$$\sum_{i=1}^{n} i^5 = \tfrac{1}{6}n^6 + \tfrac{1}{2}n^5 + \tfrac{5}{12}n^4 - \tfrac{1}{12}n^2$$

$$\sum_{i=1}^{n} i^6 = \tfrac{1}{7}n^7 + \tfrac{1}{2}n^6 + \tfrac{1}{2}n^5 - \tfrac{1}{6}n^3 + \tfrac{1}{42}n.$$

These expressions agree with those appearing in handbooks. The fact that a simple computer program can generate this sequence of expressions yields programming simplification when evaluating $\sum_{i=1}^{n} i_p$ for all values of p from 1 to N. Instead of having to program each expression, a simple loop which generates the successive expressions can be used to perform the evaluations.

Comments on "An Extension to Gradient Matrices"

ROBERT G. RAINS

In the above correspondence,[1] one of the matrix differentials listed by Vetter is

$$d\lambda_i(Y) = \frac{\text{tr } \{\text{adj } (\lambda_i I - Y)\, dY\}}{\text{tr } \{\text{adj } (\lambda_i I - Y)\}}. \tag{1}$$

It should be noted that if the eigenvalues of Y are distinct, then Jacobi's sensitivity formula

$$d\lambda_i(Y) = \frac{v_i^T\, dY u_i}{v_i^T u_i} \tag{2}$$

where λ_i is an eigenvalue of Y, u_i is the corresponding eigenvector of Y, and v_i is the corresponding eigenvector of Y^T, is generally a more efficient formula for eigenvalue sensitivity. For a derivation of (2) and some examples of its usefulness, the reader is referred to [1]–[3].

Manuscript received June 28, 1971.
The author is with the School of Engineering, University of California, Irvine, Calif. 92664.

Reply[2] *by William J. Vetter*[3]

Equation (1) of Rains was used in the original correspondence[1] primarily to exemplify some gradient matrix results. Mr. Rains asserts that the alternate expression for eigenvalue incrementals, his equation (2), is generally more efficient for computation than (1), but he does not support his claim with references to comparisons. In order to make a comparison, both expressions have been used for some sample computations.

Equation (1) was evaluated by use of Leverrier's algorithm (e.g., [1], [4]–[6]), roots of the characteristic polynomial so obtained as part of the routine being evaluated by a library subroutine which uses successive quadratic factorization by Bairstow iteration [7].

Equation (2) was evaluated by use of a standard eigenvalue–eigenvector subroutine which involves element scaling and a similarity transformation to upper Hessenberg form [8]. The matrix of left eigenvectors was obtained as the transposed inverse of the matrix of right eigenvectors.

The pertinent comparisons, for programming in WATFIV, double precision, and execution on an IBM 360/75 are given in Table I. Use of Fortran G/H would decrease execution time and core requirements, with an attendant increase in compiler time. The results in the table suggest that for matrices of order to about 12, the superiority of either method is at best marginal, and that additional criteria, such as error effects and success on ill-conditioned matrices, should be considered in a comparison. For matrices of large dimensions, the eigenvector approach is indeed superior, as suggested by Mr. Rains.

Additional pertinent references to the eigenvalue incremental problem are [6]–[11].

TABLE I

		Compile Time (s)	Execution Time (s)	Object Code (byte)	Array Area (byte)	Total Core (byte)
5 × 5, Real eigenvalues [12]	*a*	1.37	0.49	15 808	3144	18 952
	b	2.30	0.46	35 384	2536	37 920
5 × 5, Complex eigenvalues	*a*	1.37	0.53	15 808	3144	18 952
	b	2.30	0.53	35 384	2536	37 920
12 × 12, Complex eigenvalues [13]	*a*	1.37	5.21	15 808	24 136	39 944
	b	2.30	4.09	35 384	12 696	48 080
25 × 25, Complex eigenvalues [14]	*a*	1.37	3.13 min	15 808	167 144	182 952
	b	2.30	39.72	35 384	52 376	87 760

a—obtained using the Leverrier algorithm.
b—obtained using eigenvectors.
[·]—source of test matrices; incrementals were chosen as +10 percent on nonzero entries of the matrices.

