Sensors, Controls, and Quality Issues
in Manufacturing
ASME 1991

LEARNING CONTROL BASED ON GENERALIZED SECANT METHOD AND OTHER NUMERICAL OPTIMIZATION METHODS

Homayoon S. M. Beigi
International Business Machines
Thomas J. Watson Research Center
Yorktown Heights, New York

C. James Li and R. W. Longman
Department of Mechanical Engineering
Columbia University
New York, New York

ABSTRACT

When control systems are given the same task to perform repeatedly, they will usually repeat the same errors in executing the command, except for certain random noise effects. The field of learning control has developed to produce controllers that can improve their performance at a given task with each repetition of the task. To date, learning control algorithms have concentrated on use of analogues of integral control to eliminate errors, and recently on conversions of adaptive control to the learning control problem. This paper studies an alternative of using numerical optimization techniques to determine ways to improve controller performance from one repetition to the next. Quasi-Newton methods, the direction set method, certain derivative free algorithms and the generalized secant method are studied. It is shown analytically that the generalized secant method converges to zero tracking error in fewer repetitions than other methods aimed at direct minimization of the Euclidean norm of the error. Simulations show good results when applied to the control of nonlinear systems that include a pendulum and a non-linear mass-spring-dashpot system.

I. INTRODUCTION

A large percentage of the practical applications of control theory involve systems which are repeatedly asked to perform the same task. Examples include tracking problems for robots on assembly lines, as well as a large number of manufacturing applications. Standard controller design methods produce systems that do an imperfect job of executing a command -- and when the command is repeated the system repeats these same errors everytime. It is perhaps a bit primitive to persist in repeating the same errors. And in the last few years a theory of learning control has been developed in the literature, generating controllers that can learn from previous experience at performing a specific task, see for example [1-7]. The fact that a specific task is involved repeatedly, distinguishes the learning control problem from the adaptive control problem.

To date, the learning control literature has been based on either analogues of integral control in the repetition domain [1-4], or more recently adaptive control ideas have been converted to apply to the learning control problem in [5-7]. This paper is devoted to a preliminary investigation of a third option -- the use of numerical optimization techniques as a method of accomplishing learning in repetitive tasks.

The approach bears a certain philosophical similarity to a few optimal control investigations that use the real world rather than a computer model to evaluate each iteration in the attempt to converge on the optimal control. Such an approach can avoid the sometimes lengthy process of obtaining a good model before the optimization problem can start. Reference [3] develops such an idea for minimizing the integral of the absolute value of the error in continuous systems.

Here, we consider the learning control problem for discrete systems, a system of linear algebraic equations relating the change in the state of the system to the change in the control action from one repetition of the task to another. From this set of linear equations which gives the transition between any two repetitions, we formulate an optimization function which is quadratic in the error. We study numerical optimization methods to determine how to adjust the control history from one repetition to the next in order to decrease this quadratic function of the error as the repetitions of the task proceed.

Furthermore, we utilize a generalized secant method to solve the system of linear algebraic equations directly for the appropriate control action and parameter estimates that will minimize the tracking error of the controlled dynamic system, without any prior knowledge of the system parameters except the order of the system. It is shown analytically that the generalized secant method requires the least number of repetitions to converge among all the methods given in this paper. Hence, simulations were conducted to study practical issues of this controller.

II. PROBLEM FORMULATION

Consider a general discrete-time linear time-varying or time-invariant system,

$$x^k(t+1) = A(t)\, x^k(t) + B(t)\, u^k(t) + w^k(t) \tag{1a}$$

$$y^k(t+1) = C(t+1)\, x^k(t+1) \qquad t = 0, 1, 2, \ldots, p-1 \qquad k = 0, 1, 2, \ldots \tag{1b}$$

where the state, x, is n-dimensional, the control, u, is m-dimensional, the output, y, is q-dimensional $(q \geq m)$, t is the time step in the p-step repetitive operation, and k is the repetition number. For simplicity, the dimension n of the system is assumed known, but generalization to just knowing an upper bound on n is easily considered. Also, A, B, C, and w^k are assumed unknown -- otherwise one could determine in advance what control to use to minimize tracking error, and there would be no need for learning control. In the learning control problem (as contrasted with the repetitive control problem in [2]) the system is assumed to always start from the same initial state in each repetition of the task. Matrix A includes any state or output feedback control present in the system and the symbol u^k is reserved for the signal added to the control for learning purposes. A time varying model is considered because many applications, such as in robotics, involve nonlinear dynamic systems which when linearized produce linear models with coefficients that vary with time, and vary in the same manner each repetition. In such repetitive operations it is often the case that there will be disturbances $w^k(t)$ that repeat with each repetition of the task, and learning can be made to also correct for this source of errors in a natural way. We will limit ourselves to repetitive $w^k(t)$. Therefore the superscript k will be dropped. One of the purposes of the feedback control is to handle any non-repetitive disturbances, and these will be ignored for purposes of designing the learning control (some analysis of these random noise effects in learning control can be found in [4]).

