Comparison and Combination of Learning Controllers: Computational Enhancement and Experiments

Gina J. Lee-Glausers
Clarkson University, Potsdam, New York 13699-5725
Jer-Nan Juang
NASA Langley Research Center, Hampton, Virginia 23681
and
Richard W. Longman
Columbia University, New York, New York 10027

Five discrete-frequency linear learning-control laws are compared and experimentally tested. These include simple integral-control-based learning using a single learning gain, phase-cancellation learning control, a contraction-mapping learning-control law with monotonic decay of the error norm, and learning controllers that invert the system model and the observer model. The inversion designs converge the fastest initially, but phase cancellation with identification updates and the contraction-mapping method with model updates have better stability robustness properties. The learning control approaches are combined to obtain the advantages of each, by using inversion methods for the first few repetitions, followed by a more robust method. It is demonstrated that the computation of the learning action can be made in the frequency domain using fast Fourier transform methods, with as much as 94% reduction in computation time. In experiments on a Robotics Research Corporation robot, the learning-control laws result in a reduced rms tracking errors for all joints, by a factor of close to 3 orders of magnitude.

Introduction

A TYPICAL industrial manipulator robot performing a pre-assigned repetitive task of tracking a desired trajectory makes the same mistakes repeatedly. This is an ideal application for use of learning-control methods that can eliminate the deterministic part of the tracking error. A learning controller improves the tracking error by learning from previous repetitions of the task. Various learning-control laws can eliminate the majority of the tracking errors with a minimum of a priori knowledge of the system. However, additional knowledge about the system can improve the convergence rate and improve the learning transients, as shown by Elci et al.1,3,9 and in this paper.

Some of the earlier basic theory and convergence analyses in learning and repetitive control have been reported in Refs. 4—9. Recently, with increasing interest in learning and repetitive control, many approaches to learning have been developed, among them, continuous time,14,16 discrete time,1,3,12,17 frequency domain,2,3,18 neural networks,19,20 and adaptive control.21-25

In this paper, a zero-phase filter is used to limit the learning-controller bandwidth, and thus improve the stability robustness of the system. An alternative that also increases the robustness of the learning process performs identification updates, and we use this on some learning-control approaches. The updates are obtained using the System/Observer/Controller Identification Toolbox (SOCIT),25-28. Several linear learning-control laws have been proposed that have good learning properties,3,17,29 but they require the use of many learning gains. Here we show how linear learning controllers can be efficiently computed using fast Fourier transform (FFT) methods. The methods are applied in learning-control experiments on a commercial robot, shown in Fig. 1.

Time-Domain Models

The learning controllers in this paper are used to modify the command to a closed-loop control system. These learning controllers are based on discrete time-domain formulations of a feedback system model and on an observer model of the closed-loop system. The mathematical formulations of the learning controller developed in Refs. 15 and 29 are reviewed in this section.

System Model

Consider a general multivariable time-invariant discrete-time system expressed as

\[
\begin{align*}
\mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{w}(k) \\
\mathbf{y}(k) &= C\mathbf{x}(k) + D\mathbf{u}(k)
\end{align*}
\]

where \( \mathbf{x} \) is the \( n \times 1 \) state vector, \( \mathbf{u} \) is the \( m \times 1 \) learning-control input vector, \( \mathbf{y} \) is the \( 1 \times 1 \) output vector, and \( \mathbf{w} \) is a forcing function that includes repetitive time-varying disturbances and the desired trajectory input to the feedback control. The matrix \( A \) represents the closed-loop system of an existing feedback controller. The direct transmission term \( D \) is considered to be zero because the feedback control system will not be able to react instantaneously to a change.
in input when the output is position or velocity (such a term could be needed if acceleration measurements were used). Here, an \( I \) time-step repetitive process is considered, with the same initial condition \( x(0) \) for each repetition. The output solution of Eq. (1) for each time step, in terms of the inputs and disturbances, is