ACKNOWLEDGMENT

The computational results of this note were obtained by J. Inglis.

REFERENCES

[1] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*. San Francisco, Calif.: Freeman, 1963, pp. 228–229.
[2] J. E. Van Ness, J. M. Boyle, and F. P. Imad, "Sensitivities of large, multiple-loop control systems," *IEEE Trans. Automat. Contr.*, vol. AC-10, pp. 308–315, July 1965.
[3] R. T. N. Chen and D. W. C. Shen, "Sensitivity and optimal control of multivariable systems," *1969 Joint Automatic Control Conf., Preprints*, pp. 467–468.
[4] F. R. Gantmacher, *The Theory of Matrices*, vol. 1. New York: Chelsea, 1959.
[5] J. S. Frame, "Matrix functions and applications, part IV," *IEEE Spectrum*, vol. 1, pp. 123–131, June 1964.
[6] B. S. Morgan, Jr., "Sensitivity analysis and synthesis of multivariable systems," *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 506–512, July 1966.
[7] J. H. Wilkinson, "The evaluation of the zeros of ill-conditioned polynomials; parts 1 and 2," *Numer. Math.*, vol. 1, pp. 150–180, 1959.
[8] ——, *The Algebraic Eigenvalue Problem*. Oxford, England: Clarendon Press, 1965, pp. 515–569, 619–637. (Algorithm 343 in Collected Algorithms of the A.C.M.)
[9] H. H. Rosenbrock, "Sensitivity of an eigenvalue to changes in the matrix," *Electron. Lett.*, vol. 1, p. 278, Dec. 1965.
[10] B. S. Morgan, Jr., "A computational procedure for the sensitivity of an eigenvalue," *Electron. Lett.*, vol. 2, p. 197, June 1966.
[11] D. C. Reddy, "Sensitivity of an eigenvalue of a multivariable control system," *Electron. Lett.*, vol. 2, p. 446, Dec. 1966.
[12] R. S. Martin and J. H. Wilkinson, "Reduction of the symmetric eigenproblem $Ax = \lambda BK$ and related problems to standard form," *Numer. Math.*, vol. 11, pp. 99–110, 1968.
[13] J. H. Wilkinson, "Rigorous error bounds for computed eigensystems," *Comput. J.*, vol. 4, pp. 230–241, 1961.
[14] P. J. Eberlein and J. Boothroyd, "Solution to the eigenproblem by a norm-reducing Jacobi type method," *Numer. Math.*, vol. 11, pp. 1–12, 1968.

[1] W. J. Vetter, *IEEE Trans. Syst. Man Cybern.* (Corresp.), vol. SMC-1, pp. 184–186, April 1971.
[2] Manuscript received November 1, 1971.
[3] The author is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, Nfld., Canada.

A Two-Level System of Stochastic Automata for Periodic Random Environments

KUMPATI S. NARENDRA AND R. VISWANATHAN

***Abstract*—A class of nonstationary environments with unknown but periodically changing probabilistic characteristics is considered. It is proposed to optimize the performance in an environment from this class by using a two-level system of variable-structure stochastic automata. The first level estimates the unknown period, while the second level operates suitably in the environment for one cycle, assuming that this estimate is the true period of the environment. The average output of the environment in this cycle is used as the input to the first level to determine the next estimate of the period. The optimal performance of this two-level system of automata in periodic random environments is demonstrated through computer simulations.**

Manuscript received July 26, 1971; revised November 5, 1971. This work was supported by the NSF under Grant GK-20580.
The authors are with the Becton Center, Yale University, New Haven, Conn.

