A Heuristic Approach to Reinforcement Learning Control Systems

M. D. WALTZ AND K. S. FU, MEMBER, IEEE

Abstract—**This paper describes a learning control system using a reinforcement technique. The controller is capable of controlling a plant that may be nonlinear and nonstationary. The only a priori information required by the controller is the order of the plant. The approach is to design a controller which partitions the control measurement space into sets called control situations and then learns the best control choice for each control situation. The control measurements are those indicating the state of the plant and environment. The learning is accomplished by reinforcement of the probability of choosing a particular control choice for a given control situation. The system was stimulated on an IBM 1710-GEDA hybrid computer facility. Experimental results obtained from the simulation are presented.**

I. Introduction

IN RECENT YEARS, the application of learning to automatic control systems has become an important area of research [1]–[8]. The system described in this paper is a learning control system in the sense that it is capable of developing and improving a control law in the case where there is very little a priori information about the plant and environment available. It has a greater capability than a conventional adaptive system due to the fact that it recognizes similar recurring situations and uses and improves the best previously obtained control law for this control situation. In addition, the identification of the plant characteristics is not required.

Consider a plant which is disturbed by the environment and obeys a differential equation of the form

$$\dot{X} = \psi(X, u, V, N, t) \tag{1}$$

where

$X = (x_1, \cdots, x_n)$ is the state vector defined in state space Ω_x,

$u =$ is the control signal here called the "control choice,"

$V = (v_1, \cdots, v_m)$ is the environment vector defined in space Ω_v, and

$N = (n_1, \cdots, n_n)$ is the output disturbance vector including output measurement noise, defined in space Ω_x.

The index of performance of the system is of the form

$$IP = \sum_{r=1}^{n} r(\,|\,X_1(rT)\,|\,)^a \tag{2}$$

where "a" is a constant selected by the designer. The block diagram of the system is shown in Fig. 1. The environment vector is obtained from the measurements of the environmental parameters which affect the plant dynamics. Thus the complete state of the plant at any particular instant can be specified by a measurement vector $M = (x_1, \cdots, x_n, v_1, \cdots, v_m)$[1] in space Ω_M. $u \in \Omega_u$, where Ω_u is a set with a finite number of admissible control signals. It is assumed that N is insignificant ($\|N\| < \epsilon$) during most of the operating time. The occurrence of any significant disturbance will cause a signal N' to be detected by the controller. ψ is an unknown function which may be nonlinear. A measurement vector M is obtained and a new control choice u is determined every T seconds. T, the sampling period, must be long enough to allow for a significant change in X for a typical u. The value of T may be preset (if the approximate system response time is known) or learned by a trial and error procedure.

The controller learns to drive the state vector X from any set of initial conditions to the vicinity of the origin in the state space in a way which approaches the optimum as defined by the system IP. The learning is accomplished by establishing a stimulus-response type relationship or mapping between elements of the space

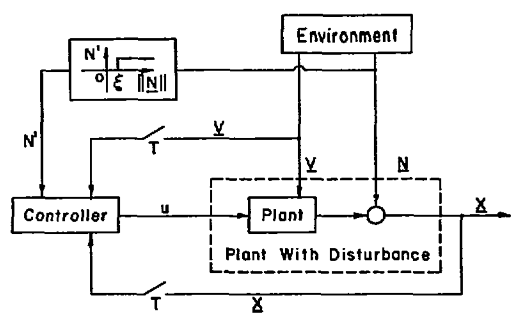

Fig. 1. Simplified block diagram for system.

Manuscript received November 3, 1964; revised April 5, 1965, and July 23, 1965. The work in this paper was supported in part by the Air Force Office of Scientific Research, Control AF AFOSR 62-351, by the National Science Foundation, Grant GP-2183, and by a Collins Radio Fellowship.

M. D. Waltz is with the Youngstown Sheet and Tube Company Research Center, Youngstown, Ohio. He was formerly with the Control and Information Systems Laboratory, School of Electrical Engineering, Purdue University.
K. S. Fu is with the Control and Information Systems Laboratory, School of Electrical Engineering, Purdue University, Lafayette, Ind.

[1] $v_1, \cdots, v_m$ are the measurable environmental parameters, for example, the surrounding temperature which affects the plant characteristics.

Ω_M and elements of the space Ω_u. The first step is to partition the space Ω_M into classes or sets called control situations. The best control choice from the set Ω_u is considered the same for all members in one class or one control situation. The next step is to determine which control choice is the best for each control situation. This is done through a reinforcement procedure which will be described in detail later.

II. Sample Set Construction

The partitioning of the measurement space Ω_M into sets called control situations will be discussed in this section. First consider the partitioning of the state space Ω_x, assuming no environmental effects ($V=0$). This space is partitioned by the establishment of hyperspherical sets S_1' as the measurements are made. Let S_i' be the "set vector" locating the center of the ith set which has been established in the state space. Initially there are no sets. Let $S_1' = X_1$, the first measured state vector, then

$$X_2 \in S_1' \quad \text{if } \|X_2 - S_1'\| < D, \tag{3}$$

where

$$\|X - Y\| = \left[\sum_{i=1}^{n} (x_i - y_i)^2 \right]^{1/2},$$

and D is a prespecified distance. If

$$\|X_2 - S_1'\| \geq D, \quad \text{let} \quad S_2' = X_2. \tag{4}$$

This process is continued until the entire space in which actual state vector measurements occur, denoted by Ω_x', is covered with overlapping sets. (This process is similar to that described by Sebestyen in the recognition of speakers [9].) If a measured X_i falls within distance D of two or more set vectors it is considered a member of the closest set. This scheme does not waste memory by establishing sets in regions where measurements never occur. This is a significant advantage over a pregridded type partitioning especially if the space Ω_x is significantly greater than the space Ω_x'. A typical example of the set construction for a second-order system with two control choices is shown in Fig. 2. The set radius D was made large so that a relatively small number of sets would cover the space Ω_x'. Due to this large radius (and, therefore, large quantization), the resulting switching boundary between two control choices will be a crude approximation to the optimum one.