The solution to (1) for the p-steps of the repetitive process can be written as

$$x^k(t+1) = \Big(\prod_{j=0}^{t} A(j)\Big) x^k(0) + \sum_{j=0}^{t} \Big(\Big(\prod_{r=j+1}^{t} A(r)\Big)\big(B(j)\, u^k(j) + w(j)\big)\Big) \tag{2}$$

where the product symbol is taken to give the identity matrix if the lower limit is larger than the upper limit. Let p be the total number of time steps. Defining a difference operator $\delta_k z(t) = z^k(t) - z^0(t)$ for any variable z, and using the fact that $x^k(0)$ and w(t) are repetitive, one can write,

$$y^k = y^0 + P\, \delta_k u \tag{3}$$

where,

$$y^k = [y^{kT}(1)\ \ y^{kT}(2)\ \ \ldots\ \ y^{kT}(p)]^T$$

$$u^k = [u^{kT}(0)\ \ u^{kT}(1)\ \ \ldots\ \ u^{kT}(p-1)]^T$$

$$P = \begin{bmatrix} C(1)B(0) & 0 & \cdots & 0 \\ C(2)A(1)B(0) & C(2)B(1) & & 0 \\ & & \cdots & . \\ C(p)A(P-1)A(P-2)\ldots A(1)B(0) & \cdots & \cdots & C(p)B(p-1) \end{bmatrix}$$

The objective is to find a change in the control at each repetition k that will, as k progresses, minimize the quadratic error function

$$f(\delta_k u) = (y^k - y_D)^T (y^k - y_D) \tag{4}$$

where y_D is the desired output history for the p-step process. In previous work on learning control in discrete systems [4-7], as well as in the extensions of these papers for journal publication, attention had to be directed to the problem of ensuring that the special desired trajectory was in fact a feasible trajectory for the system to perform. For example, one method involved assuming knowledge of an upper bound on the system dimension n, and assuming the system is controllable. Then specifying the desired measured output variable history every n steps is guaranteed by modern control theory to be a feasible specification. In the present approach these conditions can be relaxed if one is satisfied with minimizing (4) rather than insisting that (4) be driven to zero.

Substitute (3) into (4),

$$\begin{aligned} f(\delta_k u) &= (y^0 + P\,\delta_k u - y_D)^T (y^0 + P\,\delta_k u - y_D) \\ &= \delta_k u^T (P^T P)\,\delta_k u + 2\,\delta_k u^T [P^T (y^0 - y_D)] + (y^0 - y_D)^T (y^0 - y_D) \end{aligned} \tag{5}$$

Note that the gradient of this objective function and the Hessian matrix of second partials are,

$$\frac{df}{d(\delta_k u)} = 2P^T P\,\delta_k u + 2P^T (y^0 - y_D) \tag{6}$$

$$\left[\frac{d^2 f}{d(\delta_k u)^2}\right] = 2P^T P \tag{7}$$

It is now possible to characterize the nonlinear optimization problem represented by our learning control objective. We wish to

1. minimize a function which is known to be quadratic in the control change variable $\delta_k u$,
2. but the first derivative of this function is not directly available, because we do not know the system matrices A, B, C, and hence do not know P,
3. and the second derivative is similarly unavailable.

Having characterized the problem, we present an overview and assessments of the possible approaches to the learning control problem using numerical optimizing method.

III. LEARNING CONTROL BY NUMERICAL OPTIMIZATION METHODS

3.1 Quasi-Newton Methods

Over the last couple of decades various finely tuned quasi-newton based methods have been generated for the minimization of unconstrained nonlinear functions. Among these methods are the rank 1 Broyden method, DFP, BFGS, and other members of the Broyden family of methods as well as SSVM methods. An important characteristic of these methods is the use of the iterative process to approximate the Hessian matrix so that no explicit expression for the second derivative is needed. If we could evaluate the first derivative $df/d(\delta_k u)$, then the BFGS method would converge to the optimal $\delta_k u$ in approximately mp repetitions. One might note that if one actually knows the $df/d(\delta_k u)$ of (6), then one could equate it to zero and solve for $\delta_k u$ and obtain the optimal $\delta_k u$ in one step -- something which could be classified as a Newton method in numerical optimization.