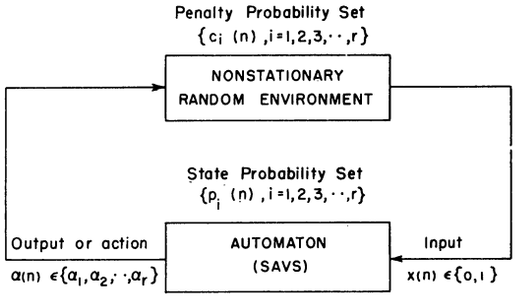

Fig. 1. Automaton–environment feedback configuration.

I. Introduction

In this correspondence we consider a two-level system of variable-structure stochastic automata for optimization of performance in a specific class of nonstationary environments, namely, those whose probabilistic characteristics change in time according to a periodic law with unknown period. The first level makes a decision on the value of this unknown period, and the second level accordingly arranges itself to operate in the random medium for one cycle. The average output of the environment in this cycle gives, in the expected sense, the relative worth of the decision made by the first level and is therefore used as the input to the first level to make the next decision on the value of the period.

The problem of automata functioning in stationary random environments has been considered very extensively in the past [1]–[9]. However, only a few authors concerned themselves with nonstationary environments [1], [10], [11]. The pioneering work of Tsetlin [1] in this area dealt with the performance of deterministic automata in random media whose probabilistic characteristics vary in accordance with a Markov chain. He established the existence of an optimal memory capacity (i.e., number of internal states) for the automaton. The performance of variable-structure stochastic automata in such nonstationary environments was investigated by Varshavskii and Vorontsova through computer simulations [2]. Recently, Varshavskii *et al.* [10] considered the behavior of deterministic automata in periodic random environments. Here this investigation is extended to the case of variable-structure stochastic automata.

II. Problem Statement: A Two-Level System of Automata

The problem that is considered here can be simply stated as follows. A nonstationary random environment with periodically changing (in time) probabilistic characteristics is given. The actual value of the period is unknown *a priori*, but its upper limit (maximum value) is given. The input set of the environment is finite. It is desired to determine that sequence of input values which minimizes the expected value of the average response of the environment over one period.

It is proposed to design a two-level system of variable-structure stochastic automata in the feedback path across the environment (see Fig. 1). The first level estimates the unknown period, while the second level operates suitably assuming that this estimate is the true period of the environment.

III. System Description

Fig. 1 depicts a closed-loop structure consisting of a stochastic automaton with a variable structure (SAVS) and a nonstationary random environment. The automaton at any time instant n performs an action α_i, $i \in \{1,2,\cdots,r\}$, governed by the probability set $\{p_1(n),\cdots, p_r(n)\}$. The environment in turn responds with a penalty ($x = 1$) with probability $c_i(n)$ and a nonpenalty ($x = 0$) with probability $1 - c_i(n)$. The environment is said to be periodic if the penalty probabilities $c_i(\cdot)$, $i = 1,2,\cdots,r$, are periodic functions of time. For simplicity, it is assumed that the automaton has only two actions (i.e., $r = 2$) and $c_i(\cdot)$, $i = 1,2$, are periodic piecewise-constant functions. Let c_{ni}, $i = 1,2$, denote penalty probability at time n for the action α_i. Thus the environment is completely characterized by the penalty probability set $\{c_{11},c_{12},c_{21},c_{22},\cdots,c_{T1},c_{T2}\}$, where T is the period of the functions $c_i(\cdot)$, $i = 1,2$. (As we have assumed that the environment operates in discrete time, the period T is necessarily an integer.) The expected output of the environment over one period $M(n)$ is defined as

$$M(n) = \frac{1}{T} \sum_{k=n}^{n+T-1} \sum_{i=1}^{2} \overline{p_i(k)} c_{ki} \tag{1}$$

where the overbar refers to expected value and $p_i(k)$ is the probability of performing the action α_i at time k. The asymptotic value of $M(n)$ is denoted as M: $M = \lim_{n\to\infty} M(n)$. Clearly, $M_{\min} \le M \le M_{\max}$, where

