IFAC

INTERNATIONAL FEDERATION
OF AUTOMATIC CONTROL

WARSZAWA 1969

Learning Systems and Pattern Recognition

Fourth Congress of the International Federation of Automatic Control
Warszawa 16–21 June 1969

TECHNICAL SESSION
6

Organized by
Naczelna Organizacja Techniczna w Polsce

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

Learning Systems and Pattern Recognition

TECHNICAL SESSION No 6

FOURTH CONGRESS OF THE INTERNATIONAL
FEDERATION OF AUTOMATIC CONTROL
WARSZAWA 16 – 21 JUNE 1969

Organized by
Naczelna Organizacja Techniczna w Polsce

Constents

Paper No			Page
6.1	USA	- L.E.Jones, K.S.Fu - On the Selection of Subgoal and the Use of Apriori Information in Learning Control Systems.....	3
6.2	SU	- V.A. Jacubovich - On Adaptive /Self-Learning / Systems of Some Class.....	35
6.3	CS	- S.Petráš - On the Algorithm of Learning with Accumulation of Experience in Optimum Control.....	47
6.4	USA	- H.H.Yeh, J.T.Tou - On the Ergodicity and Dynamic Behavior of Finite-State Markov Chains.....	69
6.5	JA	- B.Kondo, S.Eiho - Statistical Min-Max Decision Methods and Their Application to Learning Control	87

Wydawnictwa Czasopism Technicznych NOT
Warszawa, ul. Czackiego 3/5 — Polska

Zakład Poligraficzny WCT NOT. Zam. 33/69.

ON THE SELECTION OF SUBGOAL AND THE USE OF APRIORI INFORMATION IN LEARNING CONTROL SYSTEMS*

L. E. Jones, III
Graduate Engineering
Education System, GENESYS
University of Florida
Port Canaveral, Florida

K. S. Fu
School of Electrical Engineering
Purdue University
Lafayette, Indiana

ABSTRACT

Numerous methods have been proposed for the design of control systems which learn to function in unknown or partially known environments. Most learning schemes are radical departures from the techniques using continuous adjustment of parameters which grew out of early developments in model reference systems. Principal contributions to the area have been controller models and algorithms. In studying these models, the system is abstracted to such an extent that there is quite often a loss of contact with practical considerations. The objective of this paper is to present some results in the theory of learning control, but also to look again at some of the practical problems encountered in applying a learning controller to a problem.

This paper defines the subgoal as a subordinate to the primary goal of minimizing the performance index. It must evaluate each decision one control interval after it is instituted. The subgoal problem is to choose a subgoal which will direct the learning process to the optimal as prescribed by the given performance index. An analytical solution is presented and extended heuristically for the general case. This extended method makes use of the apriori information about the plant.

Two other problems are also discussed. A fixed grid is used to partition the state space into control situations, and a method of extending the grid is proposed and evaluated. The controller is initialized using the apriori information, too. A full scale simulation confirms that the proposed methods of choosing the subgoal, extending the fixed grid and initializing the controller are improvements over previous methods.

* This work was supported in part by National Science Foundation, Grant GK-1970

I. INTRODUCTION

In the current decade, there has been a surge of interest in designing systems which exhibit learning behavior and research has progressed rapidly on probabilistic models and learning algorithms. The control problem has been abstracted to allow one to isolate the decision problem and to study the convergence properties of learning or reinforcement algorithms.

This paper is intended to bridge the gap in the design problem. On one extreme is the system proposed by Fu and Waltz[1] which assumes only the order of the plant is known, a more or less black-box approach. On the other extreme is the case where the plant equations are known and the designer solves an optimal control problem. The problem, posed as a question, is: How is the theory used to design and mechanize a learning control system? Several facets of the design are considered in the ensuing sections following some additional background.

On-line learning occurs with the controller embedded in a closed loop control system.[2] A learning controller collects some pertinent information during its operation about the random variables or functional which describe the controlled process or plant-environment relation, and processes it according to an algorithm to optimize a pre-specified performance index (PI).[3] Many of the pioneering contributions to the area of learning control originated from the approach of considering a learning system as an adaptive system with additional memory.[2,4] More recently, contributions to the area of learning control have sprung from stochastic approximation[6,9] and automata theory.[7,8,10,12]

II. THE CONTROL PROBLEM

The general control problem is a classical optimal control problem. That is, it is desired to design a controller for a plant described by an ordinary differential equation as Equation (1) to minimize a performance index specified by Equation (2).

$$\dot{\underline{x}} = \underline{f}(t,\underline{x},\underline{u}) \qquad \underline{x}(0) = \underline{x}_o \tag{1}$$

$$PI(u,x_o) = \int_0^T F(t,\underline{x},\underline{u})dt \tag{2}$$

In general, the state $\underline{x}$ is an n-vector and the control input $\underline{u}$ is an m-vector.

The primary goal is to design a controller which minimizes a given PI. The learning control designer, in general, does not have complete knowledge of $\underline{f}$. Instead, he must measure the PI as the system operates and use these measurements and his incomplete or inaccurate mathematical model to guide

after-the-fact decisions.

The physical constraint on the control input is that it is bounded as shown in Equation (3).

$$|u_i(n)| \leq U_{M_i} \qquad i = 1,\ldots,m; \quad n = 0,1,\ldots \tag{3}$$

In order to develop some of the analytical results in Sections IV and V, the constraint will be relaxed, but it is not ignored. In fact, as evidenced by the following presentation, the type of constraints form an integral part of the investigation.

There are two often cited classes of control constraints which lead to different implementation and application results, but which appear the same to the decision making element of the control system.

(1) Parameter Choice - Partition the i-th of M parameters in a specific form of controller into K_i levels. Learn the best values from the set of K_p allowable decisions, where

$$K_p = \underset{i=1}{\overset{M}{\times}} K_i \tag{4}$$

(2) Control Action Choice - Partition the closed interval $[-U_M, +U_M]$ into K levels. Learn the best u_i for each state $\underline{x}$ from the set of K allowable control actions.

One example of option (1) is: learn the best set of gain values in a control law constrained to be of the form

$$u(n) = \underline{k}'\underline{x}(n) \tag{5}$$

Option (2) is an attempt to learn $u^*(\underline{x})$, itself, subject to quantization of both state and control. The proposed system uses this option.

The following are steps for the design and mechanization of the learning controller:

1. Sample time to allow time for making and reinforcing control decisions.
2. Quantize the control input into a finite collection of allowable control actions.
3. Partition the state space into a finite collection of regions called control situations.
4. Choose a reinforcement algorithm and a subgoal to direct the learning process.

The reinforcement learning control system[1,14] is realized by these steps.

The primary control problem is to design a controller for the plant in Equation (6) which satisfies the primary goal of minimizing the performance index in Equation (7).

$$\underline{x}(n+1) = \emptyset\underline{x}(n) + \underline{h}u(n) \quad n = 0,1,\ldots \quad \underline{x}(0) = \underline{x}_o \tag{6}$$

$$PI(u,\underline{x}_o) = \sum_{n=1}^{N} [\underline{x}'(n)\, Q\, \underline{x}(n) + \alpha u^2(n-1)] \tag{7}$$

Matrix Q is at least positive semidefinite and $\alpha \geq 0$. Plant coefficients $\emptyset$ and $\underline{h}$ are, in general, unknown or partially known and might depend upon the operating conditions of the plant. Sampling period τ is fixed and problem time $T = N\tau$ is fixed or infinite. Initial state $\underline{x}_o$ is considered fixed for the purposes of solving the optimal control problem, but during normal operation of the plant, it can assume any value in a compact subset of the state space. Control input u is to be chosen from the finite set U of control actions, formed as indicated in Step 2.

$$u(n) \in U = \{u_1,\ldots,u_K\} \quad n = 0,1,\ldots,N-1 \tag{8}$$

This is not a completely general problem, but the results indicate that it is of general interest in demonstrating the design techniques.

III. THE LEARNING CONTROL SYSTEM

The learning control system belongs to the general class of systems shown in Figure 1, in which the decision making element of the controller is a variable structure, finite, stochastic automaton A. All other system components are combined into E, the stochastic environment of A. E contains the plant and its environment, the control input mechanization and the performance evaluator. This model is well suited to an investigation of the convergence properties of reinforcement or learning algorithms. Some researchers[10,11] have used the model for examining the convergence and expediency of automata, and some[7,8,12] have already applied it to the adaptive and learning control system problems.

This general model is structured to a particular application to control problems by defining the three pertinent terms:

(i) Control Decision - made in A, sent to E

(ii) Control Decision Evaluation - made in E, sent to A

(iii) Control Interval - time for E to evaluate a decision

It is usually assumed that control decisions require negligible time. This time is small compared to the control interval, but it is not exactly zero. However, this is a discrepancy that the learning system can automatically compensate for, provided it doesn't become excessive.[1]

Figure 2 is the schematic diagram of the proposed learning control system. The plant is assumed to obey physical laws which lead to a mathematical model as Equation (1) which is then sampled to yield Equation (6). In a classical sampled data control system, the sampling period is an im-

portant control parameter. Here it is even more important because, as is made clear in Section IV, the sampling period is also the control interval. Several authors[15,18] have demonstrated that there is an optimal sampling rate for obtaining data to use in digital identification techniques. Though the present application does not perform an explicit identification, the controller inherently identifies as it learns to make the best decisions. Based on this, it is reasonable to expect that there is some optimal sampling rate, which is not zero. However, since there is no unique way to choose the optimal τ, one was selected by trial and error for the experimental work in Section VI.

A control situation is a collection of states for which the same control decision is optimal.† These states can be generalized to include measurable but uncontrollable inputs as well as measurable state variables. It is emphasized that the purpose of partitioning the state space into control situations is to make successive trials as nearly alike as possible.[13] Viewed in a general sense, the system is accumulating experience from a succession of trials which are effected by an uncontrollable parameter $\underline{x}_o$. It performs best when a control decision is compared only to other decisions made in like circumstances. The fineness of the grid determines the amount that the $\underline{x}_o$ effect is reduced.

Two factors influenced the selection of a fixed grid for the partition of state space: simplicity and speed. Figure 3 illustrates the technique for a two dimensional case. The grid partitions the finite region bounded by $|x_1| \leq 50$, $|x_2| \leq 50$ into 200 square sets. Symmetry allows quadrants 3 and 4 to be folded onto 1 and 2, respectively. In higher dimensions, the squares would be hypercubes. A state is classified by multiplying its elements by appropriate scale factors and truncating to integer values. Section V considers the classification of states located outside the grid.

Since the system can learn only by trying, the learning time depends on the number of possible trials $K_p L_p$ or KL and the trial time T or τ. Option (1) might use T or τ, option (2) uses τ, so the time to perform one trial of each decision is $L_p K_p T$, $L_p K_p \tau$ or $LK\tau$ seconds. Learning time will be multiples of this minimum. Based on experiments reported in Section VI, a typical learning pattern is that the worst decisions are ruled out with only one trial and the two or three better ones are tried several times. A representative estimate is that within each control situation it would

---

†The statement is idealized; in reality there is an inherent averaging over the states in a given situation.

take $K + 3x3 = K + 9$ trials to select the best. This corresponds to a minimum learning time of $L(K + 9)\tau$ seconds.

A stochastic automaton is used as a model for the learning controller. The primary goal of the controller is to learn to make control decisions which cause the PI to be minimum. It is conceivable that the PI could be used to evaluate decisions and to direct the learning process. However, it is not a suitable evaluator for the system presented here. The controller chooses one of K admissible control actions to act for one control interval, τ. Therefore, it is necessary to have a per-interval (per-decision) evaluator or a subgoal to guide the reinforcement. A detailed description of the operation of the controller and the reinforcement algorithm, which is similar to that proposed in Reference 1, is given in Appendix A.

IV. THE SUBGOAL PROBLEM

The subgoal problem for the proposed reinforcement learning control systems is formulated as follows.[14] The plant is assumed to be described by a vector difference equation.

$$\underline{x}(n+1) = \underline{f}[\underline{x}(n),u(n),n] \qquad n = 0,\ldots,N-1 \qquad \underline{x}(0) = \underline{x}_o \tag{9}$$

The state $\underline{x}(n) = \underline{x}(n\tau)$ is an n-vector, $u(n) = u(n\tau)$ is a scalar control input, $n = n\tau$ is time, τ is the sampling period, and $\underline{f}$ is an n-vector function of $\underline{x}(n)$, $u(n)$ and n. The primary goal is to minimize a performance index of the form

$$PI(u,\underline{x}_o) = \sum_{n=1}^{N} F[\underline{x}(n),u(n-1),n] \tag{10}$$

where F is a scalar function of its arguments. The solution of this optimal control problem is subject to the constraint that the control must be chosen from a finite set of admissible actions as in Equation (8).

The primary problem has its primary goal of minimizing the PI, but the term subgoal is used to refer to both the sub-problem and its criterion. Mechanization of the system requires that the subgoal (the criterion) have these characteristics:

(i) It must evaluate each decision separately.

(ii) It must be related to the PI so that minimizing the subgoal with each decision also minimizes PI.

The problem posed in Equations (8), (9) and (10) is a specific optimal control problem which the learning system is to solve on-line. Solution involves successively trying the admissible control actions until the "best" one is learned. If the PI of Equation (10) is used to evaluate the trials, rather than a subgoal satisfying (i), then a control decision is a choice of a sequence of N inputs $\{u(n);\ n = 0,\ldots,N-1\}$, $u(n)\in U$. There are K^N

such sequences and it takes $N\tau$ seconds to evaluate each trial. Furthermore, PI depends on $\underline{x}_o$, so step 3 in the mechanization procedure is still required to partially eliminate (or at least desensitize) this dependency. Partitioning the state space into L control situations creates L simultaneous experiments, in each of which the $\underline{x}_o$ dependency is assumed to be negligible. As N increases, both the time to complete a trial and the number of possible sequences increase. The control decision in (i) on the other hand consists of choosing a single control input from the K admissible actions in U. The subgoal must be capable of evaluating this decision and may be called a per-interval PI. Requirement (ii) is obviously necessary since the objective is to solve the primary problem.

A sub-goal is a function of $\underline{x}(n)$ and $u(n)$ which is minimized with respect to $u(n)$. Step 3 in Section II is still required to handle the $\underline{x}(n)$ dependency, i.e., trials of $u(n)$ are compared for all $\underline{x}(n)$ in a particular control situation. Consider

$$SG\,[\underline{x}(n),u(n),n] = F_1\,[\underline{x}(n+1),u(n),n] \quad n = 0,\ldots,N-1 \tag{11}$$

where $\underline{x}(n+1)$ depends on $\underline{x}(n)$ and $u(n)$ by Equation (9), and F_1 is a scalar function of its arguments. The form of the subgoal in Equation (11) satisfies (i). However, it remains to find relationships between $\underline{f}$, F and F_1 to satisfy (ii). Finding these relationships is precisely the subgoal problem. The F_1 satisfying these relationships is the exact subgoal, otherwise, it is a subgoal referred to as arbitrary, sub-optimal or inexact. Only the exact subgoal is expected to direct the learning controller to the optimal PI.

Relationships between $\underline{f}$, F and F_1 can be obtained via dynamic programming for the special case of a linear plant, quadratic PI and unconstrained control, as in Equations (6) and (7). For the unconstrained case with N fixed and $\underline{x}(N)$ free, the optimal control law is found to be[19]

$$u^*(n) = \underline{k}'(n+1)\underline{x}^*(n) \quad n = 0,\ldots,N-1 \tag{12}$$

and the minimum value of the PI is

$$PI^*(\underline{x}_o) = PI(u^*,\underline{x}_o) = \underline{x}_o'P(0)\underline{x}_o \tag{13}$$

where the gain vector $\underline{k}(n)$ and the matrix $P(n)$ (an nxn symmetric, time-varying matrix) are computed by iterating Equations (14) and (17) backward in time with starting condition $P(N) = \|0\|$.

$$R(n) = P(n) + Q \tag{14}$$

$$\underline{k}'(n) = -\frac{\underline{h}'R(n)\underline{\phi}}{\underline{h}'R(n)\underline{h} + \alpha} \tag{15}$$

$$\underline{\phi}(n) = \underline{\phi} + \underline{h}\underline{k}'(n) \quad n = N,\ldots,1 \tag{16}$$

$$P(n-1) = \underline{\phi}'(n)R(n)\underline{\phi}(n) + \alpha\underline{k}(n)\underline{k}'(n) \tag{17}$$

Consider a subgoal of the form

$$SG(n) = \underline{x}'(n+1)G(n)\underline{x}(n+1) + \lambda u^2(n) \tag{18}$$

where, in simplified notation, the arguments of SG are represented by n. Substituting Equation (6) into Equation (18) and minimizing with respect to u(n) yields the solution which minimizes the subgoal at time nτ.

$$u(n) = -\frac{\underline{h}'G(n)\emptyset}{\underline{h}'G(n)\underline{h} + \lambda}\,\underline{x}(n) \tag{19}$$

The exact subgoal should cause Equations (12) and (19) to be identical, so, Equations (12), (15) and (19) are compared to obtain these relationships between the PI and the subgoal.

$$\left.\begin{aligned} G(n) &= R(n+1) = 0 + P(n+1) \quad n = 0,\ldots,N-1 \\ \lambda &= \alpha \end{aligned}\right\} \tag{20}$$

Several significant observations bear on this result. First, a learning controller is being used because of some lack of information about the plant-environment. Yet, $\emptyset$ and $\underline{h}$ are required in computing the exact G. Section V presents and evaluates a procedure for choosing an inexact subgoal when the known values of $\emptyset$ and $\underline{h}$ are in error. It is stressed here that the learning system uses the subgoal in Equation (18), but does not use the analytical expression for the control law in Equation (19) which minimizes it. The system learns the control law using only the subgoal and past experience to reinforce current decisions.

A constant G matrix is desirable because of the method for storing past experience. Otherwise, an additional state, time, must be included in the partition forming control situations. It so happens that a constant G forms the exact subgoal if $N \to \infty$ or if Q is properly time variable. Though the latter is not too likely or meaningful, the infinite time problem is an often cited case. Even a knowledge of the form of the exact subgoal is of some value. It is especially useful to know that the subgoal is a time-variable quadratic of the states for finite N. Then, any arbitrarily chosen constant G is sub-optimal except when $N \to \infty$. In the solution, P(n) converges after relatively few iterations. So, even though P is unknown, it is known to be nearly constant until the last few sampling periods. And, it is reasonable to expect an inexact, constant G to yield near-optimal performance.

V. USES OF APRIORI INFORMATION

In many practical situations, the designer has nominal values and expected ranges of the plant parameters at his disposal. His job is to make the best use of this apriori information in his attempt to completely solve the primary problem. The most important problem confronting the designer

is still the choice of a subgoal. Section IV solved the problem for one class of systems, with the result depending on exact knowledge of $\emptyset$ and $\underline{h}$. In the following, a practical method of selecting a subgoal is suggested and compared to other uses of the same apriori information. Then, two other aspects of the design are considered: fixed grid extension and controller initialization.

Choice of a Sub-Optimal Subgoal

Begin with the ideal case: no control constraints and no state space partitions. Let the plant be represented by Equation (6) with actual parameters $\emptyset$ and $\underline{h}$. The apriori information is contained in a model composed of Equation (6) with given or guessed nominal values $\tilde{\emptyset}$ and $\tilde{\underline{h}}$. The suggested choice of a subgoal for the primary problem of minimizing the PI in Equation (7) is

$$SG(n) = \underline{x}'(n+1)G\underline{x}(n+1) + \alpha u^2(n) \tag{21}$$

where the constant G matrix is computed from Equations (14) and (17) with $\emptyset = \tilde{\emptyset}$, $\underline{h} = \tilde{\underline{h}}$ and $N \to \infty$. These equations become (22) and (25) in their steady state condition.

$$G = \tilde{P} + Q \tag{22}$$

$$\underline{k}_F' = - \frac{\tilde{\underline{h}}' G \tilde{\emptyset}}{\tilde{\underline{h}}' G \tilde{\underline{h}} + \alpha} \tag{23}$$

$$\tilde{\phi} = \tilde{\emptyset} + \tilde{\underline{h}} \underline{k}_F' \tag{24}$$

$$\tilde{P} = \tilde{\phi}' G \tilde{\phi} + \alpha \underline{k}_F \underline{k}_F' \tag{25}$$

The fixed gain $\underline{k}_F$ is the gain in the optimal control law for the model. The learning system, directed by the sub-optimal subgoal in Equation (18), learns $\underline{k}_L$ by making on-line trials and reinforcements. Assuming the learning process converges, the completely learned gain is given by

$$\underline{k}_L' = - \frac{\underline{h}' G \emptyset}{\underline{h}' G \underline{h} + \alpha} \tag{26}$$

Neither $\underline{k}_L$ nor $\underline{k}_F$ is optimal, except as a special case, but the learning controller is preferable if the following inequality is satisfied.

$$PI(u^*, \underline{x}_o) \leq PI(u_L, \underline{x}_o) \leq PI(u_F, \underline{x}_o) \tag{27}$$

The control inputs u_L and u_F are given below for this ideal case.

$$u_L(n) = \underline{k}_L' \underline{x}(n) \tag{28}$$

$$u_F(n) = \underline{k}_F' \underline{x}(n) \tag{29}$$

Next, consider the primary problem posed by Equations (6) and (7) with a bounded control input.

$$|u(n)| \leq U_M \tag{30}$$

This leads to a computationally difficult two-point boundary value problem, which is not likely to have a unique solution. The complications are due to the discrete-time formulation, and they are especially serious when $N \to \infty$. But, knowing that the form of the optimal control law is a saturating amplifier[22], dependent on $\underline{x}_o$, a procedure for choosing a subgoal can be suggested. Ignore the control bound and calculate G using $\tilde{\emptyset}$, $\tilde{h}$ and infinite N, as above. The fixed gain, calculated at the same time, can be used for comparison. Equations (31) and (32) are the resultant learned and fixed control laws, respectively.

$$u_L(n) = U_M \text{ sat} \left[\frac{\underline{k}_L' \underline{x}(n)}{U_M} \right] \tag{31}$$

$$u_F(n) = U_M \text{ sat} \left[\frac{\underline{k}_F' \underline{x}(n)}{U_M} \right] \tag{32}$$

In general, both of these are sub-optimal. In fact, if $\emptyset = \tilde{\emptyset}$ and $h = \tilde{h}$, then Equation (32) is the Letov solution.[23] It, too, is sub-optimal except for those initial states for which the trajectories enter (or originate in) the linear region and never leave.[24]

An identical approach is suggested for choosing a subgoal when the control input is quantized as in Equation (8). Compute G (and $\underline{k}_F$ for comparison) using $\tilde{\emptyset}$, $\tilde{h}$ and infinite N, still ignoring the constraints. The subgoal and its constant G matrix are given by Equation (21) and (25) for the primary problem posed by Equations (6), (7) and (8). Solutions to the primary and sub-problems are switching boundaries which separate the state space into regions. In each of these regions, one control action u_i is the best. And, the switching boundary[1] separating the region in which $u^* = u_i$ from the region in which $u^* = u_j$ is the set of all states $\underline{x}$ for which u_i and u_j are equally good. Equivalently, it is the locus of points $\underline{x}$ yielding constant $u = \frac{1}{2}(u_i + u_j)$.

As $N \to \infty$ in Equations (14) and (18), the gain $\underline{k}$ in Equation (15) becomes constant at the optimal value for the infinite time problem with unconstrained control. The optimal switching boundaries for the primary problem with constraints are conjectured to be the hyperplanes in Equation (33).

$$\underline{k}'\underline{x} = \frac{1}{2}(u_i + u_j) \tag{33}$$

Only the boundaries between adjacent values of u are required. With no loss of generality, order the elements in the set so that u_i and u_{i+1} are adjacent numerically as well as in position in U. Then, Equation (34)

gives the (K-1) switching boundaries.

$$\underline{k}'\underline{x} = \frac{1}{2}(u_i + u_{i+1}) \qquad i = 1,\ldots,K-1 \tag{34}$$

The learned and fixed switching boundaries are the following hyperplanes.

$$\underline{k}_L'\underline{x} = \frac{1}{2}(u_i + u_{i+1}) \qquad i = 1,\ldots,K-1 \tag{35}$$

$$\underline{k}_F'\underline{x} = \frac{1}{2}(u_i + u_{i+1}) \qquad i = 1,\ldots,K-1 \tag{36}$$

Here, as before, $\underline{k}_L$ and $\underline{k}_F$ are given by Equations (26) and (23). Equation (35) is the optimal solution to the sub-problem of controlling Equation (6) with actions from U to minimize Equation (21).

Reference 20 contains numerous comparisons between the learned and fixed controllers using the equations presented above. The next section contains simulation results to compare them.

Before proceeding to the simulation however, two other uses of apriori information are considered.

Extension of the Fixed Grid

A fixed grid covers a subset of the state space, as discussed in Section III. The states encountered during system operation will either (i) exactly coincide with, (ii) be contained in, or (iii) contain the subset. Presumably, (i) is the design objective, avoiding either the uneconomical use of memory locations accompanying (ii) or the degraded performance of (iii). Of the latter two, (iii) is preferred, provided a means of mapping outside states into boundary sets (i.e., for extending the grid), is available. It should cause little degradation in performance. Several schemes can perform this extension. The simplest method to implement is to extend the grid lines outward from the boundary parallel to the coordinate axes, as shown in Figure 3 for a second order system.

The method proposed here uses the apriori information to calculate $\underline{k}_F$ and G. This vector $\underline{k}_F$ predicts the positions and slopes of the switching boundaries and can be utilized to extend the boundary sets. Figure 3 also shows this form of extension in two dimensional space. The slope of the predicted switching boundaries (loci of constant $u = \underline{k}_F'\underline{x}$) used in this case is -2. Results of simulation studies are reported in Section VI for a large number of plants and wide variations of assumed knowledge confirming that this form of extension yields superior performance.

An extension in two dimensions can be programmed by examining the geometry of Figure 3. Systems of third and higher order are more difficult and the classification time for higher dimensional space using an extended fixed grid could become greater than for the variable grid it replaced. However, it is unlikely that it would be necessary to grid a

very high dimensional state space even for high order systems. One reason is that only the measurable states or outputs would be gridded. Besides, the majority of the operation time is with $\underline{x}$ inside the grid.

The two learning systems used in the experiments in Section VI differ only in their method of extending the state space grid.

LERN: Extends parallel to the axes

LARN: Extends parallel to the predicted switching boundaries

Initialization of the Controller

The controller was defined in Appendix A in Equations (A1) and (A11). Of the variables involved, $p_{ij}(0)$ and $\tilde{d}_{ij}(0)$ must be initialized. With no knowledge of the plant, the controller is initialized by setting all $P_{ij}(0) = 1/K$, $\tilde{d}_{ij}(0) = C_{ij}(0) = 0$, and requiring that each action be chosen deterministically in each S_j (as it is encountered). No reinforcement can take place in the j-th column of P(n) until the state has been set K times, slowing down the learning process.

