JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 20, No. 6, November–December 1997

Generalized Holds, Ripple Attenuation, and Tracking Additional Outputs in Learning Control

Richard W. Longman and Chun-Ping Lo
Columbia University, New York, New York 10027

Iterative discrete-time learning control can be very effective at improving the tracking performance of control systems. Three different limitations or potential difficulties of discrete-time learning control are addressed. 1) The number of output variables for which zero tracking error can be achieved is limited by the number of input variables. 2) Every variable for which zero tracking error is sought must be a measured variable. 3) Because the control action is digital, the intersample behavior may have undesirable error from ripple. The first issue is addressed by two approaches, either by seeking zero error for different output variables at alternating time steps or by skipping time steps. The latter corresponds to using a generalized hold device. To address the second issue, it is shown how these techniques can be combined with an observer when one wishes to have improved tracking of not only measured outputs but also unmeasured output variables. To improve ripple, one can ask for good tracking of the output, and in addition good tracking of its derivative or derivatives, and it is shown how methods developed here can accomplish this. These generalizations are made for three different effective learning control laws: integral control-based learning with zero phase filtering, which is particularly simple; a contraction mapping learning control, which is particularly robust; and a phase cancellation learning control, which is particularly effective at producing small final error levels in experiments.

Introduction

LEARNING and repetitive control are relatively new fields that develop controllers that learn from previous experience performing a specific command to improve their performance in the next execution of the command. The main motivation for iterative learning control was robots performing repetitive tasks. In Ref. 1, the section by Arimoto on the early history of learning control for robot tracking states, "Not a single but several persons struggling in the same research frontiers were gifted simultaneously and independently with a common idea," first Uchiyama[2] in Japan and second but independently Craig[3] in the United States, Casalino and Bartolini[4] in Italy, and Arimoto et al.[5] in Japan, all in 1984. In addition, Ref. 6 was submitted in 1984 from Australia. Repetitive control has seen similar independent developments, as in Ref. 6 and the substantial contributions of Tomizuka (e.g., Ref. 7). The literature in the field has now become quite extensive. In a series of publications (for example, Refs. 8–22) the authors and co-workers have sought to develop an overall framework for learning and repetitive control methodologies, and this is summarized in Refs. 23–25. References 16–18 develop several learning methods and test them on a commercial Robotics Research Corporation robot. The best results improve the tracking error performing a high-speed, large-angle maneuver by a factor of nearly 1000 in a small number of repetitions of the learning process. This is close to the repeatability level of the hardware, which represents the minimum possible tracking error level obtainable.

Although, these methods can be very effective in eliminating the tracking error at the sample times, there are three limitations, which will be addressed. 1) The number of output variables for which zero tracking error can be obtained is limited to be less than or equal to the number of input or control variables. 2) Every output variable for which one seeks to obtain zero tracking error must be

Richard W. Longman received his Ph.D. at the University of California, San Diego, and then joined Bell Telephone Laboratories. In 1970 he accepted a faculty position at Columbia University. Since then, he served as visiting faculty at the Massachusetts Institute of Technology and six other universities in Germany, Taiwan, and Australia and has been consultant or parttime employee at 17 organizations, including NASA Langley Research Center, NASA Goddard Space Flight Center, Lockheed Missiles and Space Company, General Electric Research Laboratory, the U.S. Naval Research Laboratory, Martin Marietta, RAND, and the European Space Operations Centre. He served as Vice President for Publications, Vice President Technical, and First Vice President of the American Astronautical Society (AAS) and as a member of their Board of Directors. He is a Fellow of AIAA and AAS.

Chun-Ping Lo received a Bachelor of Engineering degree from Tamkang University, Taipei, in 1989 and then worked at An Feng Steel Co., Ltd., Kaohsiung, as an Assistant Engineer in the Hydraulics Division. In 1992 he came to Columbia University, where he received his M.S. in Mechanical Engineering in 1994. He is currently a Ph.D. student conducting research in learning control.

Received Nov. 22, 1996; presented as Paper 97-0685 at the AIAA 35th Aerospace Science Meeting, Reno, NV, Jan. 6–9, 1997; revision received July 14, 1997; accepted for publication July 16, 1997.

a measured variable. 3) Although many learning control algorithms are formulated in continuous time, the implementation is necessarily done in discrete time. Hence, what one can obtain is zero tracking error at the sample times. As with all digital control methods, there is the possibility of unacceptable ripple, i.e., unacceptable motion of the output variables between the sample times. It is the purpose of this paper to develop methods of learning control that address each of these three issues. We will make use of methods suggested in Refs. 8 and 9 to allow us to control more output variables than input variables, either by cycling through the set of variables of interest as the time steps progress or by skipping an appropriate number of steps between those steps for which zero tracking error is sought. The latter approach is related to the use of generalized hold devices (see, for example, Ref. 26), and Ref. 27 studies the use of such holds for ripple attenuation in repetitive control. Several different learning control laws will be generalized to address these three issues.

Linear Learning Control Laws

In this section we review the general linear learning control formulation presented in Ref. 8 and then present the three learning control laws, developed in Refs. 6, 8, 12, 16, and 18, which will be generalized to handle the described problems. All three of these laws have been tested in experiments on a Robotics Research Corporation robot, and all produced improvements in tracking accuracy by a factor of 100 or more in a small number of repetitions.[16–18,20] Each control law has its own set of advantages and disadvantages.

Consider the general multi-input/multi-output (MIMO) linear discrete-time system given by

$$x(k+1) = Ax(k) + Bu(k) + w(k), \qquad k = 0, 1, \ldots, p-1$$
$$y(k) = Cx(k), \qquad k = 1, 2, \ldots, p \tag{1}$$

where x is the n-dimensional state vector, u is the m-dimensional learning control input vector, y is the g-dimensional output vector, and w is a forcing function, including disturbances. We assume that a feedback controller is operating so that the matrix A is the closed-loop system matrix. Then w contains the command to the control system. A repetitive process of p time steps is considered that starts from the same initial condition $x(0)$ each repetition. Any disturbances in w are also considered to be repetitive (nonrepetitive disturbances are discussed in Ref. 8). We are limiting ourselves to considering time-invariant systems, but the first two of the learning laws to be treated immediately generalize to time-varying systems. This can be important when considering nonlinear systems and modeling them as linearized about the desired trajectory. Because a feedback controller is in operation, it is likely that the trajectory will start within a neighborhood of the desired trajectory within which linearized equations are valid.