$$M_{\min} = \frac{1}{T} \sum_{j=1}^{T} \min \{c_{j1},c_{j2}\}$$

$$M_{\max} = \frac{1}{T} \sum_{j=1}^{T} \max \{c_{j1},c_{j2}\}.$$

The automaton is said to be *expedient* if

$$M < \frac{1}{2T} \sum_{j=1}^{T} (c_{j1} + c_{j2})$$

and *optimal* if $M = M_{\min}$. Discussions relating to the significance of expediency and optimality for the case of stationary environment apply to the present case also (see [6]–[8]).

It is proposed in this correspondence to obtain optimality using variable-structure stochastic automata. It is assumed that the penalty probability set is not known *a priori* so that the problem becomes nontrivial. When the period T of the environment is known *a priori*, the design of an optimal automaton becomes simple and reduces to organizing a switching pattern among a group of T automata, each of which is optimal in a stationary environment, in such a way that each automaton operates in a stationary environment [10]. When the period T is unknown but its maximum value $T_{\max}$ is known *a priori*, a two-level structure of automata is developed as follows. The first level consists of a single automaton having $T_{\max}$ actions and is responsible for making a decision about the value of the unknown period, while the second level comprising $T_{\max}$ automata is organized in such a way that if $T(n)$ is the period chosen by the first level, then the first $T(n)$ automata in the second level start operating in the environment in a sequential fashion, to be described later, and the arithmetic mean of the output values of the environment in these $T(n)$ operations is fed back to the first level as input for the next stage. The optimal performance of this two-level system of automata in periodic random environments is demonstrated through computer simulations.

IV. Optimal System of Automata

We shall assume that the period T of the environment is unknown. In the procedure suggested by Varshavskii *et al.* [10], a composite automaton consisting of two groups A and B, each having $T_{\max}$ deterministic automata, was used. Each automaton A_i in group A has two actions, 0 and 1. At every stage n, the number of automata in A which select the action 1 becomes the estimated value of the unknown period and those automata B_i in group B which correspond to A_i choosing the action 1 are connected[1] to the environment in sequence. The connection scheme is realized by means of a pair of rotating commutators [10]. The arithmetic mean of the responses of the environment in this one cycle is used as the penalty probability for the next stage for each of the automata in group A. Computer simulation results were presented to demonstrate the optimal performance of the composite automaton.

It is proposed in this correspondence to extend this approach to the case of variable-structure stochastic automata. The reason for doing this is that deterministic automata are at best only asymptotically optimal, i.e., optimal performance results only when the number of

[1] By "connection" we mean the following: the automaton under consideration chooses its output in accordance with its current state probability distributions; this output is then applied to the environment and the resulting response is given to the automaton as input for updating its state probability distributions. It should be noted that each automaton uses a prespecified optimal scheme (for details about optimal schemes refer to [8]) for updating its state probabilities.

states grow indefinitely large. First, exactly the same design as that just given, but using optimal variable-structure stochastic automata, was simulated on the computer. The results were not satisfactory. However, a modified design that will be described proved successful in computer simulations.

In the modified design, the group A has only one optimal S-model automaton having $T_{\max}$ actions, while the group B still has $T_{\max}$ optimal P-model automata. (For the discussion of P and S models, refer to [6], [9].) At any stage n, the automaton (or group) A estimates the value of the period by selecting a number $T(n)$ from 1 to $T_{\max}$. Accordingly, the first $T(n)$ automata $B_1,B_2,\cdots,B_{T(n)}$ in group B are connected to the environment in the following sequence. Assuming that the environment has a fixed period $T(n)$ and that the group B was operating in this environment from the beginning, the indices of the automata that have to be connected sequentially to the environment are computed. The procedure of connecting the automata in group B to the environment introduces the necessary synchronization for processing the past information very effectively, but it requires remembering the total number of time instants t_{n-1} up to the current stage n. The index l_n of the automaton in B to be connected first to the environment is given by

