Linear Decentralized Learning Control for the Multiple Dynamic Subsystems

Soo Cheol Lee*

Abstract

The new field of learning control develops controllers that learn to improve their perfomance at executing a given task, based on experience performing this task. The simplest forms of learning control are based on the same concept as integral control, but operating in the domain of the repetitions of the task. This paper studies the use of such controllers in a decentralized system, such as a robot with the controller for each link acting independently. The basic result of the paper is to show that stability of the learning controllers for all subsystems when the coupling between subsystems is turned off, assures stability of the decentralized learning in the coupled system, provided that the sample time in the digital learning controller is sufficiently short.

I. Introduction

When a control system is required to execute the same command repeatedly, the error in following the command will be repeated (except for random disturbances). It seems a bit primative to produce the same errors every time the command is given. The new field of learning control refers to controllers that can learn from previous experience executing a command in order to improve their performance. They learn what input should be given to the system in order to have the response be the desired response. They eliminate

---

* Department of Industrial Engineering, Taegu University

the errors of the control system in executing the command, and they eliminate errors due to disturbances that repeat each time the command is given. Learning controllers aim to accomplish this with minimal knowledge of the system being controlled, and base their adjustments to the command on previous experience performing the command without relying on an a priori model of the system dynamics. There has been considerable research activity in this field in the last few years, some examples of which are given in the references [1-8].

The usual application of learning control, and the application that motivated the development of the field in the last few years is robotics performing repeated tracking commands, for example on an assembly line. Nearly all robot controllers are designed with each joint axis having its own controller, and this controller knows only feedback information about its joint angle or angle rate and nothing about the other joint variables. The effect on the motion of one joint due to motion of other joints, such as through centrifugal force effects, is treated as a disturbance that the feedback control law must take care of. Furthermore, with proper choice of the joint variables, the equations for each link can be made to involve only the control action for that link. Then the only coupling between the dynamic equations for the links is a dynamic one, with no interaction between the axes in the input and output coefficient matrices.

The question arrises, what happens if a learning controller is used with each of the separate feedback controllers of the robot arm. Such an application represents use of a decentralized learning control. A serious issue is whether the dynamic interactions in the dynamics of the systems governed by the separate learning controllers could cause the learning processes to fail to converge. It is the purpose of this paper to study under what conditions such a decentralized learning control is stable.

The system equations are nonlinear in the robotics problem. Hence, the aim of the present investigation is to obtain an understanding of the stability of a decentralized learning control system applied to a linear time varying system of differential equations with disturbances that repeat each repetition. We will approach this problem by starting with a simpler situation, considering decentralized difference equations first, both time invariant and time varying, and then progressing to both time invariant and time varying decentralized differential equation systems.

II. The Learning Control Method and Convergence Analysis

The simplest form of learning control produces the analog of this integral for every time step k, and adds it to the control action for that step. This added "integral" term is given as a sum of the error histories in all past repetitions in the discrete repetition domain, which can easily be calculated in recursive form

$$u^{j+1}(k) = \phi \sum_{i=1}^{j} e^{i}(k+1)$$
$$u^{j+1}(k) = u^{j}(k) + \phi e^{j}(k+1) \tag{1}$$

The superscripts give the repetition numbers j and j+1, and ϕ is the learning gain which the control system designer can adjust. Note that the errors involved are one step ahead of the control signal. In discrete systems the control being chosen at step k will not influence the error immediately. Here we assume that there is a one step delay. Note also that the dimension of the error column matrix, and hence the dimension of the desired trajectory matrix are both equal to the dimension of the input matrix. Thus the desired trajectory must be specified in terms of measured output variables, and the dimension chosen to match that of the input vector.

The conditions for convergence to zero tracking error for such a digital control system are given in [Phan, 1988] in a more general setting. Since they form the basis for the present work, they are summarized here. Use an underbar to denote the column matrix containing the history of a variable for all p steps of the p step repetitive operation. Then the learning control (1) is a special case of the following

$$\underline{u}^j = \underline{u}^0 + \delta_1\underline{u} + \delta_2\underline{u} + \cdot\cdot\cdot + \delta_j\underline{u}$$

$$\delta_j\underline{u} = L\underline{e}^{j-1}$$

$$L = \begin{bmatrix} \phi_1(1) & 0 & \cdots & 0 \\ \phi_2(1) & \phi_1(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_p(1) & \phi_{p-1}(2) & \cdots & \phi_1(p) \end{bmatrix} \qquad (2)$$

$$\underline{e}^j = \underline{y}^* - \underline{y}^j$$

where $\underline{y}^*$ is the column matrix of the desired trajectory history for steps 1 through p, and $\underline{u}$ is the matrix of the learning control signal for steps 0 through p-1. The difference operator δ_j operating on any quantity represents the value of that quantity at repetiton j minus the value at repetition *j-1*. Learning control law (2) reduces to learning control law (1) if L is chosen in the specialized form

$$L = diag[\phi\ \phi \cdots \phi]$$

Consider the modern control model

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k)$$
$$y(k) = C(k)x(k) \qquad (3)$$

where $W(k)$ is any disturbance or forcing function that repeats each repetition.