\[
y(k) = C \hat{A}^k x(0) + \sum_{i=0}^{k-1} C \hat{A}^{k-i-1} B u(i) + \sum_{i=0}^{k-1} C \hat{A}^{k-i-1} w(i) \tag{2}
\]

for \( k = 1, 2, \ldots, I \). A backward difference operator \( \delta_j(\cdot) \) is defined to convert the time domain into the repetition domain in two successive repetitions as

\[
\delta_j z(k) = z_j(k) - z_{j-1}(k)
\]

where \( j \) denotes the repetition index. Substituting Eq. (3) into Eq. (2) and noting that the initial conditions and the forcing function repeat each repetition, so that \( \delta_j x(0) = 0 \) and \( \delta_j w(k) = 0 \), yields

\[
\delta_j y = P \delta_j u \tag{4}
\]

where \( y = [y^T(1) \ y^T(2) \ \ldots \ y^T(I)]^T \), \( u = [u^T(0) \ u^T(1) \ \ldots \ u^T(I-1)]^T \), and

\[
P = \begin{bmatrix}
C B & 0 \\
C A B & C B \\
\vdots & \vdots \\
C A^{I-1} B & C A^{I-2} B & \cdots & C A B & C B
\end{bmatrix}
\]

The above equation shows that the building blocks of the \( P \) matrix are the system Markov parameters. Therefore, if the system Markov parameters are known, then the change in the learning-control signal from the previous repetition needed to produce zero tracking error can be calculated directly. However, if the system Markov parameters are not known precisely, a learning-control law can be used to converge the tracking error to zero.

We consider learning controllers that are linear functions of the error history:

\[
\delta_j u = L e_{j-1} \tag{5}
\]

where \( L \) is the learning-gain matrix and \( e_{j-1} = y^* - y_{j-1} \) is the tracking error, \( y^* \) is the desired output history. Substituting the above two equations into Eq. (4) results in a formula for the change in the tracking-error history from one repetition to the next:

\[
e_j = (I - PL)e_{j-1} \tag{6}
\]

To guarantee convergence of the tracking error to zero, the matrix \( I - PL \) must be asymptotically stable, i.e., its eigenvalues must satisfy \( |\lambda| < 1 \). There are many possible choices for the learning-gain matrix \( L \) that satisfy this stability criterion. In particular, the choice of \( L = P^{-1} \) converges to zero error in one repetition, provided one knows the \( P \) matrix.

**Observer Model**

Juang et al.\(^{28}\) facilitate system identification by identification of an observer for the system directly from data as the first step in the process of identifying the system. Phan and Juang\(^{29}\) study the use of observer models in learning control. Consider an observer of the form

\[
x(k+1) = \hat{A} x(k) + [B \ -M] \begin{bmatrix} u(k) \\ y(k) \end{bmatrix} + w(k) \tag{7}
\]

where \( \hat{A} = A + MC \) and \( M \) can be represented as an observer gain without the presence of disturbances. The observer-model output is

\[
y(k) + \sum_{i=0}^{k-1} C \hat{A}^{k-i-1} M y(i) = C \hat{A}^k x(0)
\]

\[
+ \sum_{i=0}^{k-1} C \hat{A}^{k-i-1} B u(i) + \sum_{i=0}^{k-1} C \hat{A}^{k-i-1} w(i) \tag{8}
\]

for \( k = 1, 2, \ldots, p \) where \( p \) represents the number of observer Markov parameters. For \( k \geq p \), \( \hat{A}^k \) is assumed to be identically equal to zero for the chosen value of \( p \). After applying a backward difference operator to Eq. (8), the following equation is obtained:

\[
Q \delta_j y = R \delta_j u \tag{9}
\]

where

\[
Q = \begin{bmatrix}
1 & 0 \\
CM & 1 \\
C \hat{A}^{p-1} M & \cdots & CM & 1 \\
0 & \cdots & \cdots & \cdots
\end{bmatrix}
\]

\[
P = \begin{bmatrix}
C B & 0 \\
C \hat{A} B & C B \\
\vdots & \vdots \\
C A^{p-1} B & \cdots & C A B & C B
\end{bmatrix}
\]

The system and observer Markov parameters are related, such that \( P = Q^{-1} R \). The \( Q^{-1} \) always exists because the \( Q \) matrix is always lower triangular with identity matrices on its main block diagonal; hence it is square and full rank.