The solution to Eq. (1) is

$$y(k) = CA^k x(0) + \sum_{i=0}^{k-1} CA^{k-i-1}Bu(i) + \sum_{i=0}^{k-1} CA^{k-i-1}w(i) \qquad k = 1, 2, \ldots, p \tag{2}$$

Define the difference operator $\delta_j z(k) = z_j(k) - z_{j-1}(k)$ that differences values of any variable in two successive repetitions. When applied to Eq. (2) the first and last terms on the right are eliminated, and the resulting equation can be written in matrix form as

$$\delta_j \underline{y} = P\delta_j \underline{u} \tag{3}$$

where

$$P = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & CA^{p-3}B & \cdots & CB \end{bmatrix}$$

and the underbars indicate the matrix of histories of variables

$$\underline{x} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(p) \end{bmatrix}, \qquad \underline{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(p) \end{bmatrix}, \qquad \underline{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(p-1) \end{bmatrix} \tag{4}$$

Note that Eq. (3) can be written in the alternate form

$$\underline{y}_j = I\underline{y}_{j-1} + P\delta_j \underline{u} \tag{5}$$

which can be thought of as a modern state-space representation of the system behavior in the repetition domain. The system matrix in this representation is the identity matrix, and the input is the change in the learning control signal from the previous repetition. Some advantages of this repetition domain representation are the following. 1) The repetitive disturbances are eliminated in the differencing, making the learning control laws developed equally applicable to systems with or without such disturbances (for example, the rather large gravity disturbance in a robot). 2) Time-varying systems become repetition invariant so that control concepts associated with time-invariant systems can be applied. 3) The computation of the learning control action can be made based on all of the information of the preceding repetition and, therefore, need not be restricted to operations satisfying causality in the time domain.

Linear learning control has the form

$$\underline{u}_j = \underline{u}_{j-1} + L\underline{e}_{j-1}, \qquad \underline{e}_{j-1} = \underline{y}^* - \underline{y}_{j-1}, \qquad j = 1, 2, 3, \ldots \tag{6}$$

where L is the learning gain matrix and $\underline{e}_j$ is the error history for repetition j, i.e., the difference between the desired output and the actual output at repetition j. From Eq. (6) it is clear that we need at least as many input variables as measured output variables whose values we wish to track specified trajectories. Hence, L must either be square or have more rows than columns. Substituting Eq. (6) into Eq. (5) shows that for feasible desired output histories the error history vector as a function of repetition number satisfies[8]

$$\underline{e}_j = (I - PL)\underline{e}_{j-1} \tag{7}$$

and converges asymptotically to zero when all eigenvalues of $I - PL$ satisfy

$$|\lambda_i(I - PL)| < 1 \tag{8}$$

There is a great deal of freedom in selecting the learning matrix L to satisfy Eq. (7), and three choices that have proved very effective are now discussed.

Integral Control-Based Learning Control with Zero Phase Filtering

The most basic form of learning control uses integral control concepts applied in the repetition domain (see Refs. 6, 8, and others). Zero phase filtering is introduced to limit the bandwidth of learning[16,17] and to avoid poor transients during the learning process.[10,11] Let the learning gain matrix be $L = \mathrm{diag}(\Phi, \Phi, \ldots, \Phi)$; then the learning control law written for each time step becomes

$$u_j(k) = u_{j-1}(k) + \Phi e_{j-1}(k+1) \tag{9}$$

This law was called a p-integrator in Ref. 6. The continuous-time version is sometimes called proportional (P-type) learning control.[28,29] P-type learning control laws are preferable to derivative (D type) learning control laws because they do not involve differentiation of a signal.[28] Convergence of the learning process for this special form of learning control is guaranteed according to Eq. (7) if Φ satisfies the much smaller eigenvalue problem

$$|\lambda_i(I - CB\Phi)| < 1, \qquad i = 1, 2, \ldots, m \tag{10}$$

which reduces to a scalar inequality in the single input/single output case. The learning control in Eq. (6) has a large amount of freedom in selecting learning gains, which can be used for various performance objectives. By constraining L to block diagonal form, we have used much of this freedom in exchange for an easily applied learning rule with a simple convergence criterion.

This rule can be very effective, very practical, and very simple to use. The experiments in Ref. 16 decreased the robot tracking error by a factor of 100 in about seven repetitions, and by adding a compensator in the learning control, the tracking error is decreased by a factor of nearly 1000 (Ref. 17). A low-pass zero phase filter can be applied to the error signal before putting it into the learning control law, as developed in Refs. 16 and 17. This makes the learning process blind to error components above the cutoff. A methodology is given to choose the cutoff frequency as a tradeoff with the learning gain, based on the closed-loop steady-state frequency response of the system. This information is known, for example, when the same person designs the feedback controller as designs the learning control. An alternative to using the low-pass filter is to learn until the error reaches a minimum and then freeze the learning signal for future repetitions.[10,11]

Contraction Mapping Learning Control Law

The contraction mapping learning control gain matrix is given by[12–14]

$$L = sP^T \tag{11}$$

which corresponds to the learning control law

$$u_j(k) = u_{j-1}(k) + s \cdot \sum_{i=1}^{p} CA^{i-1}Be(k+i)$$
$$e(k+i) = 0 \qquad \text{for} \qquad k+i > p$$
$$k = 0, 1, \ldots, p-1, \qquad j = 1, 2, 3, \ldots \tag{12}$$

When this is substituted into Eq. (7), one obtains

$$\underline{e}_j = (I - sPP^T)\underline{e}_{j-1} \tag{13}$$

which produces a contraction mapping of the Euclidean norm of the error with repetitions, provided the spectral norm $\|I - sPP^T\|_s$ is less than 1, i.e., the maximum singular value is less than 1. This corresponds to the range of learning gains $0 < s < 2/\|P\|_s^2$ producing monotonic decay of the error norm with repetitions.