Since we do not know $df/d(\delta_k u)$, one can consider the use of Quasi-Newton methods with approximation of the gradient obtained by a finite difference method. In order to obtain these differences for all of the mp elements of the gradient vector it would require approximately mp repetitions of the task to make one evaluation of the gradient vector. The total number of repetitions required for convergence would exceed $(mp)^2$. This makes such methods a poor choice in learning control. Methods that do not require the use of a gradient in picking the search direction are called for. Note that the same difficulty eliminates the use of steepest descent and conjugate gradient methods.

3.2 A Direct Search Method

A method which does not require knowledge of gradients, is the direct search method of optimization as expressed by Wood [9] and later by Hooke and Jeeves [10]. The direct search method, which takes advantage of the quadratic nature of the objective function, is as follows:

1. Pick mp orthogonal directions in the $\delta_k u$ space. For each of these directions in succession, perform the line search as in the next steps.
2. Take a step along the chosen direction in $\delta_k u$ space, and apply the resulting control in the next repetition. Evaluate f from the data of this repetition.
3. If f decreased pick another step in the same direction, if it increased pick a step in the opposite direction, and apply to the system.
4. Since the quadratic objective function surface in the plane of the chosen direction is a parabola, the data from 2 and 3 determines this parabola, and can be used to find the minimizing $\delta_k u$ for this direction. This completes the line search.
5. Return to 2 with the next direction.

This algorithm will improve the tracking every three repetitions, provided there is no noise in the measurements. There is no guarantee for a finite convergence unless the directions of search are mutually conjugate about the Hessian matrix of the quadratic objective function. Care must be taken to avoid large disturbances to the repetitive process during the line search.

3.3 Conjugate Direction Methods

If the unidimensional search of the last section were used along mp mutually conjugate directions, then in the absence of noise, the system would converge after minimizing the quadratic objective function in all these directions. There are a few conjugate direction methods available in the literature [11-13]. These methods start with a set of mp orthonormal directions which could be coincident with the columns of an mpxmp identity matrix. Then, by doing unidimensional searches in these directions, a new set of directions is generated.

Rosenbrock's [11] method provides a direction coincident with one of the eigenvectors of the Hessian (H) matrix. Eigenvectors of H are mutually conjugate and point in the direction of the minimum of the quadratic objective function. One such eigenvector is reached after mp unidimensional searches along the orthonormal directions. The rest of the mp-1 directions are found by using the Gram-Schmidt [14] orthogonalization procedure. This cycle is then repeated to provide another eigenvector of H. There is no guarantee that the eigenvectors are not repeated. Therefore, there is no general finite convergence proof available.

Rosenbrock's method might converge to a non-minimum point in which case the method breaks down. To avoid or delay the failure of the minimization procedure, the method due to Davies, Swann, and Campey [12] (DSC) uses a renumbering system for the directions of search. Another method due to Powell [13] provides a direction set with one of the directions conjugate to the complementary mp-1 directions. This set is obtained after a cycle similar to that of Rosenbrock and DSC. However, Powell's method does not require any orthogonalization procedure after every cycle and it never breaks down as Rosenbrock and DSC methods do.

None of the above methods guarantees a finite convergence since the main direction after each cycle might be linearly dependent on directions obtained in previous cycles. However, if the set of directions are linearly independent, then the function minimization will converge within mp(2mp+1) repetitions. This applies to Powell's second method.

3.4 An Identification Test Method

It is always possible to identify a system from impulse response tests. In this section we study the use of such tests, and then in the next section we will study how to accomplish improvement with each repetition. If data is taken for a sequence of repetitions (j=0,1,2,...,N) then eq. (3) gives

$$\left[\delta_N y \; \delta_{N-1} y \; \ldots \; \delta_1 y\right] = P \left[\delta_N u \; \delta_{N-1} u \; \ldots \; \delta_1 u\right] \qquad (8)$$

Once N is sufficiently large, this can be solved for P by use of an inverse or pseudo-inverse. If one makes each $\delta_j u$ a different column of the identity matrix, which is the discrete equivalent of an impulse response test, then one can eliminate the need for inverting a matrix. In other cases, one can take advantage of the lower block triangular nature of P to efficiently compute the inverse. The noise level can be important in deciding the best approach. Once P is determined, then solving (6) gives

$$\delta_{j+1} u = P^{\dagger} (y^0 - y_D) \qquad (9)$$

where the superscript † indicates the Moore-Penrose pseudoinverse.

In the case that the desired trajectory is feasible for system 1, and the desired trajectory exactly specifies all the freedom in the system, then the solution to (9) is the same as solving (3) for v_j to obtain $v_j = P^{-1} (y^j - y_D)$. This approach requires mp+1 steps to optimize the tracking in the no noise case.

A Modification to Obtain Improvement Each Repetition: In practice one usually has a model of the system, call it P^m, when one starts the repetitive control, although one may not be very confident about the model. As the data from each repetition arrives one could modify the model to match the data, using the smallest possible change in the elements in the least squares sense. At step j, one would choose ΔP to satisfy

$$\left[\delta_j y \; \delta_{j-1} y \; \ldots \; \delta_1 y\right] = \left[P^m + \Delta P\right] \left[\delta_j u \; \delta_{j-1} u \; \ldots \; \delta_1 u\right]$$

or

$$\Delta P = \left\{ \left[\delta_j y \; \delta_{j-1} y \; \ldots \; \delta_1 y\right] - P^m \left[\delta_j u \; \delta_{j-1} u \; \ldots \; \delta_1 u\right] \right\} \left[\delta_j u \; \delta_{j-1} u \; \ldots \; \delta_1 u\right]^{\dagger} \qquad (10)$$

where the superscript † indicates the Moore-Penrose pseudoinverse. One can show that this pseudoinverse will minimize $tr(\Delta P^T \Delta P)$, i.e. the sum of the squares of the changes of all elements of ΔP.

The method becomes:

1. Pick a direction for $\delta_1 \mathbf{u}$ and make the first repetition to obtain the $\delta_1 \mathbf{y}$, $\delta_1 \mathbf{u}$ pair.
2. Solve for ΔP from (10).
3. Obtain $\delta_{j+1} \mathbf{u}$ from (9) substituting $P^m + \Delta P$ for P. The singular value decomposition of $[\delta_{j+1} \mathbf{u} \; \ldots \; \delta_1 \mathbf{u}]$ involved in the pseudo-inverse will determine if $\delta_{j+1} \mathbf{u}$ is linearly independent. In the event that it is not, choose a linearly independent direction at this step.
4. Repeat steps 2 and 3 until convergence is obtained.

This approach takes full advantage of a priori information about the system, improves the control system behavior with every repetition, handles noise in a natural way that smooths its influence, handles in a logical manner specified desired trajectories that are not feasible for the system to perform, and in the no noise case produces convergence in mp+1 repetitions or less. The effectiveness of the method by comparison to the direction set method above is fundamentally related to the fact that the quadratic cost (5) is not a general quadratic function, but rather the linear and the quadratic matrix coefficients P^T and P^TP are related, and this extra information has been incorporated in the new algorithm. Rather than trying to identify the quadratic surface of f as in the direction set method, the linear relation (3) from which the quadratic is derived is identified, with a resulting savings in repetitions. A numerical optimization method with this same property is presented in the next section.

The above method involves identification of the system model in the repetition domain, which by analogy to learning control can be called learning identification. Various approaches to identification in the repetition domain are presented in [7,15,16], for both time-varying and time-invariant cases.

When the system is time-invariant one can accelerate the learning control convergence by using the identification of elements of P to compute the time domain description A, B, C, from which one can complete the elements of P before enough data has been taken to obtain all elements directly. The algorithm is as follows:

1. While applying the above algorithm, use the lower block triangular nature of P to determine successively CB, CAB, CA^2B, etc.

2. Form the Hankel matrix of these Markov parameters for the appropriate dimension, if one knows the order n of the system, for any repetition number large enough to have generated $CA^{2n-1}B$. Later repetitions allow more smoothing. If n is not known, monitor the rank of the Hankel matrix with repetitions.

3. Use the Eigensystem Realization Algorithm (see for example [17], or use the Ho Kalman algorithm) to obtain A, B, and C. Note that one does not need to identify w(t) of the time domain model.

4. Use A, B, and C to construct the full P matrix.

5. Find the optimizing control using this in (9).

Note that if the full state is measured, then the Eigensystem Realization Algorithm can be bypassed, by simply identifying A and B using a pseudoinverse solution of the sequence of eq. (1) for succeeding steps, after eliminating w(t) through differencing.

IV. THE GENERALIZED SECANT METHOD

Define the error, e^k, as the difference between the output y^k and the desired trajectory y_D,

$$e^k = y^k - y_D \tag{11}$$

Define the change of input between two consecutive repetitions as

$$v^k = u^{k+1} - u^k \tag{12}$$

Then from eq. (3), one could obtain

$$Pv^k = e^{k+1} - e^k \tag{13}$$

Since our goal is to reduce the tracking error, e_k, as the task is being repeated, we aim at finding a change in our input which would lead to zero tracking error at the next repetition, $e^{k+1}=0$. For that input change, eq. (13) becomes

$$P\bar{v}^k = -e^k \tag{14}$$

Equation (14) suggests that the change of input, v^k, can be deduced from P and e^k. Since P is not available, an approximation to P at the k^{th} repetition, P^k, has to be used. Hence

$$P^k v^k = -e^k \tag{15}$$

To obtain v^k, the Moore-Penrose pseudo-inverse can be used to minimize the following norm,
$\| P^k v^k + e^k \|$

namely,

$$v^k = -P^{k\dagger} e^k \tag{16}$$

Hence, the control action at the (k+1)st repetition, u^{k+1}, can be calculated according to eq. (12), $u^{k+1} = u^k + v^k$. In the following, the method for finding P^k will be described.

At repetition k, denote the difference between the actual system parameters P and the approximation P^k by the matrix D^k.

$$P = P^k + D^k \tag{17}$$

Substituting eq. (17) into (13) and rearranging,

$$D^k v^k = e^{k+1} - e^k - P^k v^k \tag{18}$$

Since D^k is a qp by mp matrix and the other terms in eq. (18) are vectors, one can not directly solve the complete D^k from eq. (18).

A "solution" that would solve D^k along a direction defined by z^k is,

$$D^k = \frac{(e^{k+1} - e^k - P^k v^k)\, z^{kT}}{z^{kT} v^k} \tag{19}$$

while z^k are chosen in the following way[18]:

- If $k \geq n-1$ then z^k is chosen orthogonal to the previous n-1 control steps $v^{k-n-1}, \dots, v^{k-1}$.
- If $k < n-1$ then z^k is chosen orthogonal to the available k steps, $v^0, \dots, v^{k-1}$, where $n \leq mp$ is the maximum number of linearly independent v's which could be found.

A few orthogonalization methods are available in the literature which may be used for the actual evaluation of the z vectors. These include the well-known Gram-Schmidt orthogonalization process[14] and a more advanced technique due to Fletcher [19] in which the number of vectors in the set may be changed.

Together with an orthogonalization method, eqs. (16), (17), and (19) form a learning control algorithm,

$$v^k = -P^{k\dagger} e^k \tag{20}$$

$$P^{k+1} = P^k + D^k$$

$$= P^k + \frac{(e^{k+1} - e^k - P^k v^k)\, z^{kT}}{z^{kT} v^k} \tag{21}$$

Proof of Convergence

The set of z^k picked by an orthogonalization method for eq. (21) has the property that $z^{kT} v^j = 0$ for $0 < k-j \leq n-1$. Consequently

$$D^k v^j = 0 \qquad 0 < k-j \leq n-1 \tag{22}$$

Therefore,

$$P^{k+1} v^j = [P^{j+1} + D^{j+1} + \dots + D^k]\, v^j$$

$$= P^{j+1} v^j \qquad 0 < k-j \leq n-1 \tag{23}$$

Since $n \le mp$ is the maximum number of linearly independent v's, v^n has to be a linear combination of all previous n linearly independent control steps

$$v^n = \alpha_0 v^0 + \ldots + \alpha_{n-1} v^{n-1} = \sum_{j=0}^{n-1} \alpha_j v^j \tag{24}$$

Premultiply eq. (24) by P^n, one has

$$P^n v^n = \sum_{j=0}^{n-1} \alpha_j P^n v^j \tag{25}$$

Let k in (23) be n-1,

$$P^n v^j = P^{j+1} v^j = (P^j + D^j) v^j \qquad 0 \le j < n-1 \tag{26}$$

Substituting (18) into (26),

$$P^n v^j = e^{j+1} - e^j \tag{27}$$

According to (13), the right hand side of (27) must be equal to Pv^j. Hence

$$P^n v^j = P v^j \tag{28}$$

Substituting (28) into (25),

$$P^n v^n = P \sum_{j=0}^{n-1} \alpha_j v^j \tag{29}$$

According to (24), the summation in the right hand side of (29) is equal to v^n. Hence

$$P^n v^n = P v^n \tag{30}$$

Substituting (13) into (30),

$$\begin{aligned} P^n v^n &= P v^n \\ &= e^{n+1} - e^n \end{aligned} \tag{31}$$

If v^n satisfies (15), then,

$$e^{n+1} - e^n = -e^n \tag{32}$$

which means that e^{n+1} must be zero. However, if v^n only satisfies (15) in a least square sense, then,

$$e^{n+1} = P v^n + e^n \tag{33}$$

has minimum norm.

If n (the rank of P^0-P) is equal to mp, then after mp+1 repetitions of the task, a minimum norm tracking error will be achieved. (mp+1) is the maximum number before convergence. But if P^0 is a good guess, it can converge faster.

V. Simulation and Results

The learning control algorithm based on the generalized secant method was tested on a linear and two nonlinear dynamical systems. The first system is a linear mass-spring-dashpot system. The second system is a non-linear mass-spring-dashpot system and the third system is a pendulum with damping.

5.1. Governing Equations of Simulated Systems

System 1 is a linear Mass-Spring-Dashpot system. This system is described by the following differential equation

$$m\ddot{\alpha} + c\dot{\alpha} + k\alpha = T$$

where m, c, k are 1.0 kg, 1.0 Ns/m, 1.0 N/m. (34)

System 2 is a nonlinear Mass-Spring-Dashpot system. The simulated system dynamics is given by the following differential equation:

$$m\ddot{\alpha} + c(1+\alpha^2)\dot{\alpha} + (k|\alpha|)\alpha = T \tag{35}$$

where m, c, k are the same as those of system 1.

System 3 is a one-degree-of-freedom pendulum of mass m, length l, angle from vertical α, and damping c. The dynamics of the simulated pendulum obeyed the following differential equation:

$$m l^2 \ddot{\alpha} + c\dot{\alpha} + m g l \sin\alpha = T \tag{36}$$

where m, l, c are 1.0 kg, 0.1 m, 1.0 Ns/m.

The desired trajectory is given in Fig. 1. As shown in Fig. 1, the desired trajectory swing through two radians so that the system nonlinearity of system 3 can not be neglected.

5.3 Simulation Results

System 1 was first simulated to test the generalized secant controller. The results of this simulation are given in Fig. 2 which plots the sum of squares of the errors versus the repetition number. The first repetition was done using a self-tuning regulator with forgetting factor 1 (a complete theory of the self tuning regulator is given in [7]). The sum of squares of errors in this simulation converges to 1.08 and did not go to zero. The flat part of the curve is the result of control steps orthogonal to the direction of z_k, in which the P matrix is to be updated.

Barnes [18] proposes a practical stability criterion which translates to imposing a small positive lower limit to the inner product of the direction in which the P matrix is to be updated and the control step taken. Namely, a control step is checked before it is actually applied and if it does not satisfy the inequality (37) it will not be applied.

$$\frac{|z_k^T v_k|}{\|z_k\|_E \, \|v_k\|_E} > \rho \qquad 0 < \rho < 1 \tag{37}$$

This ensures the control step, v_k, is not going to be applied in a direction orthogonal to the direction of z_k, in which the P matrix is going to be updated. If a control step is rejected for not meeting (37), then a new step v_k should be found. The best choice for the new step is a step in the same direction as the P update direction, z_k, then the inequality of (37) is always satisfied.

Barnes [18] suggests taking the magnitude of the new step to be equal to that of the rejected step. This suggestion was shown to trigger instabilities in practice when applied to simulated nonlinear systems in this paper. This instability suggests that the size of the new steps ought to be small enough to keep the state of the controlled system from being driven too far away from the nominal desired trajectory. We find that a step size a little greater than the numerical accuracy of the computer usually works well.

The introduction of the checking of control steps worked well in reducing the tracking error to zero for system 1 (see Fig. 3). We then applied the controller to a set of more practical dynamic systems involving nonlinear dynamics such as those described in eqs. (35) and (36) which when linearized can be approximated by linear models with coefficients that vary with time step, and vary in the same manner each repetition. Figures 4 and 5 show the plots of the sums of squares of errors for system 2 and system 3. In both of these simulations the ρ in eq. (37) was set to 10^{-4}.

In application to the system 1 and system 2, zero tracking error was achieved after 8 repetitions which is less than the ceiling provided by the theory (10 repetitions). Similarly, the pendulum reached a zero trajectory error after only 9 repetitions which is again one repetition less than the ceiling provided by the theory.

The flat part of the curves in Figs. 3 through 5 are caused by rejecting of control steps. Since the amplitude of the new control steps replacing the rejected ones has to be kept very small, there is no significant change of state from one repetition to the next repetition. After a few repetitions involving the rejection of control steps, updates done to the P matrix bring it closer to the ultimate P and therefore a control action calculated from it usually leads to a significant improvement in tracking errors.

It should be noted that the theory in this paper was derived for control of a time-varying dynamic system while simulations were done on highly nonlinear systems. These simulations provide practical evidence on the robustness of this learning controller.

VI. CONCLUSION

This paper has given an overview of the possible numerical unconstrained optimization methods which might be used to generate learning control laws, and has identified the most applicable one among them. The lack of knowledge of the derivatives of the performance criterion constrains the range of the algorithms. It was shown that faster convergence can be obtained by identifying the system in the repetition domain and minimizing based on the current model, than can be obtained by a direct attempt to minimize the quadratic function of the error. The advantage comes from the fact that the quadratic function is not a general one, but one in which the coefficients are related. Knowledge of P is sufficient to determine all the coefficients of this function. The algorithms for minimizing the quadratic function do not take advantage of this special relationship whereas the generalized secant method constructs the coefficients from the knowledge of P. The fact that there are fewer unknown coefficients in P than there are in the quadratic cost function allows fast convergence of the generalized secant method.

Simulation results show that the learning controller based on secant method works well with some practical modifications such as the rejection of noninformative control steps.The generalized secant learning controller was shown to be very stable and robust in a practical sense when applied to the control of truly nonlinear systems.

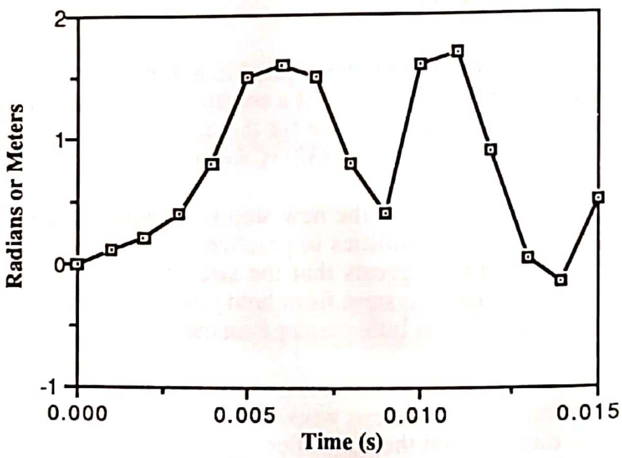


Figure 1 Desired Trajectory

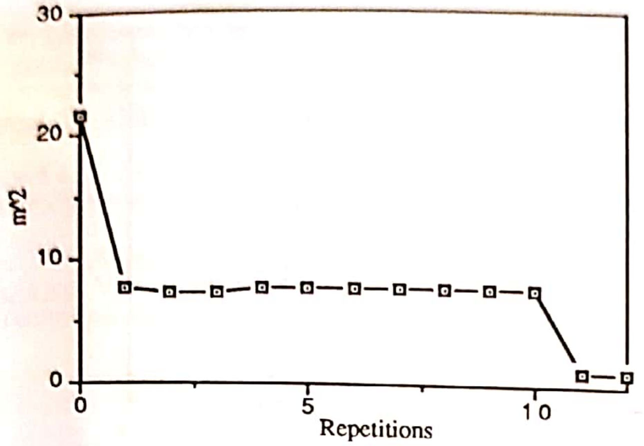
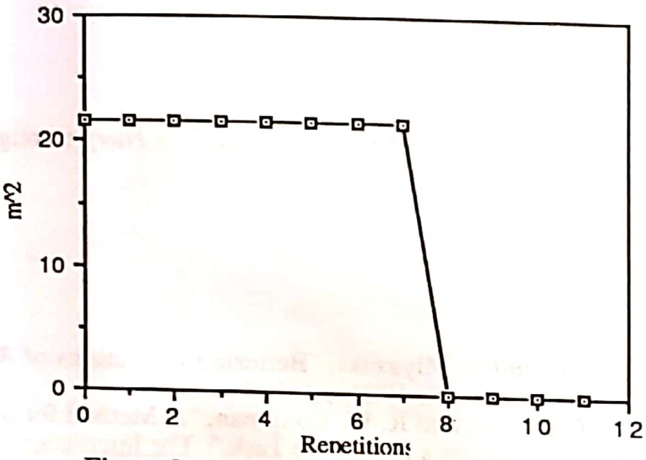


Figure 2 Squares of Errors for System 1
Using the Generalized secant Learning Controller without Rejections



Repetitions

Figure 3 Squares of Errors for System 1
Using the Generalized secant Learning Controller with Rejections

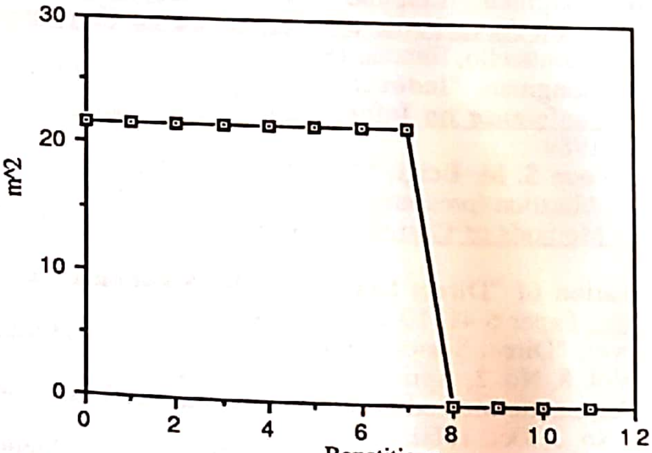


Figure 4 Squares of Errors for System 2 Using the Generalized secant Learning Controller with Rejections)