Equation (9) in terms of the error is

\[
Q \delta_j \varepsilon = - R \delta_j u \tag{10}
\]

Here, we define \( Q e \), as a filtered error \( \varepsilon \), so that

\[
\delta_j \varepsilon = - R \delta_j u \tag{11}
\]

The observer learning law is chosen as a linear function of the filtered error:

\[
\delta_j u = L \varepsilon \tag{12}
\]

Upon substitution of Eq. (12) into Eq. (11), the change in the filtered-error history from one repetition to the next is given by

\[
e_j = (I - RL) e_{j-1} \tag{13}
\]

The filtered error converges to zero when the learning-gain matrix \( L \) is selected to satisfy \( |\lambda| \ (I - RL) < 1 \). The convergence of filtered error to zero also implies convergence of the actual error to zero because they are related through the nonsingular \( Q \) matrix. In particular, the choice \( L = R^{-1} \) produces convergence to zero error in one repetition, provided the matrix is known precisely.

**Frequency-Domain Models**

The learning control updates in Ref. 3 can be computed in the time domain or computed in the frequency domain and then an inverse Fourier transform is used. The latter computation is seen to be much faster. Here, we show that various other learning-control algorithms that are not based on frequency response thinking also can be computed using the frequency domain, with significant benefit. Therefore, the frequency-domain formulation of the learning-control problem is given in this section, for both system and observer models.

The one-sided \( z \) transform,

\[
X(z) = Z[x(k)] = \sum_{k=0}^{\infty} x(k) z^{-k}
\]

of the system model, Eq. (1), with zero initial conditions is

\[
X(z) = (z I - A)^{-1} B U(z) + (z I - A)^{-1} W(z)
\]

\[
Y(z) = G(z) U(z) + R w(z) \tag{14}
\]
where \( G(z) = C(zI - A)^{-1}B \) and \( R_w(z) = C(zI - A)^{-1}W(z) \).

The reader is referred to Refs. 30 and 31 for treatments of the relationships employed here, between \( z \) transforms, discrete Fourier transforms (DFT), inverse DFT (IDFT), and FFT.

### System Model

The \( z \) transform of the learning controller is

\[
d_1 U(z) = U_j(z) - U_{j-1}(z) = L(z)E_{j-1}(z) \tag{15}
\]

The above equation shows that a convolution in the time domain is translated into a multiplication in the frequency domain. Both sides of Eq. (14) are multiplied by \( G(z) \), added to \( R_w(z) \), and then subtracted from \( Y^*(z) \) to acquire an equivalent of Eq. (6) in the frequency-domain representation as

\[
E_j(z) = [I - G(z)L(z)]E_{j-1}(z) \tag{16}
\]

where the unit pulse response \( G(z) \) coefficients in the time domain are the Markov parameters. Equation (16) will converge to the desired trajectory monotonically and geometrically when the unit pulse response component of the filtered error is

\[
\text{Observer Model}
\]

\[
\text{where the output tracking error in the frequency domain by acknowledging that Q is a nonsingular matrix as}
\]

\[
E_j(z) = Q^{-1}(z)[I - R(z)L(z)]Q(z)E_{j-1}(z) \tag{18}
\]

The condition for monotonic decay of each steady-state frequency-response component of the filtered error is \( |\lambda_i[I - R(z)L(z)]| < 1 \), \( i = 1, 2, \ldots, q \), for each frequency component of the error for \( z = e^{j2\pi n/N} \), \( n = 0, 1, \ldots, N - 1 \). This performance requirement corresponds to the Nyquist plot of \( G(z) \) lying within a unit circle centered at \( z = 1 \). This is not the stability boundary but is used as a condition to be imposed to have the learning process exhibit good transient behavior.