In order to improve the effective quantization in the measurement space, subsets are established in the sets that lie on the switching boundary. The subsets with radius $D' < D$ are established in the same way and with

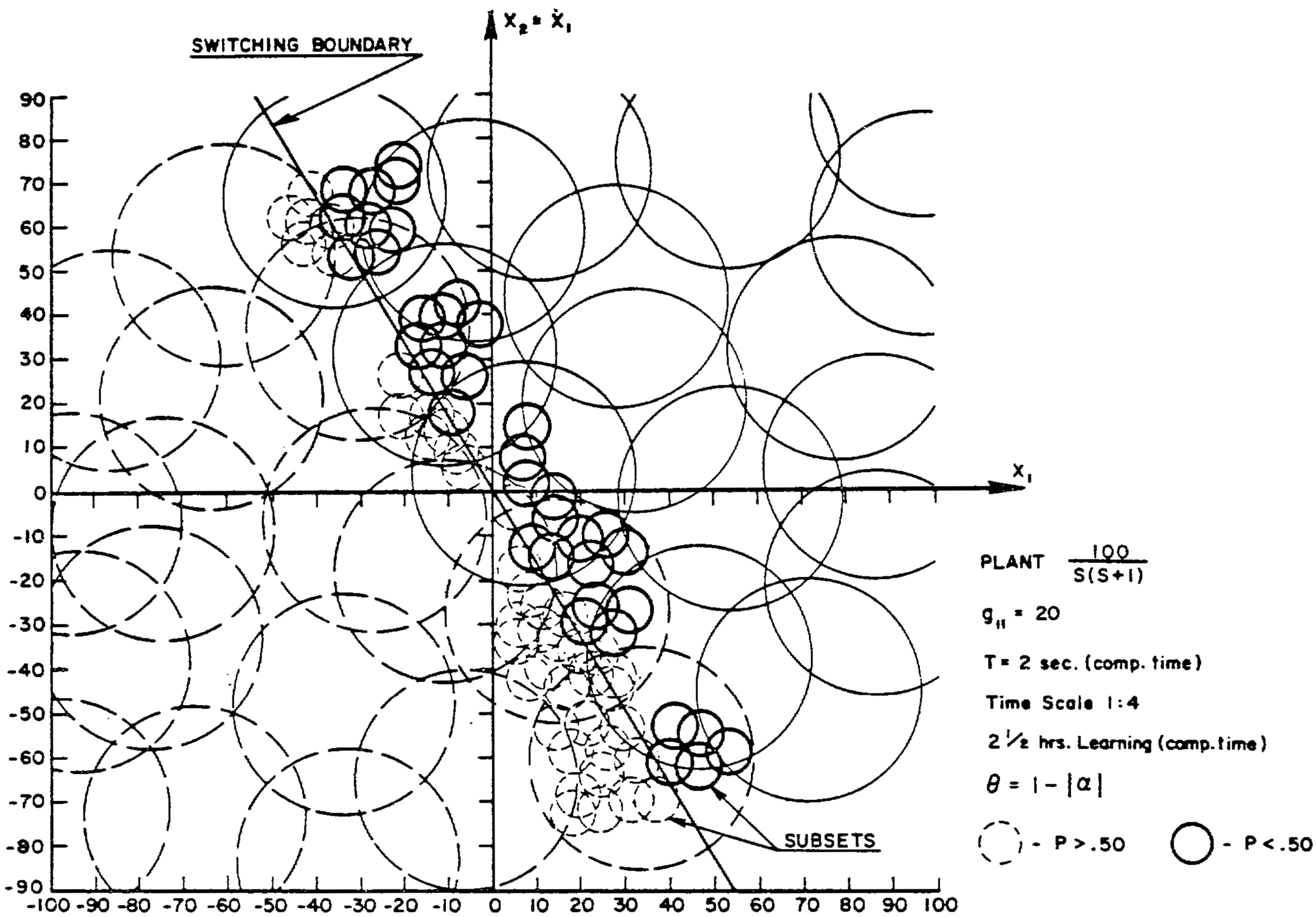

Fig. 2. Distribution of sets and subsets in the measurement space.

the same characteristics as the primary sets. This gives the system the advantage of fine quantization near the switching boundary where it is needed without the excessive memory requirements of fine quantization for the entire space. The range of each parameter $v_i \in V$ is quantized into a number of discrete levels q_i. The quantization levels need not be uniform. They may be either preassigned or self-adjusted so as to maintain a balance between the frequency of occurrence of measurements in each quantization level and the similarity of switching boundaries. Thus a specific measurement M_J is considered a member of the set S_I if $X_J \in S_I'$ and V_J fall in quantization levels $Q_I = \{q_1^I, \cdots, q_m^I\}$ where q_i^I are the specific quantization levels for the parameters v_i in set S_I. If the set S_i has no subsets, all the members of S_i constitute a control situation.

III. Reinforcement Scheme

Once the sample sets (control situations) are established it is necessary to determine which control choice from the class Ω_u is the best for each control situation. The system IP that is usually given is of integral form where the integration is over a number of sampling periods. In this case it is difficult to determine which of the series of control choices is most responsible for an improvement in performance. A similar problem was at least partially solved by Samuel [10] in his checker playing machine through the establishment of subgoals which are related to the main goal. A similar approach was used here.

The subgoal was to choose $u(rT) \in \Omega_u$ so as to maximize

$$\Delta(\overline{r+1}T) = \frac{IPS(rT) - IPS(\overline{r+1}T) - \lambda u^2(rT)}{\max \text{ of } [IPS(\overline{r+1}T), IPS(rT)] + \lambda (u_{\max})^2}$$
$$\lambda \geq 0. \quad (5)$$

$IPS(rT)$ is defined as

$$IPS(rT) = \boldsymbol{X}'(rT) G \boldsymbol{X}(rT) \quad (6)$$

where G is a positive definite diagonal matrix with elements $(g_{11}, \cdots, g_{nn})$. In (5), λ is a cost of control weighting term and $u_{\max}$ equals the maximum value of $|u_i|$.