Also, in the learning control problem, it is assumed that the initial condition is t he same every repetition, and that it is on the desired trajectory. Then (3) impl ies that

$$\begin{aligned} \delta_j \underline{y} &= P\delta_j \underline{u} \\ \underline{y} &= [y^T(1) \quad y^T(2) \quad \cdots \quad y^T(P)]^T \\ \underline{u} &= [u^T(0) \quad u^T(1) \quad \cdots \quad u^T(P-1)]^T \end{aligned} \tag{4}$$

where

$$\begin{bmatrix} C(1)B(0) & 0 & \cdots & 0 \\ C(2)A(1)B(0) & C(2)B(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots \\ C(p)(\Pi_{k=1}^{p-1}A(k))B(0) & C(p)(\Pi_{k=2}^{p-1}A(k))B(1) & \cdots & C(p)B(p-1) \end{bmatrix} \tag{5}$$

The product notation represents a matrix product going from larger arguments on the left to smaller arguments on the right. Then recognizing that $\delta_j \underline{y} = -\delta_j \underline{e}$ and using the learning control law (2) gives the following rule for the error history as a function of the repetition number j:

$$\begin{aligned} \delta_j \underline{e} &= -PL\,\underline{e}^{j-1} \\ \underline{e}^j &= (I-PL)\underline{e}^{j-1} = E\underline{e}^{j-1} \end{aligned}$$

or

$$\underline{e} = (E)^j \underline{e}_0 \tag{6}$$

Here the j superscript on the parentheses represents the jth power. Thus the learning control law will converge to zero tracking error as the repetitions progress provided the lower block triangular matrix E has all eigenvalues less than one in magnitude, i.e. provided the eigenvalues λ_i of the diagonal blocks are all less than one in magnitude:

$$| \lambda_i(I - C(k+1)B(k)\phi_1(k+1) | < 1$$
$$k = 0,1,2,\cdots,p-1 \qquad (7)$$

III. Decentralized Learning Control in Discrete-Time Systems

We first consider a time-varying or time-invariant digital system of the following form

$$x_{o,i}(k+1) = A_{o,ii}(k)x_{o,i}(k) + \sum_{\substack{j=1 \\ j \neq i}}^{s} Ao, ij(k)x_{o,j} + B_{o,i}(k)v_i(k) + w_{o,i}(k)$$
$$y_i(k) = C_{o,i}(k)x_{o,i}(k) \qquad (8)$$

This represents s subsystems, with uncoupled input and output matrices. The dynamic interactions between the subsystems are represented by the coupling matrices $A_{o,ij}$. The control input to subsystem i is v_j , its state is $x_{o,i}$, and its measured output is y_i . The subscript o refers to the open loop system model. In a later section we consider differential equation models with the same structure.

Now consider that each subsystem has its own decentralized feedback controller with feedback of only that subsystem's measured output. The feedback controllers can be direct output feedback controllers or dynamic output feedback controllers, whose equations are

$$v_i(k) = v_{FB,i}(k) + u_i(k)$$
$$v_{FB.i}(k) = C_{FB.i}(k)x_{FB.I}(k) + K_i(k)[y_i(k) - y_i^*(k)]$$

$$x_{FB,j}(k+1) = A_{FB,ii}(k)\,x_{FB,i}(k) + B_{FB,i}(k)\,[y_i(k) - y_i^*(k)] \tag{9}$$

In the case of dynamic output feedback controllers, there is a state vector for the controller for each system i. The input v_i is divided into the feedback control signal $v_{FB,i}$ determined by the controller dynamic equations or output feedback equations in (9), and the learning control signal u_i. Equations (8) and (9) can be combined to form the closed loop system dynamic equations relating the learning control input to the system response

$$x_i(k+1) = A_{ii}(k)\,x_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^{s} A_{ij}(k)\,x_j(k) + B_i(k)\,u_i(k) + w_{i(k)} \tag{10}$$

w the state vector for system i is augmented as $x_i(k) = [x_{o,i}^T(k) \quad x_{FB,i}^T(k)]^T$ and the w_i s still an input that repeats every time the command is given to the system, but now it contains the repetitive command as well as the repetitive disturbance

$$w_i(k) = \begin{bmatrix} w_{o,i}(k) - B_{o,i}(k)\,K_i(k) y_i^*(k) \\ -B_{FB,i}(k)\,y_i^*(k) \end{bmatrix} \tag{11}$$