Assuming that some knowledge of the plant is available, this technique is no longer necessary. Then, the following procedure is suggested:

1. Use $\tilde{\emptyset}$, $\tilde{\underline{h}}$, Q to compute G, the subgoal to be used henceforth.
2. Use $\tilde{\emptyset}$, $\tilde{\underline{h}}$ and a representative $\underline{x}_j = \underline{x}(0)$ for each S_j and compute $d_{ij}(0)$ for each u_i. This initializes the estimators $\tilde{d}_{ij}(0) = d_{ij}(0)$ and setting $C_{ij}(0) = 1$ initializes the counters, off-line.
3. Initialize $p_{ij}(0)$ based on $\tilde{d}_{ij}(0)$, i = 1,...,K for each S_j, using the knowledge that if some u_i is the best in S_j, then u_{i+1} and u_{i-1} are the next best.
4. Make control decisions as in Equation (A3) and reinforce based upon evaluations with the subgoal using G.

The typical $\underline{x}(0)$ used in Step 2 was the center of S_j, j = 1,...,L. Using this as the initial condition, each control action in turn was used to compute $\underline{x}(1)$ and then $d_{ij}(0)$ by Equation (A5).

Three methods for initializing $p_{ij}(0)$ for Step 3 are compared here.

a. Set equally likely, making no use of apriori information.

$$p_{ij}(0) = 1/K, \quad i = 1,\ldots,K; \quad j = 1,\ldots,L. \tag{37}$$

b. Set proportional to the estimates, the method used in Reference 7 for all time, but here only for initializing. Since $-1 \leq \tilde{d}_{ij}(0) \leq +1$, translate it to the unit interval and set the probabilities as follows,

$$\tilde{d}_{ij}(0) = \frac{1}{2}\left(\tilde{d}_{ij}(0) + 1\right) \quad \text{all } i,j$$

$$P_{ij}(0) = \frac{\bar{d}_{ij}(0)}{\sum_{\ell=1}^{K} \bar{d}_{\ell j}(0)} \qquad \text{all } i,j \tag{39}$$

c. Set to fall off from the most likely in a linear fashion. Find M_j, the index of the maximum $d_{ij}(C)$ as indicated in Equation (A11) for each j. The equations below cause probabilities adjacent to the largest to be (K-1)/K times as large, etc.

$$P_{M_j j}(0) = \frac{2K}{K^2 - K - 2KM_j + 2M_j - 2M_j^2} \tag{40}$$

$$P_{ij}(0) = \begin{cases} P_{M_j j}(0) \cdot \dfrac{(K-M_j+1)}{K} & 1 \leq i < M_j \\ P_{M_j j}(0) \cdot \dfrac{(K+M_j+i)}{K} & M_j \leq i \leq K \end{cases} \tag{41}$$

For the conditions in Section VI, Figure 4 depicts these three techniques for the 96-th control situation of those experiments for Plant 1, Model 1, Condition 1, with $x_1(0) = x_2(0) = 27.5$, $G_{11} = 37.8078151$, $G_{12} = G_{21} = 2.2047662$, $G_{22} = 1.2922614$, $\alpha = 1$ and the control sets U given in Equation (45). For this case $M_{96} = 1$, and as can be observed, there is little difference in the latter two methods. The linear reinforcement technique was used in the simulations reported in Section VI, but method b could have been used with little difference. The main improvement comes in setting them so that the controller may begin to make probabilistic decisions and be reinforced immediately, rather than having a period of deterministic decisions in order to initialize the controller on-line.

VI. EXPERIMENTAL RESULTS

Purposes of the Experiments

The basic learning control system has been presented with a formulation of the subgoal problem, and some conclusive results for several special cases. The suggested method for selecting a subgoal is a heuristic extension from the ideal case, and uses a model of apriori information to make the selection. Two other aspects of the design were given special considerations above. The motivation for seeking answers about the grid and initialization was supplied automatically when the first simulation comparison of subgoals was begun.[20] The most drastic need was to cut down the computer time. The switching from hyperspheres to a fixed grid reduced approximately twenty minute programs on IBM 7094 to about one

minute.

The purposes of the experiments reported below are to evaluate the proposals in Section V on a full scale simulation, accounting for quantization effects. Learning time was of incidental importance, which is the reason the algorithm from [1] was left unmodified. The primary purpose is to establish that in many cases the subgoal chosen as suggested yields better performance than other controllers designed with the same apriori information.

Description of the Experiments

A plant is controlled by several methods in each experiment, differing in their use of the model of apriori information, and data is presented to compare them.

All plants and models are described by the differential equation

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ b \end{bmatrix} u \qquad \underline{x}(0) = \underline{x}_o \tag{42}$$

with parameter values given in Table 1, including exact and poor information. The primary goal is to control the plant subject to control constraints, given only the model parameters, so as to minimize the PI of Equation (7) with $\alpha = 1$. Though the sampling period, which is also the control interval, is τ seconds, the performance index evaluates response over $T = N\tau$ seconds. Results are presented for two conditions:

$$\text{Condition 1: } \underline{x}_o = \begin{bmatrix} 50 \\ 0 \end{bmatrix}, \ \tau = 0.25 \text{ sec.} \quad N = 15, \ Q = \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix} \tag{43}$$

$$\text{Condition 2: } \underline{x}_o = \begin{bmatrix} 40 \\ 0 \end{bmatrix}, \ \tau = 0.15 \text{ sec.} \quad N = 25, \ Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{44}$$

Control input u is assumed to be bounded by $-20 \leq u \leq 20$, and this interval is quantized into K even levels. Two sets of allowable control actions were used.

$$K = 5: \quad u(i) \in U = \{-20, -10, 0, 10, 20\} \tag{45}$$

$$K = 9: \quad u(i) \in U = \{-20, -15, -10, -5, 0, 5, 10, 15, 20\} \tag{46}$$

Equations (22) and (25) and the model data were used to compute G, which is the method suggested, and which assumes T and N are infinite and

TABLE 1. PLANT AND PLANT PARAMETERS

	MODEL NO.			MODEL NO.								
	1	2	3	1	2	3	4	5	6	7	8	9
a	+1.0	0.0	-1.0	+1.0	+0.5	+2.0	+1.0	+1.0	0.0	-1.0	+1.0	0.0
b	25.0	25.0	25.0	25.0	25.0	25.0	12.5	50.0	25.0	25.0	25.0	12.5

u unconstrained. The learning systems are directed by the subgoal of Equation (21) with the computed G, and $\alpha = 1$. Several fixed controllers using the $\underline{k}_F$ in Equation (23) were used to control the plants. Table 2 lists the controller gains and switching boundary slopes just computed. Using these $\underline{k}_F$ values, the following control laws were used with their respective plants: Fixed-Free (u unconstrained), Fixed-Bounded (using the same gain, apply a saturation as in the sub-optimal Letov solution), and Fixed-Quantized (quantize using the allowable control actions and the same gain value). The fixed control law equations are:

$$u_{FF}(n) = \underline{k}_F' \underline{x}(n) \tag{47}$$

$$u_{FB}(n) = 20 \text{ sat} \left[\frac{u_{FF}(n)}{20} \right] \tag{48}$$

$$u_{FQ}(n) = u_i \text{ if } \frac{u_{i-1} + u_i}{2} \leq u_{FF}(n) \leq \frac{u_i + u_{i+1}}{2} \tag{49}$$

In the learning experiments, normal operating conditions were simulated by choosing an $\underline{x}_o$ with uniform distribution on the region bounded by the fixed grid in Figure 3 (and outside a circle of radius 5), controlling the plant for N control intervals, and reinforcing each control decision as discussed above. Every sixth $\underline{x}_o$ was chosen deterministically as the test $\underline{x}_o$ for that condition [either Equation (43) or (44)], and PI values were recorded for learning curves.

Program output, after 50 trials with the test $\underline{x}_o$ for K = 5 (75 for K = 9), included a grid which showed the most likely u_i for each S_j and the learned trajectory. This furnished data to compare the learned to the theoretical switching boundaries, as well as the PI values. A measure of the correctness of the learning is the Number of Incorrectly Learned Sets (NILS) given in Table 3. In order to compare LERN to LARN, only the 38 border sets (j = 1,..., 20, 21, 40, 41,..., 180, 181, 200) were considered. For the most part, LERN and LARN caused identical learning inside the grid. Performance index values are given in Tables 4 and 5 for all learning configurations.

The two conditions in Equations (43) and (44) have identical T = 3.75 seconds. Different Q and $\underline{x}_o$ were chosen so that the trajectory for Condition 1 would spend less time outside the grid than that for Condition 2. As would be expected, in this case, LARN did not improve on LERN as much for Condition 2 as for Condition 1. To illustrate, compare the ratio of the NILS total from Table 3: 120/50 = 2.4 to 104/106 = 2.26 and 191/98 = 1.95 to 205/108 = 1.90. Typical learning curves and system responses are given in Appendix B.

Discussion of the Results

The following observations are made on the basis of the experimental results:

1. Increasing K yields better performance (See Tables 4 and 5) but longer learning time.[20]
2. Learned results compared with Fixed-Quantized (FQ) is fairer than with Fixed-Bounded, assuming K can be increased:
   a. Learned performance is always better than FQ when the gain differs; e.g., Plant 1, Models 4 and 5 and Plant 2 and 3, Model 9.
   b. General trends not apparent for $a \neq \tilde{a}$; e.g., Plant 2, Models 6, 7 and 8.
3. LARN is better than LERN, with a greater difference in Condition 1 than in Condition 2, as shown in NILS totals of Table 3 and as was predictable from the initial conditions (See Figures 5 and 6).
4. It is especially significant that even when the Model leads to an unstable fixed controller, the learned controller is stable, e.g., Plant 1, Model 4, Condition 1.

VII. CONCLUSIONS AND FURTHER RESULTS

Within the scope of the experiments reported in this paper, it is concluded that the learning systems directed by the subgoal compares well with a fixed controller designed with the same apriori information. The proposed method of grid extension along the predicted switching boundary slopes yields better performance than a parallel extension. It is particularly significant that the learning controller leads to stable performance even when the apriori information yields an unstable fixed controller. This means that though the learning system may not always excell, cases might occur when using the fixed controller would be disastrous.

It is often said that learning control systems, such as the one presented in this paper are too complex and that they are not realistic solutions to practical control problems. There is no doubt that such objections are valid in some sense. But, it is a very narrow and confined sense. The random search is the central part of the controller and this method of searching is time consuming. But, there is a trade-off between convergence and efficiency. This method of control is proposed for those situations in which the use of simpler methods is not possible because of lack of sufficient information. And, above all, convergence is desired and required. It behooves the designer to be on his toes to solve his problem with the least complicated technique which assures acceptable performance.

References:

1. Waltz, M.D., Fu, K.S., "A Heuristic Approach to Reinforcement Learning Control Systems", IEEE Transactions on Automatic Control, Vol. AC-10, No. 4, October, 1965, pp. 390-398.
2. Gibson, J.F., Fu, K.S., et al, "Philosophy and State of the Art of Learning Control Systems", Purdue University TR-EE63-7, Lafayette, Indiana, November, 1963.
3. Nickolic, Z., Fu, K.S., "An Algorithm for Learning Without External Supervision and Its Application to Learning Control Systems", IEEE Transactions on Automatic Control, Vol. AC-11, No. 3, July, 1966, pp. 414-423.
4. Gibson, J.E., "Adaptive Learning Systems", Proceedings of the National Electronics Conference, Vol. 18, October, 1962.
5. Fu, K.S., "Learning Control Systems", Proc. COINS Symposium, Evanston, Illinois, June, 17-18, 1963.
6. Tsypkin, Ya. Z., "Adaptation, Training and Self-Organization in Automatic Systems", Automation and Remote Control, Vol. 27, No. 1, January, 1966.
7. McMurtry, G. J., Fu, K.S., "A Variable Structure Automaton Used as a Multimodal Searching Technique", IEEE Transactions on Automatic Control, Vol. AC-11, No. 3, July,1966, pp. 379-387.
8. Fu, K.S., McLaren, R. W., "An Application of Stochastic Automata to the Synthesis of Learning Systems", Purdue University TR-EE65-17, September, 1965.
9. Fu, K.S., Nikolic, Z.J., "On Some Reinforcement Techniques and Their Relation to the Stochastic Approximation", IEEE Transactions on Automatic Control, Vol. AC-11, No. 4, October, 1966, pp. 756-758.
10. Varshavskii, V.I., Vorontsova, I.P., "On the Behavior of Stochastic Automata with a Variable Structure", Automatika i Telemekhanika, Vol. 24, No. 1, March, 1963, pp. 353-360.
11. Chandrasekaran, B., Shen, D.W.C., "On Expediency and Convergence in Variable-Structure Automata", IEEE Transaction on Systems Science and Cybernetics, Vol. SSC-4, No. 1, March, 1968, pp. 52-59.
12. Fu, K.S., "Stochastic Automata as Models for Learning Systems", Computer and Information Sciences - II, Edited by J. T. Tou, Academic Press, New York, N. Y., 1967.
13. Mahne, S.J., Fu, K.S., "Learning System Heuristics", Correspondence and Response by the Author of [1], IEEE Transactions on Automatic Control, Vol. AC-11, No. 3, July, 1966, pp. 611-612.
14. Jones, L. E., III, "On the Choice of Subgoals for Learning Control Systems", Proceedings of the N.E.C., Vol. 23, 1967, pp. 62-66, and IEEE Transactions on Automatic Control, Vol. AC-13, No. 6, December 1968.
15. Liff, A.I., Wolf, J.K., "On the Optimum Sampling Rate for Discrete-Time Modeling of Continuous-Time Systems", IEEE Transactions on Automatic Control, Vol. AC-11, No. 2, April, 1966, pp. 288-290.
16. Bekey, G.A., Tomovic, R., "Sensitivity of Discrete Systems to Variation of Sampling Period", IEEE Transactions on Automatic Control, Vol. AC-11, No. 2, April, 1966, pp. 284-287.
17. Smith, F.W., Hilton, W.B., "Monte Carlo Evaluation of Methods for Pulse Transfer Function Identification", IEEE Transactions on Automatic Control, Vol. AC-12, No. 5, October, 1967, pp. 568-576.
18. Smith, F. W., "System Laplace Transform Estimation from Sampled Data", IEEE Transactions on Automatic Control, Vol. AC-13, No. 1, February, 1968, pp. 37-44.
19. Kalman, R. E., Koepcke, R. W., "Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes", Transactions of the ASME, November, 1958, pp. 1820-1826.

20. Jones, L.E., III, "A Learning Control System-Design Considerations," Ph.D. Thesis, Purdue University, Lafayette, Indiana, January, 1969.
21. Schwarz, R.J., Friedland, B., Linear Systems, McGraw-Hill Book Co., New York, N. Y., 1965.
22. Pearson, J.B., Jr., Sridhar, R., "A Discrete Optimal Control Problem", IEEE Transactions on Automatic Control, Vol. AC-11, No. 2, April, 1966, pp. 171-174.
23. Letov, A.M., "Analytical Controller Design II", Automation and Remote Control, Vol. 21, May 1960, pp. 561-568.
24. Rekasius, Z.V., Hsia, T.C., "On an Inverse Problem in Optimal Control, IEEE Transactions on Automatic Control, Vol. AC-9, October, 1964, pp. 370-375.

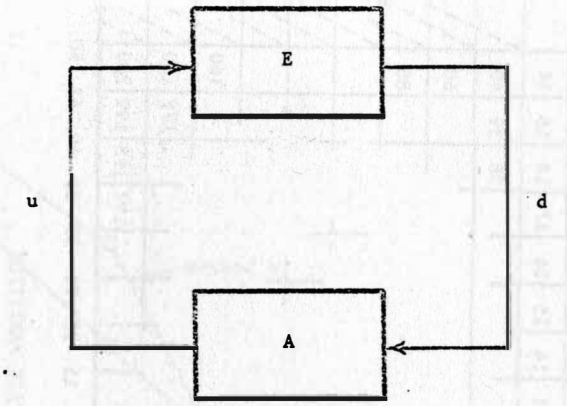


FIGURE 1 AN ABSTRACTED LEARNING CONTROL SYSTEM

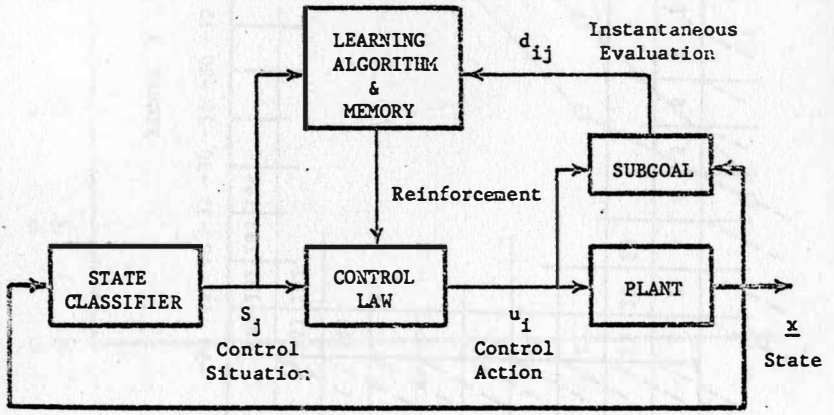


FIGURE 2 THE LEARNING CONTROL SYSTEM

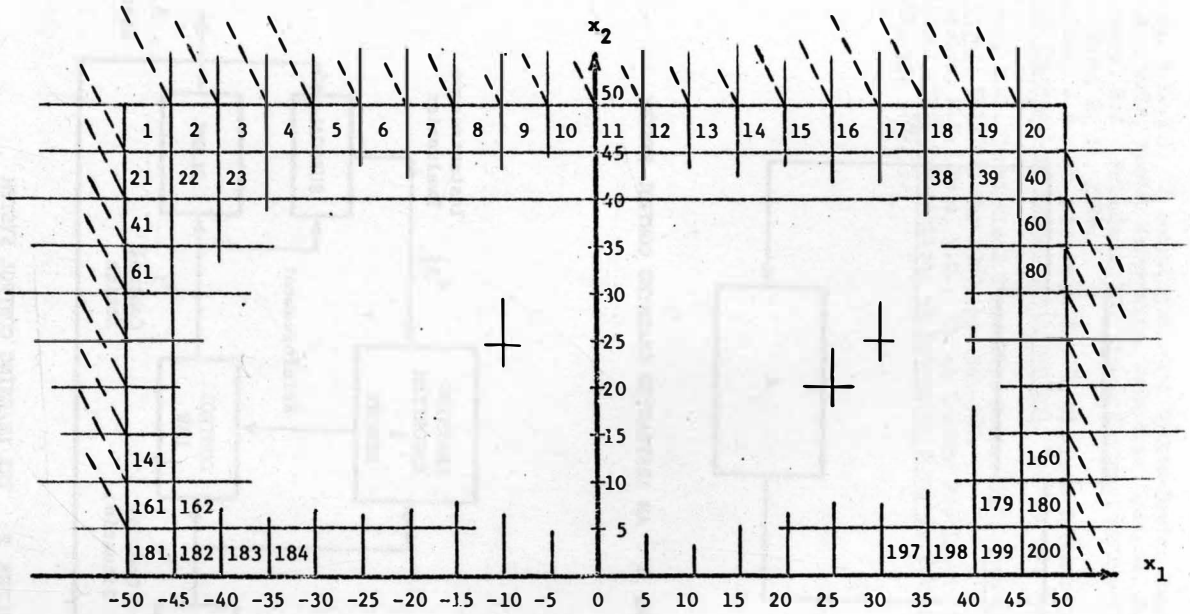


FIGURE 3 FIXED GRID STATE SPACE PARTITION AND EXTENSIONS

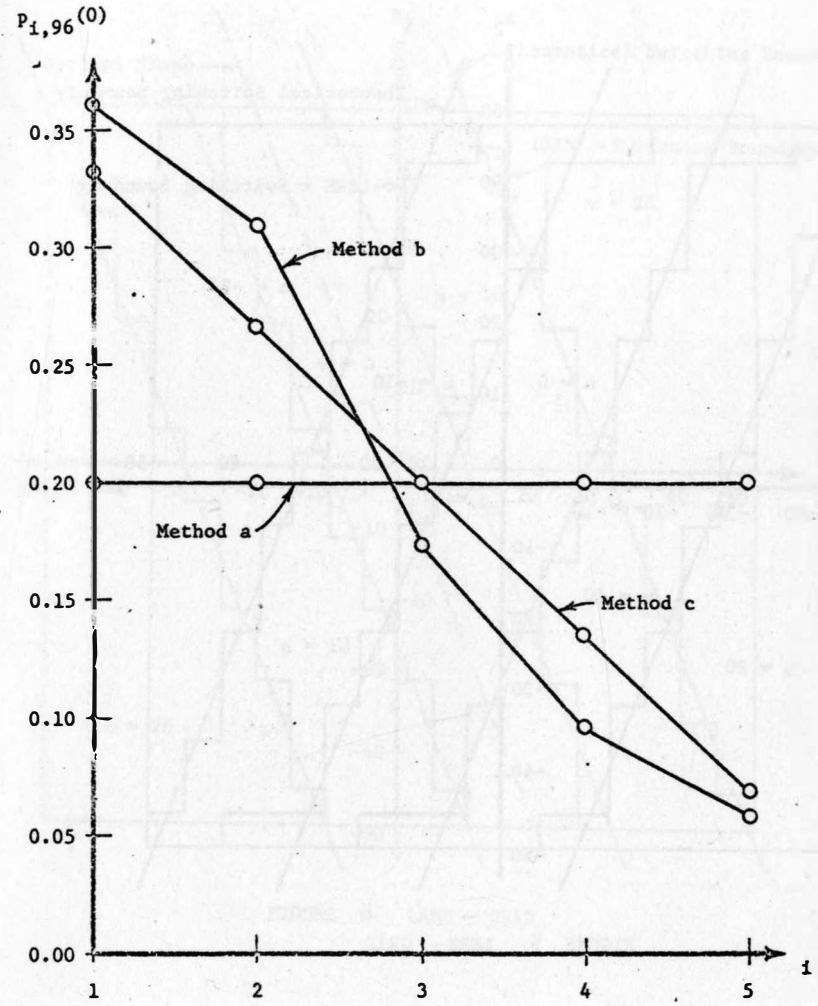


FIGURE 4 INITIAL PROBABILITIES FOR SET 96, K = 5

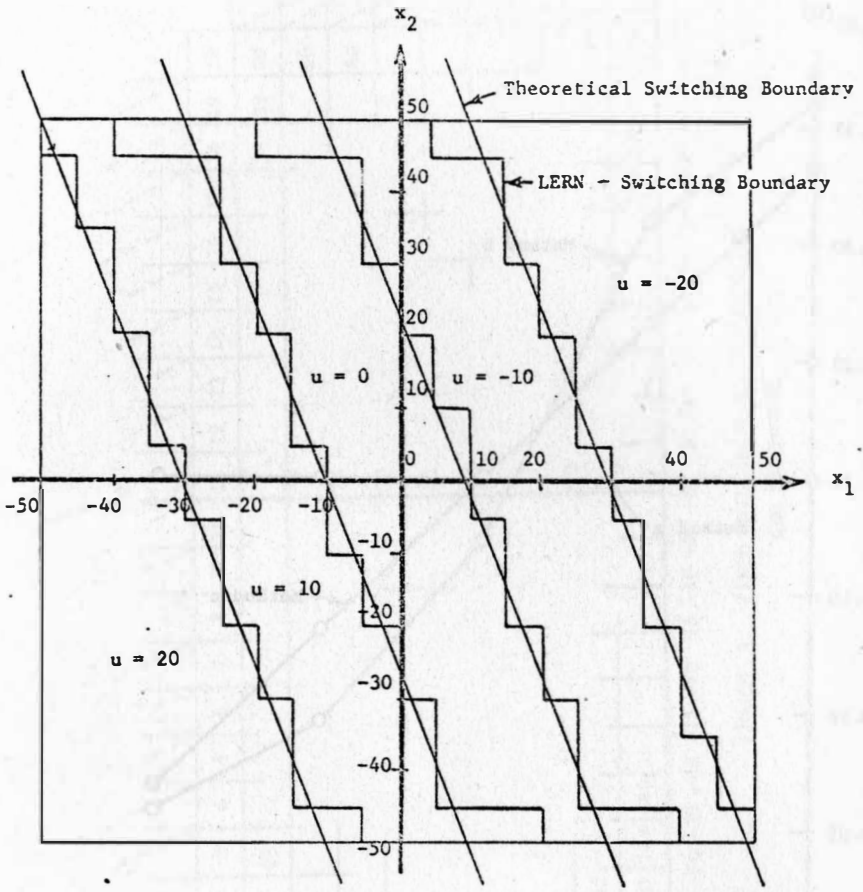


FIGURE 5. LERN - GRID

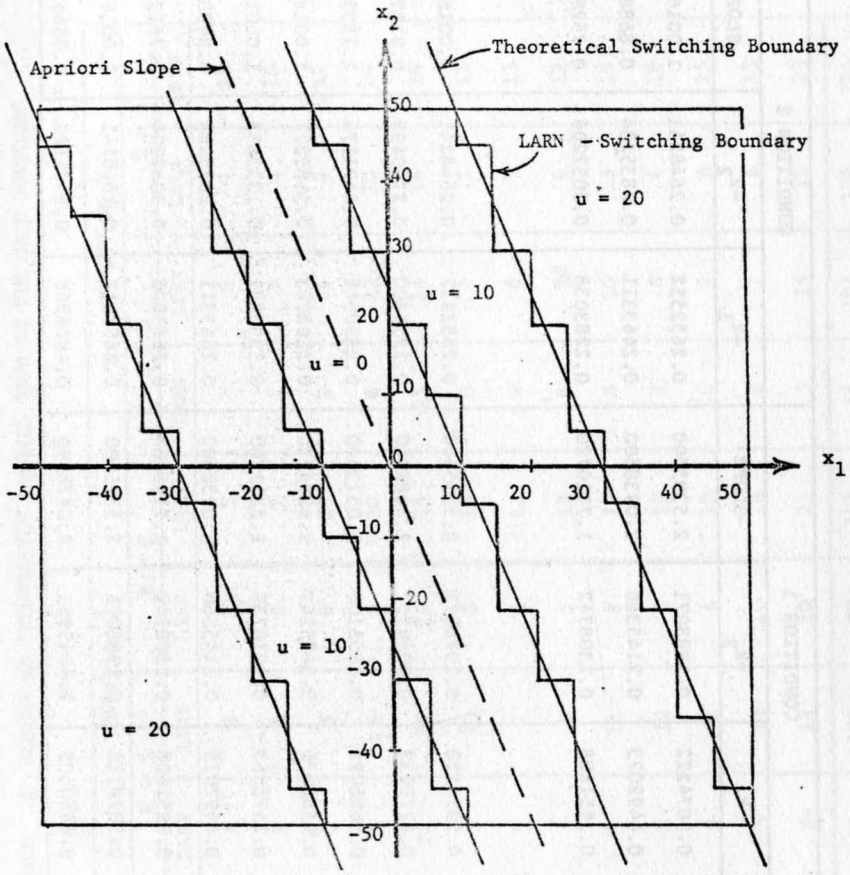


FIGURE 6 LARN – GRID

TABLE 2 CALCULATED GAINS AND SWITCHING BOUNDARY SLOPES

PLANT NO.	CONDITION 1			CONDITION 2		
	$-k_1$	$-k_2$	-SLOPE	$-k_1$	$-k_2$	-SLOPE
1	0.5074222	0.1998071	2.5395600	0.2652512	0.2648241	1.0016130
2	0.4492075	0.2145366	2.0938500	0.2463311	0.2835234	0.8688212
3	0.3951808	0.2308742	1.7116720	0.2283038	0.3032204	0.7529304
MODEL NO.						
1	0.5074222	0.1998071	2.5395600	0.2652512	0.2648241	1.0016130
2	0.4778033	0.2069751	2.3085060	0.2556800	0.2740493	0.9329703
3	0.5696028	0.1866106	3.0523600	0.2850548	0.2471174	1.1535200
4	0.9582076	0.3835247	2.4984250	0.5198223	0.5189877	1.0016080
5	0.2578663	0.1010737	2.5512700	0.1333281	0.1331133	1.0001260
6	0.4492075	0.2145366	2.0938500	0.2463311	0.2835234	0.8688212
7	0.3951808	0.2308742	1.7116720	0.2283038	0.3032204	0.7529304
8	0.5074222	0.1998071	2.5395600	0.2652512	0.2648241	1.0016130
9	0.8490405	0.4145961	2.0478740	0.4828268	0.5571122	0.8666599