Although the selection of the subgoal is rather heuristic it is felt the selected form is a reasonable choice. The system was able to select the best one from a set of possible subgoals through a trial process. Of the various subgoals that were tried, the one previously mentioned yields the best results. Consider the case where the cost of control term is zero (such as for a bang-bang system). In this case (5) reduces to

$$\Delta'(\overline{r+1}T) = \frac{IPS(rT) - IPS(\overline{r+1}T)}{\max \text{ of } [IPS(\overline{r+1}T), IPS(rT)]} \cdot \quad (7)$$

The maximization of (7) is equivalent to choosing $u \in \Omega_u$ at time rT so as to maximize the percent decrease in IPS in the time interval from rT to $\overline{r+1}T$. $IPS(rT)$ represents a weighted vector distance of the state of the system at time rT from the origin of the state space. The best value for G depends upon the plant as well as the system IP. One method to determine G is to use a "secondary learning loop" to learn the value of G that results in the control law which most nearly minimizes the system IP. This "secondary learning loop" would consist of a multidimensional search scheme. One of the elements of G may be arbitrarily selected as 1. First, an initial value for G is assigned, the corresponding control law is learned, and the system IP is evaluated. Next, G is incremented and a new control law is learned and evaluated by the system IP. This procedure is continued until the G corresponding to the control law giving the lowest system IP is found. Although this approach is time consuming, it does give satisfactory results [12], and also makes the stated goal of (2) more meaningful.

The problem now is reduced to that of matching to each control situation S_j the best control choice from the set Ω_u. This is not a straight forward deterministic problem since the control choice at time rT is dependent on the system state at time $\overline{r+1}T$. Let P_{ij} be the probability that the ith element u_i of the class Ω_u is the best control choice for members of the control situation S_j. Initially (assuming no a priori knowledge) all $P_{ij} = 1/k$, $i = 1, \cdots, k$; $j = 1, \cdots, p$. As the learning process proceeds, P_{ij} approaches 1 for one u_i and each S_j (with the possible exception of those sets located on the switching boundary). A control choice u_i is used for control situation S_j with probability P_{ij} unless some P_{ij}, exceeds a preset threshold T_P. In this case the u_i for which P_{ij} is maximum is used as the control choice for S_j.

This system uses what are called the linear reinforcement learning operators L_+ and L_- to adjust the P_{ij}'s in order to improve system performance. The positive reinforcement operator L_+ is used to increase the probability of a particular control choice u_i being used for a given control situation S_j.

$$P_{ij}(\overline{r+1}T) = L_+[P_{ij}(rT)] = \theta P_{ij}(rT) + (1 - \theta) \quad (8)$$
$$0 < \theta < 1.$$

The negative reinforcement operator L_- is used to proportionally decrease the probability of all other control choices u_q, $q = 1, \cdots, k$; $q \neq i$, being used for the given control situation S_j.

$$P_{ij}(\overline{r+1}T) = L_-[P_{ij}(rT)] = \theta P_{ij}(rT) \quad 0 < \theta < 1. \quad (9)$$

θ is the learning parameter. The larger θ is, the slower the probabilities P_{ij} change, which results in a slower learning rate. It is easily shown that repetitive application of the operators L_+ and L_- to P_{ij} causes P_{ij} to converge to 1 and 0, respectively [1], [12].

In order to determine which of the P_{ij}'s should be positively reinforced, the reinforcement must be related in some way to the subgoal. In this work a continuously updated weighted average value of the subgoal for each control situation and control choice was used as a criteria for reinforcement. Let Δ_{ij} represent the weighted average value of Δ for control situation S_j with control choice u_i. Then if $u(\overline{r-1}T)=u_I$ and $M(\overline{r-1}T)\in S_j$,

$$\Delta_{IJ}(rT) = \frac{[C_{IJ}(rT) - 1][\Delta_{IJ}(\overline{r-1}T)] + \Delta(rT)}{C_{IJ}(rT)}, \quad C_{IJ} > 0 \tag{10}$$

where

$$C_{IJ}(rT) = C_{IJ}(\overline{r-1}T) + 1 \text{ if } C_{IJ}(\overline{r-1}T) < C_m \tag{11}$$

and

$$C_{IJ}(rT) = C_m \text{ if } C_{IJ}(\overline{r-1}T) = C_m. \tag{12}$$

Thus

$$C_{IJ} = 1, 2, \cdots, C_m.$$

All other $\Delta_{ij}(rT)$ and $C_{ij}(rT)$ remain the same as for time $(r-1)T$. Thus Δ_{ij} is the actual average value of C_{ij} samples of Δ for control choice u_i in control situation S_j as long as $C_{ij}<C_m$. Once $C_{ij}=C_m$ the additional values of Δ are weighted, in determining Δ_{ij}, as if they were the C_m'th value.

The value of C_m must increase as the quantization size is increased and as the sum $(n+m)$ is increased where $(n+m)$ is the dimension of the measurement vector $\boldsymbol{M}$. Also, larger values of C_m must be used in the case where measurement noise is included in the measurement vector $\boldsymbol{M}$. In the case where the measurement noise is additive with zero mean the averaging process of (10) acts as a noise filter.

The u_i corresponding to the largest $\Delta_{iJ}(rT)$, $i=1, \cdots, k$, is apparently the control choice most nearly satisfying the subgoal for control situation S_J at time rT. Let the maximum $\Delta_{iJ}(rT)$ for set S_J at time rT be represented by $\Delta_{IJ}(rT)$. Since, according to the subgoal, u_I is the best control choice for control situation S_J at time rT, the probability of choosing u_I should be positively reinforced by applying operator L_+ to P_{IJ}. The learning parameter θ should depend upon the "certainty" that u_I is the best control choice for the control situation S_J. One measure of this "certainty" is the term α where

$$\begin{aligned} &\alpha = b \min [\Delta_{IJ} - \Delta_{iJ}]. \\ &i = 1, \cdots, k \qquad 0 < b \leq 1/2 \\ &i \neq I \qquad\qquad\; 0 < \alpha < 1. \end{aligned} \tag{13}$$

A large α would indicate that the control choice u_I has produced a much greater value for Δ than any other u_i for set S_J. Thus, one would be fairly sure that u_I was the proper control choice. Since fast learning is desired when α is large, let

$$\theta = 1 - (\alpha)^\gamma. \tag{14}$$

The "b" that is included in (13) determines the maximum possible reinforcement and forces α to lie within the range (0, 1). The exponent γ in (14) is used to either compress or expand the range of the reinforcement parameter θ. The best results were obtained using $\gamma=0.5$. If too large a value is used for "b" it is possible for the system to "jump to a false conclusion" in the sense that it learns a control choice that does not satisfy the subgoal. This occurs because of a learning rate that is too high. In general, when the parameter C_m must be large, "b" must be small since the choice of "b" is related to the reliability of the early Δ_{ij}'s. Thus "b" must decrease as the dimensionality of the measurement vector $\boldsymbol{M}$ increases and as the measurement noise increases. On the other hand, excessively small values of "b" result in unnecessarily low learning rates. Note that a significant disturbance $\boldsymbol{N}$ would introduce errors in the Δ_{ij}'s and therefore possibly cause erroneous reinforcement. To avoid this difficulty, when the controller receives an N' signal it merely holds all Δ_{ij}'s constant and prevents reinforcement during the next sampling interval.

The first scheme that was used to determine which primary sets should be further partitioned to provide finer quantization is the following. If, after a fixed number of samples C within a primary set S_J, P_{iJ} lies between two thresholds $T_2<P_{iJ}<T_1$ (typical values might be $T_1=0.90$ and $T_2=0.1$) subsets are established in S_J. Reasonable performance can be obtained for most stationary systems by proper adjustment of C, T_1, and T_2. A second scheme which can be used for stationary and nonstationary systems, uses the curvature of the approximate switching boundary to determine where subsets should be established. The chain encoding scheme described by Freeman [11] is used to determine the curvature of the switching boundary. Regions of the switching boundary with relatively high curvature in one direction are identified and the sets that are located on the inside of the curvature are further divided into subsets.

The system is capable of self-adjusting the environmental quantization levels through consideration of two criteria. First, the system tries to adjust the quantization levels so as to keep the resulting switching boundaries equally dissimilar, based on the measure of similarity suggested in the following. Second, the system tries to adjust the quantization levels so as to keep approximately the same frequency of occurrence of measurements in the various quantization intervals. Since in the general case these two criteria cannot be satisfied simultaneously with any given quantization distribution, a relative importance or weighting must be assigned to each criterion.

The difference in the directionality spectrum discussed by Freeman was used as a measure of the similarity between switching boundaries for different environmental quantization levels. The frequency of occurrence of measurements in a given quantization interval is easily found by counting the measurements falling in each interval during a fixed time interval. In actual operation, initial quantization levels are picked. The system then periodically adjusts the quantization levels in the direction indicated by the evaluation of the similarity and frequency criteria [12].

IV. Analytical Results

A number of analytical results have been obtained for this learning system with a linear plant. These are presented here with some of the experimental results for comparison purposes. The experimental procedure is discussed in Section V.

First consider the theoretical switching boundary obtained as a result of the maximum of (7) for a linear plant with the two control choices of $+1$ and -1. Since $IPS(rT)$ is independent of $u(rT)$, the $u(rT)$ which maximizes (7) is the same as the $u(rT)$ which minimizes $IPS(\overline{r+1T})$. The switching boundary is defined as the collection of all points in the state space for which positive forcing $[u(rT)=+1]$ gives the same value of $IPS(\overline{r+1T})$ as negative forcing does. Therefore, the problem reduces to finding all $\boldsymbol{X}(rT)$ for which

$$\boldsymbol{X}_p'(\overline{r+1T})G\boldsymbol{X}_p(\overline{r+1T}) = \boldsymbol{X}_m'(\overline{r+1T})G\boldsymbol{X}_m(\overline{r+1T}). \quad (15)$$

$\boldsymbol{X}_p(\overline{r+1T})$ and $\boldsymbol{X}_m(\overline{r+1T})$ equal the value of the vector $\boldsymbol{X}$ at time $\overline{r+1T}$ with positive and negative forcing, respectively, during the interval rT to $\overline{r+1T}$. Equation (15) can be written in the following form:

$$\boldsymbol{W}_p'(\overline{r+1T})H\boldsymbol{W}_p(\overline{r+1T}) = \boldsymbol{W}_m'(\overline{r+1T})H\boldsymbol{W}_m(\overline{r+1T}) \quad (16)$$

where

$$\boldsymbol{W}_p(\overline{r+1T}) = \begin{bmatrix} +1 \\ \boldsymbol{X}(\overline{r+1T}) \end{bmatrix},$$

$$\boldsymbol{W}_m(\overline{r+1T}) = \begin{bmatrix} -1 \\ \boldsymbol{X}(\overline{r+1T}) \end{bmatrix} \quad (17)$$

and

$$H = \begin{bmatrix} 0 & 0 \cdots 0 \\ 0 & \\ \cdot & G \\ \cdot & \\ \cdot & \\ 0 & \end{bmatrix}. \quad (18)$$

Now

$$\boldsymbol{W}_p(\overline{r+1T}) = \phi(T)\boldsymbol{W}_p(rT) \quad (19)$$

where $\phi(T)$ in the transition matrix for the plant. Substituting (19) into (16) gives

$$\boldsymbol{W}_p'(rT)\phi'(T)H\phi(T)\boldsymbol{W}_p(rT) = \boldsymbol{W}_m'(rT)\phi'(T)H\phi(T)\boldsymbol{W}_m(rT). \quad (20)$$

The matrix $Y(T)=\phi'(T)H\phi(T)$ is independent of $\boldsymbol{X}(rT)$. Therefore, (20) can be written as

$$\boldsymbol{W}_p'(rT)Y(T)\boldsymbol{W}_p(rT) = \boldsymbol{W}_m'(rT)Y(T)\boldsymbol{W}_m(rT). \quad (21)$$

Performing the matrix manipulation with a general matrix Y and simplifying the result gives:

$$x_1(y_{21}+y_{12}) + x_2(y_{31}+y_{13}) + \cdots x_n(y_{n+1,1}+y_{1,n+1}) = 0. \quad (22)$$

Equation (22) which represents the switching surface for a general n'th order linear plant is the equation of a hyperplane in space Ω_x passing through the origin. The foregoing derivation proves that the switching boundary is linear for a linear plant regardless of the choice of G. Although this certainly does not represent the ideal situation, this research was more concerned with the learning process itself than in determining the best subgoal. It may be that several subgoals should be used in the learning process.

In the case where G is a diagonal matrix and Y is symmetrical, the switching boundary for a second-order linear system is described by the equation

$$x_2 = -\frac{y_{12}}{y_{13}}x_1 = y_m x_1. \quad (23)$$

Consider a plant described by the differential equation

$$\frac{d^2x}{dt^2} + a\frac{dx}{dt} = (Ka)u. \quad (24)$$

Performing the matrix operation $\phi'(T)H\phi(T)=Y(T)$ and applying (23) gives a value for the slope

$$y_m = \frac{-a(\epsilon^{-aT}-1+aT)g_{11}}{[(\epsilon^{-aT}-1+aT)(1-\epsilon^{-aT})]g_{11} + [a^2\epsilon^{-aT}(1-\epsilon^{-aT})]g_{22}}. \quad (25)$$

Note that the resulting switching boundary is independent of the system gain K. It is also noted that

$$\lim_{g_{11}\to\infty} y_m = \frac{-a}{(1-\epsilon^{-aT})}, \quad \text{and} \quad \lim_{g_{11}\to 0} y_m = 0. \quad (26a)\ (26b)$$

(a) (b)

For the system in which $a=1$ and $T=1/2$, (26a) and (25) reduce to

$$\lim_{g_{11}\to\infty} y_m = -2.54 \text{ and } \begin{cases} y_m = -1.98 \text{ for } g_{11}=20 \\ y_m = -1.61 \text{ for } g_{11}=10. \end{cases}$$

This theoretical limit on the magnitude of the slope is indicated in the plot of Fig. 3 by the curve for $g_{11}=\infty$. The calculated switching boundaries with the foregoing slopes are plotted on the same axes as the learned boundaries in Figs. 4 and 5 so that they might easily be compared. Note that in both cases the learned curve leads the computed curve, that is, a plant trajectory in the state space will intersect the learned switching boundary (LSB) before it intersects the computed switching boundary (CSB). This is a result of the finite decision time (t_d') required by a controller operating in real time.[2] The controller cannot change the control choice until t_d' seconds after a measurement vector is obtained. The learning system automatically compensates for this effect by rotating the switching boundary. The effect of this decision time t_d' can be removed by calculating the plant state t_d' seconds after it hits the switching boundary. The equation of this new boundary is found by using (19) with $T=t_d$. Initial conditions on the learned switching boundary are considered by requiring that $x_2(rt_d)=y_L x_1(rt_d)$ where y_L is the slope of the learned switching boundary. For the case where $y_L=1.26$, $g_{11}=10$ and $t_d=0.075$ seconds the modified or effective, learned, switching boundary (ELSB) is described by the equation

$$x = -1.28x_1 + 7.56$$

which is plotted in Fig. 4. For the case where $g_{11}=20$, $y_L=-1.74$, and $t_d=0.07$ seconds, the effective learned switching boundary (ELSB) is described by the equation

$$x_2 = -1.83x_1 + 7.2$$

which is plotted in Fig. 5. These two figures show that there is a very close agreement between the ELSB and the CSB. This is sufficient evidence that the learning process is functioning properly in this application!

Since the system is a sampled system with a significant sampling period T, it does not actually reverse forcing when a trajectory hits the switching boundary. The actual reversal in u occurs t_d seconds after the first sampling period after the system crosses the LSB. In the limit, this could be the distance traveled along a trajectory in $T+t_d$ seconds. The equation of this upper limit in the learned switching boundary (ULLS) which is obtained using the same techniques as for obtaining ELSB was found to be

$$x_2 = -4.025x_1 + 98, \tag{27}$$

and is plotted in Fig. 5. The actual switching from $+1$ to -1 can occur when the plant state is anywhere in the region bounded by ELSB, ULLS, and a system trajectory with $u=+1$ running from the origin to ULLS. It is interesting to note that the minimum time switching boundary (MTSB) for the continuous system [13] falls

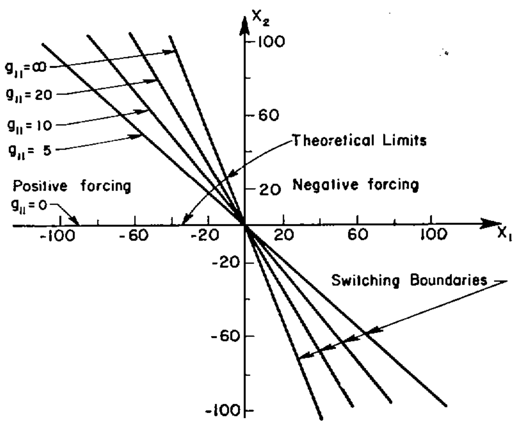

Fig. 3. Effect of change in g_{11} on switching boundaries.

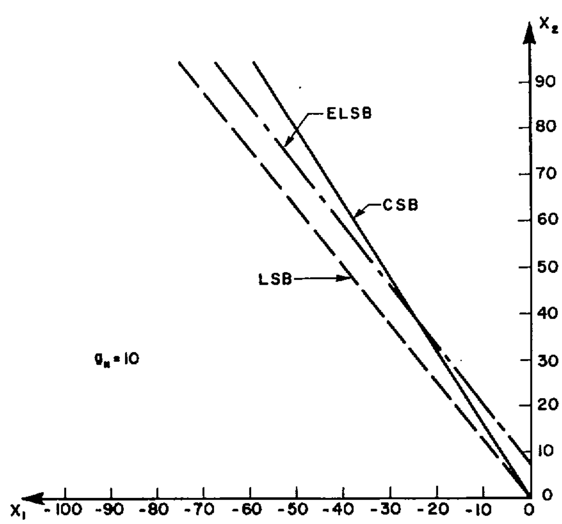

Fig. 4. Comparison of computed and learned switching boundaries for $g_{11}=10$.

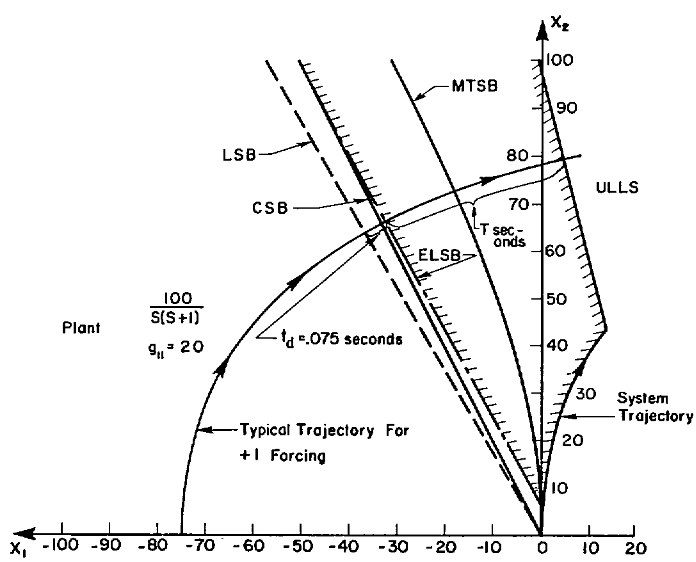

Fig. 5. Comparison of experimental and theoretical results.

[2] The decision time t_d' is the time required to obtain the measurement vector $\boldsymbol{M}$, locate the control situation S_J containing $\boldsymbol{M}$, and decide upon a control choice u_I.

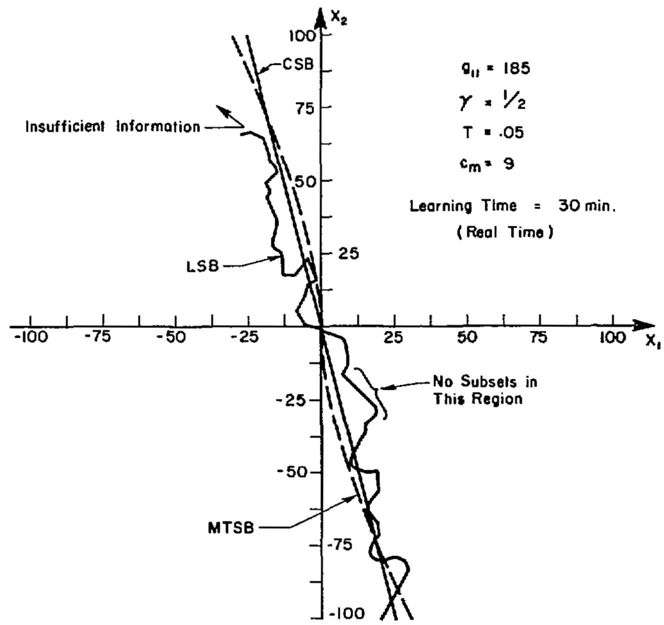

Fig. 6. Learned switching boundary with $T=0.5$ seconds.

within this region except for a very small segment near the origin. (See Fig. 5.) A run was made on an IBM 7090 with $C_m=9$, $a=1$, $K=100$, $b=1/2$, $\gamma=1/2$, and $g_{11}=185$. The learned switching boundary (LSB) is shown in Fig. 6 along with the MTSB for comparison purposes.

V. Discussion of Computer Simulation and Results

Several example systems have been simulated on hybrid computing equipment. The plant was simulated on a GEDA analog computer and the controller was simulated on the IBM 1620 digital computer which was part of the IBM 1710 control system. In addition to the 1620 computer, the IBM 1710 system also includes analog to digital and digital to analog conversion equipment. The plant simulated on the analog computer was actually controlled by the digital computer. (A simplified flow diagram for the controller is shown in Fig. 7.) The results obtained from several types of plants are presented in the following. A more detailed discussion of results can be found in Waltz and Fu [8] and Waltz [12]. In all of the following examples the state variables were limited to the range $(-100, 100)$. The control choice u, was constrained to be either $+1$ or -1 $(k=2)$. It was necessary to time-scale the problem by a factor of 4 to 1 due to the speed of the digital equipment. A sampling period of 2 seconds in computer time or 0.5 seconds in real time was used in all but Example 3, in which $T=0.75$ seconds.

Example 1: In Section IV some results for the plant described by the transfer function

$$G(s) = \frac{100}{s(s+1)}$$

were discussed. The switching boundary and typical "learning curve" for the case where $V=0$, $b=1/2$, $\gamma=1$ and $C_m=9$, are shown in Figs. 2 and 8. These curves were obtained by applying a specified initial condition to the plant every three minutes for evaluating the system *IP* and arbitrary initial conditions at all other times.

Example 2: The plant is described by the differential equation

$$\frac{d^2y}{dt^2} + 2\zeta\frac{dy}{dt} + u = Ku, \qquad x_1 = y, \qquad x_2 = \frac{dy}{dt}.$$

The gain constant $K=100$, $b=1/2$, $\gamma=1/2$, and $C_m=2$. The influence of the environment on the plant was simulated by the damping constant ζ which could be either 0.1, 0.5, or 1.0 $(V=v_1)$. ζ was held constant over three minute intervals. After each three minute interval, one of the three possible values of ζ was picked at random for the next three minute interval. The system *IP* was of the form

$$\sum_{r=1}^{n} r\,|x_1(rT)|.$$

A "learning curve" obtained in the same way as for Example 1 for this 3-environment situation is shown in Fig. 9. Portions of curves where the system is learning on a given environment are represented by solid lines which are connected together by horizontal dashed lines to form a continuous curve for each environmental situation. Results have also been obtained for the case where ζ varies continuously over the foregoing range [8].

Example 3: The plant is described by the following third-order differential equation:

$$\frac{d^3x}{dt^3} + 1.9\frac{d^2x}{dt^2} + 1.7\frac{dx}{dt} + 0.5x = 100u(t).$$

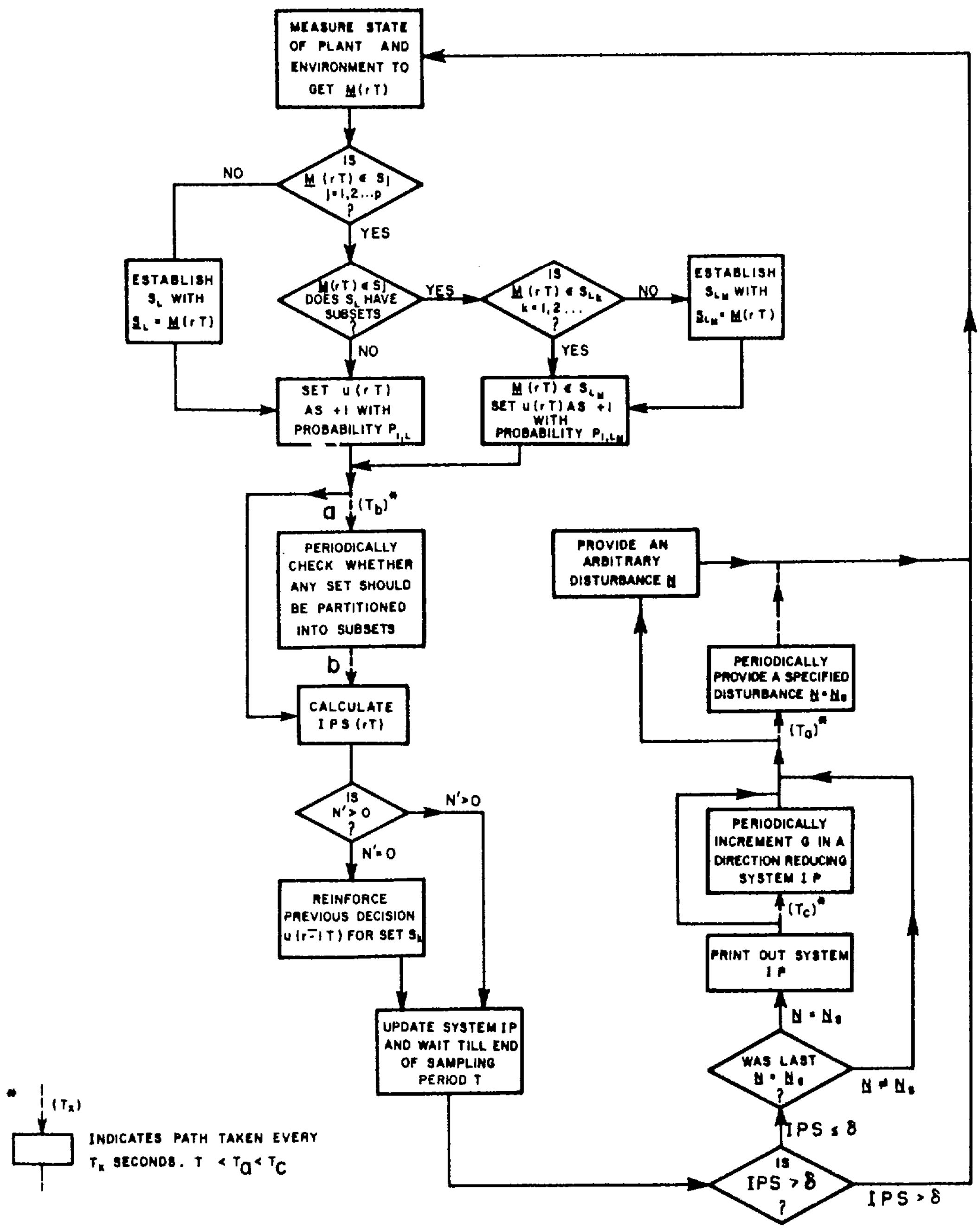

Fig. 7. A simplified flow diagram of a learning controller.

The three state variables $x_1 = x$, $x_2 = \dot{x}_1$, $x_3 = \dot{x}_2$ were monitored. b, γ, and C_m were 0.1, 0.5 and 4, respectively. In order to show the improved performance with learning for this system, four different test signals were applied to the system every 12 minutes and the system

$$IP = \sum_{r=1}^{15} r \left| x_1(rT) \right|$$

was evaluated for each test signal. The sum of the four IP's at every 12 minutes is plotted in Fig. 10. During the three hour period recorded here the total IP improved by a factor of 10.

VI. Conclusions

The previously described method appears to be one practical approach to the problem of learning control systems. One difficulty with the method is that it will require a great deal of computer memory for high-order plants. This, however, will probably be a problem with all learning systems. A second problem is that of finding

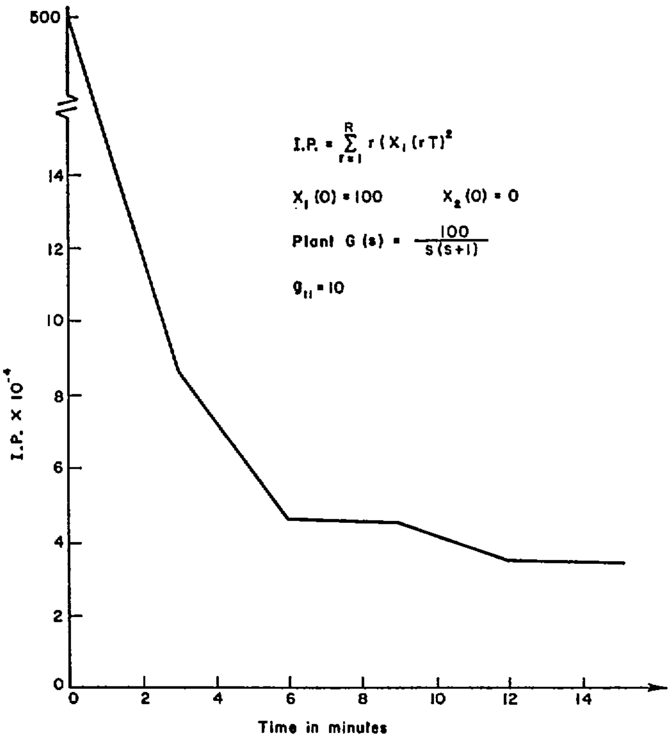

Fig. 8. Learning curves for plant in a fixed environment.

