dagger \mathcal{B}_d' & \mathcal{T}_c^\dagger & -\mathcal{T}_c^\dagger \mathcal{T}_o & -\mathcal{T}_c^\dagger \mathcal{T}_d' \end{bmatrix} \begin{bmatrix} y_p(k-p) \\ u_{cp}(k-p) \\ u_{cd}(k-p) \\ y_p(k+q) \\ u_{cp}(k+q) \\ u_{d(q+p)}(k) \end{bmatrix} \tag{29}$$

Define the following notations

$$F_c' = \begin{bmatrix} -\mathcal{T}_c^\dagger \mathcal{A}' & -\mathcal{T}_c^\dagger \mathcal{B}_c' & -\mathcal{T}_c^\dagger \mathcal{B}_d' \end{bmatrix},\ F_{co}' = \begin{bmatrix} \mathcal{T}_c^\dagger & -\mathcal{T}_c^\dagger \mathcal{T}_o \end{bmatrix},\ \text{and } F_d' = -\mathcal{T}_c^\dagger \mathcal{T}_d' \tag{30}$$

and

$$v_p(k-p) = \begin{bmatrix} y_p(k-p) \\ u_{cp}(k-p) \\ u_{cd}(k-p) \end{bmatrix} \quad \text{and } v_{cp}(k+q) = \begin{bmatrix} y_p(k+q) \\ u_{cp}(k+q) \end{bmatrix} \tag{31}$$

where F_c' is a $qr_c \times p(m+r_c+r_d)$ matrix but F_{co}' is a $qr_c \times p(m+r_c)$ matrix. The quantity $v_p(k-p)$ is a $p(m+r_c+r_d) \times 1$ column vector whereas $v_{cp}(k+q)$ is a $p(m+r_c) \times 1$ column vector. Equation (29) thus becomes

$$u_{cq}(k) = \begin{bmatrix} F'_c & F'_{co} & F'_d \end{bmatrix} \begin{bmatrix} v_p(k-p) \\ v_{cp}(k+q) \\ u_{d(q+p)}(k) \end{bmatrix} \tag{32}$$

For any given input and output data, there exists a set of F'_c, F'_{co}, and F'_d satisfying Eq. (32).

Let us assume that the input vector $u_{cq}(k)$ is chosen such that

$$u_{cq}(k) = F'_c v_p(k-p) \tag{33}$$

From Eq. (32), the output after q time steps is then governed vy

$$F'_{co} v_{cp}(k+q) + F'_d u_{d(q+p)}(k) = 0$$

or, from Eq. (29),

$$\mathcal{T}_c^\dagger y_p(k+q) = \mathcal{T}_c^\dagger \mathcal{T}_o u_{cq}(k+q) + \mathcal{T}_c^\dagger \mathcal{T}'_d u_{d(q+p)}(k) \tag{34}$$

Here we have assumed that the disturbance is measurable. If the disturbance is not predictable or measurable, Eq. (33) is not valid. Equation (34) indicates that the output vector from time $k+q$ is generated by the control vector from $k+q$ and the disturbance vector from k. If the disturbance $u_d(k)$ such as the random signal is not predictable, then we cannot use the disturbance signal for a feedforward design for the control force $u(k)$ at time k. As a result, the feedforward design included in the control law, Eq. (33), is the only way that can be implemented, because it uses only the disturbance signal before time k. From Eq. (33), the control action $u(k)$ at time k is

$$\begin{aligned} u(k) &= (\text{the first } r \text{ rows of } F'_c)\, v_p(k-p) \\ &= F'_{c1} v_p(k-p) \end{aligned} \tag{35}$$

where F'_{c1} is the control gain matrices to be determined.