TABLE 3 NUMBER OF INCORRECTLY LEARNED SETS ON THE GRID BOUNDARY

PLANT MODEL	CONDITION 1				CONDITION 2			
	K = 5		K = 9		K = 5		K = 9	
	LARN	LERN	LARN	LERN	LARN	LERN	LARN	LERN
1-1	1	5	1	8	1	3	4	11
1-2	2	5	2	7	1	5	1	13
1-3	1	5	6	10	2	5	8	12
1-4	3	6	9	10	6	11	20	24
1-5	14	15	25	25	15	14	26	24
2-6	1	10	3	16	0	4	1	11
2-7	5	7	6	13	4	6	7	13
2-8	3	8	5	13	2	10	6	11
2-9	7	11	9	18	6	10	13	19
3-6	2	12	8	15	2	6	4	16
3-7	0	9	4	19	0	9	3	15
3-8	3	13	10	16	2	7	4	13
3-9	8	14	10	21	5	14	11	23
TOTAL	50	120	98	191	46	104	108	205

TALBE 4 PERFORMANCE INDICES FOR CONDITION 1

PLANT MODEL			K = 5			K = 9		
	FIXED-FREE	FIXED-BOUNDED	FIXED-QUANTIZED	LARN	LERN	FIXED-QUANTIZED	LARN	LERN
1-1	4.4519747	4.6760632	6.1106370	6.2207673	6.2207673	4.9921990	4.9921990	4.9921990
1-2	4.5013914	4.7090902	5.9691791	6.3540898	6.2207673	4.9846682	5.0892173	6.5601948
1-3	4.6531649	4.8070325	6.0103131	6.2207673	6.2207673	5.1608747	5.2126928	6.5601948
1-4	716079.33	9.1708926	10.557323	6.2207673	6.2207673	9.4454831	4.9922000	5.2867873
1-5	5.8844292	5.8844292	9.5654211	6.2207673	7.0400880	6.1290573	5.2126928	5.8160306
2-6	4.5517828	4.6117812	4.8695703	4.8695703	4.8695703	4.8695703	4.8695703	5.9426855
2-7	4.8004273	4.8004273	5.5931641	4.8695703	8.9823047	4.9600195	4.9600195	4.8695703
2-8	4.8132765	4.7864689	4.8695703	4.8695703	8.6693359	5.2368848	5.2368848	5.9426855
2-9	65334764.	10.435745	11.873711	5.5931641	4.8695703	10.441006	4.8695703	6.5863476
3-6	5.0070006	4.9064164	6.1079473	5.9524361	10.689633	5.3276671	5.0186118	6.5163609
3-7	4.6600914	4.6600914	5.9524361	5.9524361	10.689633	5.0186118	5.0186118	7.6088591
3-8	6.1920680	5.7112098	7.6081326	7.6081326	10.104486	6.7481958	6.5163609	14.747823
3-9	281998760.	8.8255262	11.370411	5.9524361	10.689633	9.4246295	5.3339149	7.6088591

(Multiply all values by 10^4)

TABLE 5 PERFORMANCE INDICES FOR CONDITION 2

PLANT MODEL			K = 5			K = 9		
	FIXED-FREE	FIXED-BOUNDED	FIXED QUANTIZED	LARN	LERN	FIXED-QUANTIZED	LARN	LERN
1-1	10.652208	10.652208	10.739136	10.739136	10.739136	10.739136	10.739136	10.739136
1-2	10.676504	10.676504	10.739136	10.739136	10.739136	10.739136	10.739136	10.739136
1-3	10.771583	10.771583	10.739136	10.739136	10.739136	10.739136	10.739136	10.739136
1-4	23.849813	21.863010	46.638666	10.739136	10.739136	46.638666	10.739136	10.739136
1-5	11.177938	11.177938	10.739136	10.739136	10.739136	11.177420	11.177420	10.739136
2-6	10.671340	10.671340	15.540703	13.736972	15.540703	11.705210	11.387266	11.387266
2-7	10.791525	10.791525	15.789219	13.736972	15.540703	11.983990	11.387266	11.387266
2-8	10.810313	10.810313	13.551739	15.540703	15.540703	11.452744	11.452744	11.452744
2-9	796.98002	45.247660	40.227500	13.736972	15.540703	44.664253	11.387266	11.387266
3-6	10.854839	10.854839	16.353739	18.062188	18.062188	12.098742	12.097313	12.097313
3-7	10.692168	10.692168	16.353739	18.062188	18.062188	11.746322	12.097313	12.097313
3-8	11.343090	11.343090	17.684409	18.062188	18.062188	12.098742	11.660995	11.660995
3-9	928.98095	40.368934	41.693656	18.062188	18.062188	38.064325	12.097313	12.097313

(Multiply all values by 10^4)

Appendix A

A finite, stochastic automaton is used as the controller to make control decisions among K admissible actions for each of L control situations. Let S_j denote the j-th control situation. Consider that the outputs and the internal states of this automaton are identical, and the automaton is characterized by the stochastic matrix $P(n) = \|P_{ij}(n)\|$, where

$P_{ij}(n)$ = the probability that $u(n) = u_i$ is the optimal decision for $\underline{x}(n) \in S_j$ at time $n\tau$ (A1)

and

$$\sum_{i=1}^{K} P_{ij}(n) = 1.0 \qquad j = 1,\ldots,K \qquad n = 0,1\ldots \tag{A2}$$

Unlike many proposed automata model [10], this corresponds to a state probability matrix, not a state transition matrix. The elements are reinforced during learning as directed by the learning algorithm.

Each of the L columns represents an independent learning experiment. Classifying $\underline{x}(n)$ into S_j determines which experiment is being performed. Generate R from a uniform distribution on the unit interval, and the control decision is: Chooseu(n) = u_i for an i satisfying

$$\sum_{r=1}^{i-1} P_{rj}(n) \leq R \leq \sum_{r=1}^{i} P_{rj}(n) \tag{A3}$$

Learning occurs as a result of a succession of reinforcements, and is evident in P(n) when one probability in each column tends toward one, the others toward zero. Only one of the L experiments is in operation at a given time, the one determined by the present state. Using a subgoal, the present state is the initial condition for the optimization. If the grid is sufficiently fine to assure that any state will move between sets for a any allowed control action, then, it is reasonable to assume that the occurrence of any state in a given set is equally likely. Therefore, successive trials differ because of the random variable $\underline{x}_o$. The assumption is not strictly valid with a coarse grid unless measurement noise or environmental factors are present.

In order to define the reinforcement learning algroithm, let the subgoal be

$$SG(n) = \underline{x}'(n+1)G\underline{x}(n+1) + \alpha u^2(n)^{\dagger} \tag{A4}$$

---

† It satisfies the subgoal conditions (i) and (ii) in Section IV provided G is chosen appropriately.

At time $n\tau$, the state $\underline{x}(n)$ is sampled and classified in S_J. A control decision is made that $u(n) = u_I$, based on the current probabilities in the J-th column of P(n) in Equation (A1). The subgoal in Equation (A3) evaluates this decision one control interval later, comparing it to the past history of choices in this control situation, and rates it as being better (or worse) than other choices. Then the probability of making this choice the next time this control situation is encountered is positively (or negatively) reinforced.

The algorithm here is an exact duplication of Reference 1. It is briefly presented here for a description of the reported experiments. To compare to the past history, the subgoal is normalized and the minimization is converted to an equivalent maximization by introducing

$$d_{IJ}(n) = \frac{\underline{x}'(n)G\underline{x}(n) - SG(n)}{\text{Max}[\underline{x}'(n)G\underline{x}(n),\ \underline{x}'(n+1)G\underline{x}(n+1)] + \alpha u_{max}^2} \tag{A5}$$

The instantaneous evaluations $d_{ij}(n)$ are used to form estimates $\tilde{d}_{ij}(n)$ of the value of each decision, averaging over $\underline{x} \in S_j$.

$$\tilde{d}_{IJ}(n) = \frac{(C_{IJ}(n) - 1)\,\tilde{d}_{IJ}(n-1) + d_{IJ}(n)}{C_{IJ}(n)} \tag{A6}$$

$$\tilde{d}_{ij}(n) = \tilde{d}_{ij}(n-1) \quad \text{all other } i,j \tag{A7}$$

$C_{IJ}(n)$ is the number of times u_I has been chosen in S_J and is increased each time.† The linear reinforcement is performed only on the J-th column, as follows:

$$P_{iJ}(n+1) = \theta_n\, P_{iJ}(n) + (1-\theta_n)\,\lambda_n(u_i, S_j) \tag{A8}$$

where

$$\lambda_n(u_i, S_j) = \begin{cases} 1 \text{ for } i = M & \text{(positive reinforcement)} \\ 0 \text{ for } M \neq i = 1,\ldots,K & \text{(negative reinforcement)} \end{cases} \tag{A9}$$

The learning parameter

$$\theta_n = 1.0 - 0.5\,[\tilde{d}_{MJ}(n) - \tilde{d}_{NJ}(n) \tag{A10}$$

depends on

$$\tilde{d}_{MJ}(n) = \underset{i}{\text{Max}}\ \tilde{d}_{iJ}(n) \tag{A11}$$

---

† In the systems simulated $C_{IJ}(n)$ is increased up to a maximum of 9. After this, the estimates are weighted averages, emphasizing the latest 9 trials.

$$\tilde{d}_{NJ}(n) = \underset{i \neq M}{\text{Max}}\, \tilde{d}_{iJ}(n) \qquad \text{(A12)}$$

the largest and next largest, or equivalently, the best and next best choices in S_J. This provides reinforcement which is dependent upon the relative superiority of one action, $0 \leq \theta_n \leq 1$. Section V describes how to initialize the estimators and counters in the controller.

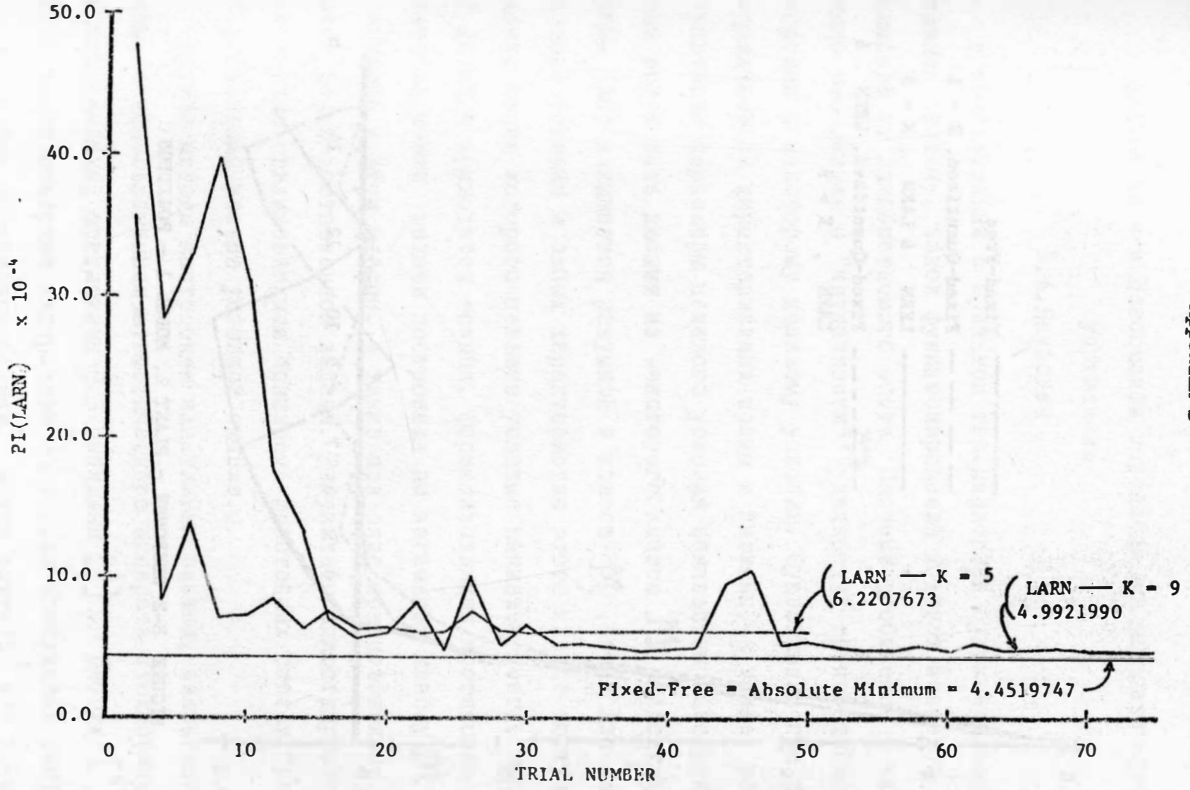


FIGURE B-1 LEARNING CURVE - PLANT 1, MODEL 1

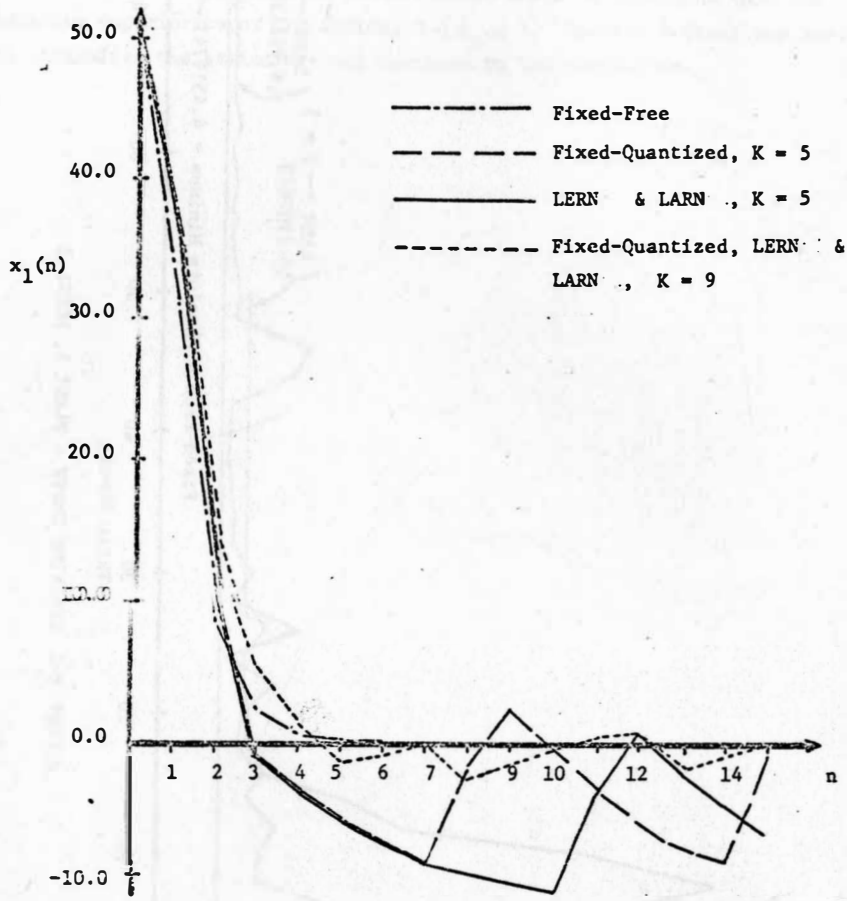


FIGURE B-2 RESPONSE - PLANT 1, MODEL 1 - POSITION

ОБ ОДНОМ КЛАССЕ АДАПТИВНЫХ (САМООБУЧАЮЩИХСЯ) СИСТЕМ

Доклад на 4-м Всесоюзном совещании по автоматическому управлению

В.А.Якубович

В соответствии с принятой терминологией будем называть адаптивной систему, закон функционирования которой меняется в зависимости от приобретаемого опыта. Системе сообщается в какой-либо виде информация о "неудачности" или "удачности" ее поведения по отношению к некоторому целевому условию. Существенно при этом, что определенные характеристики среды и системы, а также, возможно, некоторые параметры целевого условия неизвестны конструктору, - они могут быть любыми из некоторого класса M . Адаптивная система (АС) называется разумной в классе M , если для любого целевого условия и любых характеристик этого класса наступает момент, после которого целевое условие начинает всегда выполняться. В докладе приводится точная, формализованная постановка простейшего варианта задачи построения по заданному классу M системы, разумной в этом классе, а также при ряде предположений - решение этой точно поставленной задачи. Результаты иллюстрируются математически стилизованными примерами простейших систем, "разумных" в указанном, весьма условном смысле.

Другие методы построения адаптивных систем, связанные, в основном, с использованием математического аппарата стохастической аппроксимации, предложены Я.З.Цыпкиным [1]. В работе [1] имеется также обширная литература по теории адаптивных систем.

I°. Точная постановка задачи. Будем считать, что время t принимает значения $t = 0, 1, 2, \ldots$. Величины меняющиеся (вообще говоря) во времени будем называть переменными, а величины,

значения которых фиксированы для данной системы (и, следовательно, не меняются во времени), - параметрами. Среди параметров выделим так называемые варьируемые параметры $\underline{\xi}$, которые могут принимать любые значения из некоторого заданного множества M. При этом $\underline{\xi} = \|\underline{\xi}_j\|$ -многомерный вектор. (Выделение этих параметров имеет следующий смысл. Варьируемыми называются те параметры, которые могут меняться от эксперимента к эксперименту, и значение которых заранее конструктору неизвестно. Класс M поэтому будет определять класс задач, "решаемых" адаптивной системой). Заданное множество некоторых элементов z будем обозначать через $\{z\}$. Значение переменной z в момент t будем обозначать z_t. Будем считать заданными множества $\{x\}, \{s\}, \{\underline{\sigma}\}, \{u\}$ и подлежащим определению (в соответствии с условиями, сформулированными ниже) множество $\{\underline{\tau}\}$, элементы которых называются так: x -внешние координаты АС, s -среда, $\underline{\sigma}$ -сенсоры, u -управление, $\underline{\tau}$ -тактика. Пусть задана функция $\underline{\mu}(x, s, \underline{\xi})$ со значением 0 или I, называемая сигналом включения целевого условия, а также вещественная функция $F(x, s, \underline{\xi})$. Целевым условием (ЦУ) будем называть условие: если $\underline{\mu}_t = \underline{\mu}(x_t, s_t, \underline{\xi}) = 1$, то $F(x_{t+1}, s_{t+1}, \underline{\xi}) > 0$. Будем говорить, что ЦУ выполнено в момент $t+1$, если либо $\underline{\mu}_t = 1$ и $F(x_{t+1}, s_{t+1}, \underline{\xi}) > 0$, либо $\underline{\mu}_t = 0$. (Отметим, что $\underline{\mu}_t = 0$ означает, по существу, что целевое условие не поставлено). Целевое условие указанного типа будем называть одношаговым. Одношаговое ЦУ, будучи поставлено в момент t, должно быть выполнено в следующий момент $t+1$. Относительно многошаговых целевых условий см. ниже раздел 6°.

Будем считать заданными: сенсорное уравнение

$$\underline{\sigma}_t = \underline{\sigma}(x_t, s_t, \underline{\xi}) \qquad (1)$$

(определяющее то, что "видит" АС), моторное уравнение

$$x_{t+1} = X(x_t, u_t, \underline{\xi}) \qquad (2)$$

(определяющее движение АС), а также уравнение изменения среды

$$s_{t+1} = S(x_t, s_t, \underline{\xi}). \qquad (3)$$

Подлежат определению следующие "уравнения мозга" АС:

$$u_t = U(\underline{\sigma}_t, \underline{\tau}_t), \qquad (4)$$

$$\underline{\tau}_{t+1} = \underline{T}(\underline{\sigma}_t, \underline{\sigma}_{t+1}, \underline{\tau}_t). \qquad (5)$$

При заданных $x_0, s_0, \underline{\tau}_0$ уравнения (I)-(5) позволяют последовательно найти значения всех указанных переменных во все моменты времени. При этом для каждого $t = 1, 2, \ldots$ ЦУ будет выполнено или нет. Подчеркнем, что правые части уравнений (I)-(3) (в отличие от уравнений (4),(5)) зависят, вообще говоря, от варьируемых параметров. Изменение варьируемых параметров $\underline{\xi} \in M$ означает изменение задачи по выполнению целевого условия или изменение условий, при которых решается фиксированная задача. (Изменение моторного и сенсорных уравнений означает изменение в процессе эксплуатации характеристик АС, изменение функций $S, \underline{\mu}, F$ имеет место при изменении задач, решаемых АС). Начальные значения $x_0, s_0, \underline{\tau}_0$ могут также зависеть от варьируемых параметров.

Если определено все указанное выше, то будем говорить, что АС задана. АС называется раузмной в классе задач M, если для любых значений варьируемых параметров $\underline{\xi} \in M$ найдется момент t_0 такой, что для всех $t \geq t_0$ будет выполнено целевое условие и $\underline{\tau}_t = Const$ при $t \geq t_0$ [x].

После всех этих фориальных определений можно точно поставить задачу о построении "разумной" (в достаточно условном и ограни-

---

x) Требование $\underline{\tau}_t = Const$ при $t \geq t_0$ может быть отброшено. Оно, однако, по ряду соображений очень удобно, и, кроме того, оно автоматически выполняется для полученного решения.

ченном смысле) системы. Эта задача состоит в построении (по заданному классу M и заданным функциям $\underline{\mu}, F, \underline{\sigma}, X, S$)уравнений мозга (4),(5), таких, чтобы АС стала разумной системой в классе задач M (разумной в указанном выше смысле).

2°. Основные предположения и основной результат. Будем обозначать через R_n эвклидово пространство размерности n. Предположим, что $\{\underline{\sigma}\}$ -замкнутое ограниченное множество некоторого R_n, $\underline{\sigma} = \|\underline{\sigma}_j\|_{j=1}^{n}$. Будем считать выполненными следующие четыре условия:

(I) Можно ввести новые управления V, где $\{V\}$ -ограниченно множество некоторого R_q, $V = \|V_j\|_{j=1}^{q}$, так, что $u = u(v)$ - однозначные функции и так, что ЦУ в момент $t+1$ заведомо выполнено, если выполнены K неравенств

$$|(c_j, V_t) - \underline{\varphi}_t^j| < \underline{\varepsilon}_j, \quad j = 1, \ldots, K \qquad (6)$$

где $\underline{\varepsilon}_j$ -параметры, $c_1, \ldots, c_K$ -линейно независимые известные векторы и $\underline{\varphi}_t^j = \underline{\varphi}_j(x_t, x_{t+1}, s_t, s_{t+1}, \underline{\xi})$ -некоторые функции указанных аргументов.

(II). Существует функция $v = V^{\mu}(\underline{\sigma}, \underline{\xi})$, называемая идеальным управлением, такая, что для любых x_t, s_t и $\underline{\xi} \in M$ при $V_t = V^{\mu}(\underline{\sigma}_t, \underline{\xi})$ выполнено (6) с заменой $\underline{\varepsilon}_j$ на какие-либо $\underline{\varepsilon}_j^* < \underline{\varepsilon}_j$. При этом в (I) $\underline{\sigma}_t, x_{t+1}, s_{t+1}$ определяются согласно естественной цепочке соотношений:

$$\underline{\sigma}_t = \underline{\sigma}(x_t, s_t, \underline{\xi}), u_t = u(V_t), x_{t+1} = X(x_t, u_t, \underline{\xi}), s_{t+1} = S(x_t, s_t, \underline{\xi})$$

(III) Каковы бы ни были управления V_t значение φ_t^j может быть выражено через $V_t, \underline{\sigma}_t, \underline{\sigma}_{t+1}$, т.е. $\underline{\varphi}_t^j = \underline{\Phi}_j(V_t, \underline{\sigma}_t, \underline{\sigma}_{t+1})$, где $\underline{\Phi}_j$ -некоторые функции.

(IV). Для всех $\underline{\xi} \in M$, $\underline{\sigma} \in \{\underline{\sigma}\}$ существуют $\partial V^{\mu}/\partial \underline{\sigma}_j$ и $|V^{\mu}(\underline{\sigma}, \underline{\xi})| \leq Const$, $|\partial V^{\mu}(\underline{\sigma}, \underline{\xi})/\partial \underline{\sigma}_j| \leq Const$.

Поясним эти предположения. Условие (I) означает, что, во-первых, ЦУ требует, чтобы "что-то от чего-то отличалось достаточно мало",

и,во-вторых, чтобы это "что-то" линейно зависело от новых управлений. Условие (П), грубо говоря, равносильно принципиальной возможности решения задачи. (Непосредственно воспользоваться управлением $v_t = V^{II}(\underline{\sigma}_t, \underline{\xi})$, разумеется, невозможно, так как неизвестны значения варьируемых параметров $\underline{\xi}$). Условие (Ш) требует, чтобы ошибка в момент t могла быть измерена по данным в моменты t и $t+1$. Условие (IУ) практически не ограничительно.

Теорема I. <u>При выполнении условий (I)-(IУ) могут быть построены уравнения мозга так, чтобы полученная адаптивная система стала разумной в классе задач M.</u>

Доказательство этой теоремы конструктивно: при доказательстве получается процедура составления уравнений мозга адаптивной системы, разумной в классе M.