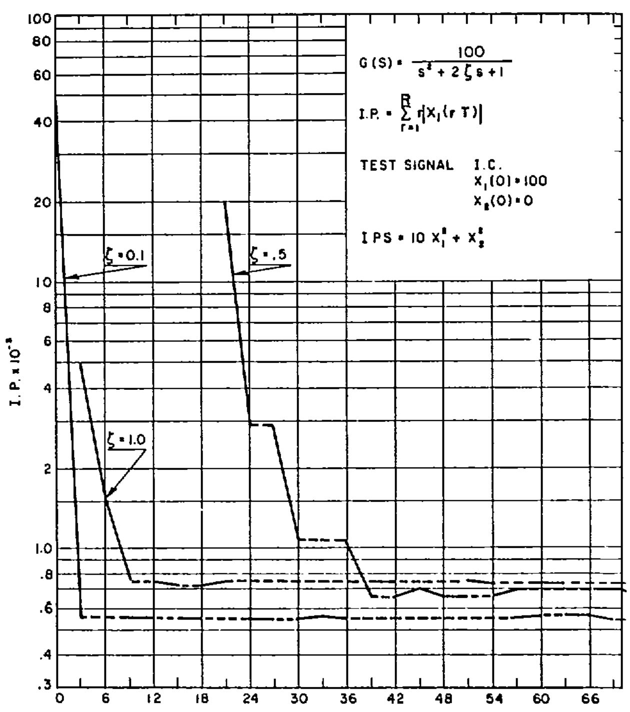

Fig. 9. Learning curve for three environment situation.

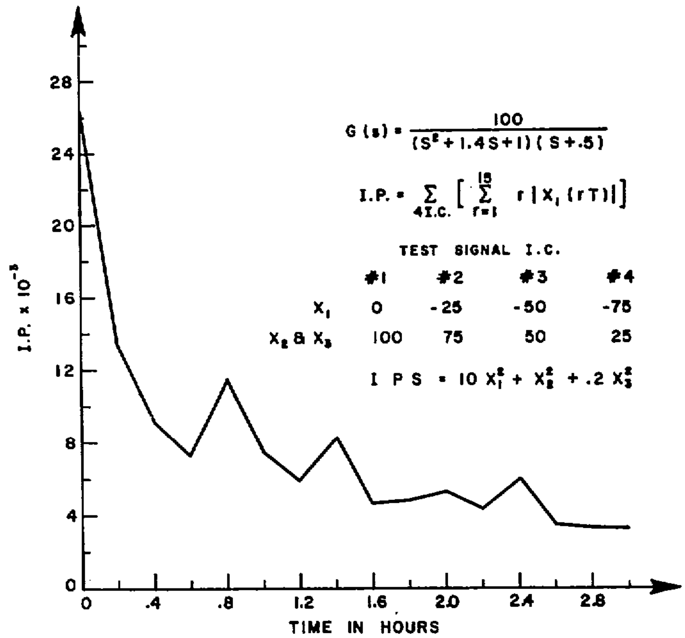

Fig. 10. Learning curve for third-order system.

a satisfactory subgoal for a given system and *IP*. This is particularly difficult when the plant is varying with the environment or with time.

The method is most useful for cases where a relatively small number of environmental parameters affect a large number of plant parameters. The advantages of the method are:

1) It can be extended to higher-order systems.
2) It can be used with plants that vary with the external environment providing the environmental parameter can be measured.
3) Very little a priori information need be known about the plant.
4) Practical systems can be controlled with available computer equipment.
5) The switching surface need not be convex, concave, or simply connected.

References

[1] K. S. Fu, "Learning control systems," *Proceedings COINS Symposium*. Washington, D. C.: Spartan, 1964.
[2] A. M. Andrew, "Learning in control systems," *Control*, September 1960.
[3] M. D. Mesarovic, "Self-organizing control systems," *1962 Proc. Symp. on Discrete Adaptive Processes*.
[4] B. Widrow, "Pattern recognition and adaptive control," *1962 Proc. Symp. on Discrete Adaptive Processes*.
[5] G. K. Krug and E. K. Letskii, "A learning automation of the tubular type," *Automation and Remote Control*, vol. 22, October, 1962.
[6] M. A. Aizerman, "Automatic control learning systems (in the light of experiments on teaching the systems pattern recognition)," *1963 Proc. of the Internat'l Federation of Automatic Control Congress*.
[7] J. E. Gibson, K. S. Fu, J. D. Hill, J. A. Luisi, R. H. Raible, and M. D. Walts, "Philosophy and state of the art of learning control systems," School of Electrical Engineering, Purdue University, Lafayette, Ind., Tech. Rept. TR-EE 63-7, November 1963.
[8] M. D. Waltz and K. S. Fu, "A computer-simulated learning control system," *1964 IEEE Internat'l Conv. Rec.*, Pt. 1, pp. 190–201.
[9] G. S. Sebestyen, "Pattern recognition by an adaptive process of sample set construction," *IRE Trans. on Information Theory*, vol. IT-8, pp. S82–S91, September 1962.
[10] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. and Develop.*, July 1959.
[11] H. Freeman, "On the digital computer classification of geometric line patterns," *1962 Proc. NEC*, vol. 18.
[12] M. D. Waltz, "A study of learning control systems using a reinforcement technique," Ph.D. dissertation, School of Electrical Engineering, Purdue University, Lafayette, Ind., August 1964.
[13] J. F. Coales and A. R. M. Norton, "An ON-OFF servo mechanism with predicted change-over," *Proc. IEE (London)*, Pt. 3, vol. 103, p. 449, July 1956.