The first r rows of Eq. (32) produces

$$\begin{aligned} u(k) &= \left(\text{the first } r \text{ rows of } \begin{bmatrix} F'_c & F'_{co} & F'_d \end{bmatrix}\right) \begin{bmatrix} v_p(k-p) \\ v_{cp}(k+q) \\ u_{d(q+p)}(k) \end{bmatrix} \\ &= \begin{bmatrix} F'_{c1} & F'_{co1} & F'_{d1} \end{bmatrix} \begin{bmatrix} v_p(k-p) \\ v_{cp}(k+q) \\ u_{d(q+p)}(k) \end{bmatrix} \end{aligned} \tag{36}$$

A similar equation to Eq. (19) can then be obtained

$$\begin{bmatrix} F'_{c1} & F'_{co1} & F'_{d1} \end{bmatrix} = U(k) \begin{bmatrix} V_p(k-p) \\ V_{cp}(k+q) \\ U_{d(q+p)}(k) \end{bmatrix}^{\dagger} \tag{37}$$

where

$$\begin{aligned} U(k) &= \begin{bmatrix} u_c(k) & u_c(k+1) & \cdots & u_c(N-p-q+1) \end{bmatrix} \\ V_p(k-p) &= \begin{bmatrix} y(k-p) & y(k-p+1) & \cdots & y(N-2p-q+1) \\ u_c(k-p) & u_c(k-p+1) & \cdots & u_c(N-2p-q+1) \\ u_d(k-p) & u_d(k-p+1) & \cdots & u_d(N-2p-q+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(k-1) & y(k) & \cdots & y(N-p-q) \\ u_c(k-1) & u_c(k) & \cdots & u_c(N-p-q) \\ u_d(k-1) & u_d(k) & \cdots & u_d(N-p-q) \end{bmatrix} \\ V_{cp}(k+q) &= \begin{bmatrix} y(k+q) & y(k+q+1) & \cdots & y(N-p+1) \\ u_c(k+q) & u_c(k+q+1) & \cdots & u_c(N-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(k+q+p-1) & y(k+q+p) & \cdots & y(N) \\ u_c(k+q+p-1) & u_c(k+q+p) & \cdots & u_c(N) \end{bmatrix} \\ U_{d(q+p)}(k) &= \begin{bmatrix} u_d(k) & u_d(k+1) & \cdots & u_d(N-p-q+1) \\ \vdots & \vdots & \ddots & \vdots \\ u_d(k+q+p-1) & u_d(k+q+p) & \cdots & u_d(N) \end{bmatrix} \end{aligned}$$

Equation (37) can also be computed recursively.

6 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

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

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

and $m_i, k_i, \zeta_i, i = 1, 2, 3$ 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 and single-output case where the control input to the system is the force on the first mass (i.e., $u_c = u_1$), 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. Note that this is a non-minimum phase system.

Consider the case where the controller is computed with $q = 6$. Let the initial guess for $P_p(0)$ and $\widehat{F}(1)$ shown in Eqs. (22) and (23) be $1000 I_{24\times 24}$ and $0_{1\times 24}$, respectively. The input signal is a sequence of normally distributed random numbers with zero mean and unit variance. Let the control action be turned on at the data point 18. In other words, we wait no time to close the system loop as soon as the first nonzero vector $\bar{v}_p(k)$ defined in Eq. (21) is formed. Figure 1 shows the open-loop and closed-loop histories of input and output. The solid curve is the open-loop response and the dashed curve is the closed-loop response. The control gain starts with a zero vector and ends with the controller

$$\begin{aligned} u_c(k) \quad = \quad & -0.3848u_c(k-1) + 0.7217u_c(k-2) + 0.2536u_c(k-3) \\ & -0.0681u_c(k-4) - 0.0150u_c(k-5) + 0.0000u_c(k-6) \\ & -0.9828y(k-1) - 0.7304y(k-2) - 1.2904y(k-3) \\ & -0.1818y(k-4) + 0.1908y(k-5) - 0.0007y(k-6) \end{aligned}$$

The controller converges to the one using the batch approach shown in Eq. (19) without any control action.