The contraction mapping learning control law makes use of additional information about the system in the form of the system Markov parameters. A major advantage of this learning law is that it has good transient properties during the learning process, i.e., monotonic decay of the tracking error norm. In addition, Ref. 13 shows that for sufficiently small learning gain it is robust to arbitrarily large errors in the Markov parameter model. It is, therefore, a very safe type of learning control to apply in practice. No zero phase low-pass filter or identification need be used for robustness, as is done in the two other learning control laws discussed here. The disadvantage is that it learns slowly at high frequency, with a learning rate related to the square of the amplitude of the frequency response of the system. Experiments using this law on the Robotics Research robot resulted in a reduction in tracking error by approximately two and one-half orders of magnitude.[20]

Phase Cancellation Learning Control Law

This law makes use of steady-state frequency response information about the system (something that would be available to the feedback control system designer) to supply a phase lead to the error signal as a function of frequency such that the phase lag produced going through the feedback system from command to response is canceled.[18,19] The z transform of the learning control law has the form

$$U_j(z) = U_{j-1}(z) + \Phi(z)E_{j-1}(z) \tag{14}$$

with obvious meanings for notations in the z domain. The error propagation from repetition to repetition appears as

$$E_j(z) = [I - G(z)\Phi(z)]E_{j-1}(z) \tag{15}$$

The $\Phi(z)$ is chosen as a function of frequency. In single input/single output systems, for each discrete frequency up to the Nyquist frequency it is set equal to a unit magnitude complex number with phase equal to minus the phase change of the input–output discrete frequency response of the system. The change in the steady-state frequency components from one repetition to another is obtained by looking at the amplitude of the transfer function $[I - G(z)\Phi(z)]$ with z set to $e^{i\omega T}$, where T is the sample time. The discrete frequencies involved are $\omega T = 2\pi n/(2p-1)$, $n = 0, 1, 2, \ldots, 2p-2$. In the event that the amplitude of the frequency response goes above one, the magnitude of $\Phi(z)$ is also cut by the reciprocal. In practice, it is easiest to perform the product on the right of Eq. (14) in the transform domain[20] and then take the inverse fast Fourier transform to obtain the learning control signal, but one can look at the learning law in the time domain by taking the inverse transform of Eq. (14) to obtain a convolution product

$$u_j(k) = u_{j-1}(k) + \sum_{n=-\infty}^{\infty} \phi(k-n)e_{j-1}(n) \tag{16}$$

The $\phi(k)$ represent the learning gains in the matrix L. Infinitely many gains are required, as indicated, to realize the steady-state frequency response design. They are truncated to the appropriate number of gains to fill the matrix L when implemented by Eq. (16), so that the limits in the summation become 0 to $p-1$. This learning law gives monotonic decay of the steady-state response of each frequency component of the error signal. A limitation of the approach is that, when the total time in a repetition becomes too short, steady-state frequency response thinking does not apply. In this case, the way to account for the finite-time aspect when generating the control law is given in Ref. 14.

The MIMO version of this algorithm is as follows.[18] Suppose that for each n the transfer function $G\{\exp[i2\pi n/(2p-1)]\}$ is diagonalizable by matrix $V(n)$, $V^{-1}\Lambda V$ with the eigenvalues λ_i. Then form the learning control matrix as

$$\Phi(n) = V(n)^{-1}\Omega(n)V(n) \tag{17}$$

where $\Omega(n)$ is a diagonal matrix formed from $\Lambda(n)$ by replacing each eigenvalue by a unit magnitude complex number with phase angle equal to the negative of that of the corresponding eigenvalue. In the event that an eigenvalue is larger than one in magnitude, the magnitude of the entry in $\Omega(n)$ is reduced by this factor.

This learning law is somewhat less robust than the contraction mapping law. To ensure convergence and good learning transients, one can either use a zero phase filter to limit the frequency range of learning to one in which one is confident of one's knowledge of the phase change of the feedback control system[16,17] or perform system identification using the data as part of the learning control process.[18,19] A major advantage of the algorithm is that its rate of learning at high frequencies is much faster, being related to the amplitude to the first power of the frequency response of the system. The experiments using this algorithm resulted in nearly a factor of 1000 improvement in the robot tracking accuracy, and the error was continuing to improve slowly at repetition 40 when the experiments were stopped.[18,20]

Controlling More Measured Outputs Than Inputs

Given m inputs in u, it is possible to control at most m output variables to exactly follow a desired trajectory at every sample time. If one has more measured output variables, and would like to achieve zero tracking error following desired trajectories for more than m output variables, then one can cycle through the set of variable, controlling one set the first time step, a second set the next time step, etc., and then repeating. This cycling through the output variables, or alternating among them, was suggested in Ref. 8. In multi-input systems this may be an underspecified problem at some time steps, allowing multiple solutions. An alternative to this alternating approach is to skip asking for zero tracking error for enough time steps that the set of control variables accumulated from the last controlled time step is greater than or equal to the number of output variables to be controlled.[9] Here we study these two concepts for the three different learning control laws. In the following section we study their use for ripple attenuation. For simplicity of exposition, we will often limit ourselves to considering that there are two scalar variables that we want to control and only one input. We also assume that each variable can be controlled at the next time step by adjusting the input, which is typical in discretized versions of continuous-time systems. However, in theory, it may take more time steps to control a given output, with the number of steps limited by n or $n - m$ for a controllable system. We will not discuss these special cases.