The closed loop system matrices are

$$A_{C,ii}(K) = \begin{bmatrix} A_{o,ii}(k) + B_{o,i}(k)\,K_i(k)\,C_i(k) & B_{o,i}(k)\,C_{FB,i}(k) \\ B_{FB,i}(k)\,C_i(k) & 0 \end{bmatrix}$$

$$A_{ij}(k) = \begin{bmatrix} A_{lo,ij}(k) & 0 \\ 0 & 0 \end{bmatrix} \tag{12}$$

$$B_i(k) = [\,B_{o,i}^T(k) \quad 0\,]^T$$

$$C_i(k) = [\,C_{o.i}(k) \quad 0\,]$$

1. The Decoupled Learning Control Systems

Now consider the design of the learning controller for subsystem i which is done assuming that this system is totally isolated without dynamic coupling to the other subsystems. The dynamic equations under consideration are then

$$x_i(k+1) = A_{ij}(k)X_i(k) + B_i(k)u_i(k) + w_i(k) \tag{13}$$

In specifying the desired response for the learning controller, the variables specified at each time step must be measured variables, and their number cannot excede the number of inputs (there is an alternative formulation, which we will not consider here, that specifies more variables than inputs but the desired trajectory is not specified every time step). This requirement means that the desired trajectory is specified as a desired history of some, but not necessarily all measured variables. The subscript R will be introduced to represent quantities with various rows deleted, keeping only those rows associated with variables specified in the desired trajectory for the learning controller. An example of this distinction which occurs in the robotics problem is as follows. Consider a single robot link of a robot arm having its own input torque and only this input torque appearing in its dynamic equation. Let the feedback controller for this link not only use position measurements but also velocity measurements in order to adjust the damping. Thus the measured information contains two variables, but the desired trajectory specified to the learning controller can contain only one variable because there is only one input. Normally this is specified as the desired velocity history. For simplicity of presentation, we will limit ourselves to the case where the number of measured variables specified in the desired trajectory at every time step is equal to the number of input variables. It is a simple matter to alter the development to handle the case were fewer variables are specified.

Using equation (2), the learning controller for the ith system becomes

$$\delta_j \underline{u}_i = L_i \underline{e}_{iR}^{j-1}$$

$$L_i = \begin{bmatrix} \phi_{i1}(1) & 0 & \cdots & 0 \\ \phi_{i2}(1) & \phi_{i1}(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots \\ \phi_{ip}(1) & \phi_{ip-1}(2) & \cdots & \phi_{i1(p)} \end{bmatrix} \tag{14}$$

$$\underline{e}_{iR}^{j} = \underline{y}_{iR}^{*} - \underline{y}_{iR}^{j}$$

Equation (4) for system i becomes $\delta_j \underline{y}_i = P_i \delta_j \underline{u}_i$ where the system matrices in P_i are from equation (13). When only the rows associated with variables specified to the learning controller are retained in $\underline{y}_i$ and in P_i one writes

$$\delta_j \underline{y}_{iR} = P_{iR} \delta_j \underline{u}_i \tag{15}$$

and then the error history for the learning controller operating on the uncoupled system (13) is given by

$$\underline{e}^{i} = (E_i)^j \underline{e}_0 \tag{16}$$

$$E_i = I - P_{iR} L_i$$

Hence, the stability condition for the learning controller for system i to give zero tracking error on the uncoupled dynamic equation (13) is that all of the eigenvalues λ_i of the indicated matrices satisfy

$$| \lambda_i (I - (C_i(k+1) B_i(k))_R \phi_{i1}(k+1)) | < 1 \tag{17}$$

$$k = 0, 1, 2, \cdots, p-1$$

We now suppose that the learning controllers for all subsystems satisfy this condition for convergence to zero tracking error that assumes there is no coupling between the subsystems, and we ask the question, what happens when these learning controllers are applied to each subsystem in the coupled dynamic equations (10)?

2. Stability of the Decentralized Learning Control

The coupled system (10) can be written in the form of equation (3), which determines the A,B, and C matrices to be considered in the learning control stability analysis. The stability of any learning control law of the form (2) applied to the coupled system (10) written in the form (3) is governed by the condition (7), after performing the appropriate deletion of rows in the product of C with B, retaining only those rows associated with outputs whose desired values are specified to the learning controller (i.e. introduce the subscript R as appropriate). However, the learning control has been designed in a decentralized manner, and is given for each subsystem by equation (14). We must combine these learning controllers to express the learning control law for the combined system (10) or (3).