### Observer Model

The \( z \) transform of the observer-model learning controller, Eq. (12), for the actual tracking error is

\[
U_j(z) = U_{j-1}(z) + L(z)Q(z)E_{j-1}(z) \tag{17}
\]

Substituting Eq. (17) into the \( z \) transform of Eq. (10) produces the output tracking error in the \( z \) domain by acknowledging that Q is a nonsingular matrix as

\[
E_j(z) = Q^{-1}(z)[I - R(z)L(z)]Q(z)E_{j-1}(z) \tag{18}
\]

The condition for monotonic decay of each steady-state frequency-response component of the filtered error is \( |\lambda_i[I - R(z)L(z)]| < 1 \), \( i = 1, 2, \ldots, q \), for each frequency component of the error for \( z = e^{j2\pi n/N} \), \( n = 0, 1, \ldots, N - 1 \). This performance requirement corresponds to the Nyquist plot of \( G(z) \) lying within a unit circle centered at \( z = 1 \). This is not the stability boundary but is used as a condition to be imposed to have the learning process exhibit good transient behavior.

### Learning Controllers

Here, four learning-controller designs together with simple single-gain learning control are studied to evaluate their effectiveness in producing zero tracking error with a good convergence rate.

#### Design 1: Single-Gain Controller

Let the learning gain matrix be simple block diagonal, i.e., \( L = \text{diag}(\Phi, \Phi, \ldots, \Phi) \), such that the eigenvalues of \( I - PL \) depend only on \( C \) and \( \Phi \). Therefore, as long as the eigenvalues of \( I - CB\Phi \) are less than 1, the stability criterion is satisfied. Equation (5) for this design can be written in the time domain as

\[
u_j(k) = u_{j-1}(k) + \Phi[f'(k + 1) - y_{j-1}(k + 1)] \tag{19}
\]

Such a learning controller is classified as an integral-control-based learning control\[15,16\] i.e., it can be obtained by applying integral-control concepts in the repetition domain for each time step in the trajectory. In the repetitive-control context, it is called a \( p \)-controller.\[8\] In continuous time, it is referred to as the proportional learning law.\[9\]

In the experiments performed here to compare different learning controllers, a constant value \( \Phi = 1 \) is used, which is considered as a natural learning gain.\[5\] As in Refs. 1 and 2, a zero-phase low-pass filter is used to guarantee monotonic and geometric convergence below the cutoff frequency (chosen as \( 3 \) Hz).

#### Design 2: Phase-Cancellation Controller

This learning-control law was designed by Elci et al.\[2\] to cancel the phase of the system for all frequencies whose magnitude is attenuated by the closed-loop system. Let \( G(W_N) \), where \( W_N(n) = e^{j2\pi n/N} \) in Eq. (16), be square and diagonalizable for all frequencies of interest. Then, each element \( \Phi(n) \) of the learning gain \( L(W_N) \) in Eq. (15) for a multi-input/multi-output (MIMO) system learning-control law is

\[
L(W_N) = V^{-1}(W_N)A_R(W_N)V(W_N) \tag{20}
\]

where \( V \) and \( A_R \) are the eigenvector and the eigenvalue matrices of the transfer matrix \( G(W_N) \), respectively. Here, \( N = 2l - 1 \) is in the time domain. The eigenvalues of the matrix \( A_R \) are

\[
\lambda_R(n) = \begin{cases} -\lambda_i[G(W_N)] & |\lambda_i[G(W_N)]| \leq 1 \\ -\lambda_i[G(W_N)] & |\lambda_i[G(W_N)]| > 1 \end{cases}
\]

where \( n = 0, 1, \ldots, l - 1 \) and \( i = 1, 2, \ldots, q \). To obtain a time-domain sequence of learning gains \( \Phi(k) \) from the IDFT, the complex conjugate of \( L(W_N) \) in reverse order \( \{l, l - 1, \ldots, 1 \} \) is used.

#### Design 3: Contraction-Mapping Controller

The contraction-mapping learning-control law was developed in the time domain by Jang and Longman\[7\] to monotonically decrease the rms error histories at each repetition. When each repetition contains many data points, the computations needed in this control law can require considerable computer time because of the large number of learning gains. Here, we show that under appropriate conditions one can make the computation in the frequency domain, with significant benefit in computation time.