Alternating Output Method

Suppose that we have two outputs that we wish to control, $y_1 = C_1 x$ and $y_2 = C_2 x$, and we ask for zero tracking error for the first on odd-numbered time steps and the second for even steps. Then the desired trajectory and the actual trajectory are

$$\underline{y}^* = \begin{bmatrix} y_1^{*T}(1) & y_2^{*T}(2) & y_1^{*T}(3) & y_2^{*T}(4) & \cdots \end{bmatrix}^T$$
$$\underline{y} = \begin{bmatrix} y_1^{T}(1) & y_2^{T}(2) & y_1^{T}(3) & y_2^{T}(4) & \cdots \end{bmatrix}^T \tag{18}$$

from which the error history vector is defined. The P in Eq. (3) becomes

$$P = \begin{bmatrix} C_1 B & 0 & 0 & 0 & \cdots \\ C_2 AB & C_2 B & 0 & 0 & \cdots \\ C_1 A^2 B & C_1 AB & C_1 B & 0 & \cdots \\ C_2 A^3 B & C_2 A^2 B & C_2 AB & C_2 B & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{19}$$

There exists a solution having zero tracking error provided $C_1 B$ and $C_2 B$ are both full rank. We now examine how to apply each of the three learning control laws to this situation.

When using the integral control-based learning control, one picks $L = [\phi_1 \;\; \phi_2 \;\; \phi_1 \;\; \phi_2 \;\; \cdots]$. The eigenvalues of the matrix in Eq. (7) become those of $I - C_1 B\phi_1$ and $I - C_2 B\phi_2$, and having these less than unity determines the range of learning gains for convergence. For smoothness of the control inputs during the learning process, it may be desirable to have each type of output learn at the same speed. This suggests picking ϕ_1 and ϕ_2 such that the given eigenvalues are approximately equal. When a zero phase low-pass filter is applied to the error signal, it should be applied to each type of output signal separately, not to the error history vector. Furthermore, improved results might be obtained by applying it to the output history for all available sample times and then picking out the filter results at the sample times of interest.

To use the contraction mapping learning controller in this situation, one simply inserts the P of Eq. (19) into Eq. (11). One can generalize the gain s to a matrix with diagonal elements s_1 and s_2, and this allows one to weight the diagonal elements for each output, but the off-diagonal terms get mixed weighting. True balancing of the learning rates for y_1 and y_2 requires adjusting the scaling in the definition of one of the output variables. For example, if y_2 is velocity, it can be scaled by any factor we choose. This becomes obvious when one considers that velocity can be measured in meters per second, meters per hour, feet per month, or furlongs per fortnight. Make up your own units if you want a different scaling. Thus, one scales the relative size of C_1 and C_2.

To apply the phase cancellation learning control law, we need to find the transfer function from each input to each of the outputs in the $\underline{y}$ of Eq. (18) at their associated sample times. This can be done as follows. Define $\bar{u}$ and $\bar{y}_a$ as

$$\bar{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix}, \qquad \bar{y}_a(k+2) = \begin{bmatrix} y_1(k+1) \\ y_2(k+2) \end{bmatrix} \tag{20}$$

Using the state equation (1) for two time steps (ignoring the repetitive disturbance because it will drop when we difference successive repetitions), we can write

$$x(k+2) = A^2 x(k) + [AB \quad B]\bar{u}(k)$$
$$\bar{y}_a(k+2) = \begin{bmatrix} C_1 A x(k) + [C_1 B \quad 0]\bar{u}(k) \\ C_2 x(k+2) \end{bmatrix} \tag{21}$$

We then renumber the time arguments from k to $k' = k/2$ to eliminate skipped steps and take the z transform to obtain the 2×2 z-transfer function matrix

$$\bar{Y}_a(z) = G_a(z)\bar{U}(z) \tag{22}$$

Then one applies the MIMO phase cancellation learning control law using this transfer matrix.

Skip Step Method Using a Generalized Hold

In this section we skip specifying a desired trajectory for a sufficient number of time steps that the number of accumulated control inputs is greater than or equal to the number of output variables to be controlled. We then specify desired values of all variables at the same time step. For the two-variable set discussed earlier, we control y_1 and y_2 at the same time steps, but we do so only every other time step. Rather than using a zero-order hold from step k to $k + 2$, we have two zero-order hold values to pick, one for the first half of this interval and a second one for the second half of the interval. This process creates a special case of a generalized hold between the tracking sample times, one having a single extra degree of freedom.[26]

The appropriate system equations for every two time steps become

$$x(k+2) = A^2 x(k) + [AB \quad B]\bar{u}(k) = \bar{A}x(k) + \bar{B}\bar{u}(k)$$
$$y_g(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x(k) = Cx(k) \tag{23}$$

The output and input histories become

$$\underline{y} = [\, y^T(2) \quad y^T(4) \quad y^T(6) \quad \cdots]^T$$
$$\underline{u} = [\bar{u}^T(0) \quad \bar{u}^T(2) \quad \bar{u}^T(4) \quad \cdots]^T \tag{24}$$

for Eq. (3), and the associated P becomes $\bar{P}$ with the A and B replaced by $\bar{A}$ and $\bar{B}$.

Integral control-based learning is then accomplished using $L = \mathrm{diag}(\Phi, \Phi, \ldots, \Phi)$ with diagonal matrix partitions, and convergence according to Eq. (10) is determined by $|\lambda_i(I - C\bar{B}\Phi)| < 1$. Because of the need to choose a matrix learning gain, the task of finding appropriate gains becomes harder than in the alternating step method described first. One can try $\Phi = \mathrm{diag}(\phi_1, \phi_2)$ giving different weighting for the two types of outputs, but there is now mixing between these weights when learning both outputs at the same time step. For convergence it can be necessary to use off-diagonal terms, and the difficulty of picking appropriate gains makes this learning rule less desirable for the generalized hold case.

When applying the contraction mapping law, the modifications are obvious, using $\bar{P}$ in place of P and scaling C_1 and C_2 as needed. The phase cancellation law requires the development of a different z-transfer matrix for this generalized hold case:

$$\bar{Y}_g(z) = G_g(z)\bar{U}(z) \tag{25}$$

by renumbering the time arguments from k to $k' = k/2$ to eliminate the skipped steps and taking the z transform of system (23).

