JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 19, No. 2, March–April 1996

Designs of Learning Controllers Based on Autoregressive Representation of a Linear System

Minh Q. Phan*
Princeton University, Princeton, New Jersey 08544
and
Jer-Nan Juang†
NASA Langley Research Center, Hampton, Virginia 23681

Learning controllers that improve tracking performance through repeated trials are derived. The design is based on an autoregressive representation of a linear system. This input–output model can be interpreted in terms of an observer in state-space form. The control input is modified at every repetition as the system learns to produce a desired response, even in the presence of unknown repetitive disturbances. The coefficients of a nominal autoregressive model are first identified from input–output data. Using the identified coefficients, simple linear feedback learning controllers are designed that can correct for the errors that remain. An optimal learning gain matrix is also derived given the identified model. Numerical examples are provided to illustrate the proposed learning approach.

Introduction

WHEN a system is required to perform a certain task, it is rarely true that it will deliver the desired performance exactly. The difference between the desired performance and the actual performance is known as mistakes or errors. When asked to perform the same task again, it is likely to repeat the same mistakes. Often, the mistakes will be repeated very accurately because of high repeatability in modern hardware. Such examples can be found in various manufacturing processes, where it seems a bit primitive to allow the same mistakes to be repeated over and over again. A learning controller can be implemented to remove this type of error. During each repetitive operation, there may be unknown repetitive disturbances present, and the learning control system should be able to correct for these errors as well. Learning control can be thought of as a method to produce the desired response through successive correction of the command, which may be supplied by an existing controller. This learning from one repetition to the next distinguishes the concept of learning from adaptive control. A substantial body of results in learning control has been established, for example, see Refs. 1–14, which propose various forms of learning controllers for dynamic systems and prove their convergence properties in the tracking control problem. Possible aerospace applications of learning control include precision scanning motion of satellite-mounted instruments, rejection of disturbances caused by control moment gyros on fine pointing spacecraft, or repetitive slewing control of a satellite-mounted solar panel.

Although learning control was originally developed for nonlinear systems such as robots, it is both of theoretical and practical interest to understand how the theory handles linear systems. A formulation of learning control is presented in Ref. 7 using the modern discrete-time state-space variable approach. Experimental validation of the proposed formulation is shown in Ref. 14. The theory is applicable to both linear time-invariant and time-varying systems. Linear feedback forms of the learning controllers are derived with explicit stability conditions for convergence of the learning process. The proportional learning law, which is one of the most popular and most successful learning laws, appears as a special case among several permitted choices. The formulation shows that linear time-varying systems viewed in the repetition domain appear as repetition-invariant, and this fact allows time-varying systems to be treated with the more complete theory associated with time-invariant systems. It also shows how the effect of unknown repetitive disturbances can be easily eliminated by learning control. Furthermore, for a particular choice of the learning controller gain, the formulation reveals the role of the Markov parameters in determining the stability of the learning process. Knowledge of the Markov parameters also allows one to design optimal learning laws. In system identification it is well known that the Markov parameters are the system pulse response samples from which a state-space representation can be constructed. For this reason, the problem of determining the Markov parameters from experimental input–output data is also an important one in the field of system identification.

Recent advances in system identification led to the development of the observer Markov parameters that offer several fundamental advantages over the traditional system Markov parameters as a way to represent a linear system.[15,16] First, a linear system can be described by a finite set of observer Markov parameters regardless of its stability and/or initial conditions. Second, the observer Markov parameters can be factored to obtain not only a system model but also an associated observer gain that can be used for modern control application. Third, unlike the Markov parameters that may exhibit very slow decay for a lightly damped system, the observer Markov parameters can be easily computed from general input–output data. In this paper, the advantages of the observer Markov parameters as seen in the system identification problem are carried over to the formulation and design of learning controllers. The extent to which system identification is involved in this formulation of learning control is limited to the identification of the observer Markov parameters from input–output data. If desired, further extraction of a state-space model and an associated observer from these parameters is possible, but this step is not necessary for learning control purposes.

The basic outline of this paper is as follows. First, an input–output description of the system is described in the repetition domain, which offers a convenient way of viewing the learning control problem. The role of the system Markov parameters and the observer Markov parameters in this process will be explained. Second, linear forms of learning laws based on the observer Markov parameters will be described, and stability conditions for the asymptotic convergence of the learning process are derived. The observer Markov parameters can be easily identified from input–output data, and this computation will be described. Third, an optimal learning gain is

Received July 12, 1993; revision received Dec. 12, 1994; accepted for publication Dec. 20, 1994.

*Assistant Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.

†Principal Scientist, Structural Dynamics Branch, Mail Stop 297. Fellow AIAA.

derived that minimizes a particular quadratic cost function involving the tracking error and the amount of allowed correction in the control input history at each repetition. Finally, numerical examples are used to illustrate and discuss the proposed design approach.

Time-Domain Representation

In modern control theory, a discrete state-space model is used to describe the input–output relationship of a linear dynamic system. Through a mathematical construct called an observer, it is shown in the following that the same input–output map can also be described by an observer system that possesses certain interesting properties. The relationship between the Markov parameters of the system and its associated observer is also described in this section.

State-Space Model

The input–output relationship is expressed in terms of a set of n simultaneous first-order difference equations of the form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w_1(k) \\ y(k) &= Cx(k) + Du(k) + w_2(k) \end{aligned} \tag{1}$$

where the dimensions of A, B, C, and D are $n \times n$, $n \times m$, $q \times n$, and $q \times m$, respectively. The vectors $u(k)$ and $y(k)$ represent the input to the system and the output of the system, respectively. The vector $x(k)$ represents the state of the system, and $w_1(k)$ and $w_2(k)$ are the process and measurement disturbances, respectively. The matrix D is known as the direct transmission term. Typically, this term is present if the output vector involves acceleration measurements. Solving for the output $y(k)$ in terms of the inputs and disturbances yields

$$\begin{aligned} y(k) &= CA^k x(0) + \sum_{i=1}^{k} CA^{i-1}Bu(k-i) + Du(k) \\ &+ \sum_{i=1}^{k} CA^{i-1}w_1(k-i) + w_2(k) \end{aligned} \tag{2}$$

for $k = 1, 2, \ldots$. The parameters, $Y(0) = D$, $Y(k) = CA^{k-1}B$, $k = 1, 2, \ldots$, are called the system Markov parameters of the system described by Eq. (1). The role of the system Markov parameters in describing the input–output relationship of a linear system is well known. In the following, the use of an observer model for this task is briefly described.