Taking the norms of each side of Eq. (16) in \( L_2 \), i.e., using the Euclidean norm, one can guarantee monotonic convergence of the error norm when

\[
\frac{||I - G(z)L(z)||E(z)||}{||E(z)||} < 1 \quad \forall \text{E}(z) \neq 0 \tag{21}
\]

where \( ||x||_2 = \sqrt{x^T x} \). The above equation can be expressed in terms of the spectral norm as \( ||I - G(z)L(z)|| < 1 \).

One way to view this learning law is to consider trying to minimize the sum of the squares of the tracking errors, \( f[U_j(z)] = \frac{1}{2} E_j(z)E_j(z) \), assuming that the future-time learning-control signals do not change. Using the steepest descent method to minimize the above cost function results in

\[
U_j(z) = U_{j-1}(z) - c\frac{\partial f}{\partial U_{j-1}} = U_{j-1}(z) - c\frac{\partial E_{j-1}}{\partial U_{j-1}}E_{j-1}(z) \tag{22}
\]

where \( c \) is a scalar that influences the convergence rate and \( \frac{\partial f}{\partial U_{j-1}} \) contains the steepest descent gradient components. The learning control gain of the above equation in the frequency domain is

\[
L(W_N) = \frac{\partial E_{j-1}}{\partial U_{j-1}} = cG^T(W_N) \tag{23}
\]

This corresponds in the time domain to \( L = cP^T \). The inverse transform of \( L(W_N) \) corresponds to the \( \Phi(k) \) of learning matrix \( L \). Substituting this learning-gain matrix into the spectral norm \( ||I - G(z)L(W_N)||_2 \), the convergence criterion becomes

\[
\frac{||I - cG(W_N)G^T(W_N)||}{||G(W_N)||^2} = \max_i|\lambda_i[I - G(W_N)G^T(W_N)]| < 1 \tag{24}
\]

for all \( W_N(n) \) for \( n = 0, 1, \ldots, N - 1 \). The range of acceptable \( c \) values is evaluated by rewriting the above inequality for all \( I, 0 < c < 2/||G(W_N)||^2 \), so that the repetition process is monotonically decaying in the Euclidean norm of the tracking-error history for each repetition. In this study, a constant value of \( c = 1 \) is used.
Design 4: Direct-System-Inversion Controller

The direct-system-inversion learning-control law assumes that \( G(W_N) \) is square and its inverse exists. Then, the learning-gain matrix is

\[
L(W_N) = G^{-1}(W_N)
\]

(25)

This satisfies the convergence criterion for Eq. (16), giving convergence in one repetition. In practice, direct inversion of a dynamic system is likely to encounter various difficulties: One seldom has complete and accurate knowledge of the system; the system dynamics attenuate at high frequencies, producing high gains; and the data are corrupted with noise, often at high frequencies. However, because of the attractiveness of its potential to produce zero tracking error in one repetition, it is of interest to explore the usefulness of the direct-system-inversion controller.

A zero-phase fifth-order Butterworth filter with a cutoff frequency of 3 Hz is used to remove dependence or poorly known higher-frequency system dynamics.

Design 5: Direct-Observer-Inversion Controller

The direct-observer-inversion learning-control law has advantages and disadvantages similar to those of the direct-system-inversion learning control. The observer Markov parameters are computed from the input and output time histories, using SOCIET. Assuming that \( R(W_N) \) is square and invertible, the learning gain for Eq. (17) is

\[
L(W_N) = R^{-1}(W_N)
\]

(26)

which satisfies the convergence criterion for Eq. (18), producing zero error in one repetition.

Once again, a zero-phase fifth-order Butterworth filter with a cutoff frequency of 3 Hz is used.

Computation Enhancement

This section discusses the enhancement of the computation of the learning-control actions, for any linear learning-control law of the form of Eq. (5) or (12), by taking advantage of the speed of computation of convolution products using FFT methods.