Example

We can illustrate these methods by taking a multiple output system in controllable canonical form, which makes certain things evident:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -3 & 0 \\ 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix} u$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{26}$$

In this system we can control either output every time step, or both outputs every other time step, either at the same steps or alternating,

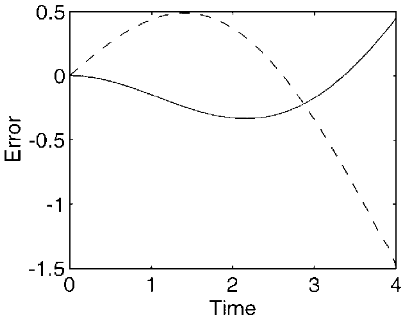

Fig. 1 Error in ——, y_1 after achieving zero error every time step for y_2 and – – –, y_2 after achieving zero error every other time step for y_1.

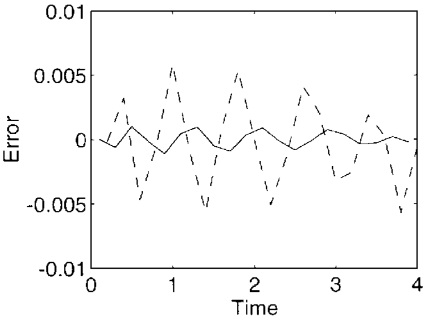

Fig. 2 Tracking errors for ——, y_1 and – – –, y_2 after 30 repetitions of alternate time step learning.

producing zero tracking error to some desired trajectory at these steps, e.g., $y_1^*(t) = 1 - \cos(\pi t/2t_f)$ and $y_2^*(t) = \sin(\pi t/t_f)$, where t_f is the final time.

If we apply learning control to achieve zero tracking error of $y_1(y_2)$ every sample time, then the output $y_2(y_1)$ is totally uncontrolled, and the resulting error histories are shown in Fig. 1. We now apply the alternating output method together with integral control-based learning to control both variables simultaneously. Usually, in applying learning control we start in the zeroth repetition by using feedback control only and commanding the desired output. It is not immediately obvious what scalar command to give the system when we have two different desired outputs, one for y_1 and the other for y_2. Here we set the command starting from step 0 and going to step p-1 to equal the successive entries in the $\underline{y}^*$. Thus, the scalar command to the control system is alternating between the two desired outputs. Note the one time step shift in the command compared to $\underline{y}^*$, which reflects the one time step delay going through the control system. Setting the learning gains to produce eigenvalues of 0.5 and 0.7, the error histories at repetition 30 are the small errors shown in Fig. 2.

Ripple Suppression

In digital control systems, even if the output at the sample times has zero error, the intersample tracking can be poor. To decrease variations from the desired trajectory between sample times, often people ask to have not only the right output at the sample times but the right derivative of the output at the sample times. In higher-order systems, one can imagine trying to match still further derivatives if they can be measured. If not, we can still attempt to match them based on a model using methods in the next section.

Now we consider a single input system with both position and velocity as outputs, and we wish to control both. These output variables are very closely related, in contrast to the two output variables in the preceding example. It is instructive to consider the frequency content of the ripple signal. In the initial time steps there can be some transient behavior, and in steady state, because the ripple is not visible at the sample times, the minimum frequency is the Nyquist frequency, and one can have all harmonics. That corresponds to periods of two time steps, one time step, etc. Because the errors are at the Nyquist frequency or harmonics, the digital control is unable to influence these position errors directly, but by controlling velocity at the sample times one does have some control over these errors.

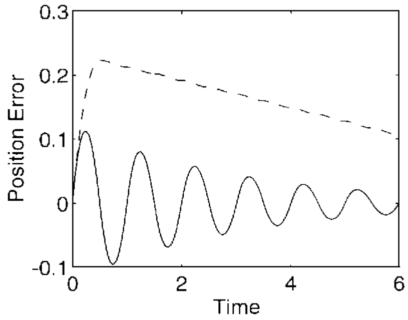

Fig. 3 Ripple between sample times when position error is zero at sample times (——) and position error when velocity error is zero at sample times (– – –).

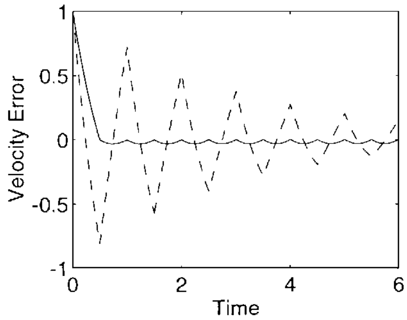

Fig. 4 Velocity error histories corresponding to Fig. 3 (– – –, position learning and ——, velocity learning).

Here we investigate the ability of the learning control methods of the preceding section to simultaneously eliminate tracking errors at the chosen sample times and to control ripple between sample times.

Obtaining Zero Tracking Error of Both Position and Velocity at Sample Times

Consider the following system:

$$\ddot{y} + \dot{y} + y = u \tag{27}$$

where input u is the command $y^*(t) = t$ coming through a zero-order hold and the sample time is $T = \frac{1}{2}$. For comparison purposes, Fig. 3 shows the position error for all time when the discrete-time learning control has achieved zero position error at every sample time. Note the rather substantial ripple between the sample times. In the early learning control literature, there was much discussion of the difficulty of learning position in second-order mechanical systems, and instead zero position error was sought by starting at the correct initial condition and learning to have zero velocity error. Also in Fig. 3 is the position error that results from asking for zero velocity error at each time step. Because this is a digital system, and there is ripple between sample times, having zero velocity error at every sample time does not guarantee zero position error, as can be seen in Fig. 3. Figure 4 shows the corresponding velocity errors after each learning process has converged.

Figure 5 gives the results when we apply the methods of the preceding section to this problem, to attempt to decrease the ripple. Figure 6 gives the corresponding velocity errors. When a generalized hold is used to learn both position and velocity at even numbered sample times, as shown by the solid lines, the ripple is greatly attenuated. Thus, the approach can be quite effective.