Observer Model

The system input–output relation can also be described by a set of state-space equations derived as follows. Adding and subtracting the term $My(i)$ to the right-hand side of the state equation in Eq. (1) yields

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + My(k) - My(k) + w_1(k) \\ &= (A + MC)x(k) + (B + MD)u(k) - My(k) \\ &\quad + Mw_2(k) + w_1(k) \end{aligned} \tag{3}$$

If M is a matrix such that $A + MC$ is deadbeat of order p, that is, $(A + MC)^k \equiv 0$, $k \geq p$, then the relationship between the input $u(k)$ and output $y(k)$ can be described as

$$\begin{aligned} y(k) + \sum_{i=1}^{p} C(A+MC)^{i-1}My(k-i) &= C(A+MC)^k x(0) \\ &+ \sum_{i=1}^{p} C(A+MC)^{i-1}(B+MD)u(k-i) + Du(k) \\ &+ \sum_{i=1}^{p} C(A+MC)^{i-1}[Mw_2(k-i) + w_1(k-i)] + w_2(k) \end{aligned} \tag{4}$$

In the absence of disturbances, the role of the matrix M in the preceding development can be interpreted as an observer gain. To see this, consider the system given in Eq. (1) without the disturbance terms. It has an observer of the form

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) - M[y(k) - \hat{y}(k)] \\ &= (A + MC)\hat{x}(k) + (B + MD)u(k) - My(k) \end{aligned} \tag{5}$$

where $\hat{y}(k) = C\hat{x}(k) + Du(k)$. Defining the state estimation error $e(k) = x(k) - \hat{x}(k)$, the equation that governs $e(k)$ is $e(k+1) = (A + MC)e(k)$. For an observable system, the matrix M may be chosen to place the eigenvalues of $A + MC$ in any desired (symmetric) configuration. If the matrix M is such that $A + MC$ is asymptotically stable, then the estimated state $\hat{x}(k)$ tends to the true state $x(k)$ as k tends to infinity. From this analysis, the matrix M can be interpreted as an observer gain. The parameters defined as

$$\begin{aligned} \bar{Y}(0) &= D \\ \bar{Y}(k) &= C(A + MC)^{k-1}[B + MD, -M], \qquad k = 1, 2, 3, \ldots \end{aligned} \tag{6}$$

are the Markov parameters of an observer system; hence, they are referred to as observer Markov parameters.

Note that if M is a deadbeat observer gain, then the observer Markov parameters become identically zero after a finite number of terms. This deadbeat observer gain matrix is the one used to derive Eq. (4). At this point, it is important to emphasize that we are not concerned with finding the deadbeat observer gain M given A and C, but rather the existence of such a gain matrix for the system described by Eq. (1) so that Eq. (4) can be justified. For any observable system, the existence of such a deadbeat gain matrix is always assured because the deadbeat gain matrix simply places all of the eigenvalues of $\bar{A} = (A + MC)$ at the origin in the complex plane. Therefore, the system can be described by a finite number of observer Markov parameters. The implication of this fact for a lightly damped system is that the system can be described by a relatively small number of observer Markov parameters $\bar{Y}(k)$, instead of an otherwise large number of the usual system Markov parameters $Y(k)$. For this reason, the observer Markov parameters are useful in the representation of a linear system. As later shown, the observer Markov parameter combinations that include A, B, C, D, and M are directly computed from input–output data. This paper will show how this can be accomplished and how the observer Markov parameters can be used in the design of various learning controllers.

The system Markov parameters are related to the observer Markov parameters via a simple relationship, which is given as

$$Y(k) = \bar{Y}^{(1)}(k) + \sum_{i=1}^{k} \bar{Y}^{(2)}(i)Y(k-i), \qquad Y(0) = \bar{Y}(0) \tag{7}$$

where

$$\begin{aligned} \bar{Y}^{(1)}(k) &= C(A + MC)^{k-1}(B + MD) \\ \bar{Y}^{(2)}(k) &= -C(A + MC)^{k-1}M \end{aligned}$$

where $\bar{Y}(k) \equiv 0$ for $k \geq p$. To describe a system of order n, the number of observer Markov parameters p must be such that $qp \geq n$ where q is the number of outputs. Furthermore, the maximum order of a system that can be described with p observer Markov parameters is qp. Therefore, for multiple-output systems, the number of observer Markov parameters required can be less than the true order of the system. Specifically, the minimum number of observer Markov parameters that can describe the system is $p_{\min}$, which is the smallest value of p such that $qp_{\min} \geq n$ (Ref. 16).

Repetition-Domain Representation

For learning control, it is advantageous to describe the system behavior in the repetition domain. Basically, the repetition-domain representation describes how a change in the control input from one repetition to the next will affect the system response from one repetition to the next. Appropriate learning laws can then be devised to make the system learn to produce the desired response. A backward difference operator in the repetition domain is defined that permits the conversion of the time-domain equations to the repetition domain. For clarity of presentation, the repetition-domain representation formulated in Ref. 7, using the system Markov parameters, will be briefly reviewed. The results will then be generalized to allow the use of observer Markov parameters. The difference between the two representations will be discussed.

Repetition-Domain Representation Using System Markov Parameters

Define a backward difference operator $\delta_j()$ in the repetition variable j, applied to any variable $z(k)$ at time step k in the time domain to be

$$\delta_j z(k) = z_j(k) - z_{j-1}(k) \tag{8}$$

In words, $\delta_j z(k)$ represents a change in the value of the variable $z(k)$ between two successive repetitions j and $j-1$. Applying the backward difference operator to the input–output representation using the Markov parameters yields

$$\delta_j y(k) = CA^k \delta_j x(0) + \sum_{i=1}^{k} CA^{i-1} B \delta_j u(k-i) + D\delta_j u(k) + \sum_{i=1}^{k} CA^{i-1} \delta_j w_1(k-i) + \delta_j w_2(k) \tag{9}$$

Assuming that the system starts from the same initial condition at every repetition and the disturbances are also repetitive, that is, $\delta_j x(0) = 0$, $\delta_j w_1(k) = 0$, $\delta_j w_2(k) = 0$. Equation (9) becomes

$$\delta_j y(k) = \sum_{i=1}^{k} CA^{i-1} B \delta_j u(k-i) + D\delta_j u(k) \tag{10}$$