The error history for the ith subsytem, $\underline{e}^j_{iR}$, is a column vector of the errors at repetition j for time steps 1 through p. The error history vector associated with the coupled system (10), $\underline{e}^j_R$, contains all elements from the error vectors for each of the subsystems, but regrouped. The column matrix starts with the errors for time step 1 for all subsystems going from 1 to s, and progresses to time step 2, etc. Hence, the block partitions $\phi_m(k)$ of the learning control gain matrix L for the coupled system (3)

$$\delta_j \underline{u} = L \underline{e}^{j-1}_R \tag{18}$$

can be written for all m in terms of the block partitions $\delta_{im}(k)$ of the learning control matrices in (14) for the s subsystems as follows

$$\phi_m(k) = diag(\phi_{1m}(k), \phi_{2m}(k), \cdots, \phi_{sm}(k)) \tag{19}$$

Equation (4) becomes $\delta_{ij}\underline{y}_R = P_R \delta_j \underline{u}$ where the diagonal partitions of P_R can be written as

$$(C(k)B(k-1))_R = diag((C_1(k)B_1(k-1)_R, (C_2(k)B_2(k-1)_R, \cdots, (C_s(k)B_s(k-1))_R) \tag{20}$$

by making use of the decoupled block diagonal nature of the input and output m atrices in system (10).

According to equation (7), zero tracking error for system (10) in the form (3) is achieved if the eigenvalues of the diagonal blocks of $I - P_R L$ are all less than unity in magnitude, i.e. if the eigenvalues of

$$I - (C(k)B(k-1))_R \phi_1(\varepsilon) \tag{21}$$

are all less than one in magnitude for all k. Using the diagonal decoupled struc ture of both matrices involved, according to equations (19) and (20), produces th e stability condition that all eigenvalues of the matrices

$$I - (C_i(k)B_i(k-1))_R \phi_{i,1}(k) \tag{22}$$

must be less than one in magnitude, for all systems and all time steps

$$i = 1, 2, \cdots, s$$
$$k = 1, 2, \cdots, p$$

However, the conditions (22) are simply the set of all stability conditions associated with the s decoupled learning control systems. We can sumerize in the following result.

<u>Result 1</u>: Suppose that the learning controllers (14) for each of the s subsystems satisfy the stability conditions (17) for stability of the learning process when there is no coupling between the subsystems, i.e. for stability of learning in system (13). Then, when these learning controllers are applied to the coupled system (8) with the decoupled feedback controllers operating on each subsystem, the resulting decentralized learning control system will converge to zero tracking error as the repetitions of the operation progress. This result holds independent of the magnitude and nature of the dynamic coupling terms A_{ij}.

IV. Decentralized Learning Control in Differential Equation Systems

The problem of interest for application to robot problems has the same form as equation (8) except that instead of being a difference equation, it is a differential equation

$$\dot{x}_{o,i}(k) = A_{co,ii}(t)\, x_{o,i}(t) + \sum_{\substack{j=1 \\ j \neq i}}^{s} A_{co,ij}(t)\, x_{o,i}(t) + B_{co,i}(t)\, v_i(t) + w_{co,i}(t) \tag{23}$$

$$y_i(t) = C_{o.i}(t)\, x_{o.i}(t)$$

Again we consider a set of decentralized output or dynamic feedback controllers, one for each of the s subsystems

$$
\begin{aligned}
v_i(t) &= v_{FB,i}(t) + u_i(t) \\
v_{FB,i}(t) &= C_{FB,i}(t)\, x_{FB,i}(t) + K_i(t)\,[\,y_i(t) - y_i^*(t)\,] \qquad (24) \\
x_{FB,i}(t) &= A_{FB,ii}(t)\, x_{FB,i}(t) + B_{FB,i}(t)\,[\,y_i(t) - y_i^*(t)\,]
\end{aligned}
$$

The equations are converted to closed loop form as before

$$
\dot{x}_i(t) = A_{C,ii}(t)\, x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{s} A_{C,ij}(t)\, x_j(t) + B_{c,i}(t)\, u_i(t) + w_{c,i}(t) \qquad (25)
$$

$$
y_i(t) = C_i(t)\, x_i(t)
$$

The new aspect of the problem is that this is a set of differential equations rather than a set of difference equations with the input and output matrices decoupled, but with the system dynamics coupled. The learning controller is a digital controller, which means that before it can be applied to the problem, we must discretize equation (25) using a zero order hold on the learning control input. This process causes coupling of the subsystems in the input influence matrix. However, we can still prove the following result.

Result 2: **Suppose that the learning controllers for each subsystem are asymptotically stable for all sufficiently small sample time T when applied to that subsystem without coupling to other subsystems. Thus they converging to zero tracking error at the sample times as the repetitions progress when applied to system (25) when all coupling terms in the system matrix are set to zero. Then, when these learning controllers are applied to system (25) with the coupling present, there exists a sample time T sufficiently small that the resulting decentralized learning**

controller will converge to zero tracking error at the sample times as the repetitions of the command tend to infinity. This result is independent of the size of the dynamic coupling terms between the subsystems.