Let the output be added with some measurement noise so that the signal to noise ratio is 11.4 (i.e., the output norm divided by the noise norm). The noise is random normally distributed with zero mean. Set the values of p and q to the same as that for deadbeat, i.e., $p = q = 6$. The open-loop and closed-loop time histories are shown in Fig. 2. The control

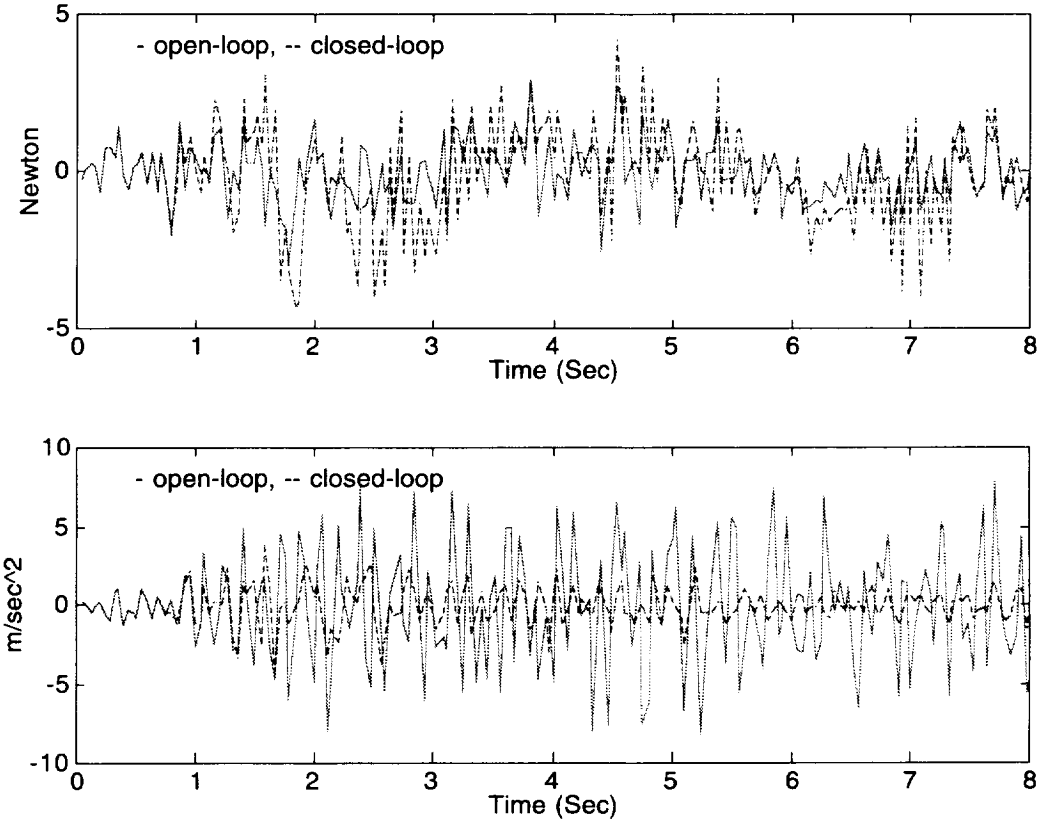

Figure 1: Input and output time histories

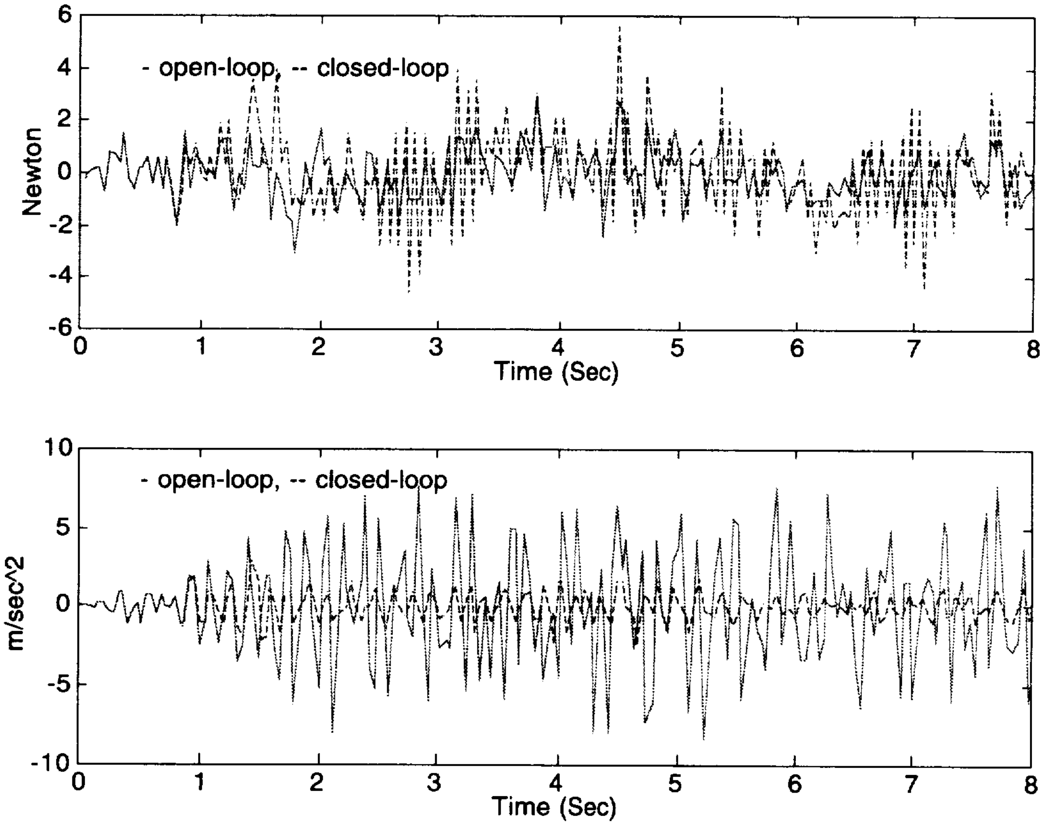

Figure 2: Input and output time histories

gain starts with a zero vector and ends with the controller

$$\begin{aligned}u_c(k) \;=\; & -0.1208u_c(k-1) + 0.6969u_c(k-2) - 0.02486u_c(k-3) \\ & -0.0780u_c(k-4) + 0.0704u_c(k-5) + 0.0762u_c(k-6) \\ & -0.5724y(k-1) - 0.3321y(k-2) - 0.7017y(k-3) \\ & +0.0593y(k-4) + 0.2075y(k-5) - 0.1076y(k-6)\end{aligned}$$

Although the final control gain looks considerably different from the one shown earlier for the noise-free case. The performance for both noise-free and noisy cases is quite similar. Increasing the value of q does not seem to improve the performance. In some cases with q larger than p, the performance is worse than the case with $p = q$ when the control action is turned on too early. Given sufficient time steps for the control gain to converge to a reasonable level, the performance may be improved somewhat particularly for the case where $q > p$. One may raise the question whether the recursive controller design works for the case where the order of the controller is smaller than the order of the system, i.e., p is smaller than 6 for this example. Let us choose $p = q = 4$. Using the same set of data for the previous case, the input and output time histories are shown in Fig. 3. Obviously, the performance is not as good as the ones shown earlier but it is still stable and somewhat acceptable. In practice, the order of a system is unknown and thus there is a great possibility that the values of p may be smaller than the true one.

7 Concluding Remarks

A new recursive predictive control method has been presented in this paper. System identification was reformulated in such a way that it fits better for predictive controller designs. In other words, the conventional thinking in system identification has been re-oriented to focus on the control design process. The conventional procedure for any controller designs includes two steps, i.e., first perform system identification within an acceptable level of ac-