Writing the result for each of the N-step process and packaging it in matrix form produces

$$\delta_j \underline{y} = P \delta_j \underline{u} \tag{11}$$

where for compactness of notation, the following time histories of input and output data and the transfer matrix P are defined:

$$\underline{y} = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}, \qquad \underline{u} = \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N) \end{bmatrix}$$

$$P = \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \ddots & \ddots & \ddots & \\ CA^{N-1}B & \cdots & CAB & CB & D \end{bmatrix}$$

Equation (11) is a representation of the original system in the repetition domain. This representation is expressed in terms of the system Markov parameters that make up the entries in the transfer matrix P. Interestingly, the repetitive initial conditions and the repetitive disturbances are absent in this presentation, allowing the learning controllers designed from this model the capability to handle these repetitive effects. This differenced form is very natural for learning control because if one has the result of one repetition, then one knows how much change in the control input history is needed to produce the desired output history, provided that the system Markov parameters are known. On the other hand, if the Markov parameters are not known, then this equation will motivate different learning control laws that will eventually drive the tracking error to zero by feeding forward the tracking errors generated from previous repetitions.

Repetition-Domain Representation Using Observer Markov Parameters

Analogous to the development presented in the preceding section, the repetition-domain representation using the observer Markov parameters can be derived. Applying the backward difference operator to Eq. (4) yields

$$\delta_j y(k) + \sum_{i=1}^{p} C(A+MC)^{i-1} M \delta_j y(k-i) = \sum_{i=1}^{p} C(A+MC)^{i-1}(B+MD)\delta_j u(k-i) + D\delta_j u(k) \tag{12}$$

where terms involving the repetitive initial conditions and the repetitive disturbances are automatically eliminated from the equation. Writing in matrix form for a N-step process yields

$$Q\delta_j \underline{y} = R\delta_j \underline{u} \tag{13}$$

where

$$Q = \begin{bmatrix} I & & & & & \\ CM & I & & & & \\ \vdots & CM & I & & & \\ C\bar{A}^{p-1}M & \cdots & CM & I & & \\ & \ddots & \vdots & \ddots & \ddots & \\ & & C\bar{A}^{p-1}M & \cdots & CM & I \end{bmatrix}$$

$$R = \begin{bmatrix} D & & & & & \\ C\bar{B} & D & & & & \\ \vdots & C\bar{B} & D & & & \\ C\bar{A}^{p-1}\bar{B} & \cdots & C\bar{B} & D & & \\ & \ddots & \vdots & \ddots & \ddots & \\ & & C\bar{A}^{p-1}\bar{B} & \cdots & C\bar{B} & D \end{bmatrix}$$

In the preceding matrices, $\bar{A} = A + MC$ and $\bar{B} = B + MD$. The matrices Q and R are formed by the first and second partitions of the observer Markov parameters $\bar{Y}(k)$. Because of the deadbeat conditions imposed for $\bar{A}$, the Q and R are both banded matrices. In fact, comparing Eq. (13) to Eq. (11) immediately reveals that $P = Q^{-1}R$. Since Q is always square and full rank, its inverse always exists.

Learning Control

In this section, linear forms of learning controllers are derived. It is assumed that the desired trajectory is specified such that it is feasible, i.e., there exists a control input history that would produce the desired response. First, simple linear feedback learning controllers will be briefly reviewed here. Generalizations to the observer model then follow.

Linear Forms of Learning Control Laws

Consider the system representation in the repetition domain in Eq. (11). Let the learning controller takes the linear form for any two successive repetitions $j-1$ and j,

$$\delta_j \underline{u} = L\underline{e}_{j-1} \tag{14}$$

The tracking error history is defined as $\underline{e}_j = \underline{y}^* - \underline{y}_j$, where $\underline{y}^*$ denotes the desired output history. Such a law looks at the tracking error at all points in the output history of the repetition $j-1$ and adjusts the control for the next repetition according to the rule

$$\underline{u}_j = \underline{u}_{j-1} + \delta_j \underline{u} \tag{15}$$

The question is then to determine L so that eventually as the repetitions progress, the control $\underline{u}_j = \underline{u}_0 + \delta_1\underline{u} + \delta_2\underline{u} + \cdots + \delta_j\underline{u}$ converges to the necessary input history $\underline{u}^*$ that would produce the desired output history $\underline{y}^*$. The system dynamics in the repetition domain given by Eq. (11) can be written in terms of the change in the tracking error history $\delta_j\underline{e} = \underline{e}_j - \underline{e}_{j-1}$ since $\delta_j\underline{e} = -\delta_j\underline{y}$:

$$\delta_j e = -P\delta_j \underline{u} \tag{16}$$

Substituting the learning control given in Eq. (14) into Eq. (16) yields

$$\underline{e}_j = (I - PL)\underline{e}_{j-1} \tag{17}$$

Hence, for asymptotic stability of the tracking error history $\underline{e}_j$ in the repetition domain, the eigenvalues of $I - PL$ must have magnitudes less than one. Asymptotic stability in the repetition domain implies tracking convergence in the time domain, i.e.,

$$\lim_{j\to\infty} \underline{e}_j = 0 \Rightarrow \lim_{j\to\infty} y_j(k) = y^*(k)$$

for $k = 0, 1, \ldots, N$. For a given system, it is thus seen that this learning control law will eventually make the system learn to produce zero tracking error with considerable freedom in the learning gain matrix L. This result is independent of the initial conditions and of the unknown repetitive disturbances. More interesting results can be obtained by exploiting the lower triangular block structure of the transfer matrix P. Let L be of the form

$$L = \begin{bmatrix} \Phi & & & \\ & \Phi & & \\ & & \ddots & \\ & & & \Phi \end{bmatrix} \tag{18}$$

Then the eigenvalues of $I - PL$ depend on D and Φ alone. Specifically, the learning gain L in Eq. (18) will produce an asymptotically stable process, provided that the eigenvalues of $I - D\Phi$ have magnitudes less than one. This result is independent of the remaining part of the system characterized by A, B, and C. Note that the learning control law in the repetition domain given by Eq. (14) can be easily written in the time domain. For example, the case where L is given in Eq. (18) has a learning controller written in the time domain of the form

$$u_j(k) = u_{j-1}(k) + \Phi e_{j-1}(k) \tag{19}$$

for $k = 0, 1, \ldots, N$. Equation (19) is known as the proportional learning law in learning control. Refer to Ref. 7 for a more complete investigation of other possible choices for the learning matrix L.