Proof: Write system (25) in the combined form

$$\dot{x}(t) = A_C(t)x(t) + B_c(t)u(t) + w(t) \quad (26)$$
$$y(t) = C(t)x(t)$$

Consider time invariant case first. Then using the Taylor series definition of the exponential of a matrix

$$\Phi(T-\tau) = \exp(A_c(T-\tau))$$
$$= I - A_c(T-\tau) + (1/3!)A_c^2(T-\tau)^2 + \cdots$$

in the expression for the input matrix in the discrete time system equations (3) produces

$$B = \int_0^T \Phi(T-\tau)d\tau\, B_c$$
$$= B_C T + (1/2!)A_cB_cT^2 + (1/3!)A_c^2B_cT^3 + \cdots \quad (27)$$

Write the learning control gain matrix of equation (19) in terms of gains normalized by the sample time as $\phi_1 = \overline{\phi}_1/T$. Then the matrix whose eigenvalues determine stability in equation (21) becomes

$$I - (CB)_R\overline{\phi}_1/T = diag(I - (C_iB_{c,i})_R\overline{\phi}_{i,1}) \quad (28)$$
$$-((1/2!\ CA_cB_cT + (1/3!)CA_c^2B_cT^2 + \cdots)\overline{\phi}_1$$

As we make T sufficiently small, the eigenvalues can be made arbitrarily close to the eigenvalues of the decoupled learning controllers in the first term of this expression, i.e., they can be made arbitrarily close to those of equation (22).

The time varying case is more complicated. In the discrete time input matrix,

$$B(k) = \int_{kT}^{(k+1)T} \Phi((k+1)T, \tau) B_c(\tau) d\tau$$

make use of the expansions

$$B_c(\tau) = B_c(kT) + \frac{dB_c}{dt}\Big|_{t=kT}(\tau - kT) + \frac{1}{2!}\frac{d^2B_c}{dt^2}\Big|_{t=kT}(\tau - kT)^2 + \cdots$$

$$\Phi(t, t_o) = \Phi(t_0, t_0) + \frac{d\Phi}{dt}\Big|_{t=t_0}(t - t_0) + \frac{1}{2!}\frac{d^2\Phi}{dt^2}\Big|_{t=t_0}(t - t_0)^2 + \cdots$$

$$= I + A_c(t_c)(t - t_0) + \frac{1}{2!}[A_c(t_0) + A_c^2(t_0)](t - t_0)^2 + \cdots$$

and

$$\Phi((k+1)T, \tau) = [\Phi(\tau, (k+1)T)]^{-1}$$

$$= I + A_c((k+1)T)((k+1)T - \tau) + , \cdots$$

to obtain

$$B(kT) = B_c(kT)T + (1/2)[A_c((k+1)T)B_c(kT) + B_c(kT)T^2 + \cdots \quad (29)$$

As before, substitution of this expansion for the input matrix into equation (21) produces eigenvalues arbitrarily close to those of equation (22) for sufficiently small sampling times T. This completes the proof.

This shows that the decoupled nature of the continuous time input influence matrix can produce arbitrarily small coupling in the discrete time input influence matrix because for sufficiently small sample time the linear terms in the expansion can be made to dominate. Thus stability of the decoupled learning controllers for all sufficiently small sample times is sufficient to guarantee stability of the decentralized learning in the coupled differential equation at least for sufficiently small sample time. This result explains the success of decentralized learning control on the robot problem.

V. Numerical Examples

In this section, we present two examples. The first illustrates the effect of coupling between subsystems in the system dynamics, and the second studies the application of decentralized learning control to robot problems. The latter example illustrates the application of decentralized learning control to nonlinear systems, and also studies the effect of the coupling between subsystems introduced in the input matrix by the discretization of the system equations.

Example 1 : Consider the following discrete-time system, containing two subsystems that are coupled only in the matrix of the system dynamics by the coupling factor α:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.905 & \alpha \\ 0 & 0.923 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{30}$$

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.099 & 0 \\ 0 & 0.077 \end{bmatrix} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}$$

Note that when the coupling factor is set to zero, each subsystem produces zero steady state tracking error for a constant command. Figure 1 gives the desired trajectories for the two outputs y_1 and y_2, of subsystems 1 and 2.

For subsystem 2, the desired trajectory was generated by splitting the total time interval into three parts. The first one-third and the last one-third follow sixth order polynomials that satisfy the boundary conditions for its subinterval and supplies smoothness through the second derivative. These two subintervals are connected by a constant slope segment in the central one-third. The same method was used to obtain each half of the desired trajectory in subsystem 1. The same desired trajectories will be used in the next example, where the horizontal axis indicates time in seconds. For the purposes of this example, the conversion from continuous time to the discrete time of the present example is done using a sample time of 0.02 sec, associated with $k = 1, 2, \dots, 50$.

The learning gains for each subsystem were set to produce eigenvalues of the learning process of 0.66, following the examples in Ref. [Chang, 1991]. Also in accordance with the results of that reference, we alternate the sign of the learning gain each time step, in order to improve the transient behavior.