В приводимых ниже двух простых, но типичных примерах адаптивных систем опущены (чтобы не загромождать изложение) второстепенные детали. Можно показать, что для этих примеров, выполнены условия (I)-(IУ). Поэтому, согласно теореме I, могут быть построены уравнения мозга этих систем, так, чтобы эти системы стали разумными в указанных ниже классах задач. Уравнения мозга этих систем построены; они не приводятся здесь так как, во-первых, для этого требовалось бы значительно более детальное описание этих примеров, и так как, во-вторых, уравнения мозга этих систем имеют достаточно громоздкий вид.

<u>3°. Адаптивная система "Кузнечик" (К).</u> Внешними координатами К являются $x = \|z, \varphi\|$, где z -комплексное число ($|z| < L$) (определяющее декартовы координаты К), а "курсовой" угол φ (изменяющийся в пределах $0 \le \varphi < 2\pi$) определяет ориентацию К. Среда λ отождествляется с комплексным числом λ (координаты цели), $|\lambda| \le L$. Число L -варьируемый параметр.

Системой координат К будем называть систему с центром в точке z, повернутую на угол φ. К видит ориентир в начале неподвижной системы координат и цель. Точнее, сенсорами $\sigma = \|\xi_0, \psi_0, \xi, \psi\|$ являются следующие величины, связанные с координатами цели и ориентира в системе координат кузнечика:

$$\xi_0 = \delta(|z|+\nu)^{-1},\ \psi_0 = \arg z - \varphi,\ \xi = \delta(|z-s|+\nu)^{-1},\ \psi = \arg(s-z) - \varphi.$$

Здесь $\delta > 0,\ \nu > 0$ -параметры. Движение кузнечика осуществляется так. Кузнечик поворачивается на угол f_t, затем прыгает на расстояние ζ_t. Поэтому управлениями являются $u = \|f_t, \zeta_t\|$, а уравнения движения имеют вид $\varphi_{t+1} = \varphi_t + f_t$, $z_{t+1} = z'_{t+1}$, где $z'_{t+1} = z_t + \zeta_t \exp i(\varphi_t + f_t)$, если только $|z'_t| \le L$. Если $|z'_t| > L$ (что означает, что кузнечик "хочет выпрыгнуть из круга $|z| \le L$"), то z_{t+1} определяется из условий "прилипания" к стенке $|z| = L$ или "отражения" (по некоторому закону) от этой стенки. Сигнал включения ЦУ: $\mu_t = 1$, если $|z_t - s_t| \ge \varepsilon$ и $\mu_t = 0$, если $|z_t - s_t| < \varepsilon$. ЦУ состоит в требовании поймать цель в следующий момент, если она сейчас не поймана: если $\mu_t = 1$ то $|z_{t+1} - s_{t+1}| < \varepsilon$. (Число ε -параметр, $\varepsilon < L$). Таким образом, кузнечик должен прыгнуть в ε -окрестность той точки, где окажется в следующий момент цель. Цель видит ориентир и кузнечика, и ее перемещение зависит от того, где она их видит. Предположим, что цель не обращает внимание на ориентацию кузнечика: $s_{t+1} = S(s_t, s_t - z_t, \xi)$. Здесь $\xi \in M$ - варьируемый векторный параметр. Если цель поймана в некоторый момент $(\mu_t = 0)$, то в следующий момент в круге $|s| \le L$ появляется квазислучайным образом новая цель с тем же законом функционирования. Поскольку цель видит лишь ориентир, а не связанную с ней систему координат, то функция $S(s, w, \xi)$

должна удовлетворять следующему условию:
$S(e^{ix}s, e^{ix}w, \xi) = e^{ix}S(s,w,\xi)$,где x -любое вещественное число.Кроме того, будем считать, что функция S и ее производные по $Re\,s, Im\,s, Re\,w, Im\,w$ ограничены при $|s| \le L$, $\varepsilon \le |w| \le 2L$ равномерно по $\xi \in M$ При этих условиях выполнены предположения (I)-(IУ) раздела 2^0, и, следовательно, могут быть пострроены уравнения мозга, так, чтобы система К стала разумной в указанном классе задач.

Разумность этой системы означает, что для любого фиксированного типа цели (т.е. для любого фиксированного $\xi \in M$)кузнечик в процессе преследования одной цели или, может быть, нескольких целей как бы выясняет для себя возможную реакцию цели этого типа и с некоторого момента начинает ловить любую цель этого типа за один такт. Если после этого появляется цель другого типа (с новым законом функционирования, т.е. с любым другим $\xi \in M$),то, естественно, кузнечик сначала не сможет поймать новую цель за один такт.Однако, в процессе преследования целей этого типа он "изучит" их реакцию, начнет верно предугадывать прыжок цели и с некоторого момента начнет ловить любую цель второго типа за один такт.

Указанное самообучение кузнечика будет иметь место для любых целей класса M .

4^0.Адаптивная система "глаз-рука".(ГР) .Внешними координатами ГР является пара комплексных чисел z, z' ,связанных соотношениями $|z| = l, |z - z'| = l'$, где $l > 0, l' > 0$ -варьируемые параметры.(Вектор z -"плечо", вектор $(z - z')$ - "предплечье", а точка z' -"конец руки").Среда отождествляется с парой комплексных чисел s', s'' ,где $|s' - s''| = \delta$, $|s''| \le l + l' - \delta_0$.Здесь δ_0 -параметр, δ -варьируемый параметр.Числа s', s'' определяют концы отрезка "объекта"; который

"видит" ГР .Пусть"глаз" находится в точке a ,(комплексное число a -параметр).ГР "видит" точки z', s', s'' .Точнее, пусть сенсорами являются $\sigma=\|\xi,\eta,\xi',\eta',\xi'',\eta''\|$, где $\xi=\delta[|z'-a|+\nu]^{-1}, \eta=\arg(z'-a), \xi', \eta', \xi'', \eta''$ определяются аналогичным образом по s', s'' , а $\delta>0, \nu>0$ - параметры.Пусть φ - угол, образованный плечом с фиксированным направлением, например, $\varphi=\arg z$ и ψ -угол между продолжением плеча и предплечья.Движение ГР осуществляется установкой заданных значений φ и ψ .Следовательно, φ, ψ - управления, а $z_{t+1}=le^{i\varphi_t}, z'_{t+1}=le^{i\varphi_t}+l'e^{i(\varphi_t+\psi_t)}$ -моторные уравнения.Сигнал включения ЦУ: $\mu_t=1$,если $|z'_t-s'_t|\geq\varepsilon$ и $\mu_t=0$, если $|z'_t-s'_t|<\varepsilon$ ЦУ: если $\mu_t=1$ то $|z'_{t+1}-s'_t|<\varepsilon$.Таким образом, адаптивная система ГР должна следить концом z' "руки" за точкой s' , предугадывая ее положение в следующий момент.Будем считать, что "объект" $s_t=\|s'_t, s''_t\|$ "видит" лишь конец руки z'_t ,т.е., что уравнения изменения среды имеет вид $s_{t+1}=S(s_t, z'_t, \xi)$, причем функция S имеет производные по $\mathrm{Re}\, s'_t, \mathrm{Im}\, s'_t, \ldots, \mathrm{Im}\, z'_t$ и вместе с этими производными ограничена равномерно по $\xi\in M$ когда s'_t, s''_t, z_t меняются в указанных выше пределах.При этих условиях выполнены предположения раздела 3^0, и,следовательно, могут быть построены уравнения мозга так,чтобы система ГР, стала разумной в классе M.

Как и выше, разумность этой системы означает, что для любого фиксированного закона движения объекта из класса M система ГР наблюдая за движением объекта и его реакциями на приближающуюся руку начинает предугадывать следующее положение объекта и начинает,как и требуется, "ловить" точку s',т.е.устанавливать конец руки z' в ε-окрестность той точки, в которую попадет в следующий момент точка s'.При изменении закона движения объекта система ГР начинает сама "переучиваться", и после конечного про-

межутка времени обучения снова начинает правильно предугадывать реакцию объекта и ловить точку s'. Это самообучение имеет место для любых законов движения объекта из класса M.

5°. Эксперименты на ЭВМ. Доказательство теоремы I, как было отмечено выше, конструктивно: при выполнении условий (I)-(IУ) указывается процедура уравнения мозга. Вместе с тем остается открытым вопрос о длительности времени обучения t_0. В общем случае вряд ли могут быть получены точные оценки числа t_0. Если бы значения t_0 оказались чрезвычайно большими в реальных случаях, то ценность построенных уравнений мозга была бы сомнительной. Для того, чтобы определить значение t_0 в реальных случаях, а также для того, чтобы определить влияние различных параметров на время обучения было проведено моделирование на ЭВМ адаптивных систем типа "кузнечик" для $L = 15 \div 500$, $\varepsilon = 1 \div 10^{-2}$, $\varepsilon/L = 0{,}067 \div 0{,}002$. Класс M содержал от 20 до 40 варьируемых параметров. Кузнечик считался обученным в момент t_0, если он ловил любую квазислучайным образом появляющуюся цель за один такт (что и требуется) для любого t в интервале $t_0 \le t \le t_0 + 10^4$. Значения времени обучения t_0 оказались вполне удовлетворительными. Если, для наглядности, считать, что один такт длится одну секунду, то время обучения t_0 кузнечика изменялось для различных значений параметров в пределах от одной минуты до нескольких часов.

6°. Некоторые замечания. I) Методика построения уравнений мозга адаптивных систем с описанными выше одношаговыми ЦУ переносится в ряде случаев на многошаговые ЦУ. Целевое условие называется многошаговым, если либо оно связывает переменные s_t, x_t для нескольких моментов времени, либо, если оно, будучи поставлено в момент t должно быть выполнено в некоторый момент $t' > t$.

Если, например, ввести ограничения на скорость передвижения кузнечика или на скорости перемещения плеча и предплечья, а также заменить требование поймать цель в следующий момент $t+1$ требованием существоания момента $t' > t$, для которого цель будет поймана (в прежних смыслах), то получается многошаговое ЦУ. Целевые условия этого типа могут быть снова сведены к одношаговым, смысл которых заключается в правильном предсказании поведения цели в следующий момент. Для систем с ЦУ этого типа могут быть построены уравнения мозга, так, чтобы эти системы стали разумными в указанном выше смысле.

Можно, однако, привести примеры систем с многошаговыми (и даже с одношаговыми) ЦУ для которых, хотя и удается свести задачу построения уравнений мозга к некоторым математическим задачам, но решение этих задач остается неизвестным.

2)В разделах $2^0, 3^0, 4^0$ были сделаны самые общие предположения о классе M .На самом деле среднее время обучения сильно зависит от объема класса M .Как правило, среднее время обучения t_0 увеличивается с увеличением класса M,т.е., точнее,с увеличением числа варьируемых параметров.Таким образом, если системы A' и A''-разумные, соответственно, в классах M' и M'', причем класс M' соответствует более широкому классу задач $(M' > M'')$, то в среднем, любую задачу из более узкого класса M'' "более разумная" система A' решает медленее, чем "менее разумная система" A'' (При расширении анализируемых возможностей как бы появляется некоторая нерешительность).Высказанное утверждение,впрочем, допускает ряд исключений.

3)В разделе 1^0 и для примеров разделов $3^0, 4^0$ мозг адаптивных систем должен был решать одновременно две задачи: задачу предсказания и задачу выработки сигналов управления для осуществления нужного движения. (Общая схема раздела I позволяет также рассмат-

ривать эти задачи раздельно). Подчеркнем, что решение даже только второй задачи представляет серьезные трудности. Рассмотрим, например, задачу решаемую мозгом системы ГР в случае, когда цель неподвижна. (Но положение цели может быть различным). Задача, таким образом, состоит в создании определенной совокупности сигналов для того, чтобы переместить конец руки в заданную точку. Можно наглядно представить себе k ручек ($k=20\div40$), положения которых определяют некоторую тактику. При фиксированном положении этих ручек осуществляется некоторое движение, и это движение зависит от того, что видит глаз. Существуют такие положения ручек, при которых осуществляется верное движение, т.е. рука ловит концом z' точку s', где бы эта точка s' не находилась. Эти "верные" положения ручек зависят, однако, от ряда неизвестных мозгу факторов, в частности, от длин плеча и предплечья (являющихся, по условию, варьируемыми параметрами). Задача мозга – найти эти верные положения ручек. После того как движение осуществилось, в мозг ГР поступает информация о величине ошибки. (Глаз видит точку s' и конец руки z'). По этим данным мозг должен изменить положения k ручек. Затем осуществляется новое движение и новое изменение положения ручек. После того, как рука "схватила" точку s' процесс продолжается снова для нового исходного положения руки, и нового положения точки s'. Разумность системы ГР в рассматриваемом классе M варьируемых параметров означает, что после конечного числа попыток мозг найдет верные положения ручек (верную тактику), при которых после появления цели s' в любом месте рука из любого положения сразу (за один такт) перемещается правильно, "схватывая" концом z' точку s'. (Разумеется точка s' должна быть в пределах досягаемости). При изменении длин плеча и предплечья изменяются верные положения ручек и после конечного числа

попыток мозг снова их находит.

Для описанного выше класса задач M отсутствовала необходимость правильного предсказания. В более широких классах M, когда "объект" $S = \|S', S''\|$ (Эксперимент) перемещается, необходимо предсказание его положения. Оно тем более необходимо, если объект реагирует на приближение руки. Изложенное выше означает, что в рамках указанной в разделе 4^0 идеализации может быть построен мозг так, чтобы система ГР решала "сама" все задачи указанного типа.

Цитированная литература.

I. Цыпкин Я.З. – Адаптация, обучение и самообучение в автоматических системах. Автоматика и телемеханика, т.27, № I, 1966.

ON THE ALGORITHM OF LEARNING WITH ACCUMULATION OF EXPERIENCE IN OPTIMUM CONTROL

Ing. Dr. Štefan PETRÁŠ
Assistant Professor
INSTITUTE OF ENGINEERING CYBERNETICS
SLOVAK ACADEMY OF SCIENCE
Bratislava
Czechoslovakia

1. INTRODUCTION

In solving problems of optimum control with incomplete information on the controlled object there arises the question of how to gain an appropriate algorithm. Hitherto known control algorithms based on the deterministic principle, are not suitable, especially when the object is multidimensional and subject to perturbations. In such cases stochastic methods are adequate, based on the principle of the theory of learning.

In my paper I wich to deal with some new aspects of the algorithm of learning that would consider the entire hispory or part of the history of learning as a sequential Markhovian phenomenon of the k^{th} order. I should like to refer to the particularity of such a phenomenon that can be looked upon as a matingal of semimartingal phenomenon provided that certain assumptions are satisfied.

2. THE MATHEMATICAL FORMULATION OF THE PROBLEM

The mathematical formulation of the optimum control of quasi-stationary phenomena is defined as follows:

Assume that the optimum control /Fig. 1/ given by a purpose function, is of the form

$$Q = Q(\bar{x}) \qquad (1)$$

where Q is a scalar function independent on t

$\bar{x}$ is the vector of the controlled quantity $x_1, x_2, \ldots, x_n$.

The task of optimum control is to determine such a vector of the controlled quantity $\bar{u}$, that the corresponding vector $\bar{x}^*(x_1^*, x_2^*, \ldots, x_n^*)$ may satisfy the relation

$$\sup Q = Q_{min} = Q(\bar{x}^*) \leq Q(\bar{x}) \qquad (2)$$

where $\bar{x}^* \in X$.

Let the chosen algorithm be discrete and given by the recurrent relation

$$\bar{x}^{N+1} = \bar{x}^N + \Delta\bar{x}^{N+1} \qquad (3)$$

where

$$\Delta\bar{x}^{N+1} = \begin{cases} \Delta\bar{x}^N & \text{if} \quad Q(\bar{x}^N) < Q(\bar{x}^N) \\ a\bar{\xi} & \text{if} \quad Q(\bar{x}^N) \geq Q(\bar{x}^{N-1}) \end{cases} \qquad (4)$$

where N = 1,2,...

a is the scalar step length

$\bar{\xi}$ is the realization of the random unity vector.

The realization of the unity random vector will depend on the probability $p(\bar{w}^N)$ of the storage parameter $\bar{w}^N$. This probability will vary according to the amount of experience accumulation. If the result of the test proves to be succesful, probability $p(\bar{w}^N)$ will grow and reversely. Already R.F.Arnold [1] has pointed out that the optimal strategy of the learning process depends not only on the immediate success or failure but on the entire history.

What is the substance of this assertion?

1. An arbitrary learning and simultaneously optimal system must have controlling signals - a control in arbitrary time t, determined on the basis of all observations gained uptill this time t.

2. The concept " information gained uptill time t" must be explained.

As a matter of fact two interesting cases may occur:

a/ when the processes in question are essentially ergodic, hence such phenomenon can be considered as a simple Markhovian process, as a sequential string, i.e.

evolution, the future of which depends solely on its behaviour in a given moment of time, i.e. is independent on the behaviour of the system in the past. Graphically this means as it is shown in Fig. 2, where ◯ determines the state of the system or the direction of the random vector, respectively, ▭ determines the performed test in the relevant step.

It can be seen from the abovesaid that the subsequent step is immediately determined only by the immediate state, i.e. for example by the k^{th} state, if the k+1 step is determined.

Simple sequential Markhovian processes are defined among others by conditioned probabilities

$$P\left\{\xi(t)\in A \mid \xi(t_1), \xi(t_2), \ldots, \xi(t_n)\right\} = P\left\{\xi(t)\in A \mid \xi(t_n)\right\}$$

for $t_1 < t_2 < \ldots < t_n < t$;

b/ another learning process is, however, also possible. It gives, in my opinion, a much truer picture of the actual process of learning, i.e. the working or operational step is determined by information from the preceding step, e.g. a complicated Markhovian process, for instance of the k^{th} order with complete linkages. It is true that it has been rather difficult in concrete cases to analytically express e.g. the transfer time for the system from one state into another, on the other hand the state or the step can be very expediently expressed by means of the Bayesian decision. The graphical representation is given in Fig. 3 and the four-state process shown in Fig. 4.

Stochastic processes with complete linkage represent such processes, in which the conditioned probability of the subsequent states depends on all preceding states, in our case the subsequent step is determined by all preceding steps.

$$P\left\{\xi(t)\in A \mid \xi(t_1), \xi(t_2), \ldots, \xi(t_n)\right\}$$

If a finite Markhovian process of the k^{th} order is in question, or if a finite number of steps is assumed, then the complex Markhovian process can be expressed by an ordinary Markhovian process having, however, a greater number of states.

Another approach to the solution of this problem was dealt with by authors [2][5][7].

In my opinion the realization of the random vector will, among others, depend on the conditioned probability

$$p(w) = p\left(w_i^{N+1} \mid w_i^N, w_i^{N-1}, \ldots, w_i^1\right) \qquad (5)$$

where $0 \leq p(w) \leq 1$.

The conditioned probability according to Bayes is given by the relation

$$p\left(w_i^{N+1} \mid w_i^N, \ldots, w_i^1\right) = \frac{p\left(w_i^{N+1}\right) \cdot p\left(w_i^1, w_i^2, \ldots, w_i^N \mid w_i^{N+1}\right)}{\sum_N p\left(w_i^1, w_i^2, \ldots, w_i^{N-1}, w_i^N\right)} \qquad (6)$$

where $p\left(w_i^{N+1}\right)$ and $p\left(w_i^1, w_i^2, \ldots, w_i^N\right)$ is the ápriori probability and $p\left(w_i^1, w_i^2, \ldots, w_i^N, w_i^{N+1}\right)$ at known values of $w_i^1, w_i^2, \ldots, w_i^N$ is the function of only w_i^{N+1}, that is $L\left(w_i^{N+1}\right)$.

This function is a probable function and permits the application of the maximal aposteriorial probability principle consisting in the fact that such value of w_i^{N+1}, is most probable for which the function of probability $L\left(w_i^{N+1}\right)$ has a maximum.

The increment of the purpose function

$$\triangle Q^{N+1} = Q\left(\bar{x}^{N+1}\right) - Q\left(\bar{x}^N\right) \qquad (7)$$

is looked upon as the measure of success.

It has been said that the realization of $\bar{\xi}$ will depend on the probability

$$p_i^{N+1} = p\left(w_i^{N+1} \mid w_i^N, \ldots, w_i^1\right) \qquad (8)$$

and

$$w_i^{N+1} = w_i^N - \tilde{\sigma} \operatorname{sign} \left[\Delta x_i^N \cdot \Delta Q^N \right]$$

where $\tilde{\sigma} > 0$ is the parameter of learning speed, or

$$p_i^{N+1} = p\left(w_i^{N+1}\right)$$

and (9)

$$w_i^{N+1} = w_i^N - \tilde{\sigma}\left[p\left(w_i^{N+1} \mid w_i^N, w_i^{N-1}, \ldots, w_i^1\right)\right] \operatorname{sign} \left[\Delta x_i^N \cdot \Delta Q^N \right]$$

It can be seen that the determination of optimum control at a given algorithm by the method of learning with the accumulation of experience is, as a matter of fact, the determination of such random sequence $Q(\bar{x}^1)$, $Q(\bar{x}^2)$,..., $Q(\bar{x}^k)$, which converges to $Q(\bar{x}^*)$, or to its close neighbourhood, respectively.

The sequence of random values $Q(\bar{x}^1)$, $Q(\bar{x}^2)$,..., $Q(\bar{x}^k)$ is the function of the random sequance of vectors having the form

$$\begin{aligned}
&\bar{x}^1 = \bar{x}^0 + \Delta \bar{x}^1 \\
&\bar{x}^2 = \bar{x}^1 + \Delta \bar{x}^2 = \bar{x}^0 + \Delta \bar{x}^1 + \Delta \bar{x}^2 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\vdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\bar{x}^{k+1} = \bar{x}^k + \Delta \bar{x}^{k+1} = \bar{x}^0 + \Delta \bar{x}^1 + \ldots + \Delta \bar{x}^{k+1}
\end{aligned} \qquad (10)$$

where $\bar{x}^0$ is the initial vector.

However, the increments of the separate vectors $\Delta \bar{x}^j$ are functions of probability, i.e.

$$\Delta \bar{x}^j = \bar{f}(p_i^j) = \bar{f}\left[p(w_i^j) \right]$$

$$\Delta \bar{x}^j = f\left[p(w_i^j \mid w_i^{j-1}, \ldots, w_i^1) \right] \qquad (11)$$

respectively,

$i = 1,2,\ldots,n$

$j = 1,2,\ldots,k+1.$

In addition to the relation (8) and (9), respectively, referred to above, the algorithm of learning can be applied both as shown in [6].

3. THE SOLUTION OF THE PROBLEM AS A MARTINGAL PROCESS

If our system and the process are such that the relation

$$M\left\{\left|Q(\bar{x}^j)\right|\right\} < \infty$$

and

$$M\left\{\left|Q(\bar{x}^j)\right|\right\}^2 < \infty \qquad (12)$$

respectively,

$j = 1,2,\ldots,k+1$

holds and if it is tried to investigate the process from the integral point of view, that is to find how the probability P of such a complex phenomenom is distributed, then it is found that such process is a martingal one.

According to relation (10) one can put down [4]

$$P_i\left[x_i^1, x_i^2, \ldots, x_i^{k+1}\right] = P_i\left[\Delta x_i^1(x_i^1), \Delta x_i^2(x_i^1, x_i^2) \cdot \ldots \cdot \Delta x_i^{k+1}(x_i^1, x_i^2, \ldots, x_i^{k+1})\right] \qquad (13)$$

where $i = 1,2,\ldots,n$.

Let $n = k+1$, then by relation (10) and (11) we can put down

$$P_i\left[x_i^1, x_i^2, \ldots, x_i^{k+1}\right] = P_i^1(x_i^1 - x_i^0) \cdot P_i^2(x_i^2 - x_i^1) \cdot \ldots \cdot P_i^{k+1}(x_i^{k+1} - x_i^k) = \prod_j^{k+1} P_i^j(x_i^{j+1} - x_i^j) = P_i^1(\Delta x_i^1) \cdot$$

$$\cdot P_i^2(\Delta x_i^2) \ldots P_i(\Delta x_i^{k+1}) = \prod_{j=1}^{k+1} P_i^j(\Delta x_i^j) = \prod_{j=1}^{k+1} P_i^j\left[f^j(p_i^j)\right] =$$

$$= \prod_{j=1}^{k+1} P_i^j\left\{f^j\left[p(w_i^j \mid w_i^{j-1}, \ldots, w_i^1)\right]\right\} \qquad i=1,2,\ldots,n \qquad (14)$$

When turning now to the conditioned probabilities, obtain

$$P_i\left[x_i^1, x_i^2, \ldots, x_i^{k+1}\right] = \prod_{j=1}^{k+1} P_i^j(x_i^j \mid x_i^{j-1}, \ldots, x_i^1) \qquad (15)$$

$i = 1,2,\ldots,n$.

A special case occurs if the phenomenon is Markhovian, then

$$P_i\left[x_i^1, x_i^2, \ldots, x_i^{k+1}\right] = \prod_{j=1}^{k+1} P_i^j\left(x_i^j \mid x_i^{j-1}\right) \qquad (16)$$

By means of relations (10) and (16) we can put down the mathematical hope of the purpose function

$$M\left\{Q\left(x_i^{k+1} \mid x_i^k\right)\right\} \equiv M\left\{Q\left(x_i^k \mid x_i^k\right)\right\} + M\left\{Q\left(\Delta x_i^{k+1} \mid x_i^k\right)\right\} \qquad (17)$$

The first term of equation (17) is

$$M\left\{Q\left(x_i^k \mid x_i^k\right)\right\} = M\left\{Q\left(x_i^k\right)\right\}$$

The second term of the equation is

$$M\left\{Q\left(\Delta x_i^{k+1} \mid x_i^k\right)\right\} = 0.$$

When Δx_i^{k+1} will not depend on x_i^k, i.e. when $\Delta x_i^{k+1} = f_i^{k+1}\left[p_i^{k+1}\right] = f_i^{k+1}\left[p\left(w_i^{k+1} \mid w_i^k, \ldots, w_i^1\right)\right]$ will not be the function of Δx_i^k, that is that the conditioned probability will not alter at the change of x_i^k to x_i^{k+1}.

In such case

$$M\left\{Q\left(x_i^{k+1} \mid x_i^k\right)\right\} = M\left\{Q\left(x_i^k\right)\right\}$$

or

$$M\left\{Q\left(x_i^{k+1} \mid x_i^k\right)\right\} = Q\left(x_i^k\right) \qquad (18)$$

with probability 1.

By generalizing the relations given above, we obtain

$$M\left\{Q\left(x_i^{k+1} \mid x_i^k, x_i^{k-1}, \ldots, x_i^1\right)\right\} = Q\left(x_i^k\right) \qquad (19)$$

with the probability 1, for i = 1,2,...,n.