If the system does not have a direct transmission term, then some structural modifications to the appropriate equations must be made. For example, the input history vector $\underline{u}$, the output history vector $\underline{y}$, and the transfer matrix P now take the form

$$\underline{y} = \begin{bmatrix} y(1) \\ y(2) \\ y(3) \\ \vdots \\ y(N) \end{bmatrix}, \qquad \underline{u} = \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix}$$

$$P = \begin{bmatrix} CB & & & & \\ CAB & CB & & & \\ CA^2B & CA^2B & CB & & \\ \vdots & \ddots & \ddots & \ddots & \\ CA^{N-1}B & \cdots & CA^2B & CAB & CB \end{bmatrix}$$

Consequently, the learning controller of the form given in Eq. (14) with L given in Eq. (18) will become

$$u_j(k) = u_{j-1}(k) + \Phi e_{j-1}(k+1) \tag{20}$$

for $k = 0, 1, \ldots, N-1$. Stability of the learning process is now governed by the eigenvalues of $I - CB\Phi$. When there is no direct transmission term, $u(0)$ can influence $y(1)$ but not $y(0)$, so that it is generally assumed that the system starts out with no error, that is, $y(0) = y^*(0)$.

Designs of Learning Controllers Based on the Observer Model

The described learning laws can be modified to accommodate the observer model derived in Eq. (13). First, the corresponding equation that governs the difference in the tracking error for the model given in Eq. (13) is simply

$$Q\delta_j\underline{e} = -R\delta_j\underline{u} \tag{21}$$

Define a filtered error of the form $\underline{\varepsilon}_j = Q\underline{e}_j$; the repetition-domain equation that governs the filtered error is

$$\delta_j\underline{\varepsilon} = -R\delta_j\underline{u} \tag{22}$$

since $Q\delta_j\underline{e} = \delta_j(Q\underline{e}) = \delta_j\underline{\varepsilon}$. Comparing Eq. (22) to Eq. (16) immediately reveals that R now plays the role of P and the filtered error $\underline{\varepsilon}$ now plays the role of the tracking error $\underline{e}$. Let the learning law operated on the filtered error be of the form

$$\delta_j\underline{u} = L\underline{\varepsilon}_{j-1} \tag{23}$$

Substituting the learning control given in Eq. (23) into Eq. (22) yields the equation that describes the propagation of tracking error in the repetition domain,

$$\underline{\varepsilon}_j = (I - RL)\underline{\varepsilon}_{j-1} \tag{24}$$

Asymptotic convergence of the filtered tracking error is assured if L is chosen such that the eigenvalues of $(I - RL)$ have magnitudes less than one. This is in contrast to the previous case, where for asymptotic stability the eigenvalues of $I - PL$ are required to have magnitudes less than one. Note that the learning law given in Eq. (23) operates on the filtered tracking error instead of the actual tracking error. Since the filtered tracking error is related to the actual tracking error by the nonsingular matrix Q, that is, $\underline{\varepsilon}_j = Q\underline{e}_j$, convergence of the filtered tracking error to zero also implies convergence of the actual tracking error to zero. Also, because the matrix R also possesses the lower block triangular structure, simple learning laws are also possible in this case. If the learning gain matrix L takes the special form as in Eq. (18), then the stability of the learning process is governed by the eigenvalues of the matrix $I - D\Phi$. This is exactly the same requirement as in the previous case despite the fact that the learning law now operates on the filtered error instead of the actual tracking error. Another way to look at this control law is by examining how the actual tracking error propagates. Substituting the learning law in Eq. (23) in Eq. (21) produces

$$Q\delta_j\underline{e} = -RL\underline{\varepsilon}_{j-1} = -RLQ\underline{e}_{j-1} \tag{25}$$

Since Q is nonsingular, Eq. (25) implies

$$\begin{aligned} \underline{e}_j &= (I - Q^{-1}RLQ)\underline{e}_{j-1} \\ &= Q^{-1}(I - RL)Q\underline{e}_{j-1} \end{aligned} \tag{26}$$

which is asymptotically stable, provided that eigenvalues of $(I - RL)$ have magnitudes less than one.

If one chooses the same learning gain matrix L with a lower block triangular structure of appropriate dimensions in both cases, then the eigenvalues of $(I - RL)$ are the same as those of $(I - PL)$. The stability conditions for convergence of the learning process are, therefore, identical. The difference between the two methods, namely, that of using the actual error and the filtered error, appears only in the transient behavior from one repetition to the next during learning. To see this, let L be the same in both cases. Note that

$$\delta_j \underline{u} = L\underline{\varepsilon}_{j-1} = (LQ)\underline{e}_{j-1} \tag{27}$$

The learning law using the filtered error with gain L is, therefore, identical to the learning law using the actual tracking error with gain LQ. Therefore, the transient behavior during the learning process will be different in each case even if their eigenvalues are the same.

There is still another way to view Eq. (22) in relation to Eq. (16). The system A, B, C, D is transformed into the system $\bar{A}, \bar{B}, C, D$. The transformed system is typically far more stable than the original system. In fact, it is a deadbeat system. Since it is known that simple learning laws tend to work better on more stable systems, this transformation is expected to be advantageous. This is, indeed, the case, as shown later in the numerical examples.

Identification of the Q and R Matrices

This section will describe how the coefficients in Q (and also R) can be computed from input–output data. The procedure to be described involves only a single matrix computation after the input–output data are arranged in a particular format.

For simplicity, consider the case where the system starts from zero initial conditions and no repetitive disturbances are present. Equation (4) becomes

$$y(k) = \sum_{i=1}^{p} \bar{Y}(i)v(k-i) + Du(k), \qquad v(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix} \tag{28}$$

where the observer Markov parameters $\bar{Y}(i)$ are defined in Eq. (6). Writing Eq. (28) in matrix form for a set of input–output data $N+1$ samples long yields

$$y = \bar{Y}V \tag{29}$$

where

$$y = [\, y(0) \quad y(1) \quad \cdots \quad y(p) \quad y(p+1) \quad \cdots \quad y(N)\,]$$

$$\bar{Y} = [\,\bar{Y}(0) \quad \bar{Y}(1) \quad \cdots \quad \bar{Y}(p)\,]$$

$$V = \begin{bmatrix} u(0) & u(1) & \cdots & u(p) & u(p+1) & \cdots & u(N) \\ & v(0) & \cdots & v(p-1) & v(p-2) & \cdots & v(N-1) \\ & & \ddots & \vdots & \vdots & \vdots & \vdots \\ & & & v(0) & v(1) & \cdots & v(N-p) \end{bmatrix}$$

The observer Markov parameters are then computed from $\bar{Y} = yV^{+}$, where $(\,)^{+}$ denotes the pseudoinverse of the quantity in the parentheses.