The theory developed in this chapter establishes that convergence of the learning process to zero tracking error in decentralized learning control is dependent only on the input and output influence matrices, and is independent of the coupling a between the subsystems in the system dynamics matrix. Here we study the behavior of the learning control process as a function of this coupling α. Figure 2 shows the error histories for various numbers of repetitions when the coupling is set to zero. The first repetition corresponds to the error produced by simply commanding the desired trajectory in the learning control input. The subsequent repetitions apply the linear learning control with alternating sign. The repetition 51 error histories correspond to the horizontal lines which appear to be zero error to within the resolution of the graphs presented. Figures 3 and 4 give the corresponding error histories for subsystem 1 when the coupling factor is set to α = 0.5 and 5.0, respectively. Note that the error histories for subsystem 2 remain as in Fig. 2 since this subsystem remains uncoupled.

Examination of these figures indicates that after 51 repetitions the error appears to be zero in each case, which is consistent with the theory. However, introduction of large coupling between the subsystems is seen to produce significant increases in the magnitudes of the errors during the transient part of the learning process. Thus, the coupling influences the history of convergence to zero tracking error as the repetitions progress, but cannot influence the ultimate convergence itself.

Example 2: The theory developed here applies to linear time-invariant systems, and also to linear time-varying systems. The original motivation for much of the literature in learning control is for application to robots which are nonlinear systems. The main objective of the chapter is to develop decentralized learning control for such applications, and the theory developed models the nonlinear robot equations as linearized in the neighborhood of the desired trajectory, which produces linear time-varying equations. This example illustrates this process by application of decentralized learning control to a polar coordinate robot moving in the horizontal plane. First, decentralized learning control is applied to the time-varying linearized equations model, and then application to the full nonlinear model is studied.

The nonlinear equations for motion of the polar coordinate robot in Fig. 5 are given as

$$(m_B + m_L)\ddot{r}(t) - [m_B r(t) + m_L(r(t) + l)]\,\dot{\theta}_1(t)^2 = F(t) \qquad (31)$$

$$[I_3 + m_B r(t)^2 + m_L(r(t) + l)^2]\,\ddot{\theta}_1(t) + 2[m_b r(t) + m_L(r(t) + l)]\,\dot{r}(t)\,\theta_l(t) = M_1(t)$$

where $r(t)$ is the radial extension of the prismatic joint measured from the center of the support point to the center of mass of the prismatic beam (without load), and $\theta_1(t)$ is the angle of rotation of the beam about the vertical axis.

The beam mass is $m_B = 39.28\,kg$, its half length is $l = 0.6$, and its moment of inertia about the vertical axis is $I_3 = 1.93\,kgm^2$. The mass of the point mass load located at the end of the beam is $m_L = 10\,kg$. The force and moments applied to each joint are supplied by proportional plus derivative feedback controllers given by

$$F(t) = K_1 [\, r(t) - \dot{r}(t) \,] + K_2 [\, \dot{r}(t) - r^*(t))] + u_1(t)$$

$$M_1(t) = K_3 [\, \theta_1(t) - \theta^*{}_1(t) + K_4 [\, \dot{\theta}_1(t) - \dot{\theta}_1^*(t))] + u_2(t) \qquad (32)$$

where, $K_1, K_{2.}, K_3, K_4$ are the feedback gains with values 98.6, 443.5, 450.9, 182.2 respectively, and $u_1(t)$ and $u_2(t)$ are the learning control signals.

The desired trajectory is again given by Fig. 1, where the subsystem 1 graph is $\dot{r}^*(t)$ in meters, and the subsystem 2 graph is $\theta_1^*(t)$ in radians. Decentralized learning control was applied to each axis, using a learning gain set to give eigenvalues of the learning process as 0.66 for the input influence matrix values at the initial time on the trajectory. Then this learning gain is given an alternating sign with each time step as in [Chang, 1990]. Ten time steps are used for the 1 second maneuver when the sample time is 0.1 sec, and 20 are used with the 0.05 sample time. Note that when a sample time of 0.1 sec is used, there are slightly less than two samples for the fastest "time constant" at the start of the maneuver.

Figure 10 gives the error histories for various repetition numbers when decentralized learning is applied to the linear time varying model of equation (33) with a learning sample time of 0.1 sec. Repetition 1 correspond to the first run with feedback only, and no learning control signal. Figure 11 gives the corresponding curves when the same control law is applied to the full nonlinear equations in (31) and (32). These figures are very similar in form, and it is interesting to note that the introduction of nonlinearities did not hinder the learning process. In fact learning progressed somewhat faster for the nonlinear system model. Figures 12 and 13 give corresponding curves for sample time 0.05 sec. This time the figures for the learning in the nonlinear and the linear system models are not as similar, and in some cases convergence is faster in the linearized system.