A direct discrete time-domain implementation of the learning-control laws—Eqs. (5) and (12)—requires computation of a convolution summation between the time-domain learning gains and the error signal. This can be seen easily by rewriting Eq. (5) for an arbitrary time step:

\[
u_j(k) = u_{j-1}(k) + \sum_{n=1}^{i-1} \Phi(n) e_{j-1}(n)
\]

(27)

where, \( \Phi(k) \) represents the (general time-invariant) learning matrix \( L \) by

\[
\begin{bmatrix}
\Phi(0) & \Phi(-1) & \cdots & \Phi(-l + 1) \\
\Phi(1) & \Phi(0) & \cdots & \Phi(-l + 2) \\
\Phi(l - 1) & \Phi(l - 2) & \cdots & \Phi(0)
\end{bmatrix}
\]

Equation (12) can be written similarly in time series. The above equation shows that the tracking error \( e_{j-1}(k) \) is convolved with the learning gain \( \Phi(k) \). In general, the learning matrix \( L \) has \( 2l - 1 \) nonzero coefficients. Depending on the learning law, the period of the command, and the sample time, this can involve a large number of terms in the summation.

The advantage of DFT and IDFT in practical applications is largely attributable to the existence of computationally efficient FFT algorithms, which are most efficient for a data sequence with the number of entries equal to a power of 2. A data sequence of a different length is padded with zeros to reach the closest power of 2 before performing DFT or IDFT.

For a single input/single output (SISO) system, the time-domain convolution summation takes \( 2l^2 \) multiplication and addition operations. In the frequency domain, \( 2^l (1 + 8r) \) multiply operations are required where \( r = \text{ceil}[(\log_2(2l - 1)) = \log_2(2l - 1)/\log_2(2), \) i.e., \( 2^l \) is the next power of 2 larger than \( 2l - 1 \). As the length of the trajectory increases (or as the sample rate is increased), the computation speed advantage of the frequency-domain approach increases dramatically.

There are several difficulties in the use of the frequency-domain computation approach. These include leakage problems attributable to a nonperiodic finite length trajectory. Leakage causes power in the original signal to spread to the entire frequency range. Also, when a finite time trajectory is expanded in frequency components, it is as if the trajectory were repeated indefinitely in time. If the start point and the end point of the trajectory are not the same, this repetition creates a discontinuous function, for which the Fourier expansion converges to the average. At the expense of doubling the length of the trajectory, this can be avoided by appending a mirror image of the trajectory and then repeating the combination to make a continuous periodic signal, prior to computation of the FFT. There are additional differences between time and frequency domain implementations concerning the ease of real time computations.

Performance Analyses

The performance of the aforementioned learning-controller designs is evaluated using a redundant 7-degree-of-freedom Robotics Research Corp. K-series manipulator. The analytical model of the manipulator, including the analog feedback controllers for all joints, is quite complex. However, these learning-controller designs are quite simple and require relatively little system knowledge a priori. Hence, these learning-controller designs can be attractive in practice.

The desired trajectory is a cycloidal path containing a 90-deg turn followed by a return to the starting position, as shown in Fig. 2. All seven joints execute the same commanded trajectory simultaneously to produce a large, high-speed motion through the work space. Relatively large dynamic coupling between the motions of the links is achieved by making the trajectory reach the maximum allowed speed of the largest link of 55 deg/s. The sampling rate is 400 Hz.

Learning control assumes that the initial condition is the same every repetition, and that it is on the desired trajectory. However, commanding the feedback controllers for all joints to go to the desired starting point results in some joints settling at points as much as 2 deg off from the desired starting position. This is attributable to gravity disturbances. Applying learning control in this situation results in a learning command that tries to fix the initial error in the first time step, and eventually asks for unreasonable control actions. To avoid this problem, the desired trajectory is extended using 1 sec of a smooth cycloidal path from the feedback-controller initial position to the desired initial position in the experiment.1–3 The results shown
in the figures in this paper are for the desired trajectory portion of the curves.