If the initial conditions are not zero, then a slightly different equation must be used to solve for the observer Markov parameters. Specifically, $\bar{Y} = y_t V_t^{+}$, where y_t and V_t are obtained by deleting the first p columns in y and V, respectively. This step eliminates the effect of the initial conditions due to the requirement $\bar{A}^k = (A + MC)^k \equiv 0$ for $k \geq p$. If the repetitive disturbances are also present, then two sets of input–output data are required so that the observer Markov parameters can be identified correctly, i.e., $\bar{Y} = \delta_j y_t (\delta_j V_t)^{+}$. The difference between the two sets of data will eliminate the effect of the repetitive disturbances on the computation of the observer Markov parameters.

The second partitions of the identified observer Markov parameters are the parameters $\bar{Y}^{(2)}(k) = C\bar{A}^{k-1}M$ that will be used in the Q matrix for the proposed learning law $\delta_j \underline{u} = L\underline{\varepsilon}_{j-1}$, where $\underline{\varepsilon}_j = Q\underline{e}_j$. This can be viewed as a filter equation because $\underline{\varepsilon}_j = Q\underline{e}_j$ implies

$$\varepsilon_j(k) = e_j(k) + CMe_j(k-1) + C\bar{A}Me_j(k-2) + \cdots + C\bar{A}^{p-1}Me_j(k-p) \tag{30}$$

where $e_j(k) \equiv 0$ for $k \leq 0$. The identified direct transmission term D is $\bar{Y}(0)$, which can be used to determine a learning gain for the learning controller.

Optimal Learning Gain Matrix

The preceding sections show that to design a gain Φ to be used the matrix L given by Eq. (18), one would need to know only the direct transmission term D that is in R. This is a simple law that can produce an asymptotically stable learning process, but not necessarily optimal because the single gain Φ constrains L to be a block diagonal matrix. This section shows an optimal learning gain matrix can be designed if one relaxes this restriction.

Consider a quadratic cost function of the form

$$J_j = \tfrac{1}{2}\left(\underline{\varepsilon}_j^T S^{(1)} \underline{\varepsilon}_j + \delta_j^T \underline{u} S^{(2)} \delta_j \underline{u}\right) \tag{31}$$

where $S^{(1)}$ and $S^{(2)}$ are symmetric positive definite matrices. The first term is a weighted norm of the filtered tracking error history due to a change in the control input history by learning. The second term is a weighted norm of that change in the control history. This term influences implicitly the change in the control so that excessive correction during learning does not occur. Ideally, the tracking error should converge smoothly (in the repetition domain) without undergoing excessive transient behavior. The optimal learning control law is to be determined by minimizing this cost function. It will be shown that this control law possesses the same linear structure as presented in preceding sections.

Since $\delta_j \underline{\varepsilon} = -R\delta_j \underline{u}$, the cost function in Eq. (31) can be expressed as

$$\begin{aligned} J_j &= \tfrac{1}{2}\left[(\underline{\varepsilon}_{j-1} - R\delta_j \underline{u})^T S^{(1)} (\underline{\varepsilon}_{j-1} - R\delta_j \underline{u}) + \delta_j^T \underline{u} S^{(2)} \delta_j \underline{u} \right] \\ &= \tfrac{1}{2}\left(\underline{\varepsilon}_{j-1}^T S^{(1)} \underline{\varepsilon}_{j-1} - \underline{\varepsilon}_{j-1}^T S^{(1)} R\delta_j \underline{u} - \delta_j^T \underline{u} R^T S^{(1)} \underline{\varepsilon}_{j-1} \right. \\ &\quad \left. + \delta_j^T \underline{u} R^T S^{(1)} R\delta_j \underline{u} + \delta_j^T \underline{u} S^{(2)} \delta_j \underline{u}\right) \end{aligned}$$

Taking the derivative of the cost function with respect to the change in the control input and simplifying the expression produces

$$\begin{aligned} \frac{\partial J_j}{\partial[\delta_j \underline{u}]} &= \frac{1}{2}\left(-R^T S^{(1)T} \underline{\varepsilon}_{j-1} - R^T S^{(1)} \underline{\varepsilon}_{j-1} + R^T S^{(1)} R\delta_j \underline{u} \right. \\ &\quad \left. + R^T S^{(1)^T} R\delta_j \underline{u} + S^{(2)} \delta_j \underline{u} + S^{(2)^T} \delta_j \underline{u} \right) \\ &= -R^T S^{(1)} \underline{\varepsilon}_{j-1} + \left(R^T S^{(1)} R + S^{(2)}\right)\delta_j \underline{u} \end{aligned}$$

Setting the result to zero and solving for $\delta_j \underline{u}$,

$$\delta_j \underline{u} = L_{\text{opt}} \underline{\varepsilon}_{j-1} \tag{32}$$

where the optimal learning gain matrix is given as

$$L_{\text{opt}} = \left(R^T S^{(1)} R + S^{(2)}\right)^{-1} R^T S^{(1)} \tag{33}$$

The learning control law thus has a linear feedback structure of the general form in Eq. (23) with an optimal learning gain matrix L_{opt}. In general, L_{opt} does not possess the block diagonal structure.

If one uses the cost function with the actual tracking error instead, i.e.,

$$J_j = \tfrac{1}{2}\left(\underline{e}_j^T S^{(1)} \underline{e}_j + \delta_j^T \underline{u} S^{(2)} \delta_j \underline{u}\right) \tag{34}$$

then the optimal learning gain matrix can be obtained replacing R by $P = Q^{-1}R$ as

$$\delta_j \underline{u} = \left[\left(P^T S^{(1)} P + S^{(2)}\right)^{-1} P^T S^{(1)}\right] \underline{e}_{j-1}$$
$$= \left[\left(R^T Q^{-T} S^{(1)} Q^{-1} R + S^{(2)}\right)^{-1} R^T Q^{-T} S^{(1)}\right] \underline{e}_{j-1} \quad (35)$$

where Q^{-T} denotes $(Q^{-1})^T$. In this case, the learning control law will have the feedback of the actual tracking error, not the filtered error.