On the other hand, alternately learning position and then velocity as the sample times progress results in the dashed curve (the desired position is matched at odd time steps and the desired velocity at even). The ripple is, in fact, increased by roughly a factor of two. Thus, we conclude that this alternating method for controlling the additional velocity variable is not advisable when ripple attenuation is the objective. Because ripple is at a high frequency equal to the Nyquist frequency or harmonics, it is apparently easy to match the desired velocity at the intermediate time steps without keeping the position error small. This method should be reserved for problems such as that in Eq. (26).

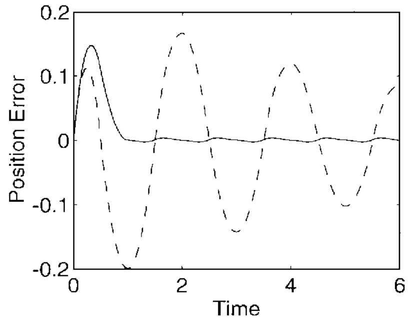

Fig. 5 Position errors after convergence when position and velocity are ——, both learned every other time step and - - -, learned alternately.

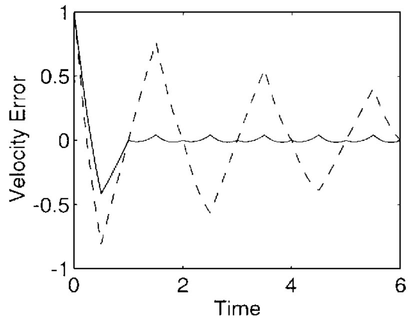

Fig. 6 Velocity error histories corresponding to Fig. 5.

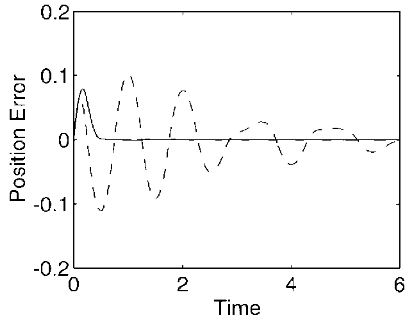

Fig. 7 Position error vs time as in Fig. 5, except that the sample time is cut in half.

The approach needed to get both zero position and zero velocity error required us to control these quantities every other time step, rather than every time step. In a sense this corresponds to cutting the Nyquist frequency in half. Because ripple is at this frequency or higher, one might wonder what price we have paid in not controlling the variables every sample time. Figure 7 answers this question for this example, by cutting the step size in half and again applying the methods. Thus, both position and velocity are now controlled every $T = \frac{1}{2}$. As expected, the already small ripple in Fig. 5 is made substantially smaller, becoming almost undetectable after the first two time steps.

Optimizing the Chosen Velocity to Minimize Ripple with a Generalized Hold

In the preceding subsection we have improved the ripple by asking for the velocity (and perhaps higher derivatives) to match the desired continuous-time position/velocity history at every other sample time. Because ripple is a high-frequency phenomenon, having the right velocity at the start of a time step is not as closely related to the error at the end of the time step as one might wish. It is possible that the ripple could be still smaller by not asking for the desired velocity at the start of a step but rather optimizing what velocity we ask for, choosing it to minimize the position error for all time. This type of optimization is sometimes done in steady-state situations where one only optimizes one value that is used repeatedly. Learning control is designed for situations when we want to reach zero error in tracking a finite-time trajectory that can be specified arbitrarily. Here we consider how the optimization of the choice of specified velocities can be applied in the learning control problem.

The problem statement is as follows. Given a continuous-time system $\dot{x} = A_c x + B_c u$, $y = Cx$, with command input u coming through a zero-order hold. Let the desired output position history be $y^*(t)$, and we impose zero tracking error every two time steps, $y(2kT) = y^*(2kT)$, $k = 1, 2, 3, \ldots, p/2$ (p even). Find the velocities $v(2kT) = \dot{y}(2kT)$ to require, such that the following position error function is minimized:

$$J = \int_0^{pT} [y^*(t) - y(t)]^2 \, \mathrm{d}t \tag{28}$$

This problem can be written as a quadratic program, requiring minimization of a quadratic function of many variables, subject to linear inequality constraints, i.e., minimize,

$$\begin{aligned} J &= \sum_{j=0}^{(p-2)/2} J(2j) \\ J(2j) &= X^T(2j)\Theta_1 X(2j) + \Theta_2(2j)X(2j) + \Theta_3(2j) \\ J(2j) &= [x^T(2jT) \quad u^T(2jT) \quad u^T((2j+1)T)]^T \end{aligned} \tag{29}$$

by choice of the set of velocities $v(2kT)$ [which can be chosen as the second state variable $x_2(2kT)$], subject to the linear equality constraint of the discrete-time version of the state equations (with system matrices A, B, and C) relating the $x(2jT)$ variables, and subject to the requirement that the control choices cause the outputs to match $y(2kT) = y^*(2kT)$,

$$y^*(2kT) = CA^2 x(2kT) + [CAB \quad CB]\begin{bmatrix} u(2kT) \\ u((2k+1)T) \end{bmatrix} \tag{30}$$

The Θ_1, $\Theta_2(2j)$, and $\Theta_3(2j)$ in Eq. (29) are determined by writing

$$J(k) = \int_{kT}^{(k+1)T} [y^*(t) - y(t)]^2 \, \mathrm{d}t + \int_{(k+1)T}^{(k+2)T} [y^*(t) - y(t)]^2 \, \mathrm{d}t$$

and substituting for $y(t)$ the solution of the differential equation on these two intervals

$$y(\tau + kT) = C\alpha(\tau)x(kT) + C\beta(\tau)u(kT)$$

$$\begin{aligned} &y(\tau + (k+1)T) \\ &\quad = C\alpha(\tau)Ax(kT) + C\alpha(\tau)Bu(kT) + C\beta(\tau)u((k+1)T) \end{aligned}$$

$$\alpha(\tau) = \exp(A_c\tau), \qquad \beta(\tau) = \int_0^{\tau} \exp[A_c(\tau - t')]B \, \mathrm{d}t'$$

and then evaluating the resulting integrals.