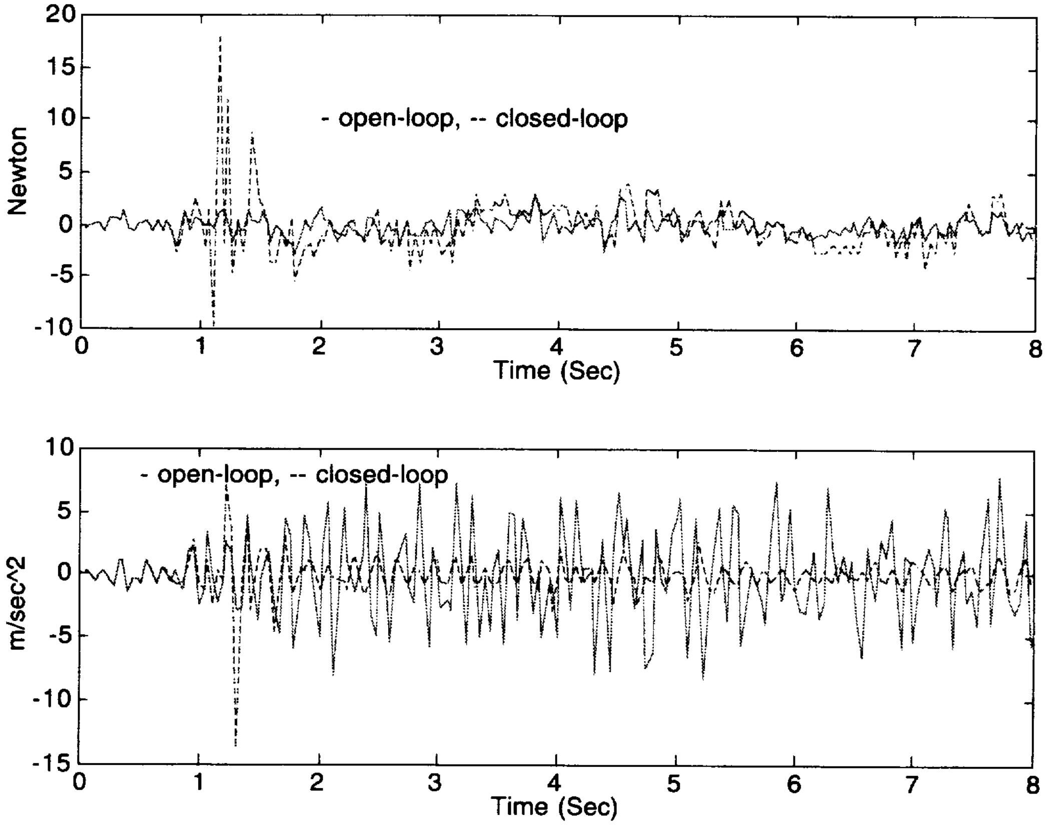

Figure 3: Input and output time histories

curacy and then conduct a controller design. The error in system identification will likely be accumulated and carried through the controller design process. As a result, the conventional approach tends to introduce more error in the controller design than system identification itself. The method derived in this paper uses a different approach for system identification to eliminate the additional controller design process. For noise-free cases, both conventional and new approaches produce an identical result if the predictive controller is unique. For non-unique controllers, the new approach provides the control gain smaller in norm than that from the conventional approach. This is due to the fact that, instead of minimizing the output error, the new approach minimizes the input error to compute the control gain. The proposed recursive method has a considerable advantage of computational speed. For noisy cases, numerical simulations have showed that the new method is more robust than the other methods.

8 References

1. Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, New York, 1972.

2. Kailath, T., Linear Systems,Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

3. Juang, J. N., Lim, K. B., and Junkins, J. L., "Robust Eigensystem Assignment for Flexible Structures," *Journal of Guidance, Control and Dynamics,* Vol. 12, No. 3, May-June 1989, pp. 381-387.

4. Juang, J. N., and Phan, M., "Robust Controller Designs for Second-Order Dynamic Systems: A Virtual Passive Approach", *Journal of Guidance, Control and Dynamics,* Vol. 15, No. 5, Sept-Oct. 1992, pp. 1192-1198.

5. Morris, K. A. and Juang, J. N., "Dissipative Controller Designs for Second-Order Dynamic Systems," *IEEE Transactions on Automatic Control,* Vol. 39, No. 5, May 1994, pp. 1056-1063.