It can be said that the algorithm of optimum control of the steady state process, expressed by the method of learning by experience accumulation is a martingal process if the following conditions are satisfied

1/ $M\left\{\left|Q\left(x_i^j\right)\right|\right\} < \infty$

2/ $p\left(w_i^{k+1} \mid w_i^k, w_i^{k-1}, \ldots, w_i^1\right) = p\left(w_i^k\right)$

$$3/\quad M\left\{ Q\left(\Delta x_i^{k+1} \mid \Delta x_i^{k-1}, \ldots, \Delta x_i^{1}\right)\right\} = 0$$

In the enclosure to the paper i give the proof of the convergence of the solution similarly as it is introduced in [9].

4. PRACTICAL RESULTS

The methods hitherto applied and this one were verified and mutually compared with the purpose function being generally of the form,

$$Q = \sum_{i=1}^{2} C_i \, e^{\sum_{j=1}^{5} a_{ij}(x_j - b_{ij})^2}$$

or the concrete model, being of the form

$$Q(x_1, x_2, x_3, x_4, x_5) = 2{,}3\, e^{-[0{,}6(x_1 - 2)^2 + 0{,}8(x_2 - 2)^2 +}$$

$$+ (x_3 - 2)^2 + 1{,}2(x_4 - 3)^2 + 0{,}7(x_5 - 2{,}5)^2] + 3\, e^{-[0{,}9(x_1 - 4)^2 +}$$

$$+ 0{,}7(x_2 - 4)^2 + 1{,}1(x_3 - 5)^2 + 1{,}3(x_4 - 4)^2 + 0{,}8(x_5 - 4{,}5)^2]$$

This function has two extremums in point $x_1 = 2$; $x_2 = 2$; $x_3 = 2$; $x_4 = 3$; $x_5 = 2{,}5$; $Q_{1\ max} = 2{,}3$. The second extremum is in point $x_1 = 4$; $x_2 = 4$; $x_3 = 5$; $x_4 = 4$; $x_5 = 4{,}5$; $Q_{2\ max} = 3$.

In Table 1 the results of the test are given.

Test 1 presents the classical method of a random test, the course of which is in Fig. 5.

Test 2 presents the method of random test with punishment, the course of which is in Fig. 6.

Test 3 presents the method of random test with the choice of the optimal result. The course is given in Fig. 7.

Test 4, 5 and 6 presents the method of random search with learning and with the uniform law of probability change. The course is in Fig. 8. The parameter of

learning speed δ and the magnitude of step "a" have changed.

Test 7, 8 and 9 presents the method of random search with learning and experience accumulation as a complex process. The course is in Fig. 9. The parameter of learning speed and the magnitude of step "a" were changing.

It can be seen from the figures that the algorithm of learning with the accumulation of experience is one applicable both for relatively sophisticated multi-parameter systems.

APPENDIX A 1

CONVERGENCE OF THE MARTINGAL PROCESS

Let the algorithm of optimum control be given by the relation

$$x^{(N+1)} = x^{(N)} - a_N \gamma_N \bar{\xi}^{(N)} \qquad (A.1)$$

where a_N - magnitude of step

γ_N - normalizing coefficient

$\bar{\xi}^N$ - random vector

if $\bar{\xi}^N$ is the unity random vector, then $\gamma_N = 1$

$x^{(1)}$ is chosen arbitrarily.

Let the following assumption be satisfied

$$\sum_{N=1}^{\infty} a_N = \infty \,, \quad \sum_{N=1}^{\infty} a_N^2 < \infty \,, \quad a_N > 0 \qquad (A.2)$$

$$M\left(\bar{\xi}^{(N)} \,\middle|\, x^{(1)}, \ldots, x^{(N)} \right) = c_N \nabla^{(N)} + m^{(N)} \qquad (A.3)$$

where $0 \leq c_N \leq C < \infty$

$m^{(N)}$ is the systematic error

$\nabla^{(N)}$ is the gradient of function $Q(x)$ in point $x^{(N)}$

$$\nabla^{(N)} = \left(\frac{\partial Q}{Qx_1}, \ldots, \frac{\partial Q}{Qx_n} \right)$$

$$\gamma_N^2 M \left(\left\| \bar{\xi}^{(N)} \right\|^2 \Big| x^{(1)}, \ldots, x^{(N)} \right) \leq k_1 < \infty \qquad (A.4)$$

where $\left\| \bar{\xi}^{(N)} \right\|^2$ is the standard of the random vector

$$\sum_{N=1}^{\infty} a_N \left\| m^{(N)} \right\| < \infty \; ; \; \sum_{N=1}^{\infty} \frac{a_N}{c_N} \left\| m^{(N)} \right\| < \infty \qquad (A.5)$$

$$D \left(\bar{\xi}^{(N)} \Big| x^{(1)}, \ldots, x^{(N)} \right) < \infty \qquad (A.6)$$

I claim that this process is a martingal or semimartingal one, respectively and the iterative process converges to the extremum with probability 1.

Proof

notation: the scalar product will denote

a/ $\langle x : y \rangle$

b/ $A(x)$ **matrix** $\dfrac{\partial^2 Q}{\partial x_i \, \partial x_j}$; $i \, . \, j = 1,2,\ldots,n.$

Procedure

Distribute $Q\left(x^{(N+1)}\right)$ **into the Taylor series**

$$Q\left(x^{(N+1)}\right) = Q\left(x^{(N)}\right) - \frac{a_N \gamma_N}{1!} \langle \nabla^{(N)} \, . \, \bar{\xi}^{(N)} \rangle +$$

$$+ \frac{a_N^2 \gamma_N^2}{2!} \langle A\left(x^{(N)} - a_N \gamma_N \bar{\xi}^{N}\right) \bar{\xi}^{N}, \bar{\xi}^{N} \rangle - \ldots \leq Q\left(x^{(N)}\right) -$$

$$- \frac{a_N \gamma_N}{1!} \langle \nabla^{(N)} \, . \, \bar{\xi}^{(N)} \rangle + \frac{a_N^2 \gamma_N^2}{2!} \, . \, K_2 \left\| \bar{\xi}^{(N)} \right\|^2$$

Introduce conditioned probabilities

$$M\left[Q\left(x^{N+1} \Big| x^{(1)}, \ldots, x^{(N)}\right) \right] \leq Q\left(x^{(N)}\right) - \frac{a_N \gamma_N}{1! \, c_N} \langle \left[M\left(\bar{\xi}^{(N)} \Big| x^{(1)}, \ldots \right.\right.$$

$$\left.\left. \ldots, x^{(N)} \right) - m^{(N)} \right] \, . \, M\left(\bar{\xi}^{(N)} \Big| x^{(1)}, \ldots, x^{(N)} \right) \rangle + \frac{a_N^2}{2!} K_2 K_1 =$$

$$= Q\left(x^{(N)}\right) - \frac{a_N \gamma_N}{1!\, c_N} \left\| M\left(\bar{\xi}^{(N)} \middle| x^{(1)},\ldots,x^{(N)}\right) \right\|^2 + \frac{a_N \gamma_N}{1!\, c_N} \cdot$$

$$\cdot \left\langle m^{(N)} \cdot M\left(\bar{\xi}^{(N)} \middle| x^{(1)},\ldots,x^{(N)}\right) \right\rangle + \frac{a_N^2}{2!} K_1 K_2$$

Due to the fact that $\| \quad \|^2$ is nonnegative and Cauchy-Bujanovský's inequality holds, it follows that

$$M\left[Q\left(x^{(N+1)} \middle| x^{(1)},\ldots,x^{(N)}\right)\right] \leq Q\left(x^{(N)}\right) + \frac{a_N \gamma_N}{1!} \left\| m^{(N)} \right\| \cdot$$

$$\cdot \left\| M\left(\bar{\xi}^{(N)} \middle| x^{(1)},\ldots,x^{(N)}\right) \right\| + \frac{a_N^2}{2!} K_1 K_2 \leq Q\left(x^{(N)}\right) + \frac{a_N}{1!\, c_N} \left\| m^{(N)} \right\| \cdot$$

$$\cdot \sqrt{\gamma_N^2\, M\left(\left\| \bar{\xi}^{(N)} \right\|^2 \middle| x^{(1)},\ldots,x^{(N)} \right)} + \frac{a_N^2}{2!} K_1 K_2 \leq Q\left(x^{(N)}\right) +$$

$$+ \sqrt{K_1}\, \frac{a_N}{1!\, c_N} \left\| m^{(N)} \right\| + \frac{a_N^2}{2!} K_1 K_2$$

due to (A.4).

We see that this last term can be substituted thus

$$z^{(N)} = Q\left(x^{(N)}\right) + \sqrt{K_1} \sum_{K=N}^{\infty} \frac{a_N}{1!\, c_N} \left\| m^K \right\| + \frac{K_1 K_2}{2!} \sum_{K=N}^{\infty} a_N^2$$

then we can write

$$M\left(z^{(N+1)} \middle| x^{(1)},\ldots,x^{(N)}\right) \leq z^{(N)}$$

If the considerations for all $z^{(i)}$ and $x^{(1)} = z^{(1)}$, are carried out, then

$$M\left(z^{(N+1)} \middle| z^{(1)},\ldots,z^{(N)}\right) \leq z^{(N)}$$

this inequality forms a semimartingal a this converges with probability 1 to the extremum, that means that

$$M\left(z^{(N)}\right) \leq \ldots \leq M\left(z^{(1)}\right) < \infty$$

hence (A.7)

$$M\left[Q\left(x^{(N)}\right)\right] < \infty$$

From this convergence ensues also the limitation $Q\left(x^{(N)}\right)$ with probability 1.

It remains to be proved that

$$P\left\{\lim_{N\to\infty} Q\left(x^{(N)}\right) = Q_{min}\right\} \quad \text{w. p. 1.}$$

Since x is limited $Q\left(x^{(N)}\right)$ is also limited and it is assumed that $Q\left(x^{(N)}\right)$ is continuous, then both $\| \nabla^{(N)} \|$ is a limited quantity, hence $\| \nabla^{(N)} \| \leq B$.

If (A.3) are satisfied, then it follows from (A.5) that $M\left(\| \tilde{\xi}^{(N)} \|^2 \mid x^{(1)},\ldots,x^{(N)} \right) < \infty$, then γ_N can always be chosen such that

$$0 < a \leq \gamma_N \leq A < \infty$$

Then

$$M\left[Q\left(x^{(N+1)}\right) \mid x^{(1)},\ldots,x^{(N)}\right] \leq Q\left(x^{(N)}\right) - \frac{a_N \gamma_N}{1!} \nabla^{(N)} \cdot$$

$$\cdot M\left(\tilde{\xi}^{(N)} \mid x^{(1)},\ldots,x^{(N)}\right) + K_1K_2 \frac{a_N^2}{2!} = Q\left(x^{(N)}\right) - \frac{a_N \gamma_N}{1!} \nabla^{(N)} \cdot$$

$$\cdot \left(c_N \nabla^{(N)} + m^{(N)} \right) + K_1K_2 \frac{a_N^2}{2!} \leq Q\left(x^{(N)}\right) - \frac{a_N \gamma_N c_N}{1!} \| \nabla^{(N)} \|^2 +$$

$$+ \frac{a_N \gamma_N}{1!} \| \nabla^{(N)} \| \| m^{(N)} \| + \frac{K_1K_2 a_N^2}{2!} \leq Q\left(x^{(N)}\right) - \frac{a_N \gamma_N a}{1!} \| \nabla^{(N)} \|^2 +$$

$$+ \frac{ABa_N}{1!} \| m^{(N)} \| + \frac{K_1K_2 a_N^2}{2!} = M(L)$$

If putting down for all N

$$M\left[Q\left(x^{(N+1)}\right)\right] \leq M\left[Q\left(x^{(N)}\right)\right] - a \sum_{K=1}^{N} a_K c_K M \| \nabla^{(N)} \|^2 +$$

$$+ M\left[AB \sum_{K=1}^{N} a_K \| m^{(K)} \| + M\left(\frac{1}{2} K_1K_2 \sum_{K=1}^{N} a_K^2 \right) \right] \qquad (A.8)$$

it will still hold

$$\sum_{K_1}^{\infty} a_K c_K M \left\| \nabla^{(K)} \right\|^2 \leq C \sum_{K=1}^{\infty} a_K M \left\| \nabla^{(K)} \right\|^2 < \infty$$

where $c_K \leq C$

due to the inequality of (A.3) since $\sum_{K=1}^{\infty} a_K = \infty$ must be

$$M \left\| \nabla^{(K)} \right\|^2 \longrightarrow 0$$

the 3rd term of the expression (A.8) $< \infty$, because

$$\sum_{K=1}^{N} a_K \left\| m^{(K)} \right\| < \infty$$

the 4th term of the expression (A.8) $< \infty$, because $\sum_{K=1}^{N} a_K^2 < \infty$

the 1st term of the expression (A.8) $< \infty$, because
$M\left[Q(x^{(N)})\right] < \infty$

i.e. we obtain the convergence of the sequance $\nabla^{(N)}$ in mean quadratic. This ensures a sufficient sequence $\nabla^{(N_K)}$ of the convergence to 0 with probability 1.

Since Q(x) is a continuous function and converges the sequence $Q\left(x^{(N)}\right)$ to the limit with probability 1 we obtain

$$P\left\{ \lim_{N \to \infty} Q(x^{(N)}) = Q_{min} \right\} \longrightarrow 1$$

However, the problem remains unsolved, if N is finite, that means the question arises when to finish the process on the computer.

LITERATURE

[1] Arnold R.F.:"A Compilier Capable of Learning". Procedings Western Joint Computer Conference, San Francisco, California, 1959, New York, 137-143.

[2] Bush R.R. and Mosteller F.: "Stochastic Models for Learning". John Wiley and Sons, New York, 1955.

[3] Doob J.L.: "Stochastic Processes". John Wiley and Sons, New York, 1953.

[4] Papoulis A.: "Probability Random Variables and Stochastic Processes". Mc Graw-Hill, New York, 1965.

[5] Растригин Л.А.: "Случайный поиск", Изд-во Зинатне, Рига 1965.

[6] Растригин Л.А.: "Поведение марковских алгоритмов случайного поиска в процессе многопараметрической оптимизации при наличии помех". Автоматика и вычислительная техника 13, Рига, 1966.

[7] Rubin A.J., Munson J.K.: "Optimization by Random Search - IRE Transactions 1959, No 2.

[8] Юдин Д.В., Хазен Е.М.: "Некоторые математические аспекты статистических методов поиска". Автоматика и вычислительная техника 13, Рига, 1966.

[9] Ермольев Ю.М., Некрылова З.В.: "О некоторых методах стохастической оптимизации". Кибернетика № 6, 1966.

[10] Ciucu G., Theodorescu R.: "Procese cu legături complete". Acad. R.P.R., Bucarest, 1960.