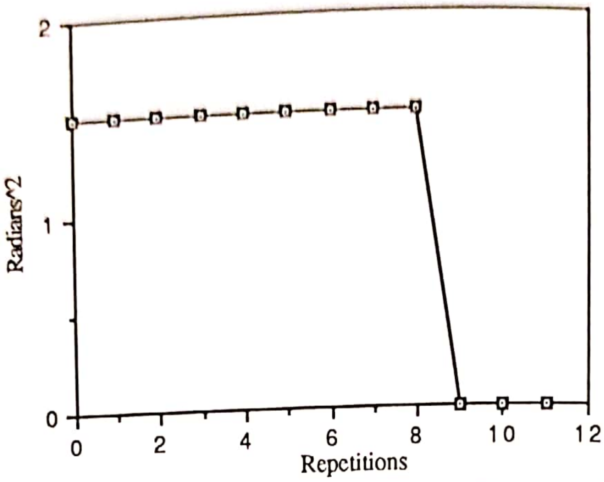


Figure 5 Squares of Errors for the Pendulum Using the Generalized secant Learning Controller with Rejections

Acknowledgment

The authors like to acknowledge the support from Department of Navy through Contract N00039-84-C-0165 P00019 and NASA through Grant NAG 1-649.

REFERENCES

1. S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering Operations 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 Accuracy of a Robot Performing a Repetitive Task," The International Journal of Robotics Research, Vol. 8, No. 5, October 1989, pp. 67-74.
3. E. G. Harokopos and R. W. Longman, "A Learning Controller for Tracking Arbitrary Inputs in Robot Motions," Proceeding of the Tenth IASTED International Symposium on Robotics and Automation, Lugano, Switzerland, June 29 - July 2, 1987.
4. M. Phan and R. W. Longman, "A Mathematical Theory of Learning Control for Linear Discrete Multivariable Systems," Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, Minnesota, August 1988.
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, Allerton House, Monticello, Illinois, 1988.
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, Baltimore, MD, 1989.
7. C. James Li & Homayoon S. M. Beigi, "A Self-tuning Regulator with Learning Parameter Estimation", submitted for publication (presented in ASME WAM, 1990)
8. R. Fletcher, Practical Methods of Optimization, Second Edition, John Wiley & Sons, New York, 1987.
9. C. F. Wood, Application of "Direct Search" to the Solution of Engineering Problems, Westinghouse Res. Lab. Sci., Paper 6-41210-1-P1, 1960.
10. R. Hooke and T. Jeeves, "Direct Search Solution of Numerical and Statistical Problems," J. Assoc. Computer Mach., Vol. 8, No. 2, April 1961, pp. 212-229.
11. H. H. Rosenbrock, "Automatic Method for Finding Greatest or Least Value of Function," The Computer Journal, Vol. 3, No. 3, Oct. 1960, pp. 175-184.
12. W. H. Swann, "Report on the Development of a New Direct Search Method of Optimization," Imperial Chemical Industries, Ltd. Central Instr. Lab. Res. Note 6413, 1964.
13. M. J. D. Powell, "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives," The Computer Journal, Vol. 7, 1964, pp. 155-162.
14. B. Noble and J. W. Daniel, Applied Linear Algebra, Second Edition, Prentice-Hall, 1977, pp. 138-143.

15. M. Phan and R. W. Longman, "Learning System Identification," Proceedings of the Modeling and Simulation Conference, Pittsburgh, PA, 1989, to appear.
16. J. N. Juang, L. Horta, and R. W. Longman, "ERA System Identification in the Repetition Domain," Proceedings of the 1989 AIAA/AAS Astrodynamics Specialist Conference, 1989, to appear.
17. R. W. Longman and J. N. Juang, "A Variance Based Confidence Criterion for ERA Identified Modal Parameters," Astrodynamics, 1987, Advances in the Astronautical Sciences, Vol. 65, Part 1, 1988, pp. 581-601.
18. J. G. P. Barnes, "An Algorithm for Solving Nonlinear Equations Based on the Secant Method," Computer Journal, Vol. 8, 1965, pp. 66-72.
19. R. Fletcher, "A Technique for Orthogonalization," J. Inst. Maths Applics, Vol. 5, 1969, pp. 162-166.