In the following, the experimental results of individual learning-control designs are presented, and then learning-control designs are compared. The reader is referred to the conference version of this paper for a much more detailed presentation of the experimental results. Also, Refs. 1–3 give earlier experiments performed on the same test bed using the same desired trajectory with several other learning control laws.

**Single-Gain Learning Controller**

The single-gain learning control is used here as the baseline learning controller for comparison. It is effective and robust within the range of the controller bandwidth. Figure 2 shows the desired trajectory (solid line), the robot’s response to commanding this trajectory (Rep 0), and the single-gain learning controller for first and second repetitions. The robot tracking error without the learning controller reaches as much as 9 deg. By two repetitions of learning, the robot’s response is indistinguishable from the desired trajectory in this plot.

Figure 3, Design 1, shows the maximum joint-1 tracking error for various learning-controller designs. The trend of the Design-2 tracking error is similar to that of Design 1, in that the tracking error decreases quickly to a small value in two to three repetitions. However, the Design 2 tracking error is half as much as that of the Design 1. The maximum tracking error of joint 1 is reduced by 97% in the first learning repetition. Figure 4, Design 2, shows the rms tracking error of all seven joints. This figure shows that the tracking error continues to decrease gradually for all later repetitions.

**Contraction-Mapping Learning Controller**

In the contraction-mapping learning-control law, Eq. (23), the gains are the Markov parameters, i.e., the pulse response history of the feedback-control system. One might find these directly from data or create a model and use its pulse response history. Therefore, the effectiveness and the convergence rate of the method is a function of the accuracy of the model. However, if the convergence criterion, Eq. (24), is satisfied, then the rms error will converge monotonically.

Figure 3, Design 3, shows the maximum joint-1 tracking error for the first four repetitions of the contraction-cancellation learning controller. The trend of the Design-2 tracking error is similar to that of Design 1, in that the tracking error decreases quickly to a small value in two to three repetitions. However, the Design 2 tracking error is half as much as that of the Design 1. The maximum tracking error of joint 1 is reduced by 97% in the first learning repetition. Figure 4, Design 2, shows the rms tracking error of all seven joints. This figure shows that the tracking error continues to decrease gradually for all later repetitions.

**Direct System Inversion/Phase Cancellation**

In the ideal case, if one knows \( G(W_N) \) exactly, then all the eigenvalues will be zero for the monotonic convergence inequality below.
Eq. (16) and the learning will reach the desired trajectory in one repetition, i.e., the fastest possible convergence rate. However, in real life the system is not known exactly, with the accuracy of the model usually deteriorating as the frequency goes up. Because the output of physical systems decays as the frequency gets large, the inverse system can result in high gains at high frequencies, with the gains being highly influenced by noise. The zero-phase low-pass filter used here eliminates much of this problem, by limiting the model to the range of frequencies for which we are reasonably confident in the model.

In the experimental verification runs made here, the direct-system-inversion learning control is used for the first two repetitions, and then this is followed by applying the phase-cancellation learning control for the rest of the learning repetitions. This approach combines the advantages of each approach, i.e., the direct inversion eliminates the majority of the error very fast, and the phase-cancellation controller is capable of learning at higher frequencies with less system model accuracy. The system identification updates allow it to resume learning whenever the phase error of the model is too large for any important error frequency components. In this combination, the direct-system-inversion learning controller is used for the fastest possible convergence for the first two repetitions, and then the phase-cancellation learning controller is substituted to take advantage of its stability robustness properties.

The first four repetitions of joint-1 tracking error are shown in Fig. 3, Design 4. This shows a significant reduction in the first direct-system-inversion learning, which is approximately half of the phase-cancellation tracking error for the first repetition of learning (Design 2). It is observed that within two repetitions of learning, about 99% of the joint-1 maximum tracking error has been eliminated. Figure 4, Design 4, shows the combination portion of the phase-cancellation learning controller. Therefore, this shows a trend similar to that of the Design-2 tracking error.