6. Goodwin, G.C. and Sin, K.S., Adaptive Filtering, Prediction, and Control, Prentice-Hall, New Jersey, 1984.

7. Ydstie, B.E., "Extended Horizon Adaptive Control," *Proceedings of the 9th IFAC World Congress,* Vol. VII, pp. 133-138, 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, New Jersey, 1992.

10. 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.

11. 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.

12. Peterka, V., "Predictor-Based Self-Tuning Control," *Automatica,* Vol. 20, No. 1, pp. 39-50, 1984.

13. 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.

14. 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.

15. Mosca, E., Optimal, Predictive, and Adaptive Control, Prentice-Hall, New Jersey, 1995.

16. Bialasiewicz, J.T., Horta, L.G., and Phan, M., "Identified Predictive Control," *Proceedings of the American Control Conference,* Baltimore, Maryland, June 1994.

17. Phan, M. G., and Juang, J.-N., "Predictive Feedback Controllers for Stabilization of Linear Multivariable Systems," *Journal of Guidance, Control, and Dynamics,* To appear.

18. Juang, J.-N. and Phan, M.G., "Deadbeat Predictive Controllers," *NASA Technical Memorandum,* TM-112862, May 1997.

19. Kida, T., Ohkami, Y., and Sambongi, S., "Poles and Transmission Zeros of Flexible Spacecraft Control Systems," *Journal of Guidance, Control, and Dynamics,* Vol. 8, No. 2, Mar.-Apr., 1985, pp. 208-213.

20. Williams, T., "Transmission Zero Bounds for Large Structures, with Applications," *Journal of Guidance, Control, and Dynamics,* Vol. 12, No.1, Jan.-Feb., 1989, pp. 33-38.

21. Williams, T., and Juang, J. N.,"Sensitivity of the Transmission Zeros of Flexible Space Structures, *Journal of Guidance, Control and Dynamics,* Vol. 15, No.2, Mar.-Apr. 1992, pp. 368-375.

22. Williams, T., and Juang, J. N.,"Pole/Zero Cancellations in Flexible Space Structures," *Journal of Guidance, Control, and Dynamics,* Vol. 13, No. 4, July-Aug. 1990, pp. 684-690.

23. Williams, T., "Model Order Effects on the Transmission Zeros of Flexible Space Structures," *Journal of Guidance, Control, and Dynamics,* Vol. 15, No. 2, Mar.-Apr., 1992, pp. 540-543.

24. Williams, T., "Transmission Zeros of Non-Collocated Flexible Structures: Finite-Dimensional Effects," *AIAA Dynamics Specialists Conference,* Dallas, Texas, April 1992.

25. Lin, J.-L., and Juang, J.-N., "Sufficient Conditions for Minimum-Phase Second-Order Linear Systems, *Journal of Vibration and Control,* Vol. 1 No.2, May 1995, pp. 183-199.

26. 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.

27. 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.

28. Juang, J.-N., Applied System Identification, Prentice-Hall, 1994.

29. 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.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE May 1997	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE
Recursive Deadbeat Controller Design

5. FUNDING NUMBERS
WU 623-20-21-11

6. AUTHOR(S)
Jer-Nan Juang
Minh Q. Phan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM-112863

11. SUPPLEMENTARY NOTES
Juang: NASA Langley Research Center, Hampton, VA; Phan: Princeton University, Princeton, NJ

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 39
Availability: NASA CASI, (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This paper presents a recursive algorithm for a deadbeat predictive controller design. The method combines together the concepts of system identification and deadbeat controller designs. It starts with the multi-step output prediction equation and derives the control force in terms of past input and output time histories. The formulation thus derived satisfies simultaneously system identification and deadbeat controller design requirements. As soon as the coefficient matrices are identified satisfying the output prediction equation, no further work is required to compute the deadbeat control gain matrices. The method can be implemented recursively just as any typical recursive system identification techniques.

14. SUBJECT TERMS
Feedback Control, Active Control, System Identification, Acoustic Noise Reduction

15. NUMBER OF PAGES
27

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT