Journal of the Korean Institute
of Industrial Engineers
Vol.23, No.1, March, 1997

Indirect Decentralized Repetitive Control for the Multiple Dynamic Subsystems

Soo Cheol Lee*

Abstract

Learning control refers to controllers that learn to improve their performance at executing a given task, based on experience performing this specific task. In a previous work, the authors presented a theory of indirect decentralized learning control based on use of indirect adaptive control concepts employing simultaneous identification and control. This paper extends these results to apply to the indirect repetitive control problem in which a periodic (i.e., repetitive) command is given to a control system. Decentralized indirect repetitive control algorithms are presented that have guaranteed convergence to zero tracking error under very general conditions. The original motivation of the repetitive control and learning control fields was learning in robots doing repetitive tasks such as on an assembly line. This paper starts with decentralized discrete time systems, and progresses to the robot application, modeling the robot as a time varying linear system in the neighborhood of the desired trajectory. Decentralized repetitive control is natural for this application because the feedback control for link rotations is normally implemented in a decentralized manner, treating each link as if it is independent of the other links.

Keywords : decentralized control, repetitive control, robot

* 대구대학교 산업공학과

1. INTRODUCTION

A large number of control systems in use execute repetitive operations, e.g., controllers for robots in assembling and manufacturing systems. Recently, two bodies of work have appeared in the literature to develop controllers that learn to improve tracking performance based on previous experience at performing a specific command. These are known as repetitive control and learning control. Some examples of learning control are given in the references [1-19], and some recent results in repetitive control are listed in [20-29].

Repetitive control and learning control aim to make explicit use of knowledge that the command, and perhaps certain disturbances, are repetitive. Repetitive disturbance functions are common in systems executing the same command repeatedly. Learning control applies to situations in which the system returns to the same initial condition before the start of each repetition. By contrast, the repetitive control problem applies to system where the command is simply a periodic function of time, and hence the system may not return to the same initial condition at the start of the next repetition, and in fact choices of the feedback control and the learning control during one period of the periodic task will affect the initial conditions seen for the next period of the task. In addition, transients can propagate from one repetition to the next. These properties make the issue of stability of the learning process very different in the repetitive control problem than in the learning control problem.

Our aim in this work is to develop decentralized repetitive control laws that have guaranteed convergence to zero tracking error. Since the initial conditions at the start of each repetition are related to the coupling of the dynamics between the decentralized subsystems, they introduce a new difficulty in the decentralized repetitive control problem that was not present in the decentralized learning control problem treated in [15]. Here we extend the adaptive control based repetitive control methods of reference [20], making use of the results of [15] in order to produce decentralized repetitive control algorithms. These have guaranteed convergence to zero tracking error for each of the subsystems in spite of the dynamic interactions between these subsystems.

Repetitive control and learning control have application to all tracking problems in which the command is given repeatedly, but the application that motivated the development of the field starting in 1984 is robots 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 and 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 address.

The question arises, what happens if a separate repetitive controller is used with each of the separate feedback controllers of the robot arm. Such an application represents use of a decentralized repetitive control. A serious issue is whether the interactions in the dynamics of the systems governed by the separate learning controllers could cause the learning processes to fail to converge. In a previous paper [13], this question was addressed for the most basic form of learning control that is based on use of integral control like concepts applied in the repetition domain. In [15], there are various more sophisticated learning control approaches including one that makes use of indirect adaptive control ideas applied to learning in the repetition domain [6]. This approach has an important advantage over the simpler learning control law related to integral control concepts, because it can guarantee convergence of the learning process to produce zero tracking error.

The system equations are nonlinear in the robotics problem. Here we will consider that the motion takes place in the neighborhood of the desired trajectory, and hence it is possible to represent the system as a linear one, linearized about the desired trajectory. Then the system equations are linear, but with time varying coefficients. By 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 in the system matrix. There is no interaction between the axes in the input and output coefficient matrices. We produce decentralized repetitive control laws for this type of decentralized system model.

One of the interesting properties of the learning control problem is that in most methods, one can handle time varying systems with essentially the same mathematical ease as time invariant systems. Note that in a robot there are various interaction forces, such as centrifugal force on one link produced by rotation of a link closer to the base of the robot arm. When linearized about the desired motion they appear as the sum of a repetitive disturbance term to the first link's controller associated with the other link's motion as if it were along the desired trajectory for that link, and another term which is a linear coupling in the system matrix representing the linear corrections for the disturbance when the disturbing link deviated from its desired trajectory. Also, external disturbances such as the torque produced by gravity on a link, when linearized about the nominal trajectory, produce a disturbance forcing term that repeats with each repetition of the task together with a linear term with time varying coefficient that corrects for deviations in the gravity effect when the link deviates from the nominal trajectory. Hence, the aim of the present investigation is to obtain decentralized repetitive control laws with guaranteed convergence to zero tracking error, when:

1) Each decentralized repetitive controller knows only measurements from its subsystem.
2) Each subsystem has its own decentralized feedback controller operating (if desired), which makes use of measurements form it's subsystem only.
3) The complete system being controlled has coupling between subsystems in the system matrix, but not in the input and output matrices.
4) The complete system is governed by a set of linear differential equations with possibly time varying coefficients, and with unknown repetitive forcing functions (which represent the disturbances from other subsystems along the nominal desired trajectory, or disturbances such as gravity torque along the desired trajectory).

This paper presents for the repetitive control problem the same kind of treatment presented in the indirect decentralized learning control problem in [15] based on indirect adaptive control ideas. A fundamental aspect of the learning control problem is that it starts each repetition of the command from the same initial condition. In the repetitive control problem where control actions and transients during one repetition influence the state at the start of the next repetition, we make the assumption that we have an upper bound on the number of time steps needed for transients to decay. This assumption allows us to reduce the decentralized repetitive control problem to one that is similar to the decentralized learning control problem and then apply the methods of [15]. However, we first consider repetitive control in centralized systems, and then progress to the question of how these control laws can be made to apply in the decentralized implementation.

2. ADAPTIVE CONTROL BASED REPETITIVE CONTROL IN CENTRALIZED SYSTEMS

Consider first, a system of the form

$$\dot{x}(t)=A_c(t)x(t)+B_c(t)u(t)+w_c(t) \qquad (1)$$
$$y(t)=C(t)x(t)$$

where the $A_c(t)$, $B_c(t)$, and $w_c(t)$ are periodic with period pT. This system can be a model that represents motion of a nonlinear robot system linearized about the desired trajectory, producing a linear system with time varying coefficients. The periodic driving term W_c can contain periodic disturbances as seen along the desired trajectory, such as from gravity torques. Equation (1) may represent a system with feedback control operating to execute a periodic command, and then the w term also contains the periodic command (see the next section). Assuming that a zero order hold (with sample interval T) is used on the input to this system, it can be converted to a difference equation without approximation

$$x(k+1)=A(k)x(k)+B(k)u(k)+w(k) \quad (2)$$
$$y(k)=C(k)x(k)$$

where again the coefficient matrices and the exogenous driving term are periodic, with period p. In terms of special cases of these equations, we can define a hierarchy of models:

1. Time-invariant linear system with $w=0$.
2. Time-invariant linear system with $w \neq 0$.
3. Time-varying linear system with $w \neq 0$ or $w=0$.

2.1 Repetitive Control for Time-Invariant Systems

Reference [18] discusses such a hierarchy for *learning* control systems using adaptive control methods. Similar statements apply to the *repetitive* control problem. In the case of a model of the Type 1 above, one can simply apply any adaptive control algorithm that produces zero tracking error in the limit as time goes to infinity in time invariant systems of Type 1. The majority of adaptive control laws are designed for this case (although their main application is in time varying or nonlinear systems). When a repetitive command is given, this controller becomes a repetitive controller. There is no need for a distinction between adaptive control and a repetitive controller using the same adaptive ideas in this case, for centralized control problems. If the input commands to an adaptive controller are sufficiently rich then it will eventually produce zero tracking error for all feasible commands. If the input commands are limited to a single periodic input, an adaptive controller will learn enough to produce zero error for this command. Knowledge that this is the only command of interest, is not extra information that can be used to simplify the problem, or speed up the convergence, in this case.

The situation is somewhat more interesting in the case of a system of Type 2. Adaptive control algorithms are not designed to handle an unknown external input. Sometimes people try to model the external input often with some ad hoc model type, using some criterion such as finding the minimum disturbance of the chosen form that can model the data. Without such an added criterion, there is not sufficient information to distinguish influence in the data produced by the system from influence in the data produced by the unknown disturbance. In the repetitive control problem we are in a much better situation. The exogenous input w is known to repeat with the same period as the command. If one differences the input and output data between the current repetition and the corresponding data from the previous repetition, then the periodic exogenous term drops from the difference equation that models the input-output relationship for this differenced data. Then using this differenced data in an adaptive control law will converge to the change in the control needed to produce zero tracking error. Hence, one can easily eliminate the unknown periodic driving term from the equations, and there is no need to try to identify it. Again, there is no major difference between adaptive control and repetitive control based on the same adaptive concept. The differences are that one must delay application of the adaptive law until after the first repetition is completed, and then use the adaptive control signal as a change in the control, rather than as the control itself.

2.2 Repetitive Control in Time-Varying Systems—the Full Dimensional Input and Output Case

Reference [20] presents a repetitive control algorithm for the case when the full state is measured. We review this algorithm here, as our starting point for developing decentralized algorithms. The solution to (2) can be written as

$$y(i)=C(i)\left[\Pi_{k=i_0}^{i-1}A(k)\right]x(i_0) + \sum_{\tau=i_0}^{i-1} C(i)\left[\Pi_{k=\tau+1}^{i-1}A(k)\right]B(\tau)u(\tau) + \sum_{\tau=i_0}^{i-1} C(i)\left[\Pi_{k=\tau+1}^{i-1}A(k)\right]w(\tau) \quad (3)$$

Here the product notation is understood to mean that the product of matrices is performed in the order of decreasing time step arguments from left to right, and a product whose upper limit is smaller than the lower limit is taken to be the identity matrix.

In the repetitive control problem the time varying coefficients repeat every repetition, and we must use this information. It is convenient to package the complete history of a repetition in matrix form. For the rth repetition, the initial condition is at time step rp, the computed values of x and y are at time steps $rp+1$ through $(r+1)p$, while the choices of control made are for time steps rp through $(r+1)p-1$. Define

$$x(rp+k)=x_r(k);k=0,1,2,\cdots,p$$

where the subscript indicates the repetition number starting from 0, and the argument k indicates the time step within the repetition. Note that the periodicity of the coefficient matrices make $A(jp+k)=A(k)$ and similarly for $B(k)$, $C(k)$. Form the matrix of the history of a variable for one repetition as

$$\underline{y}_r=[y_r^T(1)\ y_r^T(2)\ \cdots\ y_r^T(p)]^T$$
$$\underline{u}_r=[u_r^T(0)\ u_r^T(1)\ \cdots\ u_r^T(p-1)]^T$$

Note the shift between the time arguments used in the control history versus the output history (when used, the state history follows the same rule as the output history).

In order to eliminate the repetitive disturbance, we take the difference between the histories of two successive repetitions defining

$$\delta_r\underline{y}=\underline{y}_r-\underline{y}_{r-1}$$

and similarly for any other history matrix or any component. Then writing equation (3) for one repetition starting at time $(r-1)p$, and doing the same starting at rp, and then differencing the results, produces the equation

$$\delta_r \underline{y} = \underline{A}\, \delta_r x(0) + P\, \delta_r \underline{u} \tag{4}$$

where the notation $\delta_r x(0)$ means the initial value of the state at the start of repetition r (at time step rp) minus the state at the start of repetition $r-1$(at time step $(r-1)$p). The coefficient matrices are given by

$$\underline{A} = [[C(1)A(0)]^T \quad [C(2)A(1)A(0)]^T \quad \dots \quad [C(p)\Pi_{k=1,p-1}A(k)]^T]^T$$

$$P = \begin{bmatrix} C(1)B(0) & 0 & \cdots & 0 \\ C(2)A(1)B(0) & C(2)B(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C(p)\left[\Pi_{k=1,p-1}A(k)\right]B(0) & C(p)\left[\Pi_{k=2,p-1}A(k)\right]B(1) & \dots & C(p)B(p-1) \end{bmatrix} \tag{5}$$

The repetitive controller given in [20] assumes that there are n independent measurements and n independent inputs, where n is the order of the difference equation (3). This controller is easily obtained from the above formulation. Since we are measuring n independent variables, we can without loss of generality, take the measured variables as the state variables, and make no distinction between output and state vectors. Equation (4) then represents a set of ARMA models written in the repetition domain for different time steps in this time varying system. Actually, the equation is a special case in which the moving average part of the equation is missing, leaving us with an MA or Moving Average model.

An indirect repetitive controller is then easily generated using any recursive identification algorithm such as recursive least squares which aims to identify the coefficient matrices in equation (4), and based on these one applies a one repetition ahead controller

$$\begin{aligned} \underline{u}_r &= \underline{u}_{r-1} + \delta_r \underline{u} \\ \delta_r \underline{u} &= \hat{P}^{-1}[(\underline{y}^* - \underline{y}_{r-1}) - \hat{\underline{A}}\, \delta_r x(0)] \end{aligned} \tag{6}$$

Here the hat over the matrices indicates their estimated values, and the desired changes to be produced in the output histories has been substituted for $\delta_r \underline{y}$ when picking the change to make in the control history for the upcoming repetition, $\delta_r \underline{u}$. This control law is guaranteed to produce convergence to zero tracking error as the repetitions progress (in the absence of non repetitive disturbances or noise), although it need not produce convergence to the actual values of these matrices.

This control law can be applied starting with the second period of the periodic command. During the first period, one can use a feedback controller to execute the command, and then it is the job of the repetitive controller to eliminate the error that remains. If there is no repetitive driving term w, one may wish to use an adaptive control law on the during the first period. Otherwise, one could use an adaptive control law on the second period (differencing the data with the first repetition so that w is eliminated from the equations), in addition to the feedback control law. The adaptive control law is not capable of producing zero tracking error in this time varying system, but if the coefficients are slowly time varying, or more precisely, if the time needed for adaptation to changed coefficients is relatively quick compared to the speed of variation of the time varying coefficients, then the adaptive control law will improve the tracking performance during this second period. Then starting with the third period, it is the job of the repetitive controller (6) to eliminate the errors that the adaptive controller cannot eliminate due to its time lag in tracking the system parameters.

All of the centralized repetitive controllers developed above, have initial condition terms that involve the system A matrix. Here we wish to develop decentralized repetitive control laws that are appropriate for time-varying and also time-invariant systems executing repetitive commands. This initial condition term is troublesome because it will usually make it necessary for each subsystem to know other subsystem initial conditions, and thus precludes decentralized implementation. In a later section we will develop still another centralized repetitive control law, based on the assumption that we have an upper bound on how long it takes the transients of the system to decay to a negligible level. This assumption allows us to eliminate the initial condition term appearing in the above repetitive control laws, and will make it possible to generate a decentralized repetitive control law with guaranteed convergence to zero tracking error. before we do this, we first set up the system equations for the decentralized problem.

3. MATHEMATICAL FORMULATION OF THE DECENTRALIZED SYSTEM

Consider a time-varying or time-invariant discrete time system of the following form

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

$$y_i(k) = C_{o,i}(k)x_{o,i}(k)$$

$$i=1,\ 2,\ 3,\ \ldots,\ s$$

We will later consider differential equation system with a similar structure, and it is such a model that applies to robots when linearized about the desired trajectory. System (7) contains s subsystems. The input and output matrices for the different subsystems are uncoupled, but there is dynamic coupling between the subsystems represented by the coupling matrices $A_{o,ij}$. The control input to subsystem i is v_i, its state is $x_{o,i}$, and its measured output is y_i. The $w_{o,i}$ represent disturbances that repeat with each repetition of the task. The subscript o refers to the open loop system model.

Now consider that each subsystem has its own decentralized feedback controller with feedback of only that subsystem's measured output. These controllers could be simple proportional controllers with rate feedback which is a common approach in robotics, or they can be more complex controllers including controller dynamics and a controller state variable. We include all such possibilities in the following formulation:

$$\begin{aligned} v_i(k) &= v_{FB,i}(k) + u_i(k) \\ v_{FB,i}(k) &= C_{FB,i}(k)\, x_{FB,i}(k) \\ &\quad + K_i(k)\,[y_i(k) - y_i^*(k)] \\ x_{FB,i}(k+1) &= A_{FB,ii}(k)\, x_{FB,i}(k) \\ &\quad + B_{FB,i}(k)[y_i(k) - y_i^*(k)] \end{aligned} \tag{8}$$

Here, the input v_i is the sum of the feedback control signal $v_{FB,}\, i$ and the repetitive control signal u_i. The desired output of the system is

$$y_i^*(k)\ ;\ k = 1, 2, 3, \dots p \tag{9}$$

and it is the task of the repetitive control signal

$$u_i(k)\ ;\ k=0, 1, 2, \dots, p-1 \tag{10}$$

to converge on an altered input command to the feedback system that causes the actual measurements $y_i(k)$ to correspond with these desired outputs. When dynamic controllers are used, the feedback control signal for each system i is determined as the output of this controller's dynamic state variable equation in equation (8). When output feedback is employed, then the dimension of the controller state reduces to zero in equation (8), leaving only the second term on the right in the middle equation.