$$l_n = t_{n-1} - \left[\frac{t_{n-1}}{T(n)}\right] T(n) + 1(\text{mod } T(n)) \tag{2}$$

where $[x]$ refers to the largest integer contained in x. Thus the sequence of automata in B to be connected is

$$B_{l_n},B_{l_n+1},\cdots,B_{T(n)},B_1,B_2,\cdots,B_{l_n-1}.$$

The next value of time instant t_n is computed from the following:

$$t_n = t_{n-1} + T(n). \tag{3}$$

The arithmetic mean of the responses of the environment in the nth cycle lies in the closed interval [0,1] and is used directly as the input of the automaton for the $(n+1)$th cycle. It is assumed that whenever the input of an automaton is disconnected, its state (or action) probabilities remain unchanged. Here we refer to such a composite automaton as a two-level system of automata for obvious reasons.

Updating Schemes

Let each automaton B_i, $i = 1,2,\cdots,T_{\max}$, use the optimal updating scheme $N_{R-P}^{(2)}$ [8] and let the automaton A use the ε-optimal updating scheme SL_{R-I} [9].[2] Denote the output of the environment by x and the input of the automaton A by y and that of B_i by y_i. Define

$$q_i(n) = \Pr[\text{automaton } A \text{ chooses the output } i \text{ at the } n\text{th cycle}], \quad i = 1,2,\cdots,T_{\max} \tag{4}$$

$$p_{ij}(n) = \Pr[\text{automaton } B_i \text{ chooses the action } \alpha_j \text{ at the } n\text{th cycle}], \quad i = 1,2,\cdots,T_{\max},\ j = 1,2. \tag{5}$$

It should be noted that each automaton B_i in B gets connected to the environment at most once in one cycle. This accounts for the argument in $p_{ij}(\cdot)$ being chosen as the cycle number rather than the time instant. The updating algorithms for the proposed two-level system are as follows.

Automaton A:

$$y(n) = \frac{1}{T(n)} \sum_{i=1}^{T(n)} x(t_{n-1} + i) \tag{6}$$

$$q_{T(n)}(n+1) = q_{T(n)}(n) + \alpha[1 - y(n)][1 - q_{T(n)}(n)]$$

$$q_{i \neq T(n)}(n+1) = q_i(n) - \alpha[1 - y(n)]q_i(n), \quad i = 1,2,\cdots,T_{\max} \tag{7}$$

where $\alpha \in (0,1)$.

[2] Here the adjectives "optimal" and "ε-optimal" refer to the performance of the corresponding schemes in stationary random environments. The performance of an ε-optimal scheme can, by definition, be brought arbitrarily close to the optimum by a suitable choice of a parameter in the updating scheme. Thus, for practical purposes ε optimality is adequate.

Automaton B_i:

$$y_i(n) = x(t_{n-1} + i), \quad i = l_n, l_n + 1,\cdots,T(n),1,2,\cdots,l_n - 1 \tag{8}$$

where

$$l_n = t_{n-1} - \left[\frac{t_{n-1}}{T(n)}\right] T(n) + 1(\text{mod } T(n))$$

$$p_{ij}(n+1) = p_{ij}(n), \quad i = T(n)+1,\cdots,T_{\max}, \quad j = 1,2. \tag{9}$$

If the automaton B_i, $i = 1,2,\cdots,T(n)$, performs the action $\alpha_{j(i)}$, then

$$p_{ij(i)}(n+1) = p_{ij(i)}(n) + [\beta_1 - (\beta_1 + \beta_2)y_i(n)]p_{ij(i)}^{\beta}(n) \cdot [1 - p_{ij(i)}(n)]^{\beta+1}, \quad i = 1,2,\cdots,T(n) \tag{10}$$

where the parameters β_1, β_2, and β satisfy the following inequalities:

$$\beta \geq 2$$

$$0 < \beta_1 < 2^{2\beta}$$

$$0 < \beta_2 < \frac{(2\beta)^{2\beta}}{(\beta+1)^{\beta+1}(\beta-1)^{\beta-1}}. \tag{11}$$