We can also examine these figures to see the effect of decreasing the sample time. Comparing the learning in the linearized models for the two sample times (Figs. 10 and 12) shows that convergence to zero tracking error is generally faster when the sample time is larger. With the smaller sample time, the tracking error in the early parts of the trajectory are better for early repetitions, but the error can grow to be significantly larger in the latter part of the maneuver. This type of error history motivated the learning in a wave approach presented in [Chang, 1990]. The larger sample time of course introduces significantly more coupling in the input matrix. This could adversely affect the learning, but this does not appear to be the case for the example at hand where smaller sampling time accentuates transients. Of course, the faster convergence is to zero error at the sample times, and with the larger sample time there are fewer points on the desired trajectory for which one obtains zero error. This is the price one pays for the faster convergence. When the figures for learning in nonlinear equations are compared for the two sample times, the same general conclusions apply, but with more severe transients toward the end in the nonlinear case, and with somewhat longer convergence times. Decreasing the sample time further, to 0.04 significantly accentuates the transients at the end of the 1 sec time maneuver. This suggest that the detrimental influence on the transients that occurs when the sample time is shortened, overshadows the beneficial effect of this process on the coupling in the input matrix.

VI. Concluding Remarks

In this paper, the most basic form of learning control, based on integral control concepts applied in the repetition domain, is studied in decentralized control applications such as the use of learning control on each axis of a robot. This type of decentralized learning control is illustrated in examples, including motion of a polar coordinate robot in the horizontal plane. Modeling a robot as linearized about the desired trajectory, we have shown in this paper that for sufficiently small sample time, the tracking error convergences to zero as the repetitions of the task progress, provided the learning controllers would converge if there were no coupling between the axes. When there is no such coupling, the robot equations become simple and linear, making the evaluation of convergence simple. The conclusion is that for sufficiently small learning gain, and sufficiently small sample time, the simple learning control law based on integral control applied to each robot axis will produce zero tracking error in spite of the dynamic coupling in the robot equations. Of course, the results of this chapter have much more general application than just to the robotics tracking problem. Convergence in decentralized systems is seen to depend only on the input and output matrices, provided the sample time is suffiently small.

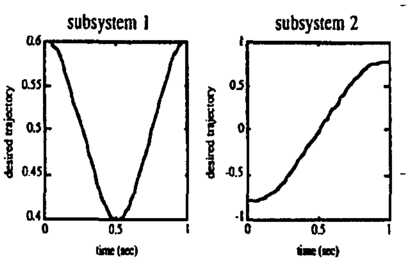

Fig. 1 Desired trajectories for subsystems 1 and 2

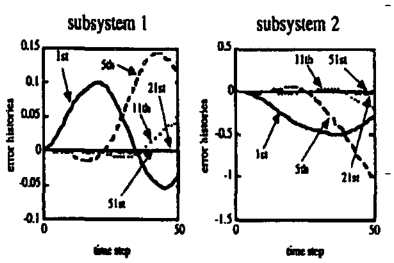

Fig. 2 Error histories for the two subsystems when the couping is set to zero

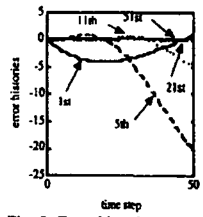

Fig. 3 Error histories for subsystem 1 when the *coupling factor is a =0.5.*

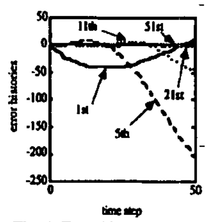

Fig. 4 Error histories for subsystem 1 when the *coupling factor is a =5.*

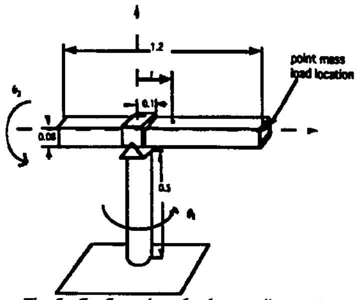

Fig. 5 Configuration of polar coordinate robot

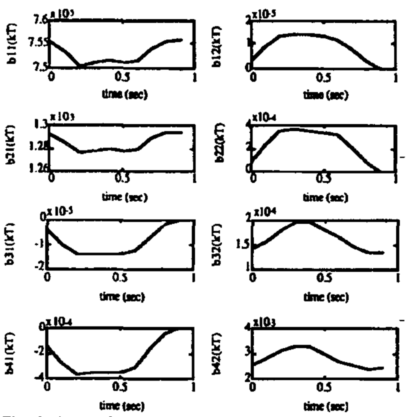

Fig. 6 Time variation and coupling between subsystems for the discrete time input influence matrix in example 2; sample time T=0.1.