The system of importance to the repetitive controller (or controllers, in the centralized case) is that relating the repetitive control signal u_i to the measured system response. This is accomplished by combining equations (7) and (8) to form the closed loop system dynamic equations

$$\begin{aligned} x_i(k+1) &= A_{ii}(k)x_i(k) \\ &\quad + \sum_{\substack{j=1 \\ j \neq i}}^{s} A_{ij}(k)x_j(k) + B_i(k)\,u_i(k) + w_i(k) \\ y_i(k) &= C_i(k)x_i(k) \\ i &= 1, 2, 3, \dots, s \end{aligned} \tag{11}$$

The closed loop system matrices in this equation are

$$A_{ii}(k) = \begin{bmatrix} A_{o,ii}(k) + B_{o,i}(k)K_i(k)C_{o,i}(k) & B_{o,i}(k)C_{FB,i}(k) \\ B_{FB,i}(k)C_{o,i}(k) & A_{FB,ii}(k) \end{bmatrix} \tag{12}$$

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

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

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

and the state vector for system i has been augmented to include the controller state as

$$x_i(k) = [x_{o,i}^T(k) \qquad x_{FB,i}^T(k)]^T \tag{13}$$

The exogenous term w_i is 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

$$w_i(k) = \begin{bmatrix} w_{o,i}(k) - B_{o,i}(k)\, K_i(k) y_i^*(k) \\ -B_{FB,i}(k)\, y_i^*(k) \end{bmatrix} \quad (14)$$

The set of equation in (11) can be combined to form one overall system equation, in the form given in equation (2). Hence, we have centralized repetitive control laws that can be applied to this problem, as developed above. However, since the feedback controllers for each subsystem are decentralized, and only use feedback from the state variables of that subsystem, it would be desirable to develop separate repetitive controllers for each subsystem as well. Each repetitive controller would learn a new command to give to the feedback controller for that subsystem, so that it produces the desired trajectory in the limit as the number of repetitions tends to infinity. And this must be accomplished using only data available to the subsystem feedback controller, and the result must apply independent of the dynamic coupling that may exist between the subsystems. With this aim in mind, we develop the following repetitive controller, and then show that in a decentralized system of the form of equation (11) it automatically produces a decentralized repetitive controller.

4. A DECENTRALIZED REPETITIVE CONTROL LAW WITH GUARANTEED CONVERGENCE TO ZERO TRACKING ERROR

The basic assumption needed, is that we know a bound on the number of time steps that are needed for transients to decay to a negligible value. Of course, if necessary, one can perform tests to observe the time of decay. This assumes that the system under consideration is asymptotically stable, which is insured by the decentralized feedback controllers in the system model described in the previous section.

The bound on the number of time steps can be any number. For simplicity in the mathematical development we will assume that the number of steps is p−1 or less − the precise assumptions are that

$$C(p+i_o)\left[\Pi_{k=i_o+1}^{p+i_o-1} A(k)\right] B(i_o) \quad (15)$$

is negligible when multiplied by any change in control that we may choose to make, for all i_o. Similarly, we will assume that

$$C(p+i_o)\left[\Pi_{k=i_o}^{p+i_o-1} A(k)\right] \quad (16)$$

is negligible when multiplied by any change in initial condition that we may generate. There will be a third similar assumption detailed later. In simplified terms we are saying that the transients are dead within one repetition of the command period, and that if the commands are not changed over the same time period, then the system state reached is independent of the initial conditions and of the control history prior to this period. An extra n steps are allowed as initial conditions for an nth order difference equation if needed.

if the feedback controller can be expected to do a reasonable job of tracking the desired command history and the repetitive control is present to eliminate remaining errors, then it is necessary that the time for decay of transients be shorter than the time associated with substantial change of the input, i.e, it should be less than -1. Thus the choice made here, is natural in many applications. In the event that one wishes to pick the number of time steps needed for decay of transients to a negligible value to be a larger number, then by setting it equal to a multiple of p, minus n , then the changes needed in the repetitive control law developed below are obvious.

The assumption that transients are gone in $p--n$ steps causes us to perform the learning process during alternate repetitions only. If we were to assume a longer time for the transients to become negligible, we would obtain the same type of repetitive control law, but we would be required to skip more repetitions between the repetitions during which we learn.

With these assumptions, we can now rewrite equation (3) in the following form, after dropping those terms which we consider to be negligible

$$\overline{\delta}_r y(i) = \sum_{\tau=i-p}^{i-1} C(i) \left[\Pi_{k=\tau+1}^{i-1} A(k) \right] B(\tau) \overline{\delta}_r u(\tau) \quad (17)$$

where the new difference operator takes the value of the variable during the rth repetition minus the value *two* repetitions earlier, for example

$$\overline{\delta}_r \underline{y} = \underline{y}_r - \underline{y}_{r-2} \quad (18)$$

Equation (17) can be rewritten in terms of the histories of the inputs and outputs during the repetitions as

$$\overline{\delta}_r \underline{y} = P' \overline{\delta}_{r-1} \underline{u} + P \overline{\delta}_r \underline{u} \quad (19)$$

which is to be compared to the previous version of this equation that appeared in equation (4). The matrix P is the same as in equation (4), while the matrix P' is a new matrix containing more impulse responses terms coming associated with changing the learning control inputs in the previous repetition.

As stated previously, we agree not to make any changes in the input $\underline{u}$ every other repetition, in which case equation (19) reduces to

$$\overline{\delta}_r \underline{y} = P \overline{\delta}_r \underline{u} \quad (20)$$

This equation has the same form as the that used in the decentralized learning control problem in reference [15], the only difference being the substitution of $\overline{\delta}_r$ for δ_r . Hence, the learning control laws developed there can be applied here as well. Two different forms were developed in [15]. Each relies on the fact that when equation (20) is written in terms of the partitions for the ith and jth subsystems,

$$\overline{\delta}_r y_i = P_{ii} \overline{\delta}_r \underline{u}_i + \sum_{\substack{j=1 \\ j \neq i}}^{s} P_{ij} \overline{\delta}_r \underline{u}_j \tag{21}$$

the diagonal partitions of the coupling matrix P_{ij} are zero. This matrix represents the pulse response of subsystem i to unit pulse inputs in the control inputs for subsystem j, while P_{ii} represents the pulse response of subsystem i to unit pulse inputs in control channels for subsystem i. One of the decentralized repetitive control algorithms follows an agreement between the subsystems that only one subsystem adjusts the repetitive control signal in any repetition, and which subsystem is doing the adjusting is rotated in a preassigned manner. This approach works here as well, but it is slower in the repetitive control case, because no adjustment of the repetitive control signal is allowed every other repetition. The approach can work well if one has a system model which is relatively close to the true system. The second method presented in [15] is the preferred method when there is poor a priori knowledge of the system. It takes advantage of the fact that the block diagonal elements of P represent the products $C(k+1)B(k)$. For the decentralized structure of the difference equations given in equations (7-14), this product decouples according to subsystems. Hence, each subsystem can learn simultaneously, one time step per repetition. First, each subsystem i uses its own input-output data to identify its $C_i(k+1)B_i(k)$. Knowing this matrix allows that subsystem to solve

$$\overline{\delta}_r y_i(k+1) = C_i(k+1)B_i(k)\overline{\delta}_r u_i(k) \tag{22}$$

for the needed change in its control to produce zero tracking error at the current step in the next repetition in which a repetitive control signal is applied. As in [15], when the governing difference equation model comes from a linear differential equation with the subsytems decoupled in the input and output matrices, or when the actual system started as a nonlinear system, this process of identifying and choosing the learning control at a particular time step may be repeated several times in order to eliminate the effects of the small coupling between subsystems in the difference equation input matrices for these cases.

To start the learning process, using (20-22) with its new difference operator, we make two repetitions with the feedback control signal only. Then in the next pair of repetitions, we use feedback only for the first, and start the learning of the first time step in the second repetition. In the third pair of repetitions, the first is again feedback only, and during the second repetition of this pair, we can complete the learning for this time step if the product of input and output matrices to be identified in (22) is a scalar, and we can progress to learning the next time step in the next pair of repetitions. If it is a multiple-input, multiple-output system, more repetitions must be allocated for this time step before progressing. Also, if there is small coupling between the subsystems in the input matrix introduced by discretization of a decentralized differential equation, or if there are nonlinear effects present, it may be prudent to allocate some extra repetitions to learn each time step.

After the wave of learning has progressed through one complete repetition, then the repetitive control signal obtained will produce zero tracking error (in the absence of random noise) for every other repetition, and for the repetitions between these, the feedback control alone is applied. This repetitive control signal produces zero tracking error during a repetition, but it is specialized in the sense that it only produces zero tracking error if the initial conditions for the repetition are those set up by a repetition in which feedback control is used. Hence, we cannot simply apply this repetitive control signal for all repetitions and expect to get zero tracking error for all repetitions. Instead, we must continue the wave of learning through p more steps, i.e., we let the wave of learning progress through the repetitions which had previously used feedback control only. The repetitive control signal obtained for such a repetition has as its initial conditions, those that are obtained by the repetitive control that produced zero tracking error for a repetition following a feedback only repetition. Because of assumptions (15) and (16), these initial conditions are along the desired trajectory. Hence, we can now apply this second learning control signal to all future repetitions in order to obtain zero tracking error at all times. Note that in analyzing the number of steps needed to eliminate the effect of different initial conditions, one must remember that not only the transients from the initial condition must be gone, but the transient effects produced by any control actions that were influenced by the initial condition, must also be gone.

Hence, the repetitive control algorithm for learning in a wave with assumptions (17) and (18) and a similar assumption discussed below, treats the repetitive signal as a signal of period *2p*, letting the wave of learning progress *2p* time steps. And then the repetitive signal obtained for the *second* set of p steps is used throughout all future repetitions, and the repetitive control signal for the first p steps can be forgotten. While learning the first p time steps, the first term in (19) is zero during the repetitions that learn, because no repetitive control signal is applied every other repetition making $\overline{\delta}_{r-1}\underline{u}$ equal to zero. During the second p time steps of learning, this term is zero because the same repetitive control signal is applied for both repetitions in this difference.

What is required for the repetitive signal obtained for the second set of p steps to be the desired signal, is that the initial conditions be those needed for steady state repetitive operation with zero tracking error. In order to see what this implies, consider equation (4) with the differencing removed and with the disturbance term included

$$\underline{y} = \underline{A}x_{FB}(0) + P\underline{u} + P_W\underline{w}$$

Here P_W is the same as P but with the input influence matrix factors removed, and the initial condition $x_{FB}(0)$ is set to that produced at the end of a feedback control only repetition. The repetitive signal learned for the first p steps, which we denote by $\underline{u}_{FB}$, produces the desired trajectory , and is given by

$$u_{FB} = P^{-1}[\underline{y}^* - \underline{A}x_{FB}(0) - P_W\underline{w}]$$

The state reached at the end of a repetition using this repetitive control signal is given by

$$x(p) = [(\underline{A})_p\, x_{FB}(0) + (P_s)_p\underline{u}_{FB} + (P_W)_P\underline{w}$$

where $(\underline{A})_p$ is the pth row partition of $\underline{A}$, $(P_s)_p$ is the pth row partition of matrix P with the output matrices deleted, and similarly for $(P_w)_p$. This state is the initial condition for the second set of p steps, and it can be written as

$$x(p) = [(\underline{A})_P - (P_S)_P P^{-1}\underline{A}]x_{FB}(O) + (P_S)_P P^{-1}[\underline{y}^* - P_W\underline{w}] + (P_W)_p\underline{w} \quad (23)$$

Examination of $[(\underline{A})_P - (P_S)_P P^{-1}\underline{A}\,]$shows that every term contains the system matrices A or $A(k)$ to the pth power. We assume that this factor is sufficiently small that the first term in equation (33) is negligible, and hence the initial condition for the start of the second p steps, after the system has followed the desired trajectory for the previous p steps, is independent of the initial conditions $x_{FB}(0)$.

This repetitive control algorithm is essentially the same for time-varying systems and for time-invariant systems. The distinction is that, in a time-varying system, repetitions for learning the product $C_i(k+1)B_i(k)$ must be allocated for every time step of the first p time steps of learning, but not for the second p steps of the wave of learning. Once the value of the product is known for any specific time step k, then in the next repetition in which repetitive control is applied, the error at step $k+1$ can be made zero. In the time-invariant case, once the product C_iB_i is learned in the first step of the repetition, the same value can be used for all future steps of the repetition without allocating repetitions for identification purposes, making the wave of zero error progress more quickly.

By comparison to learning control, the repetitive control problem has paid a price in rate of convergence to zero tracking error, because of the necessity to skip every other repetition or more, depending on the time needed for transients to decay. It was noted in [15] that the decentralized learning control would produce zero tracking error in exactly the same number of time steps as the centralized learning controller for systems of the form of equation (11) – i.e. no penalty was paid for decentralized implementation in difference equation systems (in differential equation systems of the same form, there would be some penalty). However, there is a penalty paid in the repetitive control case when using this repetitive control algorithm.

The repetitive control law is generated for a deterministic model, and is guaranteed to converge to zero tracking error in a finite number of repetitions for the time varying difference equation model with the subsystems decoupled in the input matrix.. If there is random noise in the system, then one will not have zero error at the end of this number of repetitions. One can allocate more repetitions for each time step of the wave, in order to allow averaging of the noise effects in the identification for that step. Since the learning process proceeds in a wave progressing from the beginning of a repetition to the end of the repetition, it is ambiguous what action one should take once the wave of the learning process is completed, if the noise effects have prevented one from obtaining the desired tacking accuracy. One can start a new wave of the learning process, combining the data available for identification from all past repetitions. Alternatively, one can switch to the integral control based learning from [13] after the completion of the wave, as was done in [15] .

The mathematical model of interest for application to robot problems has the same form as we have been treating, i.e. equation (11), except that instead of being a difference equation with the given structure, it is a differential equation with this structure. When such a differential equation is discretized, what was zero coupling between subsystems in the input influence matrix, becomes small coupling. It was shown in [13, 15] that if one samples sufficiently fast, then the decentralized control algorithms will still work. This statement applies here as well. If one samples fast enough, then the error at the end of the learning wave can be made arbitrarily small. Then by switching to the integral control based method, one obtains convergence to zero tracking error asymptotically as the number of repetitions tends to infinity. The price that is paid is convergence asymptotically instead of in a finite number of steps.

The development of the above algorithm for repetitive control using a wave of learning, was developed without any reference to whether the learning is centralized or decentralized. Hence, it represents a repetitive control algorithm for centralized learning as well. In centralized applications it must compete with the algorithms obtained previously in this paper, and one expects that one could obtain faster convergence with one of the previous algorithms. When applied to systems with the given decentralized structure, the learning in a wave allows each subsystem to learn in a decentralized manner. The advantage of the new algorithm is then its ability to be applied in a decentralized manner.

5. NUMERICAL EXAMPLES

In this section we apply the indirect decentralized repetitive control law of equation (22) to the linearized model of the polar coordinate robot, with PD controllers for each joint, for motion in the horizontal plane. In the repetitive control problem we need a desired trajectory that is a periodic function of time, and we pick the functions shown in Figure. 1 obtained from simple trigonometric functions.

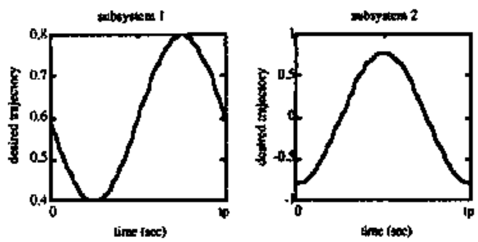

Figure 1. Desired trajectories for polar robot moving in the horizontal plane. Subsystem 1 is radial motion in meters; subsystem 2 is angular motion in radians.

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 paper 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.

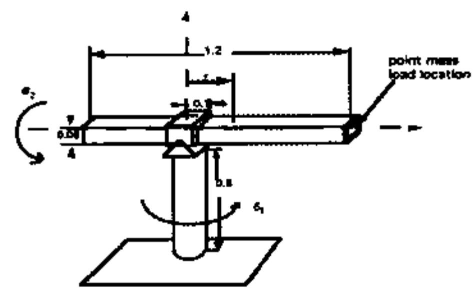

Figure. 2 Configuration of polar coordinate robot

The nonlinear equations for motion of the polar coordinate robot in Figure. 2 are given as

$$
\begin{aligned}
&(m_B + m_L)\ddot{r}(t) - [m_B r(t) + m_L(r(t) + l)]\\
&\dot{\theta}_1(t)^2 = F(t) \qquad\qquad (24)\\
&[I_3 + m_B r(t)^2 + m_L(r(t) + l)^2]\ \ddot{\theta}_1(t)\\
&+ 2[m_B r(t) + m_L(r(t) + l)]\dot{r}(t)\ \dot{\theta}_1(t) = M_1(t)
\end{aligned}
$$

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=39.28kg, its half length is l=0.6, and its moment of inertia about the vertical axis is I_3=1.93kg m². The mass of the point mass load located at the end of the beam is m_L=10kg. The force and moments applied to each joint are supplied by proportional plus derivative feedback controllers given by

$$F(t) = K_1 [r(t) - r^*(t)] + K_2 [\dot{r}(t) - \dot{r}^*(t)] + u_1(t)$$
$$M_1(t) = K_3 [\theta_1(t) - \theta_1^*(t)] + K_4 [\dot{\theta}_1(t) - \dot{\theta}_1^*(t))] + u_2(t) \qquad (25)$$

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.