Since the cost function involves the weighted tracking error of the entire time history, the weighting can be adjusted to emphasize a certain portion of the response over another. For example, if the position at the endpoint is more important, larger weight can be assigned near the endpoint so that the learning will be emphasized during that portion of the response. Also, to compute the optimal gains, specialized numerical algorithms that take advantage of the Toeplitz structure of the matrices may be needed to handle long trajectories.

Learning Control in System with an Existing Feedback Controller

A scenario of practical importance is that the system considered in Eq. (1) already has an existing feedback controller, say,

$$\begin{aligned} \boldsymbol{u}_f(k) &= G[\boldsymbol{y}(k) - \boldsymbol{y}^*(k)] \\ &= G[C\boldsymbol{x}(k) + D\boldsymbol{u}(k) - \boldsymbol{y}^*(k)] \end{aligned} \quad (36)$$

For the state feedback case, simply set $C = I$. Let the input to the system be modified such that it is the sum of the feedback input and the learning control input,

$$\boldsymbol{u}(k) = \boldsymbol{u}_f(k) + \boldsymbol{u}_\ell(k) \quad (37)$$

To derive the closed-loop equations, first substitute Eq. (37) into Eq. (36) and solve for $\boldsymbol{u}_f(k)$:

$$\boldsymbol{u}_f(k) = (I - GD)^{-1}GC\boldsymbol{x}(k) + (I - GD)^{-1}GD\boldsymbol{u}_\ell(k)$$
$$-(I - GD)^{-1}G\boldsymbol{y}^*(k) \quad (38)$$

provided that the inverse $(I - GD)^{-1}$ exists. Substituting Eq. (38) into Eq. (1) yields the following system of equations:

$$\begin{aligned} \boldsymbol{x}(k+1) &= A_c\boldsymbol{x}(k) + B_c\boldsymbol{u}_\ell(k) + \boldsymbol{w}_{c1}(k) \\ \boldsymbol{y}(k) &= C_c\boldsymbol{x}(k) + D_c\boldsymbol{u}_\ell(k) + \boldsymbol{w}_{c2}(k) \end{aligned} \quad (39)$$

where the closed-loop matrices are

$$A_c = A + B(I - GD)^{-1}GC, \qquad B_c = B + B(I - GD)^{-1}GD$$
$$C_c = C + D(I - GD)^{-1}GC, \qquad D_c = D + D(I - GD)^{-1}GD$$

and the closed-loop repetitive disturbances are

$$\boldsymbol{w}_{c1}(k) = \boldsymbol{w}_1 - B(I - GD)^{-1}G\boldsymbol{y}^*(k)$$
$$\boldsymbol{w}_{c2}(k) = \boldsymbol{w}_2 - D(I - GD)^{-1}G\boldsymbol{y}^*(k)$$

Comparing Eq. (39) with Eq. (1) immediately reveals that A_c, B_c, C_c, D_c, $\boldsymbol{w}_{c1}(k)$, and $\boldsymbol{w}_{c2}(k)$ now play the roles of A, B, C, D, $\boldsymbol{w}_1(k)$, and $\boldsymbol{w}_2(k)$ respectively. In Eq. (1), $\boldsymbol{u}(k)$ is treated as the learning control input. In Eq. (39), this is replaced by $\boldsymbol{u}_\ell(k)$. All previously developed theory applies if one makes the appropriate substitution.

Numerical Examples

The system considered is a three-degree-of-freedom mass-spring-dashpot system connected in series. A state-space representation of the system is given in the Appendix. The center mass is required to follow a desired position trajectory. The necessary (force) input profile to make this happen is to be determined by learning control. In the following figures, the solid curve represents the desired position trajectory and the solid-dashed curve is the response produced by the system in an arbitrary trial. The system is then trained to produce the desired trajectory in a number of repetitions starting from the first trial. At each repetition, the control input time history is modified and the resultant response is shown by the dashed curves. In each of these plots there is a progression of dashed curves that gradually converge to the desired trajectory (solid curve) starting from the initial trial (solid-dashed curve). Ideally, a series of rather smooth corrections is desired as the system learns to produce the desired trajectory. The following cases are considered.

Position Learning

First, it is well known that a proportional learning law, such as the one given Eq. (19), has difficulties in learning to track a position trajectory based on position error feedback when the control input is force or torque. Usually, in practice, the proportional learning law is normally applied to track a desired velocity trajectory instead. The new learning designs offer an opportunity to use direct position error for learning control when the control input is torque or force. Indeed, this is the case shown in Fig. 1 with $\phi = 20$ corresponding to an (eigen)value $\lambda = 1 - (CB)_1\phi = 0.90$. The subscript 1 denotes the parameter associated with position output in the product CB. The coefficients of the filter are obtained from a set of input–output data using random excitation, with $p = 2$. If the optimal learning gain matrix is used, then the system learns to produce the desired position trajectory quickly in just a few repetitions, which is shown in Fig. 2. A learning law obtained with $S_2^{(2)} = 0$ is equivalent to computing the exact input history that would produce the desired output history in one repetition. But Fig. 2 shows that it takes more than one repetition for the system to produce the desired position trajectory. This is because in the identification step $p = 2$ is used.

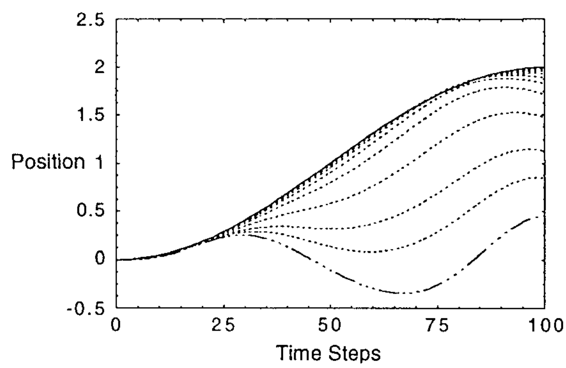

Fig. 1 Position learning with a proportional learning law using filtered error, $\phi_2 = 20$, $\lambda_2 = 0.90$, and $p = 2$.