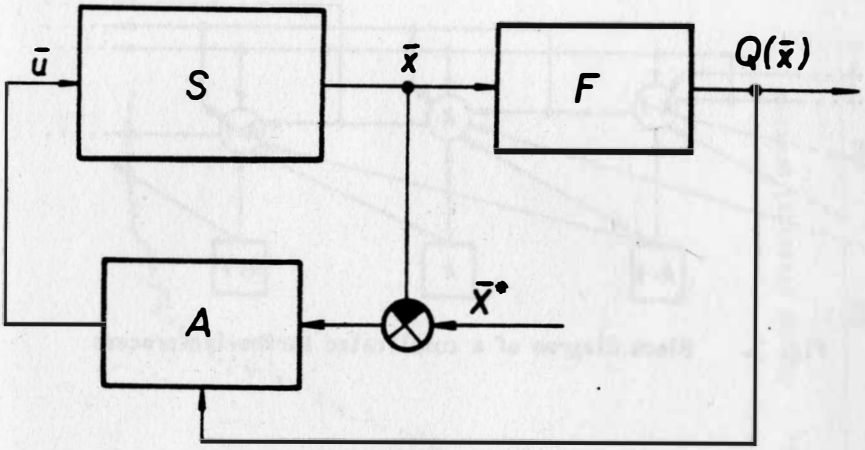


Fig. 1. Block diagram of optimum control

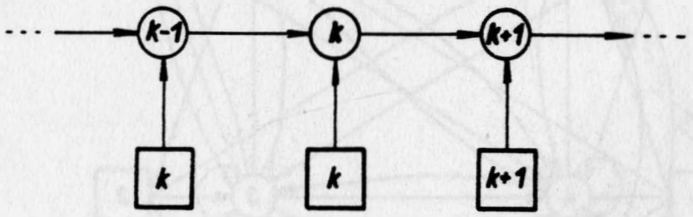


Fig. 2. Block diagram of a simple Markhovian process

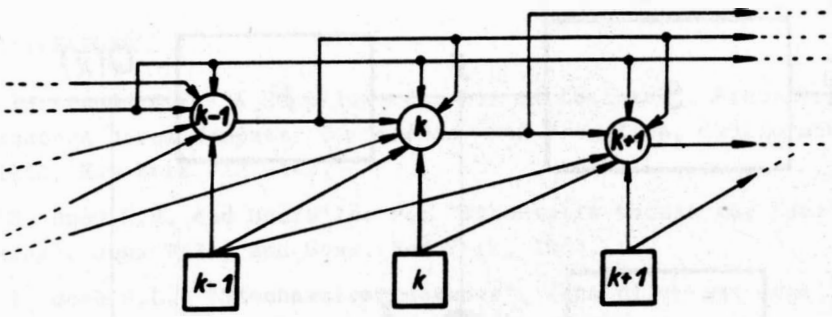


Fig. 3. Block diagram of a complicated Markhovian process

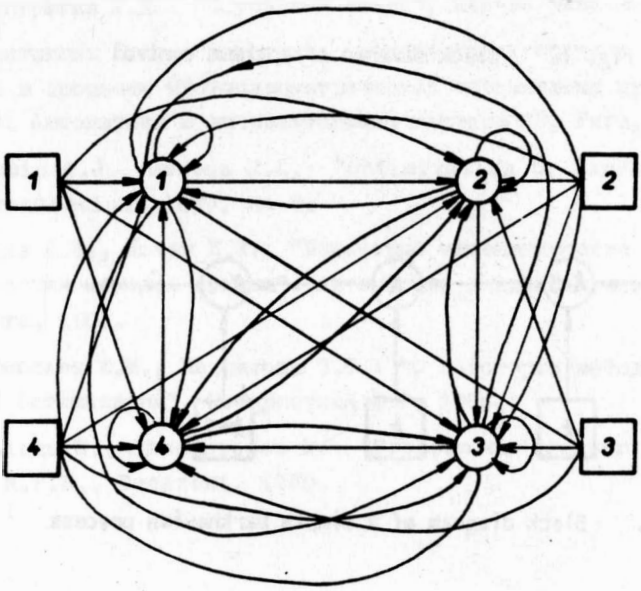


Fig. 4. Block diagram of a four-state Markhovian process

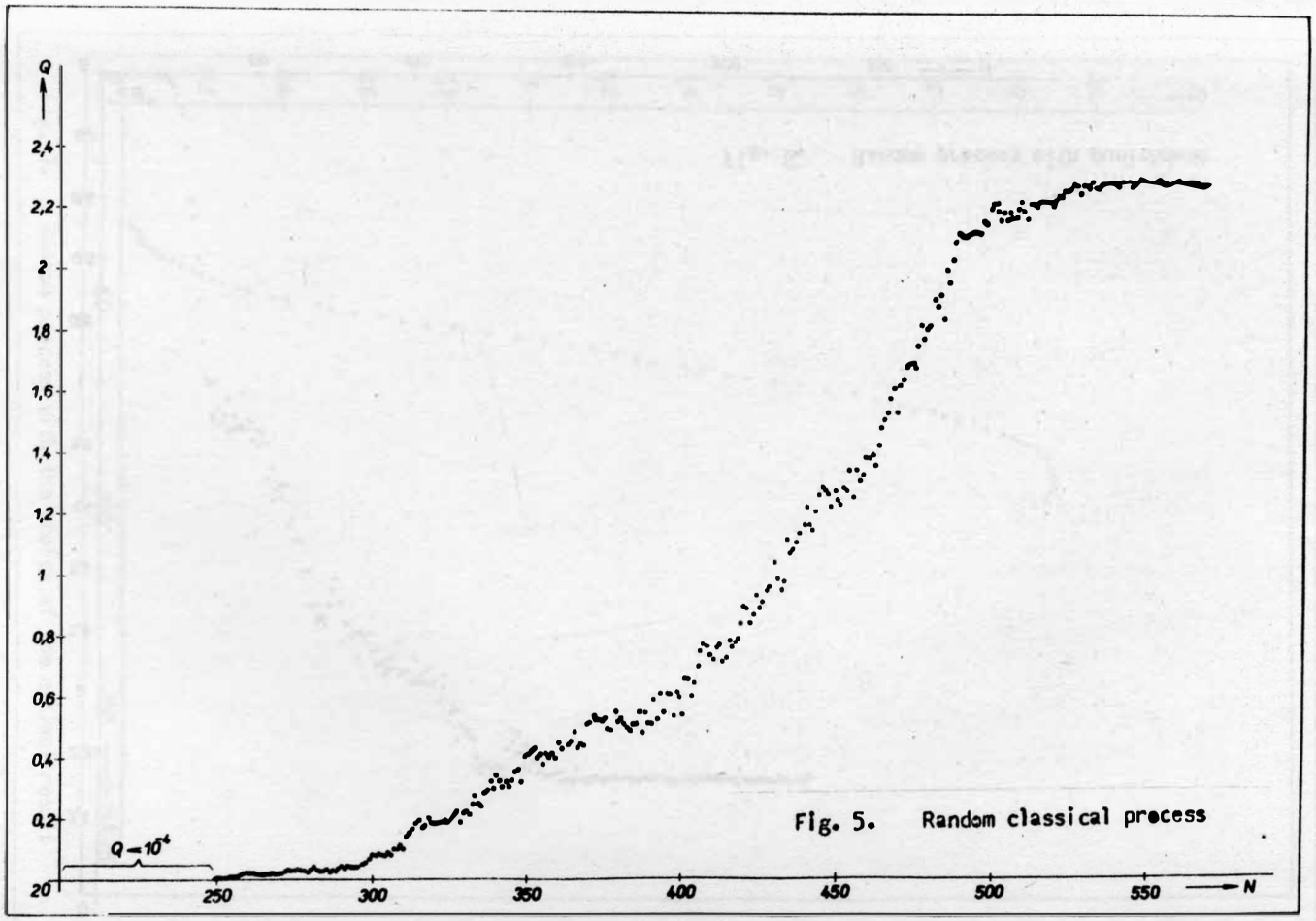


Fig. 5. Random classical process

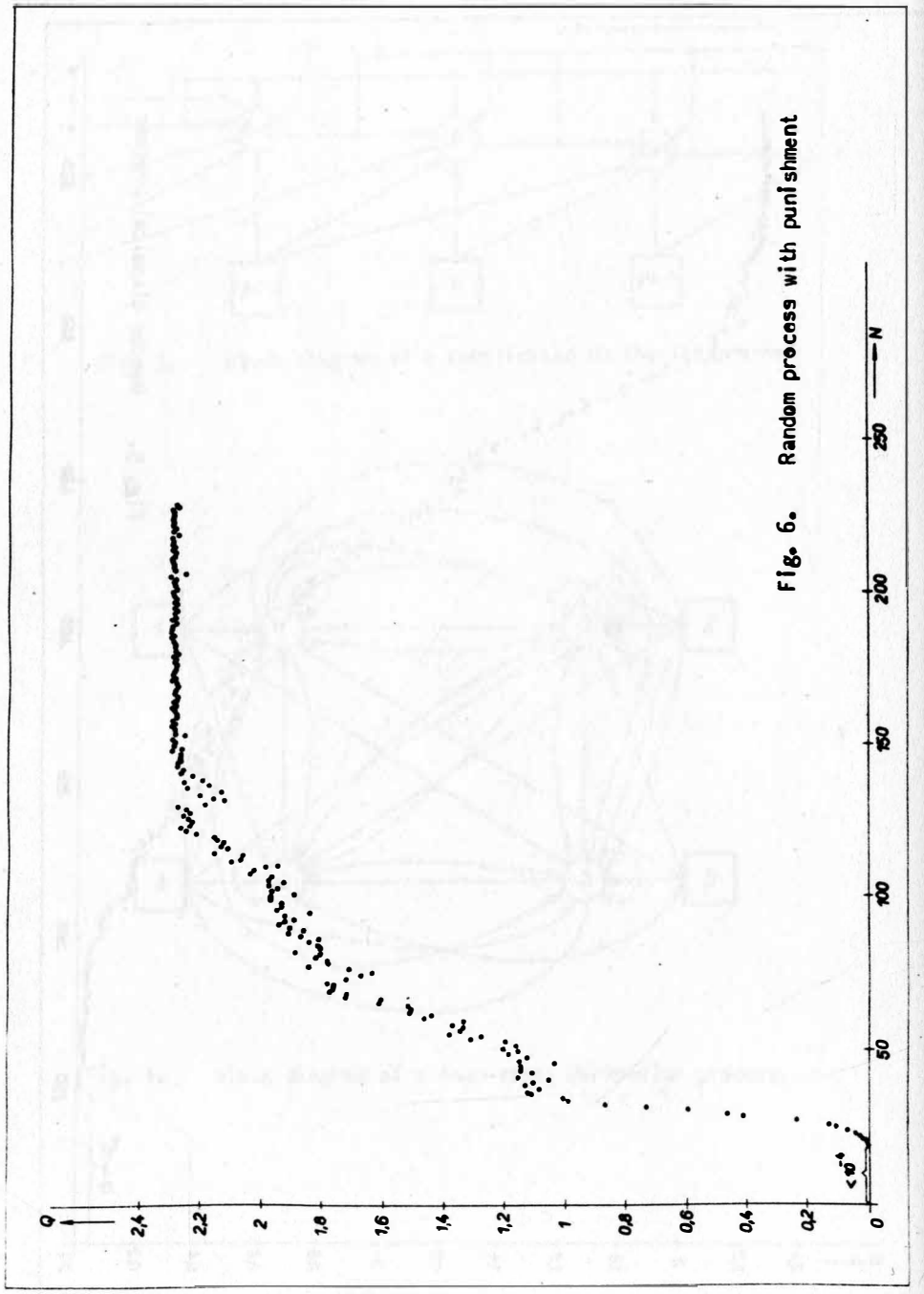


Fig. 6. Random process with punishment

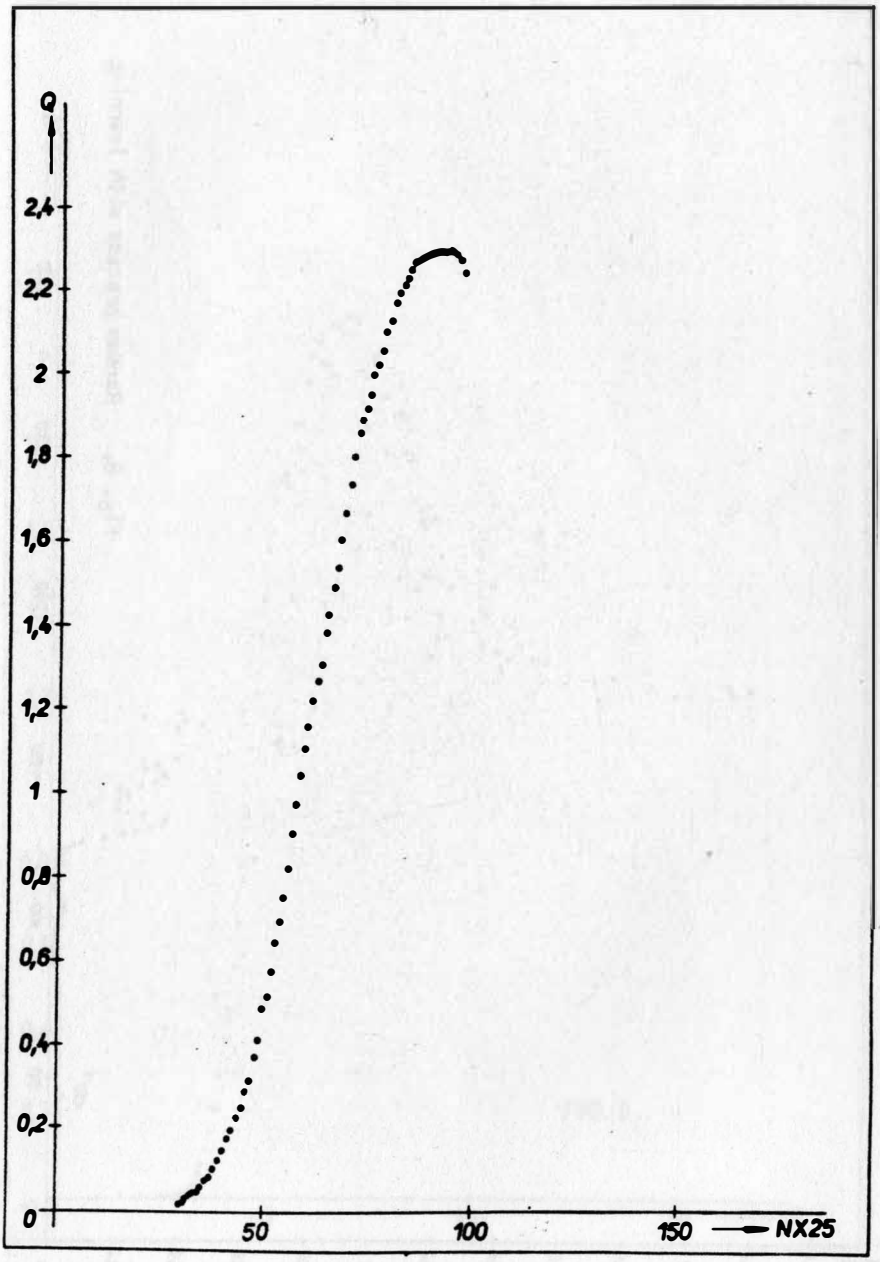


Fig. 7. Random process with the choice of the optimal result

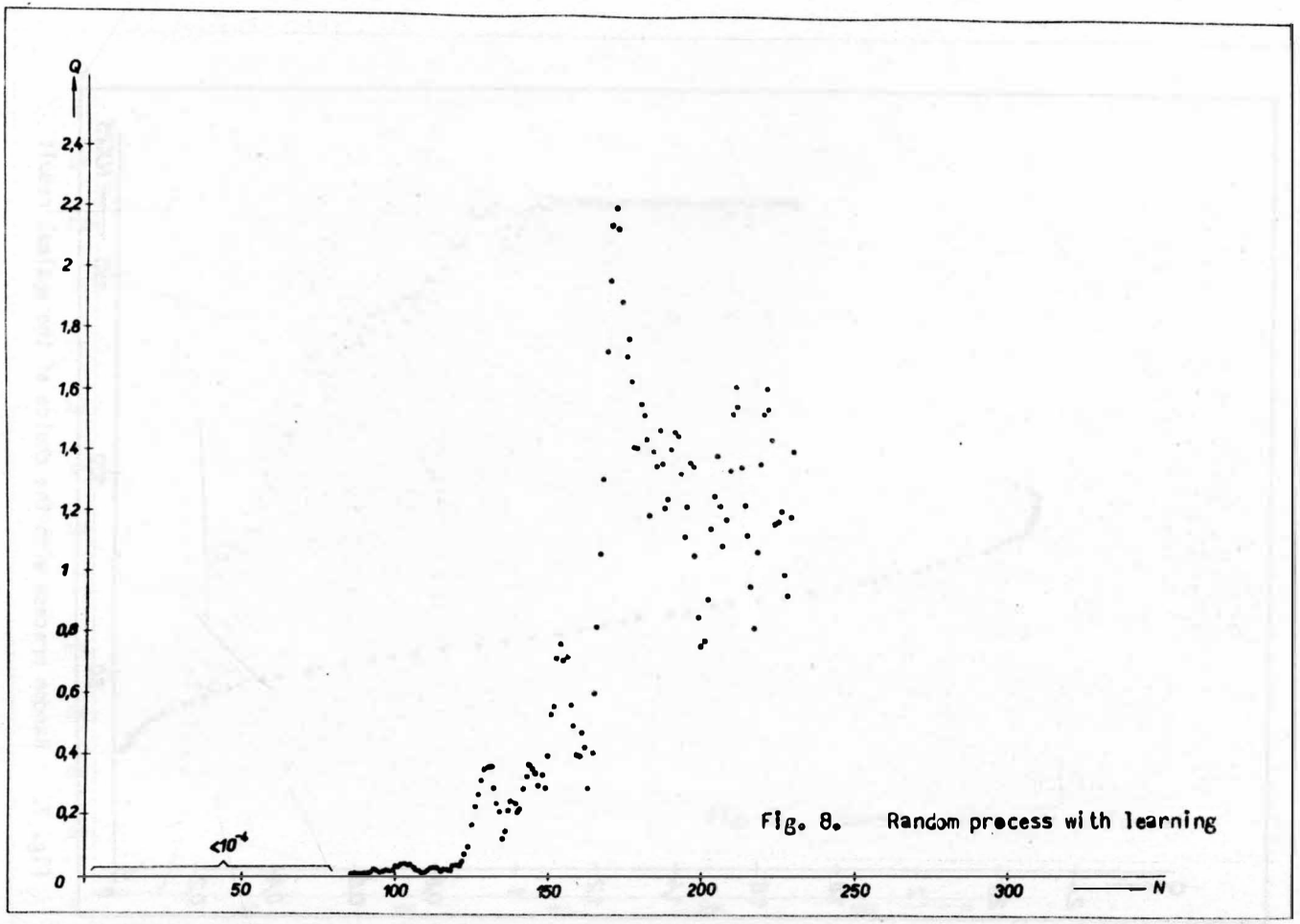


Fig. 8. Random process with learning

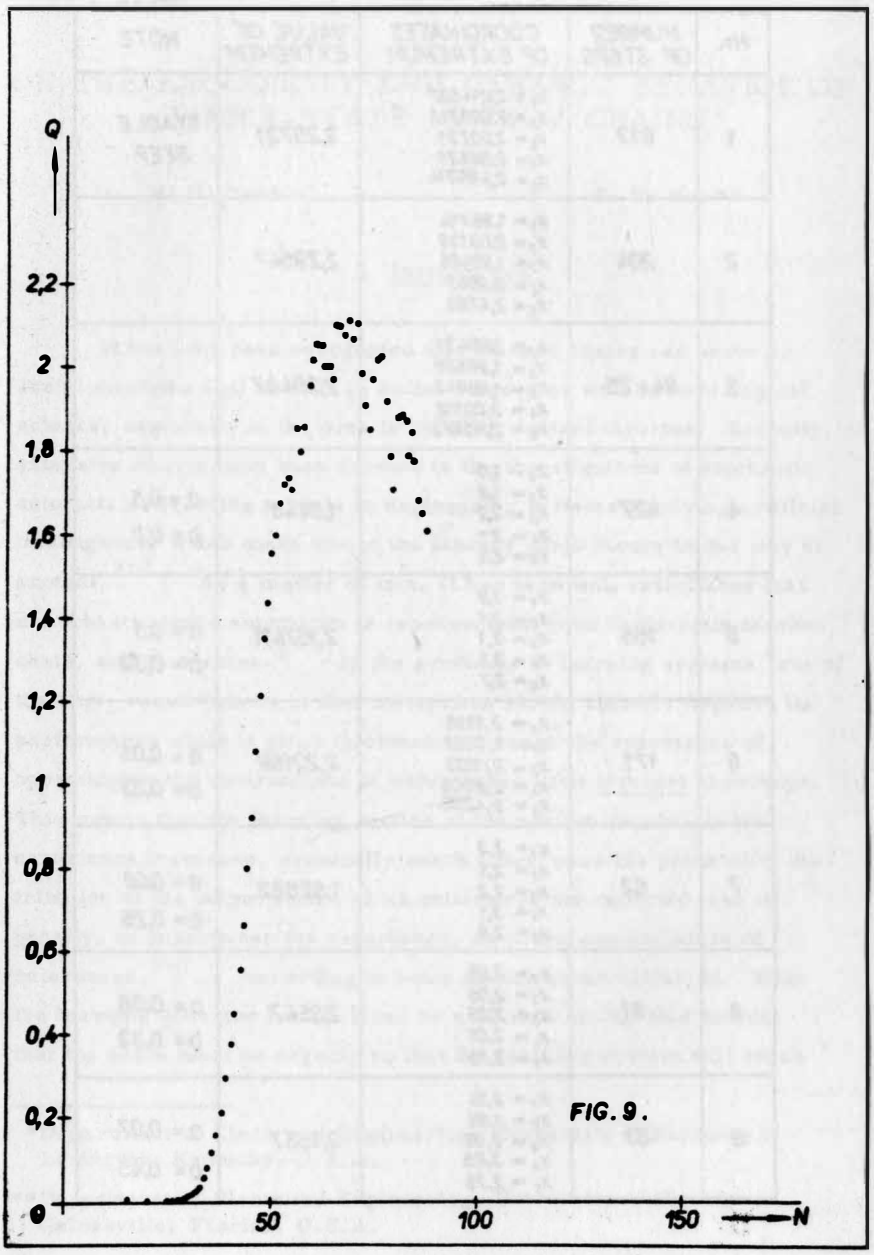


Fig. 9. Random process with learning and accumulation of experience.

TABLE 1

Nr.	NUMBER OF STEPS	COORDINATES OF EXTREMUM	VALUE OF EXTREMUM	NOTE
1	613	$x_1 = 2{,}014567$ $x_2 = 2{,}001217$ $x_3 = 2{,}00731$ $x_4 = 2{,}08521$ $x_5 = 2{,}466914$	2,29721	STABLE STEP
2	204	$x_1 = 1{,}98714$ $x_2 = 2{,}03139$ $x_3 = 1{,}98486$ $x_4 = 3{,}0047$ $x_5 = 2{,}4783$	2,29647	
3	94 x 25	$x_1 = 1{,}99431$ $x_2 = 1{,}96528$ $x_3 = 1{,}98942$ $x_4 = 3{,}00112$ $x_5 = 2{,}45872$	2,29487	
4	427	$x_1 = 1{,}6$ $x_2 = 1{,}8$ $x_3 = 2{,}2$ $x_4 = 3-$ $x_5 = 2{,}5$	1,9443	$a = 0{,}1$ $b = 0{,}2$
5	165	$x_1 = 1{,}9$ $x_2 = 2{,}1$ $x_3 = 2{,}1$ $x_4 = 3{,}1$ $x_5 = 2{,}7$	2,157411	$a = 0{,}1$ $b = 0{,}33$
6	171	$x_1 = 2{,}1116$ $x_2 = 1{,}9287$ $x_3 = 2{,}1233$ $x_4 = 2{,}8905$ $x_5 = 2{,}4396$	2,20169	$a = 0{,}05$ $b = 0{,}33$
7	62	$x_1 = 2{,}3$ $x_2 = 2{,}1$ $x_3 = 2{,}2$ $x_4 = 3{,}1$ $x_5 = 2{,}8$	1,92682	$a = 0{,}08$ $b = 0{,}25$
8	61	$x_1 = 2{,}05$ $x_2 = 2{,}05$ $x_3 = 2{,}25$ $x_4 = 3{,}05$ $x_5 = 2{,}75$	2,0547	$a = 0{,}08$ $b = 0{,}33$
9	83	$x_1 = 2{,}15$ $x_2 = 1{,}95$ $x_3 = 2{,}05$ $x_4 = 3{,}05$ $x_5 = 2{,}75$	2,1557	$a = 0{,}07$ $b = 0{,}45$

ON THE ERGODICITY AND DYNAMIC BEHAVIOR OF FINITE-STATE MARKOV CHAINS

H. H. Yeh* J. T. Tou **

I. Introduction

It has long been recognized that Markov chains can serve as useful mathematical models in social science as well as in biological science, especially in the area of learning control theories. Recently, extensive efforts have been devoted to the investigations of stochastic automata as learning models in engineering systems involving artificial intelligence, which make use of the Markov chain theory in one way or another.[1-7] As a matter of fact, it has been well established that a stochastic finite automaton is representable by a finite-state Markov chain, and vice versa.[4] In the synthesis of learning systems, one of the basic requirements is that the system should steadily improve its performance while it gains information through the experience of operating on the environment of which it has little a priori knowledge. This means that the learning section of the system should, as its experience increases, eventually reach a best possible probability distribution of the output states which minimizes the expected loss or penalty, or maximizes the expedience, as in the nomenclature of references,[5-7] according to some performance criterion. When the learning behavior is described by a Markov chain, this means that the chain must be ergodic so that the learning system will reach

*Department of Electrical Engineering, University of Kentucky, Lexington, Kentucky, U.S.A.

**Department of Electrical Engineering, University of Florida Gainesville, Florida, U.S.A.

a best final probability distribution of the output states regardless of the initial distribution which is selected arbitrarily by the designer with insufficient a priori knowledge of the media or object on which the learning system operates.

Furthermore, in an engineering system, what is important is not only the asymptotical property of the chain when time or operation step approaches infinity, but also the dynamic behavior of the system before it reaches the steady state. This includes the rate of convergence to the steady state, the extent of possible misbehavior during the transient, and sometimes the monotonicity of convergence, etc.

There has been well established relationship between the asymptotic properties of a finite-state Markov chain and the eigenvalues of its state transition probability matrix.[8] It can be shown that an irreducible finite state Markov chain is ergodic if and only if $\lambda = 1$ is the only eigenvalue with modulus 1 of the state transition probability matrix. If there exist other eigenvalues of modulus 1, then they are necessarily k-th roots of 1, for some positive integer k. In this case the chain is periodic with period k. There is indeed mathematical elegance in this theorem. Nevertheless, this knowledge is of little practical utility for detecting the asymptotic behavior of a Markov chain in engineering applications since it is often difficult to find all the eigenvalues of a matrix.

Some authors maintain that a necessary and sufficient condition of the ergodicity of a finite-state homogeneous Markov chain is that the chain is fully regular.[4] This means that the chain has only one minimal closed set of states. If transition from the i-th state to the j-th state is possible at the m-th step, then the common factor of the set of m's is one for each pair ij in the closed set. The relation between full regularity and ergodicity gives some insight into the behavior of an ergodic Markov chain, but it is of little use in testing the ergodicity of the chain however.

There also exist other methods suitable in engineering practice to determine whether a particular class of finite-state Markov chain is ergodic.[9-11] However, besides being restrictive in applicability,

these methods give no information about the convergence rate and fall short of geometrical interpretation of the dynamic behavior of a chain in the state space, which is actually the learning behavior of a system involving an equivalent stochastic automaton.

This paper analyzes the dynamic behavior of ergodic finite-state Markov chains in the finite dimensional linear space. The notion of the norms of vectors and matrices and the principle of contraction mapping are employed in this analysis which provides insight into the learning behavior of an equivalent stochastic automaton. As a result, a measure of the convergence rate is made possible. An ergodicity test procedure of general nature also results from this analysis. From this test procedure various test criteria which include many tests given in the literature [9-11] as special cases are derived. One criterion is shown to be both necessary and sufficient.

II. The Finite-State Homogeneous Markov Chain

Formal definitions of Markov chains can be found in many textbooks. In this paper it suffices to say that an r-state Markov chain is completely defined by the relation

$$\underline{x}(n+1) = \underline{P}\underline{x}(n) \tag{1}$$

where $\underline{x}(n)$ is the n-th step probability distribution vector (or simply probability vector) of dimension r, whose i-th component $x_i(n)$ is the probability of the chain being in the i-th state; $\underline{P}$ is rxr transition (probability) matrix whose element p_{ij} is the probability of transition from the j-th state to the i-th state. If $\underline{P}$ is independent of n, the chain is said to be stationary or homogeneous. Only homogeneous chain is of interest in this study. The components of $\underline{x}(n)$ are non-negative and their sum is equal to one; the elements of $\underline{P}$ are non-negative and each column sum is equal to one. Any square matrix with this property is called stochastic (or Markov). Hence the product of any two stochastic matrices is again a stochastic matrix. In the ensuing discussion, capital letters E and S denote sets and space; $\underline{A}$, $\underline{B}$, $\underline{P}$ and $\underline{G}$ denote rxr square matrices; lower-case bold-face letters denote r-dimensional column vectors; superscript t denotes the transpose of column vector or matrix; i, j, k, m, n, and r denote positive integers; and lower-case

letters denote real constants.

Let it be defined that a homogeneous Markov chain with r states is ergodic if real numbers $\pi_1, \pi_2, \ldots, \pi_r$ exist, such that for any i, j,

$$\lim_{n \to \infty} p_{ij}^{(n)} = \pi_j \tag{2}$$

where $p_{ij}^{(n)}$ is the ij element of the matrix P^n. It is readily seen that a finite-state homogeneous chain is ergodic, if and only if

$$\lim_{n \to \infty} x_i(n) = \pi_i \tag{3}$$

independent of the initial value $x_i(0)$ for i = 1, 2, ..., r. The probability vector $\underline{p}$ whose components are $\pi_1, \pi_2, \ldots, \pi_r$ is called the stationary probability vector. Equation (3) suggests that an equivalent definition of an ergodic Markov chain is that the sequence

$$\underline{x}(0), \underline{x}(1), \ldots, \underline{x}(n), \ldots \tag{4}$$

converges to $\underline{p}$ independently of $\underline{x}(0)$.

The analysis of the behavior of sequence (4) in r-dimensional linear space starts in the next section with an algebraic treatment of the Markov chain.

III. Inducement of Transition Matrix on Invariant Subspace

Let E be the r-dimensional Euclidean space. The probability vector $\underline{x}$ is a point on the hyperplane S_1, represented by

$$S_1 = \{\underline{x}: \underline{e}^t \underline{x} = 1\} \tag{5}$$

where $\underline{e}^t$ is the r-dimensional row-vector, every element of which is one. The set of points in S_1 with non-negative coordinates will be denoted by S_1^+. Since $\underline{P}$ is a stochastic matrix, it is easily seen that S_1 and S_1^+ are invariant under the transformation represented by $\underline{P}$. That is to say, for every $\underline{x} \in S_1$ (or $\underline{x} \in S_1^+$), $\underline{P}\underline{x}$ again belongs to S_1 (or S_1^+). It can be shown that for any constant c, S_c which is defined by

$$S_c = \{\underline{y}: \underline{e}^t \underline{y} = c\} \tag{6}$$

is also an invariant under transformation $\underline{P}$. To see this, let $\underline{y}(0)$ be an arbitrary vector in S_c, and let $\underline{y}(1) = \underline{P}\,\underline{y}(0)$. Then

$$\sum_{i=1}^{r} y_i(1) = \sum_{i=1}^{r}\sum_{j=1}^{r} p_{ij}\, y_j(0) = \sum_{j=1}^{r} y_j(0) \sum_{i=1}^{r} p_{ij}$$

$$= \sum_{j=1}^{r} y_j(0) = c$$

Hence $\underline{y}(1)$ is again in S_c. Of special interest to the following development is the set S_o for $c = 0$. This set forms a subspace in E since it is a hyperplane passing through the origin.

A linear transformation is called a linear operator if the image of the transformation is again contained in its domain. Thus the transformation $\underline{P}$ on E is a linear operator, and by restricting the domain of the definition of $\underline{P}$ on S_o, a linear operation $\underline{P}_o$ on S_o is induced. This linear operator is defined by

$$\underline{P}_o\underline{y} = \underline{P}\underline{y}; \quad \underline{y} \in S_o \tag{7}$$

However, $\underline{P}_o$ is different from $\underline{P}$ since its domain is on S_o, not on E.

Let $\underline{y}(1) = \underline{P}\underline{y}(0)$ and $\underline{y}(0) \in S_o$. Since $\underline{y}(0)$ satisfies

$$\sum_{i=1}^{r} y_i(0) = 0 \tag{8}$$

the i-th component of $\underline{y}(1)$ can be put in the following form by using (8):

$$y_i(1) = \sum_{j=1}^{r} (l_{ij} - a_i)\, y_j(0) \tag{9}$$

for any real number a_i. Thus the induced operator $\underline{P}_o$ on S_o is found to be an rxr matrix whose ij element is $p_{ij} - a_i$. It is worth noting that there are infinitely many matrices representing $\underline{P}_o$.

Consider the sequence of (4) where $\underline{x}(n) \in S_1+$ for all n. The sequence satisfies (1). Thus, by iteration,

$$\underline{x}(n+1) - \underline{x}(n) = \underline{P}^k[\underline{x}(n-k+1) - \underline{x}(n-k)] \tag{10}$$

Let

$$\Delta\underline{x}(n) \triangleq \underline{x}(n+1) - \underline{x}(n) \tag{11}$$

Then

$$\Delta\underline{x}(n) = P^k\, \Delta\underline{x}(n-k) \tag{12}$$

Since $\Delta\underline{x}(n)$ is in S_o for all n.

$$\Delta\underline{x}(n) = (\underline{P}^k)_o \ \Delta\underline{x}(n-k) \tag{13}$$

where $(\underline{P}^k)_o$ is the induced operator of $\underline{P}^k$ on S_o. The i, j element of $(\underline{P}^k)_o$ is found by analogy with (9) to be

$$p_{ij}^{(k)} - a_i$$

where $p_{ij}^{(k)}$ is the ij element of $\underline{P}^k$.

IV. Ergodic Chain as Contraction Mapping

Let a norm be chosen for the matrix $(\underline{P}^k)_o$. Let the vector norm of $\Delta\underline{x}$ be constructed such that the given matrix norm is consistent with it. (That this can be done for any given matrix norm has been shown in the literature.[12]) Then it follows from (13) that

$$||\Delta\underline{x}(n)|| \leq ||(P^k)_o|| \cdot ||\Delta\underline{x}(n-k)|| \tag{14}$$

It will be shown in the sequel that if a norm of a matrix of the induced operator $(P^k)_o$ can be found such that

$$||(P^k)_o|| = \sigma^k < 1 \tag{15}$$

for some positive integer k, then sequence (4) converges to a limit $\underline{x}(\infty)$ independent of $\underline{x}(0)$. In this connection, it is worth noting that convergence in every other norm. Hence in proving convergence, it suffice to choose any norm for convenience.[12]

Substituting (15) into (14) gives

$$||\Delta\underline{x}(n)|| \leq \sigma^k ||\Delta\underline{x}(n-k)|| \tag{16}$$

Let m, n_1, n_2, n_1' and n_2' be chosen such that

$$n = n_1 k + n_2 \tag{17}$$

$$n+m = n_1' k + n_2' \tag{18}$$

$$n_1' \geq n_1 \tag{19}$$

$$n_2, n_2' < k \tag{20}$$

Then

$$\underline{x}(n+m)-\underline{x}(n) = P^{n_1 k}[\underline{x}(n_1' k - n_1 k + n_2') - \underline{x}(n_2)]$$

$$= [(P^k)_o]^{n_1} [\underline{x}(n_1' k - n_1 k + n_2') - \underline{x}(n_2)] \tag{21}$$

By the definition of norm, (15) and (21) give

$$||\underline{x}(n+m)-\underline{x}(n)|| \leq \sigma^{n_1 k}||\underline{x}(n_1'k-n_1 k+n_2')-\underline{x}(n_2)||$$

$$\leq \sigma^{n_1 k}[||\underline{x}(n_2')-\underline{x}(n_2)|| + ||\underline{x}(n_2'+k)-\underline{x}(n_2')|| + \ldots$$

$$+||\underline{x}(n_1'k-n_1 k+n_2')-\underline{x}(n_1'k-n_1 k+n_2'-k)||]$$

$$\leq \sigma^{n_1 k}||\underline{x}(n_2')-\underline{x}(n_2)||$$

$$+\sigma^{n_1 k}||\underline{x}(n_2'+k)-\underline{x}(n_2)||(1+\sigma^{k}+\ldots+\sigma^{(n_1'-n-1)k})$$

$$<\sigma^{n_1 k}||\underline{x}(n_2')-\underline{x}(n_2)|| + \frac{\sigma^{n,k}}{1-\sigma^{k}}||\underline{x}(n_2'+k)-\underline{x}(n_2')||$$

where $n_1' > n_1$ has been assumed since $n_1' = n_1$ is an obvious case. Thus for every $\varepsilon > 0$ there exists an integer $N(\varepsilon)$ such that

$$||\underline{x}(n+m) - \underline{x}(n)|| < \varepsilon \quad \text{for all } n > N(\varepsilon). \tag{22}$$

Hence it is seen that (4) is a Cauchy sequence. Since the normed Euclidean space is complete, sequence (4) converges to a limit $\underline{x}(\infty)$ in the norm. It is apparent that this sequence also converges to the limit $\underline{x}(\infty)$ in the ordinary sense. That is, every component of $\underline{x}(n)$ approaches the corresponding component of $\underline{x}(\infty)$ in the limit. This is because that the norm function is continuous. However, the proof is by no means trivial.[12] Furthermore, by virtue of the continuity of the transformation $\underline{P}$

$$\underline{P}\,\underline{x}(\infty) = \underline{P} \lim_{n\to\infty} \underline{x}(n) = \lim_{n\to\infty} \underline{P}\,\underline{x}(n)$$

$$= \lim_{n\to\infty} \underline{x}(n+1) = \underline{x}(\infty)$$

Thus the limit $\underline{x}(\infty)$ is a stationary probability vector.

More can be said about the limit $\underline{x}(\infty)$. That is, the existence of $\underline{x}(\infty)$ under the condition (15) is unique. For if there exist $\underline{x}(\infty)$ and $\underline{x}'(\infty)$ in S_o such that $\underline{P}\,\underline{x}(\infty) = \underline{x}(\infty)$ and $\underline{P}\,\underline{x}'(\infty) = \underline{x}'(\infty)$, then

$$||\underline{x}(\infty)-\underline{x}'(\infty)|| = ||\underline{P}^{k}\underline{x}(\infty)-\underline{P}^{k}\underline{x}'(\infty)||$$

$$= ||\underline{P}^{k}(\underline{x}(\infty)-\underline{x}'(\infty))||$$

$$= ||(\underline{P}^{k})_{o}(\underline{x}(\infty) - \underline{x}'(\infty))||$$

$$\leq \sigma^{k}||\underline{x}(\infty)-\underline{x}'(\infty)||$$

Hence $||\underline{x}(\infty)-\underline{x}'(\infty)|| = 0$, which implies $\underline{x}(\infty) = \underline{x}'(\infty)$. Obviously, $\underline{x}(\infty)$ is identified with the stationary probability vector $\underline{p}$ in Section II.