Controlling Additional Output Variables That Are Not Measured

If we wish to use learning control to improve the tracking of a variable that is not being measured, we need a method of determining the error of that variable, and the method will necessarily depend on a model of the system. There will be no way to correct for inaccurate estimates of the variable due to use of an inaccurate model in the estimator unless a learning control method is used that involves identification[18–21] and the identification results are coupled to the desired variable by physical understanding.

Here we make use of the nominal model and construct two different ways of generating values for the unmeasured variable from the measurements. First, consider an observer

$$\begin{aligned} &\hat{x}[(k+1)T] = \\ &\qquad A\hat{x}(kT) + Bu(kT) + w(kT) + F[y(kT) - C\hat{x}(kT)] \end{aligned} \tag{31}$$

Suppose again that the variable of interest is velocity. Then one can start with the continuous-time system represented in controllable canonical form. In this form the velocity is a state variable, or in the case of a system with zeros, velocity is representable in terms of the state variables in a simple way. Then position measurements input into Eq. (31) produce an estimate of velocity as well as other states. The learning control problem involves repeating a task, always starting from the same initial conditions. Presuming that this initial state is known, then we have the correct initial condition to use in the observer, and if in addition the model is correct and there is no plant or measurement noise, then the observer produces the correct velocities every time step.

In general, there are observer transients, which would occur, for example, if we did not know all initial states, and the error in constructing the unmeasured states will converge to zero asymptotically with time during a repetition. An alternative method of constructing the desired variables that avoids this problem can be generated in the following way.[30] Use Eq. (1) to go p steps ahead

$$x(k) = A^p x(k-p) + C'\bar{u}(k-p) \tag{32}$$

where $C' = [A^{p-1}B \ \cdots \ AB \ \ B]$ and $\bar{u}(k-p)$ is a column vector with $u(k-p)$ as the top partition and ending with $u(k-1)$ as the final partition. Define $\bar{y}(k-p)$ analogously to $\bar{u}(k-p)$, and use Eq. (32) repeatedly to produce

$$\bar{y}(k-p) = O'x(k-p) + T'\bar{u}(k-p) \tag{33}$$

where O' is the observability matrix with column partitions starting with C, CA, and going through CA^{p-1}; T' is a Toeplitz matrix, which has zero partitions on and above the diagonal; the subdiagonal contains CB in every element; and the ith subdiagonal contains $CA^{i-1}B$ in every element. Adding and subtracting $M\bar{y}(k-p)$ from the right-hand side of Eq. (32) and replacing the $\bar{y}(k-p)$ in the added term by Eq. (33) produces

$$\begin{aligned} x(k) &= (A^p + MO')x(k-p) \\ &\quad + (C' + MT')\bar{u}(k-p) - M\bar{y}(k-p) \\ &= (C' + MT')\bar{u}(k-p) - M\bar{y}(k-p) \end{aligned} \tag{34}$$

We choose the value of M to satisfy $A^p + MO' = 0$, which eliminates dependence on the state $x(k-p)$. This can be done if the system is observable and the value of p is large enough that the observability matrix has reached full rank. The final expression in Eq. (34) gives the full state as a linear combination of p previous inputs and p previous measurements, and no transients are involved in the reconstruction of the unmeasured states.

Given the estimated velocities, we can now apply any of the described learning law equations (9) or (11) with Eq. (6) or (16), as formulated for either the alternating output method or the generalized hold method. We simply substitute the estimated velocity errors in place of measured velocity errors.

Consider the use of integral control-based learning using the alternating output method, and let position be controlled on the odd numbered time steps. The first time step will converge to zero position error provided that $I - C_1 B\phi_1$ is less than one in magnitude. Once this time step has converged, the estimate of the velocity for the next time step using observer (31) is a stabilized function of $u(T)$ with the same linear relationship $C_2 B$ as with true measurements of velocity. Incorrect values of the parameters in Eq. (31) (except for those in $C_2 B$) can be lumped into producing a modified, or effective, desired velocity for that step. Convergence to this effective desired velocity is accomplished provided the true $I - C_2 B\phi_2$ is less than one in magnitude. Once this time step has converged, then the next time step converges to the desired position according to $I - C_1 B\phi_1$. Hence, we can prove that this learning control method has the desirable property that it converges to zero true position error at the position time steps and to zero estimated velocity at the velocity time steps, for all sufficiently small learning gains (of the correct sign).

Conclusions

Three different learning control laws, each having its own advantages, have been generalized so that they can improve tracking of more measured output variables than input variables. The method that alternates the output variables as the time steps progress is shown to be particularly simple when using integral control-based learning. It is suggested that one may wish to adjust the gains or scaling for the different outputs to achieve similar learning rates for each. It is shown that this approach is not effective when one is interested in decreasing ripple by learning to match the desired velocity as well as the desired position. On the other hand, the generalized hold approach, which seeks to control all outputs at the same time steps, but to skip steps, is found to be significantly more complicated when using the integral control-based learning, but is equally natural when using the other learning laws. The price that is paid is that multi-input/multi-output equations are now involved. It is shown that this can be quite effective when used for purposes of ripple suppression. Tracking performance is then improved between sample times at the expense of no longer obtaining zero tracking error at some sample times. We also show how one can apply these methods when one wishes to improve the tracking of an extra variable that is not being measured. It is shown that learning control can obtain zero tracking error to all measured variables at the appropriate sample times and zero estimated tracking error of the unmeasured variables at their sample times.

References

[1]Longman, R. W., Arimoto, S., Phan, M., and Juang, J.-N., *Supplementary Material, Tutorial on Learning Control*, Inst. of Electrical and Electronics Engineers Conf. on Intelligent Control, 1991.

[2]Uchiyama, M., "Formulation of High-Speed Motion Pattern of a Mechanical Arm by Trial," *Transactions of the Society for Instrumentation and Control Engineers*, Vol. 14, 1978, pp. 706–712 (in Japanese).

[3]Craig, J. J., "Adaptive Control of Manipulators Through Repeated Trials," *Proceedings of the American Control Conference*, Vol. 3, San Diego, CA, 1984, pp. 1566–1573.