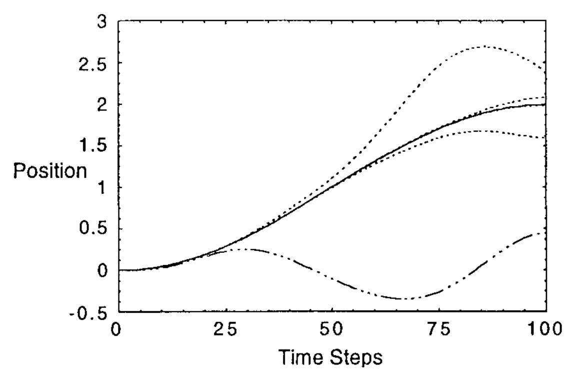

Fig. 2 Position learning using an optimal gain, $S_2^{(1)} = 1000$, $S_2^{(2)} = 0$, and $p = 2$.

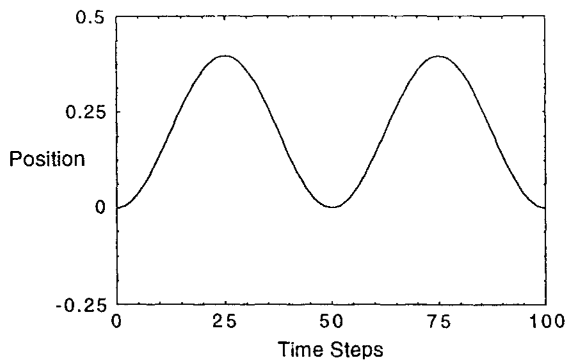

Fig. 3 Repetitive position disturbance time history.

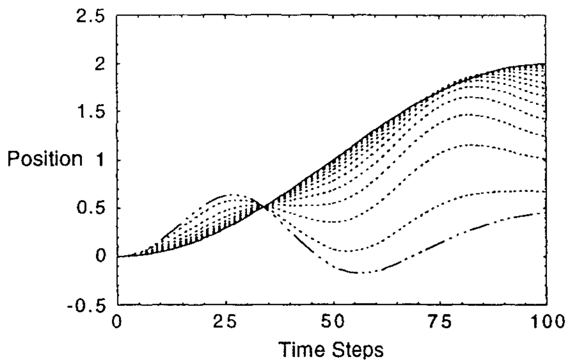

Fig. 4 Position learning in the presence of disturbance using the same controller as in Fig. 1, $\phi_2 = 20$, $\lambda_2 = 0.90$, and $p = 2$.

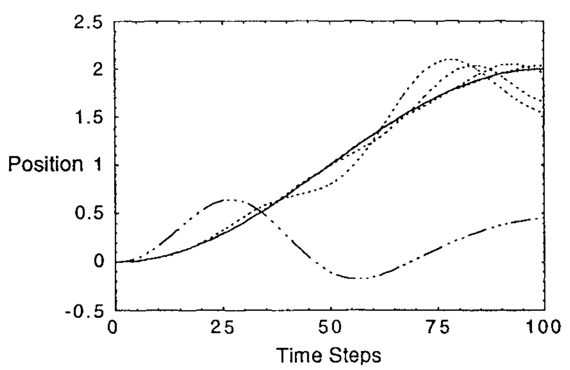

Fig. 5 Position learning in the presence of disturbance using the same optimal gain as in Fig. 2, $S_2^{(1)} = 1000$, $S_2^{(2)} = 0$, and $p = 2$.

Since the system is of sixth order with one output (position of the middle mass), this value of p does not allow sufficient freedom for complete system identification. The correct value for p is 6 or larger. This explains why the system does not produce the desired trajectory in one repetition. The system still learns to produce the desired response, however, without requiring additional updates in the parameter estimates. This example illustrates an advantage of learning control in making the system learn to produce the desired response eventually, even when the parameter estimates are in error.

Repetitive Disturbance Rejection

Another feature of learning control is its ability to handle repetitive disturbances. As an example, consider the case where at every repetition the position of the middle mass is affected by a disturbance profile shown in Fig. 3. Figures 4 and 5 are the counterparts of Figs. 1 and 2, respectively, when the disturbance is present. Recall that in the presence of repetitive disturbances, the observer Markov parameters should be computed using differences in the input–output data between two repetitions rather than using data from one single repetition.

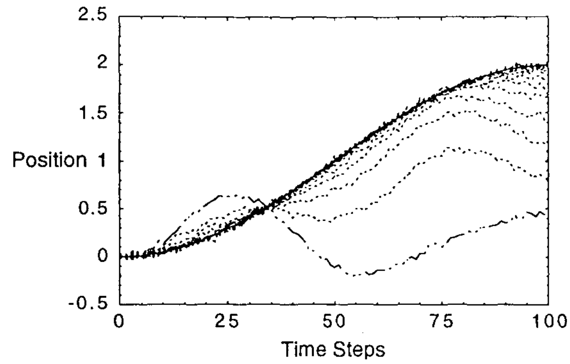

Fig. 6 Position learning with a proportional learning law using filtered error with disturbance and noise present, $\phi_2 = 20$, $\lambda_2 = 0.90$, and $p = 4$.

Learning in the Presence of Noises

Figure 6 illustrates the effect of noises on the learning process. The position learning scheme using filtered tracking error is used with a gain $\phi = 20$ corresponding to an eigenvalue of 0.90 for the learning process. The system responses that are recorded and fed back for learning control are contaminated by 3% measurement noises. This is the same case as in Fig. 1, except that the repetitive disturbance is also present and the filter order $p = 4$ is used instead of $p = 2$. Typically, when noises are present, it is expected and indeed observed that the system continues to learn to produce the desired position trajectory within the noise level. Learning laws that produce fast convergence in the noise-free case, however, tend to be more sensitive to noises. Qualitatively, fast learning implies larger correction based on a small amount of information, thus making the controller more sensitive to noise. If an optimal gain is computed, then one can introduce a weighting on the control input corrections to slow down the learning rate so that the influence of noises on the learning process can be reduced.

Concluding Remarks

This paper derives linear feedback forms of learning controllers for a system described by an autoregressive model. Besides the state-space model, the autoregressive model is another convenient way to present a linear system in the time domain. The autoregressive representation is shown to relate to an observer in the state-space formulation. The observer Markov parameters that were found to be important in recent system identification techniques are used in the formulation and design of the learning controllers presented in this paper. The new designs involve a parameter estimation step to identify the observer Markov parameters before the learning gains are designed. For a linear system, the observer Markov parameters can be easily computed from input–output data, so that any additional computational cost required by this step is quite minimal. In theory, if the system can be identified perfectly, then there is no need for learning control because the necessary control input that would produce the desired response can be computed in one repetition. In practice, unknown disturbances and modeling errors prevent this type of open-loop operation to be successful. Learning control can be used to correct for the errors that remain. The same situation applies even if there is an existing feedback controller operating. It can be argued that the original motivation of learning control is to rely on as little information about the system as possible. In practice, however, any knowledge about the system should be used to help learning process by improving the learning speed or the robustness of the learning process. This paper has shown how this can be accomplished for a system that is linear and the disturbances can be assumed to be unknown as long as they can be considered to be repetitive and deterministic. Issues that have not been addressed in this paper include stability and performance robustness, nonrepetitive disturbances, such as random process and measurement noise, and nonlinearities.