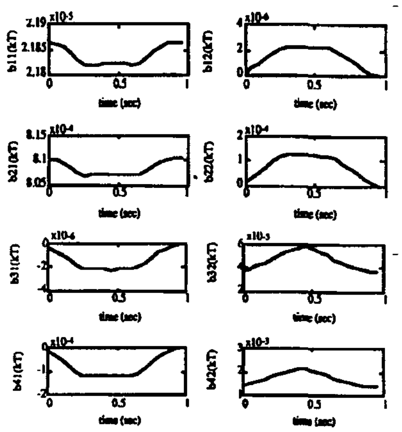

Fig. 7 Time variation and coupling between subsystems for the discrete time input influence matrix in example 2; sample time T=0.05.

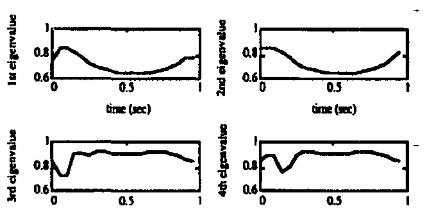

Fig. 8 Time variation and instantenous eigenvalues in examples 2; sample time T= 0.1

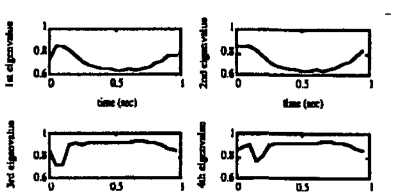

Fig. 9 Time Variation and instanteneous eigenvalues in example 2; sample time T = 0.05.

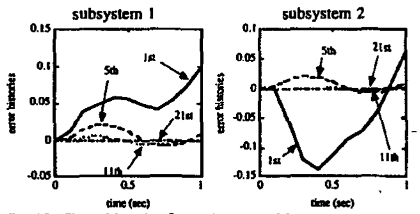

Fig. 10 Error histories for various repetitions for example 2, using a linearized time varying model with sample time T=0.1 sec.

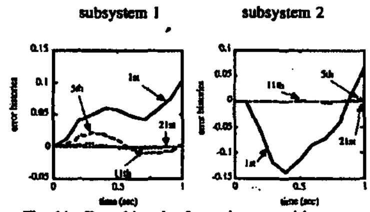

Fig. 11 Error histories for various repetitions for example 2, using the continuous nonlinear model with sample time T=0.1 sec.

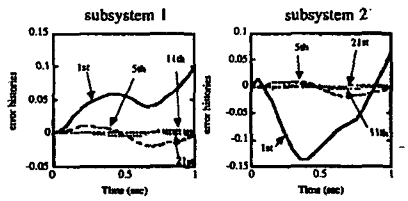

Fig. 12 Error histories for various repetitions for example 2, using a linerarized time varying model with sample time T = 0.05 sec.

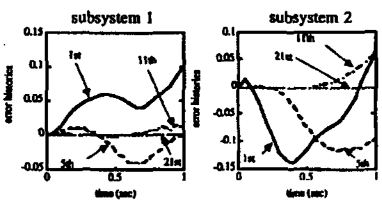

Fig. 13 Error histories for various repetitions for example 2, using the continuous nonlinear model with time T=0.05 sec.

REFERENCES

Arimoto, S., S. Kawamura, and F. Miyasaki, "Bettering Operation of Robots by Learning," Journal of Robotic Systems, Vol. 1, No. 2, 1984, pp. 123-140.

Chang, C.-K., R. W. Longman, and M. Phan, "Techniques for Improving Transients in Learning Control Systems," Proceedings of the 1991 AAS/AIAA Astrodynamics Conference, Durango, Colorado, August 1991.

Harokopos, E. G. and R. W. Longman, "A Learning Controller for Tracking Arbitrary Inputs in Robot Motions," Tenth IASTED International Symposium on Robotics and Automation, Lugano, Switzerland, June 29-July 2, 1987.

Middleton, R. H., G. C. Goodwin, and R. W. Longman, "A Method for Improving the Dynamic Accu racy of a Robot Performing a Repetitive Task," International Journal of Robotics Research, Vol. 8, No. 5, October 1989, pp.67-74.

Phan, M. and R. W. Longman, "A Mathematical Theory of Learning Control for Linear Discrete Mul tivariable Systems," Proceedings of the AIAA/AAS Astrodynamics Conference, Min neapolis, Minnesota, August 1988, pp. 740-746

Phan, M. and R. W. Longman, "Learning System Identification," Mod eling and Simulation, Instrument Society of America, Vol. 20, Part 5, 1989, pp. 1857-1864.

Phan, M., Y. Wei, L. G. Horta, J.-N. Juang, and R. W. Longman, "Learning Control for Slewing of a Flexible Panel," Dynamics and Control of Large Structures, Proceedings of the Seventh VPI&SU Symposium, L. Meirovitch, Editor, May 1989, pp. 39-52.

Phan, M., J.-N. Juang, and R. W. Longman, "Recent Developments in Learning Control and System Identification for Robots and Structures," Dynamics of Flexible Structures in Space, Proceedings of the First International Conference, Cranfield, England, May 1990