It is interesting to note that for any homogeneous finite-state Markov chain there exists a stationary probability distribution. This is readily inferred from a well known classical theorem on non-negative matrices due to Perron and Froebenius. (For a new proof see reference[13]). This classical theorem states that, for an irreducible non-negative matrix $\underline{A}$, there exists a positive eigenvalue λ which is no less than the modulus of any other eigenvalue of $\underline{A}$, and that corresponding to λ there exists an eigenvector of positive components, and this λ is the only eigenvalue of $\underline{A}$ which has a corresponding eigenvector of positive components. However, the absolute probability distribution approaches a unique stationary distribution asymptotically only when the chain is ergodic.

The principle of contraction mapping gives (15) as a sufficient condition for the ergodicity and describes the manner in which an ergodic chain converges to the stationary distribution under this condition. It will be clear in the subsequent development that condition (15) is also necessary for some k.

V. Determining the Ergodicity and the Rate of Convergence

In order to show that a finite-state Markov chain is ergodic, it suffices to show that the matrix $(\underline{P}^k)_o$ has a norm which is less than 1 for some k. A norm of a matrix cannot be less than the modulus of an eigenvalue of the matrix. For, if λ is an eigenvalue of a matrix $\underline{A}$ and $\underline{y}$ an eigenvector corresponding to λ, then $\underline{A}\,\underline{y} = \lambda\underline{y}$ and it follows that

$$||\lambda\underline{y}|| = |\lambda| \cdot ||\underline{y}|| \leq ||\underline{A}|| \cdot ||\underline{y}||$$

Therefore

$$|\lambda| \leq ||\underline{A}|| \tag{23}$$

In fact, (23) can also be proved without invoking the consistency condition between the vector norm and the matrix norm.[12] In making a test for ergodicity, it is desired to find the smallest possible norm of $(\underline{P}^k)_o$. Theoretically, the norm of a matrix can be made as closed as possible to the largest value of the modulii of all eigenvalues, which has been named spectral radium.[12] However, practical methods for constructing such norms are avilable only for matrices with non-negative elements. For

$(\underline{P}^k)_o$, in general, the best can be done is to construct the norm to be as close as possible to the spectral radius of $(\underline{P}^k)_o$. (Here and in the sequel, the absolute value symbol is applied to a matrix or vector to signify the replacement of each element by its absolute value.) Two norms which serve this purpose are the g-norm and the g′- norm. They are defined as follows: Let $\underline{G}$ be a diagonal matrix whose diagonal elements are positive numbers $g_1, g_2, \ldots, g_r$. The g-norm of a vector $\underline{x}$ is defined as the maximum modulus of the elements $\underline{G}^{-1}\underline{x}$; i.e.,

$$||\underline{x}||_g \triangleq \max \frac{|x_i|}{g_i} \tag{24}$$

Subordinate to the vector g-norm defined by

$$||\underline{A}||_g \triangleq |||\underline{A}|\underline{g}||_g \tag{25}$$

where $\underline{g}$ is the vector whose i-th component is g_i. The vector g′-norm is defined

$$||\underline{x}||_{g'} \triangleq \underline{g}^t|\underline{x}| \tag{26}$$

The matrix g′-norm which is subordinate to the vector g′-norm is

$$||\underline{A}||_{g'} \triangleq ||\underline{A}^t|\underline{g}||_g \tag{27}$$

A numerical method of computing $\underline{g}$ such that the g-norm of an irreducible matrix $\underline{A}$ is as close as possible to the spectral radius of the non-negative matrix $|\underline{A}|$ is available in the literature.[12] Reducible matrices can be treated by separate manipulation of the submatrices of the original matrix after proper permutations. An alternative way[14] of finding such a $\underline{g}$-vector is to successively transform an arbitrarily selected vector of positive elements by the non-negative matrix $|\underline{A}|$; i.e., if

$$\underline{g}' = |\underline{A}|\underline{g};\quad \underline{g}'' = |\underline{A}|\underline{g}';\quad \underline{g}''' = |\underline{A}|\underline{g}'' \cdots$$

then

$$||\underline{A}||_{\underline{g}} \geq ||\underline{A}||_{\underline{g}'} \geq ||\underline{A}||_{\underline{g}''} \geq ||\underline{A}||_{\underline{g}'''} \geq \cdots$$

The lower bound of this sequence is the spectral radius of $|\underline{A}|$
The latter method is much simpler in computation. However, when the diagonal elements of $|\underline{A}|$ are all zero, there will be no guarantee that the sequence will definitely reach its lower bound.

Comparison of (25) with (27) shows that the $\underline{g}'$-norm of a matrix $\underline{A}$ is just the g-norm of the transpose of $\underline{A}$. It also follows from (25) that the g-norm of a matrix $\underline{A}$ is the largest row sum of the nonnegative matrix $\underline{G}^{-1}|\underline{A}|\underline{G}$. Moreover, the largest column sum of this matrix is the $\underline{g}'$-norm of $\underline{A}$, with the $\underline{g}$ vector defined as $\underline{g} = [\frac{1}{g_1}, \frac{1}{g_2}, \ldots, \frac{1}{g_r}]$.
Thus either the largest row sum or the largest column sum of $\underline{G}^{-1}|\underline{A}|\underline{G}$ is a norm of $\underline{A}$.

After a chain is shown to be ergodic, the rate of convergence can be determined by the smallest norm found for $(\underline{P}^k)_o$. If a norm of $(\underline{P}^k)_o$ is found satisfying (15), then

$$||\underline{x}(n+k) - \underline{p}|| \leq \sigma^k ||\underline{x}(n) - \underline{p}|| \tag{28}$$

This means that the rate of decrease of the "distance" between the present probability vector and the stationary probability vector is at least 100 $(1 - \sigma^k)$ percent after k steps of operation. If a g-norm for $(\underline{P}^k)_o$ is found satisfying (15), then $\max |x_i(n+k) - \pi_i|/g_i$ is no greater than σ^k times $\max |x_i(n) - \pi_i|/g_i$. In particular if $\underline{g} = \underline{e}$, then the maximum of the absolute value of the components of the vector $\underline{x} - \underline{p}$ is reduced by at least $100(1 - \sigma^k)$ percent for every k steps. On the other hand, if a g'-norm is found satisfying (15), then the projection of the vector $|\underline{x}(n+k) - \underline{p}|$ on $\underline{g}$ is no greater than σ^k times the projection of the vector $|\underline{x}(n) - \underline{p}|$ on $\underline{g}$. Again σ^k marks the rate of convergence. When $\underline{g} = \underline{e}$, then the sum of the absolute value of the components of the error vector $\underline{x} - \underline{p}$ is reduced by at least $100 (1 - \sigma^k)$ percent after every k steps. Hence in the analysis of a finite-state homogeneous Markov chain it is desirable not only to know that a chain is ergodic, but also the smallest possible g-norm or g'-norm, or both, of $(\underline{P}^k)_o$ for some $\underline{g}$.

A Test Procedure

From the foregoing discussion a test for the ergodicity and convergence rate of a chain with transition probability matrix $\underline{P}$ can be summarized as follows.

(A) Form $(\underline{P}^k)_o = [p_{ij}^{(k)} - a_i]$ from $\underline{P}^k$ by assigning a_i. Trial may be started from K = 1. The choice of a_i is aimed at the minimum norm for $(\underline{P}^k)_o$.

(B) Choose r positive integers $g_1, g_2 \ldots, g_r$ such that the g-norm of $(\underline{P}^k)_o$ (the maximum row sum of $\underline{G}^{-1}|(\underline{P}^k)_o|\underline{G}$) or the g'-norm of $(\underline{P}^k)_o$ (the maximum row sum of $\underline{G}^{-1}|(\underline{P}^k)_o|^t\underline{G}$) is less than one. If this can be done then the chain is ergodic. The positive numbers $g_1, g_2, \ldots, g_r$ may be chosen by inspection.

(C) If such a $\underline{g}$-vector cannot be easily found by inspection, the numerical method proposed in Theorem 4.7 of reference 12 of finding a $\underline{g}$ vector

which gives the smallest possible g-norm or g'-norm of $(\underline{P}^k)_o$ may be employed or, alternatively, the $\underline{g}$ vector may be chosen as

$$\underline{g} = |(\underline{P}^k)_o|^m \underline{e} \tag{29}$$

Example 1. As an illustration of the above procedure, consider a Markov chain with transition probability matrix

$$\underline{P} = \begin{bmatrix} 0 & 0.3 & 0 & 0.2 & 0 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.7 & 0 & 0 & 0.5 \\ 0 & 0 & 0.4 & 0 & 0.5 \\ 0 & 0 & 0.6 & 0.8 & 0 \end{bmatrix}$$

Try k = 1, a_i = the third element of the i-th row. Then

$$|\underline{P}_o| = \begin{bmatrix} 0 & 0.3 & 0 & 0.2 & 0 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.7 & 0 & 0 & 0.5 \\ 0.4 & 0.4 & 0 & 0.4 & 0.1 \\ 0.6 & 0.6 & 0 & 0.2 & 0.6 \end{bmatrix}$$

$$\underline{g}' = |\underline{P}_o|\underline{e} = \text{Col}\begin{bmatrix} 0.5 & 0.5 & 1.7 & 1.3 & 2.0 \end{bmatrix}$$

$$\underline{g}'' = |\underline{P}_o|^2\underline{e} = \text{Col}\begin{bmatrix} 0.41 & 0.25 & 1.6 & 1.12 & 2.06 \end{bmatrix}$$

$$\underline{g}''' = |\underline{P}_o|^3\underline{e} = \text{Col}\begin{bmatrix} 0.299 & 0.205 & 1.41 & 0.918 & 1.856 \end{bmatrix}$$

It is seen that every element of $\underline{g}'''$ is less than the corresponding element of $\underline{g}''$. Hence the chain is ergodic. The g-norm of the matrix $\underline{P}_o$ using $\underline{g}''$ as the $\underline{g}$ vector is the maximum ratio of the elements of $\underline{g}'''$ to the corresponding elements of $\underline{g}''$ i.e.,

$$\|\underline{P}_o\|_{\underline{g}''} = \max\left\{\frac{0.299}{0.41}, \frac{0.205}{0.25}, \frac{1.41}{1.61}, \frac{0.918}{1.12}, \frac{1.856}{2.06}\right\}$$

$$= 0.901$$

However, if the calculation of $\underline{g}$ vector is carried on further, i.e.,

$$\underline{g}^{(4)} = \text{Col}\begin{bmatrix} 0.245 & 0.15 & 1.221 & 0.754 & 1.6 \end{bmatrix}$$

$$\underline{g}^{(5)} = \text{Col}\begin{bmatrix} 0.196 & 0.123 & 1.028 & 0.62 & 1.35 \end{bmatrix}$$

$$\underline{g}^{(6)} = \text{Col}\begin{bmatrix} 0.161 & 0.098 & 0.86 & 0.511 & 1.126 \end{bmatrix}$$

then it is found that

$$||\underline{P}_o||_{\underline{g}'''} = 0.868, \quad ||\underline{P}_o||_{\underline{g}^{(4)}} = 0.844, \quad ||\underline{P}_o||_{\underline{g}^{(5)}} = 0.838$$

Hence it is known that not only the chain is ergodic, but that the "distance" between $\underline{x}$ and $\underline{p}$ when measured in terms of g-norm with the $\underline{g}$ vector taken as $\underline{g}^{(5)}$, is reduced at least 100 (1 - 0.838) = 16.2 percent after each step.

Theoretically the smallest norm of $\underline{P}_o$ for an ergodic chain is the maximum of the modulii of the eigenvalues of P, excluding the eigenvalue $\lambda = 1$. This is because that the only eigenvalue with modulus 1 is the simple eigenvalue $\lambda = 1$ and the projection of the operator $\underline{P}$ on S_o along $\underline{p}$ has all the rest of the eigenvalues of $\underline{P}$. (Note that S_o and the subspace spanned by $\underline{p}$ are two invariant subspaces spanning E.) However, there is no practical method available for computing all the eigenvalues of a general $\underline{P}$ matrix. The scheme given in the present paper offers a way of computing a near-minimum norm for $(\underline{P}^k)_o$, for a chosen set of a_i.

The test procedure is fairly general. Note that no such restriction as reducibility has been imposed on the transition probability matrix. The matrix $(\underline{P}^k)_o$ can always be made irreducible by choosing a_i even if $\underline{P}^k$ is reducible. When used as ergodicity test, the versatility of this method lies in the freedom of choosing a_i. In the following development various test criteria of ergodicity are obtained by exercising this freedom. Among these criteria there is a necessary and sufficient condition which requires little computation. Hence to determine whether a finite-state homogeneous Markov chain is ergodic, the criteria obtained in the following paragraphs are more convenient and effective than the general procedure. However after the ergodicity is determined, the above procedure may be followed to determine the rate of convergence.

Some Criteria of Ergodicity

Let a_i be the smallest of the elements in the i-th row of the matrix $\underline{P}^k$ for some k. Then it follows that $(\underline{P}^k)_o$ is non-negative and the sum of the j-th column of $(\underline{P}^k)_o$ is

$$\sum_{i=1}^{r} (p_{ij}^{(k)} - a_i) = 1 - \sum_{i=1}^{r} \min_j p_{ij}^{(k)}$$

Note that this is the g'-norm of $(\underline{P}^k)_o$ with $\underline{g} = \underline{e}$. Hence if

$$\sum_{i=1}^{r} \min_{j} p_{ij}^{(k)} \neq 0 \tag{30}$$

then the chain is ergodic. Furthermore, (30) is also a necessary condition for the chain to be ergodic by virtue of (2). The proof is trivial. Therefore the following criterion is concluded:

Criterion 1. A finite-state homogeneous Markov chain with probability transition matrix $\underline{P}$ is ergodic if and only if there exists a positive integer k such that $\underline{P}^k$ has at least one row not containing zero elements. Moreover, for any positive $n>k$, $\underline{P}^n$ has at least one row not containing zero elements.

Example 2. The Markov chain with transition probability matrix

$$\underline{P} = \begin{bmatrix} 0 & 0.3 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & 0 \\ 0.6 & 0.7 & 0 & 0 & 0.2 \\ 0 & 0 & 1 & 0 & 0.8 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

is ergodic since the fourth row of $\underline{P}^3$ is readily seen to have no zero element. Note that the chain is reducible with two transient states.

In general, a chain with a probability transition matrix of the form

$$\underline{P} = \begin{bmatrix} 0 & X & 0 & X & X \\ 0 & 0 & 0 & 0 & X \\ X & X & 0 & 0 & X \\ 0 & X & X & 0 & 0 \\ 0 & X & 0 & 0 & 0 \end{bmatrix}$$

where x denotes non-zero elements, is not ergodic since $\underline{P}^4$ has the same form as $\underline{P}$, i.e., $p_{ij}^{(4)}$ is zero whenever p_{ij} is zero. Thus the chain is periodic with period 4.

From the above examples a corollary can readily be concluded that ergodicity of a finite state homogeneous Markov chain is determined by the form of the transition matrix. It has nothing to do with the numerical values of the elements of the matrix.

It is sometimes possible to investigate the ergodicity of a Markov chain without having to exhibit a k-step transition matrix $\underline{P}^k$, even if $\underline{P}$ has a zero in each row. Let a_i be the smallest non-zero element of the i-th

row. Then the g'-norm of $|\underline{P}_o|$ with $\underline{g} = \underline{e}$ is the maximum column sum of $\underline{P}_o$. Since each column sum of $\underline{P}$ is one, the following criterion is derived:

Criterion 2. Replace the non-zero elements of $\underline{P}$ by the negative of the smallest non-zero elements of the row, and the zero elements by the smallest non-zero element of the row. Then, if each column sum is negative, the chain is ergodic.

Example 3. Consider the transition probability matrix

$$\underline{P} = \begin{bmatrix} 0.1 & 0.1 & 0.3 & 0.3 & 0 \\ 0 & 0 & 0 & 0.2 & 0 \\ 0.5 & 0 & 0.2 & 0.3 & 0.5 \\ 0 & 0.5 & 0 & 0.2 & 0 \\ 0.4 & 0.4 & 0.5 & 0 & 0.5 \end{bmatrix}$$

Using Criterion 2 one obtains the following matrix:

$$\begin{bmatrix} -0.1 & -0.1 & -0.1 & -0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 & -0.2 & 0.2 \\ -0.2 & 0.2 & -0.2 & -0.2 & -0.2 \\ 0.2 & -0.2 & 0.2 & -0.2 & -0.4 \\ 0.4 & 0.4 & 0.5 & 0 & 0.5 \end{bmatrix}$$

Since each column sum of this matrix is negative, the chain is ergodic.

Let a_i be the largest element of the i-th row. It follows that P_o is non-positive and the j-th column sum of $|\underline{P}_o|$ is

$$\left| \sum_{i=1}^{r} (p_{ij} - a_i) \right| = \sum_{i=1}^{r} (a_i - p_{ij}) = -1 + \sum_{i=1}^{r} \max_j p_{ij}$$

Criterion 3. If

$$\sum_{i=1}^{r} \max_j p_{ij} < 2 \tag{31}$$

then the chain is ergodic.

Example 4. Let

$$\underline{P} = \begin{bmatrix} 0.3 & 0 & 0.2 & 0.3 & 0 \\ 0.6 & 0.3 & 0 & 0.6 & 0 \\ 0.1 & 0 & 0.2 & 0 & 0.2 \\ 0 & 0.7 & 0.6 & 0.1 & 0.7 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}$$

The ergodicity of this chain is determined at once using (31), i.e.,

$$\sum_{i=1}^{r} \max_{j} p_{ij} = 0.3 + 0.6 + 0.2 + 0.7 + 0.1 = 1.9 < 2$$

For the next criterion, let $a_i = p_{ij}$ for a fixed j, all i. Then the j-th column of $\underline{P}_o$ is zero. Let g_j be arbitrarily large. Then every element in the j-row of $\underline{G}^{-1}|\underline{P}_o|\underline{G}$ is negligibly small and every element in the j-th column of $\underline{G}^{-1}|\underline{P}_o|\underline{G}$ is zero.

Criterion 4. Let $a_i = p_{ij}$ for some fixed j, and i = 1, 2, . . . , r. Let $|\underline{P}_o|^*$ be the matrix obtained from deleting the j-th row and j-th column of $|\underline{P}_o|$. If a g-norm or g'-norm of $|\underline{P}_o|^*$ is found to be less than 1, the chain is ergodic.

Example 5. Let

$$\underline{P} = \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.4 \\ 0.4 & 0 & 0.2 & 0.3 & 0 \\ 0 & 0.3 & 0 & 0.3 & 0.4 \\ 0.6 & 0.7 & 0.8 & 0 & 0.2 \\ 0 & 0 & 0 & 0.2 & 0 \end{bmatrix}$$

and a_i be the fourth element of the i-th row. Then

$$|\underline{P}_o| = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0 & 0.3 \\ 0.3 & 0 & 0.3 & 0 & 0.1 \\ 0.6 & 0.7 & 0.8 & 0 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0 & 0.2 \end{bmatrix}$$

$$|\underline{P}_o|^* = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0.3 \\ 0.3 & 0 & 0.3 & 0.1 \\ 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix}$$

This chain is ergodic since both the maximum row sum and the maximum column sum of $|\underline{P}_o|^*$ are less than 1.

VI. Conclusion

When a norm of the induced transition matrix on the invariant subspace S_o whose normal is (1, 1, . . .) is found to be less than one the Markov chain is ergodic and operates as a contraction mapping on subspace S_o.

Theoretically the smallest possible norm of the induced transition matrix is less than one for an ergodic chain because $\lambda = 1$ is the only eigenvalue of the transition matrix with modulus one and all other eigenvalues have modulii less than one. Hence if a finite-state homogeneous markov chain is ergodic it is a contraction mapping on S_o. The norm of the induced transition probability matrix serves as a pessimistic estimation of the convergence rate.

A general test procedure is summarized for determining the ergodicity and convergence rate of a finite-state homogeneous Markov chain. A necessary and sufficient condition for ergodicity is derived from this procedure together with other ergodicity criteria for special cases. It is found that ergodicity is determined by the form of the transition matrix and has nothing to do with the numerical values of the elements of the matrix.

Acknowledgment

This work was supported in part by the Office of Naval Research, and by the National Science Foundation.

References

1. Tou, J. T., (Ed.), "Applied Automata Theory," Academic Press, New York, 1968.
2. Fu, K.S., "Stochastic Automata as Models of Learning Systems," in Computer and Information Sciences-2, (Tou, J.T., Ed.), Academic Press, New York, 1967.
3. Fu, K.S., and McLaren, R.W., "An Application of Stochastic Automata to the Synthesis of Learning Systems," Report TR-EE 65 - 17, Purdue University, September, 1965.
4. Cleave, J.P., "The Synthesis of Finite-State Markov Chains," Cybernatica, Vol. 5, No. 1, 1962.
5. Tsetlin, M.L., "On the Behavior of Finite Automata in Random Media," Automation and Remote Control, Vol. 22, No. 10, pp. 1210-1219, October 1961.
6. Tsertsvadze, G. W., "Certain Properties of Stochastic Automata and Methods for Synthesizing Them," Automation and Remote Control, Vol. 24, No. 3, pp. 316-326, March 1963.

7. Varshovskii, V. I., and Vorontsova, I. P., "On the Behavior of Stochastic Automata with Variable Structure," Automation and Remote Control, Vol. 24, No. 3, pp. 327-333, March 1963.

8. Frechet, M., Recherches theoriques modernes sur le calcul des probabilites, 2e livre. (Theorie des evenement en chaine dans le cas d'un nombre fini d'etats possibles), Gauthier-Villars, Paris, 1952.

9. Bellman, R., Introduction to Matrix Analysis. Ch. 14, McGraw-Hill Book Company, Inc., New York, 1960.

10. Parzen, E., Modern Probability Theory and Its Applications, Ch. 3, John Wiley and Sons, Inc., New York, 1960.

11. Doob, J. L., Stochastic Processes, Ch. 5, John Wiley and Sons, Inc., New York, 1960.

12. Householder, A. S., "The Approximate Solution of Matrix Problems," J. of the Assoc. for Computing Machinery, Vol. 5, pp. 205-243, 1958.

13. Fan, K., 'Topological Proofs for Certain Theorems on Matrices with Non-Negative Elements," Monatshefte fur Mathematik, Vol. 62, pp. 219-237, 1958.

14. Wiedlandt, H., "Unzerlegbare, nicht negative Matrizen," Math Zeit., Vol. 52, pp. 642-648, 1950.

STATISTICAL MIN-MAX DECISION METHODS AND THEIR APPLICATION TO LEARNING CONTROL

Bunji KONDO and Shigeru EIHO
Faculty of Engineering, Kyoto University
Kyoto, Japan

1. Introduction

Recently, the application of the statistical decision method to learning control systems has been advanced. However, most of works assume that the probability distribution of the system parameters are known or partially known a priori.[1-5]

This paper deals with the case where we have little knowledge about the statistical property of the system.

2. Statement of the problem

2-1 Preliminary

Consider the usual optimalizing control system with one control variable (Fig.1). Optimal control input, which brings P.I. (performance index) to the maximum point, is a function of disturbance $\mathbf{z}$. There may, however, be unobservable random disturbances. The optimal value of x may, therefore, be thought of as a random variable with a certain probability distribution function.[2]

Assume, for the sake of simplicity, that $\mathbf{z}$ is one dimensional variable, and denote the control input by d. Assume also that x, z and d take discrete values. (If they are continuous quantities, they can be changed to discrete quantities by appropriate quantization.)

The optimal control input can be determined by the statistical decision method, as follows:

Optimal decision $d_o(z_j)$ when observed value of z is z_j, is a $d(z_j)$ which minimizes $\rho(z_j)$:

$$\rho(z_j) = \sum_i L(x_i, d(z_j))\, P(x_i/z_j)$$

where $L(x_i, d(z_j))$ represents the loss associated with decision $d=d(z_j)$ when $x=x_i$, and $P(x_i/z_j)$ is the conditional probability distribution. This optimal decision is called a sinple Bayes solution.

However, $P(x_i/z_j)$ in many cases is not known beforehand. The control law when $P(x_i/z_j)$ is constant but unknown is considered in what follows. Incidentally, it is assumed that the past control experiences are tabulated as Table 1. The t-i element m_{ti} in Table 1 expresses the number of $x=x_i$ when $z=z_t$.

The problem now is how to evaluate d when $z=z_t$ is observed. Optimal decision d_o should be fixed in the light of Table 1. Here we assume that z_t is statistically independent of $z_r (r \neq t)$. Then d_o can be evaluated only by using the t-th row in Table 1. Finally the problem resolves itself into how to evaluate d when data $(m_{t1}, m_{t2}, \ldots, m_{tk})$ is given. For the sake of simplicity, suffix t is ommitted in the following.

2-2 Statement of the problem

Let us state the problem all over again. We use the following notations:

X: discrete random variable with k different possible values $x_1, x_2, \ldots$ and x_k.

d: decision. There are q different possible decisions $d_1, d_2, \ldots$ and d_q.

m_i: number of past observations that $X=x_i$. $\quad i = 1,2,\ldots,k$

n: sample size; $n = \sum_{i=1}^{k} m_i$

$\underline{m} = (m_1, m_2, \ldots, m_k)^T$: vector expression of sample (data)

M: set of $\underline{m}$, whose element is expressed by $\underline{m}_i$

L_{ij}: loss associated with the decision $d=d_j$ when $X=x_i$

$L = [L_{ij}]$: loss matrix

$P(x_i) \equiv \theta_i$: probability of $X=x_i \quad i = 1,2,\ldots,k$

$\underline{\theta} \equiv (\theta_1, \theta_2, \ldots, \theta_q)^T$: vector expression of probability distribution function of X.

The problem is : Which of $d_1, d_2, \ldots, d_q$ is to be taken when $\underline{m}$ is given? The problem can be put this way, too: With what probability should we take d_1, d_2,... and d_q, when $\underline{m}$ is given?

Let us define the randomized decision function to solve the problem just mentioned.[7] The randomized decision function means the aggregate of probability $\pi^j_{\underline{m}_i}$, with which d_j is chosen in regard to each element $\underline{m}_i$ of M, where j = 1, 2,...,q, $\sum_{j=1}^{q} \pi^j_{\underline{m}_i} = 1$ and i = 1,2,...,s. $(s = {}_{k+n-1}C_n)$

Let us use the following two notations:

$$\underline{\pi}_i \equiv (\pi^1_{\underline{m}_i}, \pi^2_{\underline{m}_i}, \ldots, \pi^q_{\underline{m}_i})^T$$

$$[\pi] \equiv (\underline{\pi}_1, \underline{\pi}_2, \ldots, \underline{\pi}_s) \quad : \text{expression of decision function}$$

The expected loss (risk) when decision $d=d_j$ is used, is:

$$r(j, \underline{\theta}) = \theta_1 L_{1j} + \theta_2 L_{2j} + \cdots + \theta_k L_{kj} \qquad (2\text{-}1)$$

which corresponds to $\int(z_j)$ in section 2-1.

The expected risk when decision function D ($[\pi]$) is used, is:

$$R(D; \underline{\theta}) = \sum_{i=1}^{s} \sum_{j=1}^{q} r(j, \underline{\theta}) \pi^j_{\underline{m}_i} P(\underline{m}_i ; \underline{\theta}) \qquad (2\text{-}2)$$

where $P(\underline{m} ; \underline{\theta})$ is the probability with which $\underline{m}$ is realized when the probability distribution of X is $\underline{\theta}$. i.e.:

$$P(\underline{m}\,;\underline{\theta}) = \theta_1^{m_1}\cdot\theta_2^{m_2}\cdot\cdots\cdot\theta_{k-1}^{m_{k-1}}(1-\theta_1-\theta_2-\cdots-\theta_{k-1})^{m_k}\,{}_nC^*_{\underline{m}} \qquad (2\text{-}3)$$

where

$${}_nC^*_{\underline{m}} = \frac{n!}{m_1!\,m_2!\cdots m_k!} \qquad (2\text{-}3)'$$

We consider three cases according as the degree of knowledge about the system:

Case 1 : $\underline{\theta}$ is known.

Case 2 : $\underline{\theta}$ is unknown but the a priori density of $\underline{\theta}$, i.e., $\lambda(\underline{\theta})$ is known.

Case 3 : Even the a priori density function of $\underline{\theta}$ is unknown or there is no a priori density function of $\underline{\theta}$.

In case 1, optimal decision function is the simple Bayes solution referred to in section 2-1. In case 2, the Bayes decision function may be used as the optimal decision function.[4,5]

It occasionally happens in conventional systems that the a priori density does not exist or is unknown if it exists (case 3). The value of $\underline{\theta}$ even in this case can be estimated if there are enough past data to draw on — that is, if n is large in number. Then the simple Bayes solution is obtainable. However, with insufficient data, it is proper to use the min-max decision function defined in the next chapter.

3 Min-max decision function

3-1 Definitiona

Definition 1

Decision function D_m which satisfies the following inequality is called the min-max decision function;

$$\max_{\underline{\theta}} R(D_m, \underline{\theta}) \leq \max_{\underline{\theta}} R(D, \underline{\theta}) \qquad (3\text{-}1)$$

where D is an arbitrary decision function.

There can be many min-max decision functions which satisfy eq.(3-1). It is clear that the optimal min-max decision function suggested in Difinition 2 is the best one, — if it exists.

Definition 2

The min-max decision function which satisfies the following inequality is called the optimal min-max decision function and is represented by D_{om}:

$$R(D_{om}, \underline{\theta}) \leq R(D_m, \underline{\theta}) \qquad \text{for all } \underline{\theta} \qquad (3\text{-}2)$$

where D_m is an arbitrary min-max decision function.

However, D_{om}, in many cases, does not exist or, if it exists, is hard to obtain directly. The following min-max decision function may, therfore, be a handy alternative:

Definition 3

The min-max decision function which satisfies the following inequality is called a sub-optimal min-max decision function and is represented by D_{sm}.

$$\int_{\underline{\theta}} R(D_{sm}, \underline{\theta}) d\underline{\theta} \leq \int_{\underline{\theta}} R(D_m, \underline{\theta}) d\underline{\theta} \equiv S(D_m) \tag{3-3}$$

where D_m is an arbitrary min-max decision function.

It is clear from these definitions that D_{om}, if it exists, is always D_{sm}. D_{sm} is easier to handle than D_{om} and is dealt with in the following paragraphs.

The quantity S(D) is calculable as follows:

$$\begin{aligned} S(D) &\equiv \int_{\underline{\theta}} R(D, \underline{\theta}) d\underline{\theta} \\ &= \sum_{j=1}^{k} \int_{\underline{\theta}} r(j, \underline{\theta}) \sum_{i=1}^{s} \pi^{j}_{\underline{m}_i} P(\underline{m}_i; \underline{\theta}) d\underline{\theta} \\ &= \sum_{i=1}^{s} \sum_{j=1}^{k} \pi^{j}_{\underline{m}_i} \int_0^1 \int_0^{1-\theta_1} \cdots \int_0^{1-\sum_{t=1}^{k-2} \theta_t} r(j, \underline{\theta}) P(\underline{m}_i; \underline{\theta}) d\underline{\theta} \\ &= \sum_{j=1}^{k} \sum_{i=1}^{s} \pi^{j}_{\underline{m}_i} K(\underline{m}_i, j) \end{aligned} \tag{3-4}$$

where $K(\underline{m}, j) = \{ n!/(n+k)! \} \cdot \sum_{i=1}^{k} L_{ij}(m_i+1)$ (3-5)

3-2 Sub-optimal min-max decision function

Transform expected loss R into:

$$\begin{aligned} R(D, \underline{\theta}) &= \sum_{i=1}^{s} \sum_{j=1}^{k} r(j, \underline{\theta}) \pi^{j}_{\underline{m}_i} P(\underline{m}_i; \underline{\theta}) \\ &= \sum_{j=1}^{k} r(j, \underline{\theta}) \sum_{i=1}^{s} \pi^{j}_{\underline{m}_i} P(\underline{m}_i; \underline{\theta}) \qquad (3\text{-}6)' \\ &\equiv \sum_{j=1}^{k} r(j, \underline{\theta}) P(j, \underline{\theta}, D) \qquad (3\text{-}6) \end{aligned}$$

where $P(j, \underline{\theta}, D) = \sum_{i=1}^{s} \pi^{j}_{\underline{m}_i} P(\underline{m}_i; \underline{\theta})$ (3-7)

The right hand side of eq.(3-7) expresses the probability with which d_j is taken in decision function D. In the light of eq.(2-1), eq.(3-7) can be transformed into:

$$R(D, \underline{\theta}) = \underline{\theta}^T L \underline{P} \tag{3-8}$$

where $\underline{P} \equiv (P(1, \underline{\theta}, D), P(2, \underline{\theta}, D), \ldots, P(q, \underline{\theta}, D))^T$ (3-9)

Eq.(3-8) may be interpreted as follows by the theory of games:[6] Eq.(3-8) expresses the expected loss of game $\Gamma(\underline{\theta}, \underline{P}, L)$ where $\underline{\theta}$ and $\underline{P}$ are mixed strategies which player I and II can take, respectively, and L is a pay-off matrix. Rewrite $R(D, \underline{\theta})$ to $E(\underline{\theta}, \underline{P})$ and assume that $\underline{\theta}^* = (\theta_1^*, \theta_2^*, \ldots, \theta_k^*)$ and $\underline{P}^* = (P_1^*, P_2^*, \ldots, P_q^*)$ is the minimax solution of game Γ. Then we get the following theorem:

Theorem

Assume that D_m is a min-max decision function and also assume that

$R(D_m, \underline{\theta})$ takes its maximum value at $\underline{\theta} = \underline{\theta}_m$. Then,

$$R(D_m, \underline{\theta}_m) = \min_{D} \max_{\underline{\theta}} R(D, \underline{\theta})$$

$$= \min_{\underline{P}} \max_{\underline{\theta}} \underline{\theta}^T L \underline{P} = E(\underline{\theta}^*, \underline{P}^*) \qquad (3\text{-}10)$$

Especially $R(D_m, \underline{\theta})$ has its maximum value at $\underline{\theta} = \underline{\theta}^*$.

The proof of this theorem is given in Appendix. The next colloraly follows this theorem.

Colloraly

A min-max decision function $[\pi]$ satisfies the following equation.

$$\sum_{i=1}^{s} \underline{\pi}_i P(\underline{m}_i ; \underline{\theta}^*) = \underline{P}^* \qquad (3\text{-}11)$$

where $\underline{\theta}^*$, $\underline{P}^*$ is the minimax solution of game Γ and

$$\sum_{j=1}^{r} \pi^j_{\underline{m}_i} = 1 \qquad (3\text{-}12)$$

Generally, there are many solutions which satisfy eq.(3-11). Some of them are not min-max decision functions. It is observed in many examples that a decision function which satisfies the condition (3-11) and minimizes S(D) is a min-max decision function and, therefore, a sub-optimal min-max decision function. The following conjecture seems to be true by a physical consideration though it is difficult to prove it mathematically.

Conjecture

The decision function $[\pi]$ that minimizes the following S(D) under the conditions (3-13) - (3-15) is a sub-optimal min-max decision function:

$$\sum_{i=1}^{s} P(\underline{m}_i ; \underline{\theta}^*) \underline{\pi}_i = \underline{P}^* \qquad (3\text{-}13)$$

where $\underline{\pi}_i \equiv (\pi^1_{\underline{m}_i}, \pi^2_{\underline{m}_i}, \cdots, \pi^r_{\underline{m}_i})^T$

$$\pi^j_{\underline{m}_i} \geq 0 \qquad \underline{m}_i \in M \qquad j = 1, 2, \cdots, r \qquad (3\text{-}14)$$

$$\sum_{j=1}^{r} \pi^j_{\underline{m}_i} = 1 \qquad \underline{m}_i \in M \qquad (3\text{-}15)$$

$$S(D) = \sum_{j=1}^{r} \sum_{i=1}^{s} \pi^j_{\underline{m}_i} K(\underline{m}_i, j) \qquad (3\text{-}16)$$

where $\underline{\theta}^*$ and $\underline{P}^*$ is the minimax solution of the game Γ, and where $K(m_i, j)$ is as given in eq.(3-5).

Note that eqs.(3-13), (3-14) and (3-15) are linear constraints about $\pi^j_{\underline{m}_i}$ and S(D) is also a linear function of $\pi^j_{\underline{m}_i}$. Therefore, the above sub-optimal min-max decision function is soluble by linear programming technique.

3-3 Simple examples

Ex.1 $L = \begin{pmatrix} 0 & 2 & 5 \\ 1 & 0 & 4 \\ 4 & 3 & 0 \end{pmatrix}$

$\underline{\theta}^* = (7/21, 3/21, 11/21)^T$

$\underline{P}^* = (11/21, 1/21, 9/21)^T$

The sub-optimal min-max decision functions in this example are shown in Fig.2.

Ex.2 $L = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix}$

$\underline{\theta}^* = (1/2,0,1/2)^T$

$\underline{P}^* = (1/2,0,1/2)^T\beta + (0,1,0)^T(1-\beta)$

$= (\beta/2, 1-\beta, \beta/2)^T \qquad 0 \le \beta \le 1$

In this case eq.(3-13) is expressible as follows:

$$\sum_{i=1}^{S} P(\underline{m}_i; \underline{\theta}^*)\,\underline{\pi}_i = (\beta/2,\ 1-\beta,\ \beta/2)^T$$

which can be transformed into the following two constraints:

$$\sum_{i=1}^{S} P(\underline{m}_i; \underline{\theta}^*)\,\pi_i^1 = \frac{1}{2}\beta \le \frac{1}{2}$$

$$\sum_{i=1}^{S} P(\underline{m}_i; \underline{\theta}^*)\,\pi_i^1 + 2\sum_{i=1}^{S} P(\underline{m}_i; \underline{\theta}^*)\,\pi_i^2 = 1$$

D_{sm} for Ex.2 are shown in Fig.3.

Ex.3 $L = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{pmatrix}$

$\underline{\theta}^* = (1/4,0,3/4)^T\alpha + (3/4,0,1/4)^T(1-\alpha) \qquad 0 \le \alpha \le 1$

$\underline{P}^* = (0,1,0)^T$

$P(\underline{m}; \underline{\theta}^*) = 0$ when m_2 in the data $\underline{m}$ is not zero, because $\theta_2^* = 0$. Therefore, eq.(3-13) is expressible as follows:

$$\sum_{i=1}^{S'} P((m_{1i}, 0, m_{3i})^T; \underline{\theta}^*)\,\underline{\pi}_i = (0,1,0)^T$$

Then the solution of the above equation is

$$\pi^2_{\underline{m}'} = 1 \qquad \pi^3_{\underline{m}'} = \pi^1_{\underline{m}'} = 0$$

where $\underline{m}'$ is a vector whose 2nd component is zero, i.e., $\underline{m}' = (m_{1i},0,m_{3i})$.

The other $\underline{\pi}_i$ have no constraints. D_{sm} here is shown in Fig.4.

Ex.4 $L = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$

$\underline{\theta}^* = (1/2,1/2)^T\alpha + (0,1)^T(1-\alpha) \qquad 0 \le \alpha \le 1$

$\underline{P}^* = (0,1)^T$

The min-max decision function is

$\pi^2_{\underline{m}} = 1, \quad \pi^1_{\underline{m}} = 0$ for all $\underline{m}$, for all n

3-4 Relaxed min-max decision functions

A min-max decision function is generally used when the statistical property of the system is unknown. The more data there is, the more accurately the property of the system can be estimated. When estimation can thus be accurately made, is it all right to use min-max decision functions to the exclusion of all other decision functions? In other words, do (sub-optimal) min-max decision functions all converge on the simple Bayes solution as n increases?

When there is only one solution to game $\Gamma(\underline{\theta}, L, \underline{P},)$, D_{sm}'s all converge on the simple Bayes solution. This can be proved by using the "law of large

numbers". If there are two or more solutions to game Γ, D_{sm}'s, in some cases, do not converge on the simple Bayes solution. (See Ex.3 and 4 in section 3-3.)

The rate of convergence of D_{sm}'s on the simple Bayes solution may, in some cases, be very slow. It is desirable, in such cases, to relax the constraint of min-max condition and bring the decision function into the (estimated) simple Bayes solution by estimating $\underline{\theta}$.

We rewrite S(D) as follows:

$$S(D) = \frac{n!}{(n+k-1)!}\frac{1}{n+k}\sum_{j=1}^{g}\sum_{i=1}^{k}\sum_{r=1}^{s}\pi^{j}_{\underline{m}r}\,L_{ij}\,(m_{ir}+1) \qquad (3\text{-}17)$$

If there is no constraint, $[\pi]$ which minimizes S(D) is caluculable independently of r. Then $\pi^{j}_{\underline{m}r}$ is fixed so as to minimize the following equation:

$$s(D,r) = \sum_{j=1}^{g}\sum_{i=1}^{k} L_{ij}\,\pi^{j}_{\underline{m}r}\,(m_{ir}+1\,/\,n+k) \qquad (3\text{-}18)$$
$$= (\pi^{1}, \pi^{2}, \cdots, \pi^{g})\cdot L\cdot (m_{1}+1/n+k, \cdots, m_{k}+1/n+k)^{T}$$

When n increases, this solution coincides with the (estimated) simple Bayes solution D_{esB} which is given by estimating $\theta_i = m_i/n$ (this is a most-likelihood estimation of $\underline{\theta}$). The corollary in 3-3 tells us that a min-max decision function needs to satisfy the following equation:

$$f^{j}(\underline{\theta}^{*}, [\pi]) = P_{j}^{*} \qquad (3\text{-}19)$$

Instead of this equation, use the following inequality:

$$|f^{j}(\underline{\theta}^{*}, [\pi]) - P_{j}^{*}| \le \gamma_{j}(n) \qquad j = 1, 2, \cdots, g \qquad (3\text{-}20)$$

or

$$-\gamma_{j}(n) \le f^{j}(\underline{\theta}^{*}, [\pi]) - P_{j}^{*} \le \gamma_{j}(n) \qquad (3\text{-}20)'$$

where $\gamma_j(n)$ is a monotone increasing function, $\gamma_j(0) = 0$, and $\gamma_j(\infty) = 1$.

When the amount of data is small, fix $\gamma_j(n)$ at a sufficiently small value. Then eq.(3-20) is almost the same constraint as eq.(3-19). When n increases, $\gamma_j(n)$ should be increased also. Then the constraint for a min-max decision function is relaxed and the decision function shifts to (estimated) simple Bayes solution D_{esB}.

The decision function, which minimizes S(D) in eq.(3-16) under the constraints of (3-14), (3-15) and (3-20) in place of (3-13), is called a "relaxed min-max decision function". This is also soluble by linear programming technique.

Function $\gamma_j(n)$ can be fixed in various ways. One way is by consulting the value of $S(D_{sm})$ (which is obtainable with the solution of linear programming) and of $S(D_{esB})$. Generally, they decrease as n increases. For example, when the rate of decrease in $S(D_{sm})$ is very small compared with the rate of decrease in $S(D_{esB})$. $\gamma_j(n)$ should be increased quickly to 1.

4. Conclusions

This paper has discussed the question of how to use the statistical decision method when the statistical properties of the system are not known. It explains fully the technique of obtaining the sub-optimal min-max decision function.

The decision mechamism in cases 1, 2 and 3 cited has the learning property as follows: With $\underline{\theta}$ known (as in case 1), there is no need for learning and, therefore, no need for data. With $\lambda(\underline{\theta})$ given (as in case 2), the calculation of the a posteriori probability density of $\underline{\theta}$ after $\underline{m}$ is observed, which is used for obtaining Bayes decision function, corresponds to the learning of the statistical property of the system. In case 3, the learning property is included in the mechaniam of minimizing S(D). Therefore, D_{sm} which does not reduce S(D) even when n increases (See Ex.4), includes no learning property. In such a case, a relaxed min-max decision function is preferable.

References

(1) Truxal, J. G. and Padalino, J. J. : "Decision Theory," Chapter 15 of Adaptive Control Systems edited by Mishkin and Braun, McGraw-Hill, 1961.

(2) Kondo, B. and Suzuki, T. : "Optimalizing and Learning Control by a Statistical Model," IFAC, 1965.

(3) Sklansky, J. : "Learning Systems for Automatic Control," IEEE trans. on AC, January 1966.

(4) Aoki, M. : "On Control System Equivalents of Some Decision Theoretic Theorems," Journal of Mathematical Analysis and Applications, January 1965.

(5) Lin, T. T. and Yau, S. S. : "Bayesian Approach to the Optimization of Adaptive Systems," IEEE trans. on Systems Science and Cybernetics, November 1967.

(6) Blackwell, D. and Girshick, M. A. : Theory of Game and Statistical Decisions, Wiley, New York, 1954.

(7) Wald, A. : Statistical Decision Functions, Wiley, 1950.

Appendix (Proof of Theorem)

Assume that

$$R(D_m, \underline{\theta}_m) > E(\underline{\theta}^*, \underline{P}^*) \tag{A-1}$$

and also assume that D^* is a decision function made by

$$\pi^j_{\underline{m}} = P_j^* \qquad \text{for all } \underline{m} \notin M \tag{A-2}$$

Substituting this relation into eq.(3-7) results in:

$$P(j, \underline{\theta}, D^*) = P_j^* \qquad \text{for all } \underline{\theta} \qquad j = 1,2,\ldots,q \tag{A-3}$$

and then

$$\max_{\underline{\theta}} R(D^*, \underline{\theta}) = E(\underline{\theta}^*, \underline{P}^*) \tag{A-4}$$

Accordingly

$$R(D_m, \underline{\theta}_m) > \max_{\underline{\theta}} R(D^*, \underline{\theta}) \tag{A-5}$$

This contradicts the fact that D_m is a min-max decision function. Therefore;

$$R(D_m, \underline{\theta}_m) \leq E(\underline{\theta}^*, \underline{P}^*) \tag{A-6}$$

On the other hand,

$$R(D_m, \underline{\theta}_m) \geq R(D_m, \underline{\theta}^*) = \underline{\theta}^{*T} L \underline{P}(\underline{\theta}^*)$$

$$\geq \min_{\underline{P}} \underline{\theta}^{*T} L \underline{P} = E(\underline{\theta}^*, \underline{P}^*) \tag{A-7}$$

$$\therefore R(D_m, \underline{\theta}_m) \geq E(\underline{\theta}^*, \underline{P}^*) \tag{A-8}$$

In conclusion,

$$R(D_m, \underline{\theta}_m) = E(\underline{\theta}^*, \underline{P}^*) \tag{A-9}$$

Now for proof of the last part of the theorem.

Assume that Θ is the set of $\underline{\theta}^*$ and assume that $\underline{\theta}_m \notin \Theta$. Then,

$$\underline{\theta}_m L \underline{P}(\underline{\theta}_m) > \underline{\theta}^* L \underline{P}(\underline{\theta}^*) \geq \underline{\theta}^* L \underline{P}^* \tag{A-10}$$

This contradicts eq.(A-9). As a result, $R(D_m, \underline{\theta})$ has its maximum value at $\underline{\theta} = \underline{\theta}^*$.

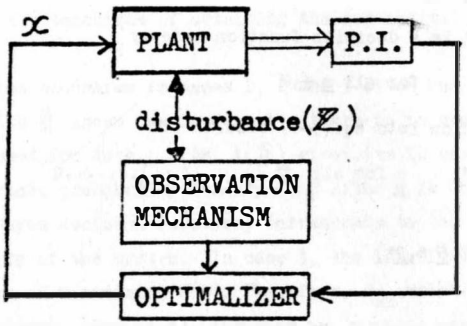
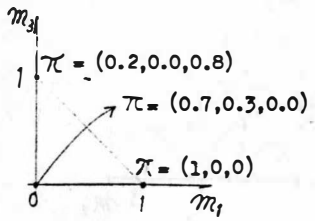


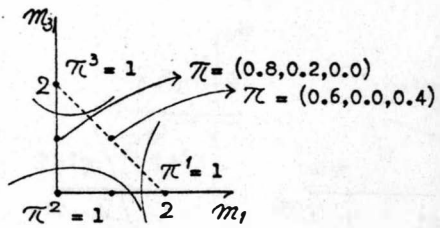
Fig.1 Optimalizing Control System

z \ x	x_1	x_2	.....	x_i	...	x_k
z_1	m_{11}	m_{12}	.....	m_{1i}	...	m_{1k}
z_2	m_{21}	m_{22}	.....	m_{2i}	...	m_{2k}
⋮	⋮					⋮
z_t	m_{t1}	m_{t2}	.....	m_{ti}	...	m_{tk}
⋮	⋮					⋮
z_r	m_{r1}	m_{r2}	.....	m_{ri}	...	m_{rk}

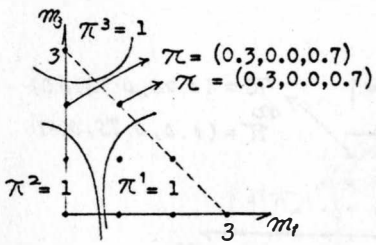
Table 1 Data of Past Control Experiences



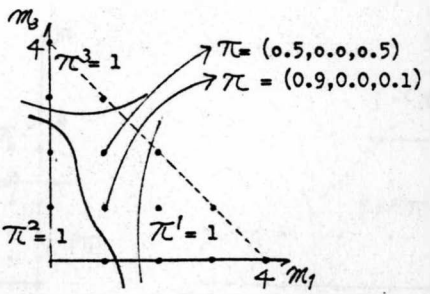
(n=1)



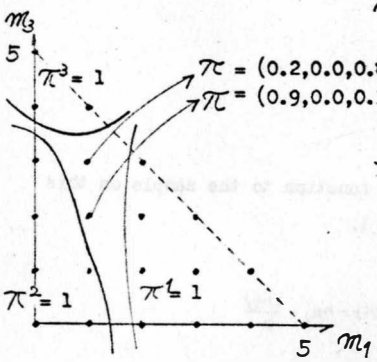
(n=2)



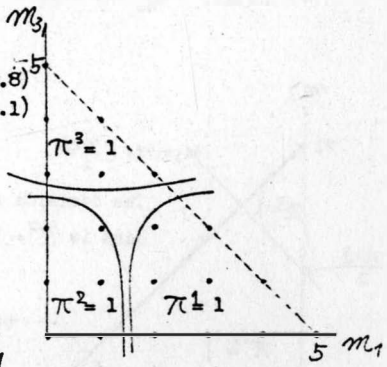
(n=3)



(n=4)



(n=5)



(n=5)

cf. Estimated Bayes solution given by estimating $=(m_1/n, m_2/n, m_3/n)$

Fig.2 D_{sm} for Ex.1

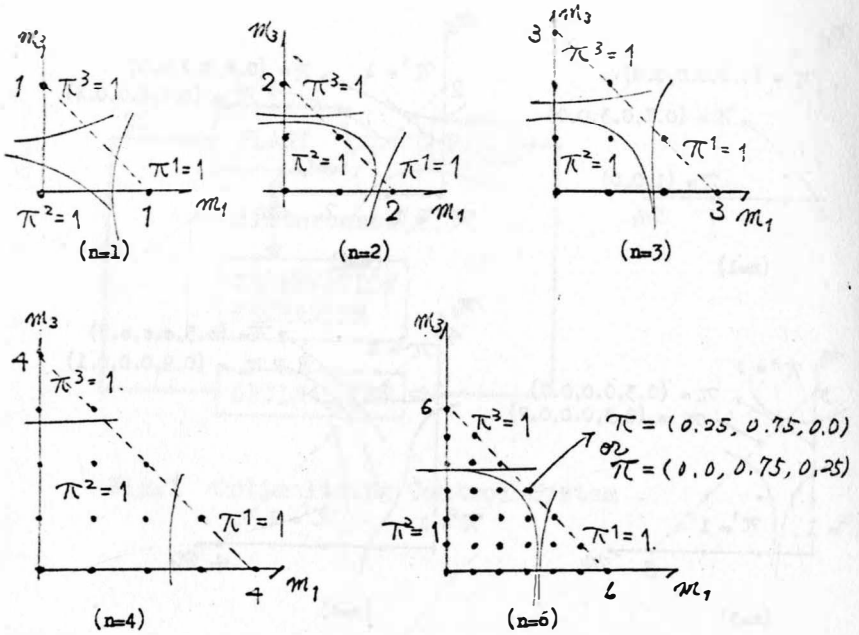


Fig.3 D_{sm} for Ex.2

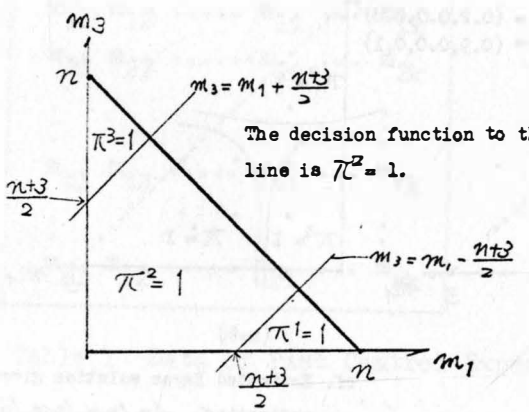


Fig.4 D_{sm} for Ex.3