Time Instant t_n:

$$t_n = t_{n-1} + T(n).$$

If the automata in group B use the L_{R-1} scheme [7], [8] for updating their state probabilities, then in the preceding the only change to be made occurs in (10). This equation, after incorporating the change, is

$$p_{ij(i)}(n+1) = p_{ij(i)}(n) + \beta[1 - y_i(n)][1 - p_{ij(i)}(n)], \quad i = 1,2,\cdots,T(n) \tag{12}$$

where $\beta \in (0,1)$.

An improvement in the scheme just described is possible. For choosing an estimate of the true period or its multiple, it is not necessary to consider all the numbers from 1 to $T_{\max}$. We can eliminate those numbers (excluding 1) which are submultiples of numbers in the range 1 to $T_{\max}$. For example, if $T_{\max}$ is 20, then it is sufficient to consider the 11 numbers: 1,11,12,$\cdots$,20. Using this procedure, the automaton A has a smaller number of actions, and so the convergence process may be speeded up. Also the action that corresponds to the true period is unique. In the original scheme, if $T = 7$ and $T_{\max} = 20$, then the actions 7 and 14 of the automaton A correspond to the true period, thus causing a nonunique answer. However, the computer simulations indicate that the sum of the probabilities of these actions goes to 1 as the number of cycles grows indefinitely large.

V. Computer Simulation Results

The purpose behind the simulation experiments to be described is to demonstrate the optimal performance in periodic random environments of the two-level system of variable-structure stochastic automata proposed in the previous section.

The following two problems are considered for simulation experiments.

Problem 1): $T = 2$, $T_{\max} = 4$. The penalty probabilities of the environment are

$$c_{11} = 0.125, \quad c_{12} = 0.875, \quad c_{21} = 0.875, \quad c_{22} = 0.125.$$

For achieving optimal performance in the environment described in problem 1), the first action should be performed at all odd instants of time and the second action at all even instants of time.

Problem 2): $T = 7$, $T_{\max} = 20$. The penalty probabilities of the environment are

$$c_{i1} = 0.25, \quad c_{i2} = 0.75, \quad i = 1,2,3$$

$$c_{j1} = 0.75, \quad c_{j2} = 0.25, \quad j = 4,5,6,7.$$

Thus, for achieving optimal performance in the environment described in problem 2), the first action should be chosen as the input for the first three time instants and the second action for the next four

time instants, this input pattern being repeated for all subsequent time instants.

In order to evaluate the performance of a two-level system of automata, a quantity M_1 is defined as follows:

$$M_1(n) = \frac{1}{T(n)} \sum_{(i,k)} \sum_{j=1}^{2} p_{ij}(n) c_{kj} \tag{13}$$

where M_1 is the average performance per cycle. If the estimate of the period $T(n)$ and the state probabilities of the automata in group B, $p_{ij}(n)$, $i = 1,2,\cdots,T(n)$, $j = 1,2$, are fixed at their present values, then $M_1(n)$ would be the expected value of the average output of the environment over the first $T(n)$ time instants. In (13) the indices i and k in the first summation run as follows:

$$i = l_n, l_n + 1,\cdots,T(n),1,2,\cdots, l_n - 1,$$

$$j = t_{n-1} + 1, t_{n-1} + 2,\cdots,t_n$$

where l_n and t_n are as defined in (2) and (3). Since in (13) the probabilities $p_{ij}(n)$ are random quantities, we should compute $E\{M_1(n)\}$ rather than $M_1(n)$ itself. In the computer simulations the basic experiment is run several times and the sample mean $\tilde{M}_1(n)$ of the values of $M_1(n)$ is computed for each n. If a given two-level system of automata performs in such a way that $\tilde{M}_1(n)$ approaches $M_{\min}$ as n becomes large, then, clearly, this two-level system performs optimally in the environment under consideration.