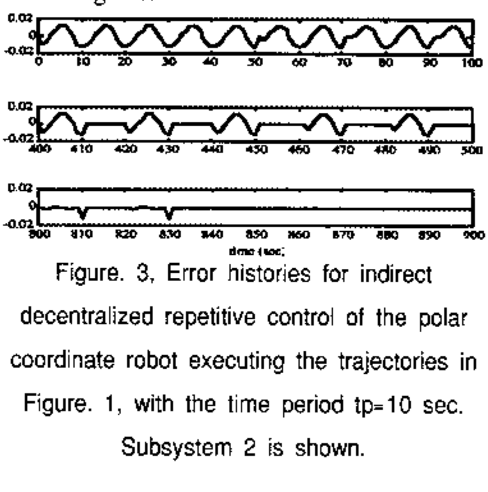

Figure. 3, Error histories for indirect decentralized repetitive control of the polar coordinate robot executing the trajectories in Figure. 1, with the time period tp=10 sec. Subsystem 2 is shown.

Figure 3 shows the resulting error history for the angular motion of the robot when the both joints are learning to execute this desired trajectory. The time period of the repetitive command is chosen as 10 sec in this example, and the sample time for the learning is 1 sec. One repetition is skipped between each repetition that learns. For those repetitions that learn, one repetition is devoted to identification, and in the next repetition that is devoted to learning, the learning signal is adjusted using the identification result. Then the learning progresses to the next time step. Examining Figure. 3, in repetitions 1 and 2, i.e. from 0 to 10 sec, and from 10 to 20 sec, feedback control alone is applied. The first repetition allows the appropriate initial condition to be set up at the beginning of the second repetition, and data from this repetition is used in the modified difference of equation (18) when computing the learning control for repetition 4. The next pair of repetitions uses feedback alone in the third repetition, and in the 4th learning is started for the first time step, which means that differenced data is obtained to identify $C_2(1)B_2(0)$. This is followed by repetition 5 with feedback only, and then in repetition 6 the identification is used to correct the error, producing the zero error point at time 51 sec. Repetition 7 (from 60 to 70 sec) uses feedback only. In repetition 8, the identified value of $C_2(1)B_2(0)$ is used to compute a learning control signal, and the resulting data is used to identify $C_2(2)B_2(1)$. This is used in repetition 10 to correct the repetitive control signal for time step 2 of the repetition. This pattern is continued until p=10 steps are learned at time 420 sec. At this time we have close to zero error for every other repetition, and in between we are using feedback only. The error is not exactly zero due to the coupling introduced between the subsystems by the discretization, and because of the presence of residual transient effects. The wave of learning continues past p to $2p$. From 420 to 430 sec feedback is used, and in 430 to 440 the same learned signal is again applied, in order to produce a zero difference in the first term of equation (29). The same process of identification followed by adjustment of the repetitive control signal was used, but in theory, we could use the already identified values of the $C_2(k+1)B_2(k)$ for all time steps, in order to increase speed of the wave of learning progressing from p to $2p$. After the wave is completed, the second repetitive signal, associated with steps p to $2p$ in the wave, is the desired repetitive control signal, since it applies when the system is on the desired trajectory at the start of the period. This signal is then used for all repetitions, after time 830 sec. The result is a very significant improvement in the tracking error, as shown in the figure.

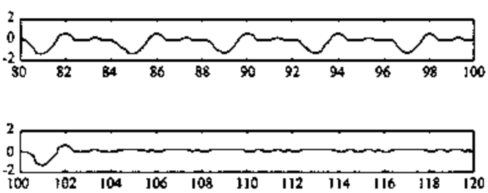

Figure. 4 Error histories using the modified indirect decentralized repetitive control on the polar coordinate robot executing the trajectories in Figure. 1, with the time period tp=2 sec. Subsystem 2 is shown.

In Figure. 4 we modify the repetitive control algorithm in order to get closer to zero tracking error sooner. Once the wave of learning has progressed through p time steps, we have a repetitive control signal that give close to zero tracking error for a repetition, provided that repetition starts with the initial conditions produced by the feedback control law. The modification used here is to let the wave of learning progress for two more time steps, from p to $p+2$, in order to adjust these first time steps of the learning to ones appropriate to the initial conditions produced by the desired trajectory. We can adjust the repetitive control signal to include these two adjusted points. Although this adjusted signal is not guaranteed to give zero error, it should be significantly better than the feedback signal. In Figure. 4, the period of the repetitive command is changed to 2 sec, and the sample time for the learning is 0.2 sec. From 80 to 82 sec is the final feedback only repetition, preceding the completion of learning of p steps in repetition 42, 82 to 84 sec. The control actions for 80 to 84 are repeated for 84 to 88 sec in order to establish appropriate zero differences. From 88 to 90 sec, the previously learned $C_2(1)B_2(0)$ is used to adjust the first time step, and the resulting data is used to improve this time step at 92.2 sec. At 102 sec the wave of learning $p+2$ steps is complete. The repetitive control signal with the first two time steps modified is used from 102 to 110 sec, and is seen to produce small error, and to do so much sooner than when using the previous algorithm. However, the learning is not yet completed, the wave of learning must continue. We use this improved signal in the repetitions that previously used feedback only. Thus, it is as if we are starting the learning process from the beginning, except that this time we are starting with much less error. Therefore, we will in theory need to progress to $3p$ before the learning process is complete. We obtained small error sooner, but at the expense of extending the learning process.

6. CONCLUDING REMARKS

In this paper, centralized indirect repetitive control laws are developed based on indirect adaptive control concepts, and these laws can be applied to time varying linear systems with periodic coefficients, and with repetitive disturbances. Thus, they can apply to robot tracking problems when the robot can be modeled as linearized about the desired trajectory. The requirement that the full state be measured that appeared in a previous repetitive control algorithm was eliminated. Then, assuming that we know an upper bound on the amount of time needed for transients to decay in the system, a decentralized indirect repetitive control law was developed which has guaranteed convergence to zero tracking error in such time varying systems. It is seen that because of the need to allow repetitions in which no learning is done, in order to allow transients to decay, it takes longer for convergence to zero tracking error in the decentralized indirect repetitive control problem than in the decentralized indirect learning control problem.

7. REFERENCES

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

[2] R. H. Middleton, 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. An earlier version is available as University of Newcastle, Newcastle, Australia, Department of Electrical Engineering Technical Report EE8546, 1985.