Appendix: Discrete Model

The discrete model used in the numerical examples is given as follows:

$$A = \begin{bmatrix} 9.901 \times 10^{-1} & 4.950 \times 10^{-3} & 7.235 \times 10^{-6} & 9.885 \times 10^{-2} & 2.582 \times 10^{-4} & 3.072 \times 10^{-7} \\ 4.971 \times 10^{-3} & 9.900 \times 10^{-1} & 4.974 \times 10^{-3} & 2.582 \times 10^{-4} & 9.947 \times 10^{-2} & 2.666 \times 10^{-4} \\ 3.755 \times 10^{-6} & 2.476 \times 10^{-3} & 9.925 \times 10^{-1} & 1.536 \times 10^{-7} & 1.333 \times 10^{-4} & 9.897 \times 10^{-2} \\ -1.974 \times 10^{-1} & 9.833 \times 10^{-2} & 2.578 \times 10^{-4} & 9.737 \times 10^{-1} & 6.792 \times 10^{-3} & 1.245 \times 10^{-5} \\ 9.896 \times 10^{-2} & -1.986 \times 10^{-1} & 9.907 \times 10^{-2} & 6.792 \times 10^{-3} & 9.862 \times 10^{-1} & 6.960 \times 10^{-3} \\ 1.330 \times 10^{-4} & 4.922 \times 10^{-2} & -1.483 \times 10^{-1} & 6.227 \times 10^{-6} & 3.480 \times 10^{-3} & 9.769 \times 10^{-1} \end{bmatrix}$$

$$B = \begin{bmatrix} 7.245 \times 10^{-6} \\ 4.985 \times 10^{-3} \\ 3.761 \times 10^{-6} \\ 2.582 \times 10^{-4} \\ 9.947 \times 10^{-2} \\ 1.333 \times 10^{-4} \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The sampling interval is 0.1 s. The first nonzero Markov parameter is $Y(1) = \bar{Y}(1) = CB$:

$$CB = \begin{bmatrix} 4.985 \times 10^{-3} \\ 9.947 \times 10^{-2} \end{bmatrix}$$

The two elements of CB corresponding to the position and velocity measurements are $(CB)_1$ and $(CB)_2$, respectively:

$$(CB)_1 = 4.985 \times 10^{-3}, \qquad (CB)_2 = 9.947 \times 10^{-2}$$

References

[1]Arimoto, S., Kawamura, S., and Miyazaki, F., "Better Operation of Robots by Learning," *Journal of Robotic Systems*, Vol. 1, No. 2, 1984, pp. 123–140.

[2]Craig, J. J., "Adaptive Control of Manipulators Through Repeated Trials," *Proceedings of the American Control Conference* (San Diego, CA), 1984, pp. 1566–1573.

[3]Casalino, G., and Bartolini, G., "A Learning Procedure for the Control of Movements of Robotic Manipulators," *IASTED Symposium on Robotics and Automation* (Amsterdam), 1984, pp. 108–111.

[4]Togai, M., and Yamano, O., "Analysis and Design of an Optimal Learning Control Scheme for Industrial Robots: A Discrete-Time Approach," *Proceedings of the 24th IEEE Conference on Decision and Control* (Fort Lauderdale, FL), Inst. of Electrical and Electronics Engineers, 1985.

[5]Atkeson, C. G., and McIntyre, J., "Robot Trajectory Learning Through Practice," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation* (San Francisco, CA), Inst. of Electrical and Electronics Engineers, 1986, pp. 1737–1742.

[6]Kawamura, S., Miyazaki, F., and Arimoto, S., "Realization of Robot Motion Based on a Learning Method," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, 1988, pp. 126–134.

[7]Phan, M., and Longman, R. W., "A Mathematical Theory of Learning Control for Linear Discrete Multivariable Systems," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference* (Minneapolis, MN), 1988, pp. 740–746.

[8]Miller, W. T., Hewes, R. P., Glanz, F. H., and Kraft, L. G., "Real-Time Dynamic Control of an Industrial Manipulator Using a Neural Network Based Learning Controller," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 1, 1990, pp. 1–9.

[9]Yoo, D. S., and Chung, M. J., "Real-Time Implementation and Evaluation of Dynamic Control Algorithms for Industrial Manipulators," *IEEE Transactions on Industrial Electronics*, Vol. 38, No. 1, 1991, pp. 26–31.

[10]Caiti, A., Cannata, G., and Casalino, G., "Implementation of Learning Control Techniques Using Descriptor Systems Methods," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* (Sacramento, CA), Inst. of Electrical and Electronics Engineers, 1991, pp. 1451–1456.

[11]Saab, S. S., Vogt, W. G., and Mickle, M. H., "Theory of P-type Learning Control with Implication for the Robot Manipulator," *Proceedings of the 1993 IEEE International Conference on Robotics and Automation* (Atlanta, GA), Inst. of Electrical and Electronics Engineers, 1993, pp. 665–671.

[12]Middleton, R. H., Goodwin, G. C., and Longman, R. W., "A Method for Improving the Dynamic Accuracy of a Robot Performing a Repetitive Task," *International Journal of Robotics Research*, Vol. 8, No. 5, 1989, pp. 67–74.

[13]Cosner, C., Anwar, G., and Tomizuka, M., "Plug-In Repetitive Control for Industrial Manipulators," *Proceedings of the 1990 IEEE International Conference on Robotics and Automation* (Cincinnati, OH), Inst. of Electrical and Electronics Engineers, 1990, pp. 1970–1975.

[14]Elci, H., Phan, M., Longman, R. W., Juang, J.-N., and Ugoletti, R., "Experiments in the Use of Learning Control for Maximum Precision Robot Trajectory Tracking," *Proceedings of the 1994 Conference on Information Sciences and Systems*, Princeton Univ., Princeton, NJ, 1994.

[15]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, 1993, pp. 59–86.

[16]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, 1993, pp. 320–329.