Direct Observer Inversion/Phase Cancellation

The difficulties involved with the direct-system-inversion also apply to the direct-observer-inversion learning control. Inaccuracies in the observer Markov parameters, especially in their high-frequency content, can cause high learning gain at high frequencies. This large gain may cause repetitive application of the process to be unstable. As in the previous learning control, we combine learning-control approaches to obtain good performance. We use the direct-observer inversion learning control for the first two repetitions only, and then the phase-cancellation learning control is applied for the rest of the learning repetitions for its stability robustness.

Effectiveness of the direct-observer-inversion learning control can be seen in Fig. 3, Design 5. Within only two repetitions of learning, the tracking error is reduced to essentially zero within the resolution of the plot. The rms tracking errors for all joints are shown in Fig. 4, Design 5. These experimental results are very similar to those of the direct-system-inversion learning-control results above.

Learning-Controller Comparison

Performance comparisons of the learning-control designs are presented in this section, using the experimental results. Figure 5 shows the maximum joint-1 tracking error of the above five learning-control designs. This figure shows that the results are rather indistinguishable, except for the single-gain learning controller and the contraction-mapping learning controller. They all have a steep convergence rate for the first two repetitions, and then have a slow learning rate thereafter. The contraction-mapping learning controller seems to stop learning after the fifth repetition, making the phase-cancellation learning controller the preferred choice in this application for use in later repetitions when stability robustness is the main objective. It is possible that the contraction-mapping learning controller has better robustness properties in general, and could be preferred in certain applications for this reason. The direct-inversion with phase-cancellation learning controllers have a slightly faster convergence rate than the phase-cancellation learning control alone. This effect is to be expected, because only the phase is being cancelled, and not the gain. Table 1 shows the tracking error percent reduction for all of the learning-control approaches. There is relatively little reduction in the error after the first few repetitions. Once one has reached a certain error level, continuing the learning process may not result in improvements that are of practical significance. Variations of tracking-error frequency contents with increasing learning repetitions for direct-observer-inversion followed by phase cancellation are shown in Fig. 6. Once again, Rep 0 refers to feedback-control error. It is observed that a large spectral amplitude has been reduced at low frequencies even with two repetitions of learning. This figure shows that as the learning repetitions increase, the tracking-error frequency becomes dominated by one frequency (1.7 Hz at repetition 15) at much lower spectral amplitude than at repetition 2.

Computational time differences using a DECstation 5000/200 Ul- trix version 4.4 between the discrete time and discrete frequency domains are shown in Table 2. This shows that the discrete-frequency approach reduces the computational time by an order of magnitude. This table also shows the stability robustness and the percent tracking-error reduction after one repetition of each learning-control design.
Fig. 6 Spectral amplitudes of combined direct-observer-inversion/phase-cancellation learning controllers with increasing learning repetition.

Conclusions

Learning control for system and observer models is derived for discrete time and frequency domains. The stability criterion for each learning-control design is addressed. Four discrete-frequency learning controls (phase-cancellation, contraction-mapping, directsystem and observer-inversion) are designed and combined, and their performance is compared to each other and to simple single-gain learning control. Based on the results presented, the following conclusions may be drawn:

1) For linear learning-control laws involving a significant number of learning gains, it is recognized that the computation of the learning-control action can be viewed as a convolution sum. And it is shown that the computation of the learning action can be significantly enhanced by use of discrete frequency-domain methods. In experimental applications the benefit was a decrease in computation time by a factor of 10.

2) Model update significantly influences the learning-control performance.

3) The direct-inversion learning controllers are combined with phase-cancellation learning to capture the benefit of fast initial convergence rate with the direct-inversion learning control and the benefit of stability robustness with the phase-cancellation learning control. It is experimentally verified in this paper that one can use the direct-inversion learning controller to significantly reduce the tracking error in one learning step. One can combine two learning controllers to achieve both the fastest convergence rate in the first few repetitions and robustness in stability.

Acknowledgments

This work is supported by the National Research Council at NASA Langley Research Center, Grant 44.50.38.03. The authors are grateful to Haluk Elci for many good technical discussions.

References