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

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

[6]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; also Dept. of Electrical Engineering, TR EE8546, Univ. of Newcastle, Newcastle, Australia, 1985.

[7]Tomizuka, M., Tsao, T. C., and Chew, K. K., "Analysis and Synthesis of Discrete-Time Repetitive Controllers," *Journal of Dynamic Systems, Measurement and Control*, Vol. 111, No. 3, 1989, pp. 353–358.

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

[9]Lee, S. C., and Longman, R. W., "Indirect Decentralized Repetitive Control," *Advances in the Astronautical Sciences*, Vol. 82, Pt. 1, 1993, pp. 93–109.

[10]Huang, Y.-C., and Longman, R. W., "The Source of the Often Observed Property of Initial Convergence Followed by Divergence in Learning and Repetitive Control," *Advances in the Astronautical Sciences*, Vol. 90, 1996, pp. 555–572.

[11]Longman, R. W., and Huang, Y.-C., "Use of Unstable Repetitive Control for Improved Tracking Accuracy," *Adaptive Structures and Composite Materials: Analysis and Applications*, AD-Vol. 45/MD-Vol. 54, American Society of Mechanical Engineers, New York, 1994, pp. 315–324.

[12]Jang, H. S., and Longman, R. W., "A New Learning Control Law with Monotonic Decay of the Tracking Error Norm," *Proceedings of the Thirty-Second Annual Allerton Conference on Communication, Control, and Computing*, Coordinated Science Lab., Univ. of Illinois, Urbana, IL, 1994, pp. 314–323.

[13]Jang, J. S., and Longman, R. W., "An Update on a Monotonic Learning Control Law and Some Fuzzy Logic Learning Gain Adjustment Techniques," *Advances in the Astronautical Sciences*, Vol. 90, 1996, pp. 301–318.

[14]Jang, H. S., and Longman, R. W., "Design of Digital Learning Controllers Using a Partial Isometry," *Advances in the Astronautical Sciences*, Vol. 93, Pt. 1, 1996, pp. 137–152.

[15]Wang, Y., and Longman, R. W., "Use of Non-Causal Digital Signal Processing in Learning and Repetitive Control," *Advances in the Astronautical Sciences*, Vol. 90, 1996, pp. 649–668.

[16]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*, Dept. of Electrical Engineering, Princeton Univ., Princeton, NJ, 1994, pp. 951–958.

[17]Elci, H., Longman, R. W., Phan, M., Juang, J.-N., and Ugoletti, R., "Discrete Frequency Based Learning Control for Precision Motion Control," *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, Inst. of Electrical and Electronics Engineers, New York, 1994, pp. 2767–2773.

[18]Elci, H., Longman, R. W., Phan, M., Juang, J.-N., and Ugoletti, R., "Automated Learning Control Through Model Updating for Precision Motion Control," *Adaptive Structures and Composite Materials: Analysis and Applications*, Vol. AD. 45/Vol. MD 54, American Society of Mechanical Engineers, New York, 1994, pp. 299–314.

[19]Longman, R. W., and Wang, Y., "Phase Cancellation Learning Control Using FFT Weighted Frequency Response Identification," *Advances in the Astronautical Sciences*, Vol. 93, Pt. 1, 1996, pp. 85–101.

[20]Lee-Glauser, G., Juang, J.-N., and Longman, R. W., "Comparison and Combination of Learning Controllers: Computational Enhancement and Experiments," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, 1996, pp. 1116–1123.

[21]Phan, M., and Longman, R. W., "Indirect Learning Control with Guaranteed Stability," *Proceedings of the 1989 Conference on Information Sciences and Systems*, Electrical Engineering Dept., Johns Hopkins Univ., Baltimore, MD, 1989, pp. 125–131.

[22]Longman, R. W., and Ryu, Y. S., "Indirect Learning Control Algorithms for Time-Varying and Time-Invariant Systems," *Proceedings of the Thirtieth Annual Allerton Conference on Communication, Control, and Computing*, Coordinated Science Lab., Univ. of Illinois, Urbana, IL, 1992, pp. 731–741.

[23]Longman, R. W., Phan, M., and Juang, J.-N., "An Overview of a Sequence of Research Developments in Learning and Repetitive Control," *Proceedings of the First International Conference on Motion and Vibration Control* (MOVIC) (Yokohama, Japan), Japan Society of Mechanical Engineers, Tokyo, Japan, 1992, pp. 952–957.

[24]Longman, R. W., "The Range of Possible Iterative Learning Control Laws to Optimize Robot Tracking," *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, Special Issue on Applied Stochastics and Optimization, Vol. 76, Supplement No. 3, 1996, pp. 287–290.

[25]Longman, R. W., "Learning Control, Repetitive Control, and Vibration Annihilation," *Proceedings of the International Conference on Structural Dynamics, Vibration, Noise and Control*, Plenary Lecture, Chinese Society for Vibration Engineering, Hong Kong, PRC, 1995, pp. 80–85.

[26]Kabamba, P. T., "Control of Linear Systems Using Generalized Sample-Data Hold Functions," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 9, 1987, pp. 772–783.

[27]Hara, S., Tetsuka, M., and Kondo, R., "Ripple Attenuation in Digital Repetitive Control Systems," *Proceedings of the 29th IEEE Conference on Decision and Control* (Honolulu, HI), Inst. of Electrical and Electronics Engineering, New York, 1990, pp. 1679–1684.

[28]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, New York, 1993, pp. 665–671.

[29]Arimoto, S., Naniwa, T., and Suzuki, H., "Robustness of P-type Learning Control with a Forgetting Factor for Robotic Motions," *Proceedings of the 29th IEEE Conference on Decision and Control* (Honolulu, HI), Inst. of Electronics and Electrical Engineers, New York, 1990, pp. 2640–2645.

[30]Goodzeit, N. E., and Phan, M. Q., "Exact System Identification in the Presence of Completely Unknown Periodic Disturbances" Dept. of Mechanical and Aerospace Engineering, Rept. No. 2096, Princeton Univ., Princeton, NJ, Jan. 1997.