It should be pointed out that optimal updating schemes for variable-structure stochastic automata operating in stationary environments are available only for the two-state case [5], [8]. In view of this, the linear reward–inaction scheme SL_{R-I}, which is ε optimal and which is applicable to any r-state case, is used in the simulations for the first-level automaton A [8], [12]. ε optimality assures that there exist values of the step-size factor α (see (7)) such that the performance of the automaton can be made arbitrarily close to the optimum.

Extensive simulation studies were conducted[3] and it was found that the proposed two-level system performed optimally in periodic random environments. For purposes of illustration, the simulation results for the two aforementioned problems will be given. Each experiment was started with the state probabilities of the automaton A being equal to $1/T_{\max}$ and those of the automata B_i, $i = 1,2,\cdots,T_{\max}$, in group B being equal to $\frac{1}{2}$.

For problem 1), the experiment was run 50 times and the average values $\tilde{M}_1(n)$ were computed. An experiment was terminated either when the number of cycles exceeded 10 000 or when any state probability of automaton A as well as that of each automaton in group B reached 0.98. Two cases of two-level systems were simulated. Each automaton in group B used the ε-optimal L_{R-I} scheme [7], [12] for the first case and the optimal $N^{(2)}_{R-P}$ scheme [5], [8] for the second case. The step-size factors used for these schemes (refer to (7), (10), and (12)) are as follows.

Automaton A:

$$SL_{R-I}: \quad \alpha = 0.02.$$

Automata B_i, $i = 1,2,\cdots,T_{\max}$:

Case 1

$$L_{R-I}: \quad \beta = 0.005$$

Case 2

$$N^{(2)}_{R-P}: \quad \beta_1 = 0.3,\ \beta_2 = 0.2,\quad \beta = 2.$$

In both cases all 50 trials of the experiment were successful in the sense that they ended with automaton A choosing the true period or its multiple with a probability larger than 0.98 and the appropriate state probabilities of the automata in group B being larger than 0.98. The plot of M_1 versus time is given in Fig. 2 for both cases. In agreement with the relative convergence properties of the L_{R-I} and $N^{(2)}_{R-P}$ schemes, it is seen from Fig. 2 that the convergence for the second case is faster in the beginning and becomes slower at the end than the convergence for the first case. As far as the total simulation time is concerned, the second case consumed a little more than twice the time taken by the first.

[3] The direct coupled system of IBM 7090 and 7094 in the Yale University Computer Center, New Haven, Conn., was used for these simulations.

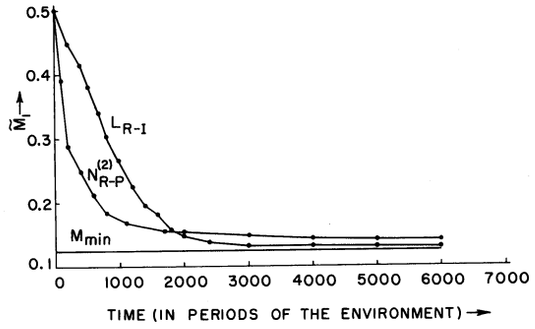

Fig. 2. Expected per cycle average performance versus time, problem 1.

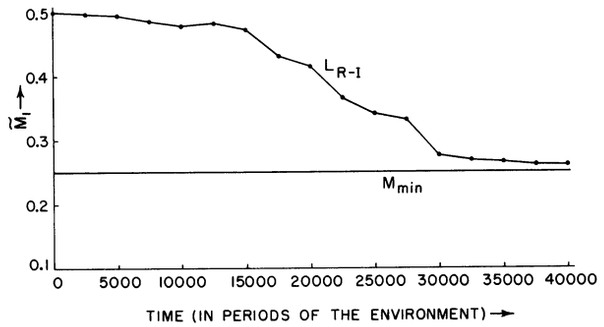

Fig. 3. Expected per cycle average performance versus time, problem 2.