[3] M. Phan 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

[4] E. G. Harokopos 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.

[5] M. Phan and R. W. Longman, "Liapunov Based Model Reference Learning Control," Proceedings of the Twenty-Sixth Annual Allerton Conference on Communication,

Control and Computing, The University of Illinois, Allerton House, Monticello, Illinois, September 1988, pp. 927-936.

[6] M. Phan and R. W. Longman, "Indirect Learning Control with Guaranteed Stability," Proceedings of the 1989 Conference on Information Sciences and Systems, Johns Hopkins University, Balti more, MD, 1989, pp. 125-131.

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

[8] R. W. Longman, H. S. M. Beigi and C. J. Li, "Learning Control by Numerical Optimization Methods," Modeling and Simulation, Instrument Society of America, Vol. 20, Part 5, 1989, pp. 1877-1882.

[9] M. Phan, 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.

[10] M. Phan, 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, C. L. Kirk, and J. L. Junkins, Editors, Computational Mechanics Publications, Southampton, Boston, and Springer-Verlag, New York, co-publishers, 1990, pp. 321-334.

[11] R. W. Longman and C.-K. Chang, "Learning Control for Minimizing a Quadratic Cost during Repetitions of a Task," AIAA/AAS Astrodynamics Conference, A Collection of Technical Papers, Part 2, Portland, Oregon, August 1990, pp. 530-536.

[12] C.-K. Chang, R. W. Longman, and M. Phan, "Techniques for Improving Transients in Learning Control Systems," Advances in the Astronautical Sciences, American Astronautical Society, Vol. 76, 1992, Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Durango, Colorado, August 1991.

[13] S. C. Lee, R. W. Longman, and M. Phan, "Linear Decentralized Learning Control," Advances in the Astronautical Sciences, American Astronautical Society, Vol. 76, 1992, Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Durango, Colorado, August 1991.

[14] H. S. M. Beigi, C. J. Li, and R. W. Longman, "Learning Control Based on Generalized Secant Method and other Numerical Optimization Methods," 1991 ASME Winter Annual Meeting, Atlanta, Georgia, Symposium on Sensors, Controls, and Quality Issues in Manufacturing, PED-Vol. 55, December 1991, pp. 163-175.

[15] R. W. Longman, S. C. Lee, and M. Phan, "Indirect Decentralized Learning Control," Advances in the Astronautical Sciences,

American Astronautical Society, to appear, Proceedings of the 1992 AAS/AIAA Spaceflight Mechanics Meeting, Colorado Springs, February 1992.

[16] C. K. Chang, R. W. Longman, and M. Phan, "Discrete Time Learning Control in Nonlinear Systems," Proceedings of the 1992 AIAA/AAS Astrodynamics Specialist Conference, an AIAA publication, Hilton Head, South Carolina, August 1992.

[17] R. W. Longman, M. Phan, and J.-N. Juang, "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, September 1992.

[18] R. W. Longman and Y. S. Ryu, "Indirect Learning Control Algorithms for Time-Varying and Time-Invariant Systems," Proceedings of the 30th Annual Allerton Conference on Communication, Control and Computing, The University of Illinois, Allerton House, Monticello, Illinois, September 1992.

[19] S. Arimoto, "Introduction: Brief History of Development of Learning Control in Robotics," appearing in Supplementary Material, Tutorial on Learning Control, Longman, R. W., Arimoto, S., Editors, 1991 IEEE Conference on Intelligent Control, Arlington, VA, August 1991.

[20] M. Phan, R. W. Longman, and J-N. Juang, "Indirect Repetitive Control for Linear Discrete Multi variable Systems," Proceedings of the 27th Annual Allerton Conference on Communication, Control, and Computing , The University of Illinois, Allerton House, Monticello, Illinois, September 1989, pp. 867-876.

[21] R. W. Longman and E. J. Solcz, "Small Gain Robustness Issues in the P-Integrator Repetitive Controller," AIAA/AAS Astrodynamics Conference, A Collection of Technical Papers, Part 2, Portland, Oregon, August 1990, pp. 537-551.

[22] E. J. Solcz, and R. W. Longman, "Disturbance Rejection in Repetitive Controllers," Advances in the Astronautical Sciences, American Astronautical Society, Vol. 76, 1992, Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Durango, Colorado, August 1991.

[23] E. J. Solcz and R. W. Longman, "Robustness of Repetitive Controllers to Frequency Uncertainties," Proceedings of the 1992 Conference on Information Sciences and Systems, Department of Electrical Engineering, Princeton University, March 1992.

[24] E. J. Solcz and R. W. Longman, "Repetitive Control in the Presence of Multiple Periodic Disturbances," Proceedings of the 1992 AIAA/AAS Astrodynamics Specialist Conference, an AIAA publication, Hilton Head, South Carolina, August 1992.

[25] T. Mita and E. Kato, "Iterative Control and Its Applications to Motion Control of Robot Arm — A Direct Approach to Servo Problems," Proceedings of the 24th IEEE Conference on Decision and Control, Fort Lauderdale, Fla., December 1985.

[26] K. Furuta and M. Yamakita, "Iterative Generation of Optimal Input of a Manipulator," Proceedings of the IEEE Conference on Robotics, 1986.

[27] S. Hara, T. Omata, and M. Nakano, "Synthesis of Repetitive Control Systems and Its Applications," Proceedings of the 24th IEEE Conference on Decision and Control, Fort Lauderdale, Fla., December 1985.

[28] M. Tomizuka, T.-C Tsao, and K.-K Chew, "Discrete-Time Domain Analysis and Synthesis of Repetitive Controllers," Proceedings of the American Control Conference, Atlanta, Georgia, June 1988.

[29] T.-C Tsao and M. Tomizuka, "Adaptive and Repetitive Digital Control Algorithms for Non circular Machining," Proceedings of the American Control Conference, Atlanta, Georgia, June 1988.

95년 7월 최초 접수, 96년 10월 최종 수정