For problem 2), the simulation experiment was run 10 times only, as the time consumed per experiment was found to be much larger than that for problem 1). For the same reason, only the case where the automata in group B use the L_{R-I} scheme was considered. For successful convergence it was necessary to choose quite small values for the step-size factors; they are as follows.

Automaton A:

$$SL_{R-I}: \quad \alpha = 0.003.$$

Automata B_i, $i = 1,2,\cdots,T_{\max}$:

$$L_{R-I}: \quad \beta = 0.001.$$

All 10 trials of the experiment were successful and the plot $\tilde{M}_1$ versus time is given in Fig. 3.

VI. Conclusions

Problems involving nonstationary environments have not been satisfactorily treated in the past. This correspondence, extending the ideas suggested in a previous work, has considered a specific class of nonstationary environments, namely, periodic random environments, and investigated the behavior of variable-structure stochastic automata in such environments.

If the investigation of multilevel systems of stochastic automata yields some positive results, then the use of stochastic automata in multimodal search would become a potent tool in many applications. To this end, a specific two-level system of variable-structure stochastic automata has been considered. The optimal performance of this two-level system in unknown periodic random environments with a known upper limit on its period has been demonstrated through computer simulations.

References

[1] M. L. Tsetlin, "On the behavior of finite automata in random media," *Avtomat. Telemekh.*, vol. 22, pp. 1345–1354, Oct. 1961.
[2] V. I. Varshavskii and I. P. Vorontsova, "On the behavior of stochastic automata with a variable structure," *Avtomat. Telemekh.*, vol. 24, pp. 353–360, Mar. 1963.
[3] K. S. Fu and G. J. McMurtry, "A study of stochastic automata as models of adaptive and learning controllers," Purdue Univ., Lafayette, Ind., Tech. Rep. TR-EE65-8, 1965.

[4] K. S. Fu and R. W. McLaren, "An application of stochastic automata to the synthesis of learning systems," Purdue Univ., Lafayette, Ind., Tech. Rep. TR-EE65-17, 1965.
[5] I. P. Vorontsova, "Algorithms for changing stochastic automata transition probabilities," *Probl. Peredach. Inform.*, vol. 1, no. 3, pp. 122–126, 1965.
[6] B. Chandrasekaran and D. W. C. Shen, "On expediency and convergence in variable-structure automata," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, pp. 52–60, Mar. 1968.
[7] I. J. Shapiro and K. S. Narendra, "Use of stochastic automata for parameter self-optimization with multimodal performance criteria," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-5, pp. 352-360, Oct. 1969.
[8] R. Viswanathan and K. S. Narendra, "On variable-structure stochastic automata," in *Pattern Recognition and Machine Learning* (Proceedings of the Japan–U.S. Seminar on Learning Process in Control Systems, 1970). New York: Plenum Press, 1971, pp. 277–287.
[9] ——, "Application of stochastic automata models to learning systdms with multimodal performance criteria," Becton Cent., Yale Univ., New Haven, Conn., Tech. Rep. CT-40, June 1971.
[10] V. I. Varshavskii, M. V. Meleshina, and M. L. Tsetlin, "Behavior of automata in periodic random media and the problem of synchronization in the presence of noise," *Probl. Peredach. Inform.*, vol. 1, no. 1, pp. 65–71, 1965.
[11] B. Chandrasekaran and D. W. C. Shen, "Adaptation of stochastic automata in nonstationary environments," in *Proc. Nat. Electron. Conf.*, vol. 23, pp. 39–44, 1967.
[12] R. Viswanathan and K. S. Narendra, "A note on the linear reinforcement scheme for variable-structure stochastic automata," Becton Cent., Yale Univ., New Haven, Conn., Tech. Rep. CT-42, July 1971.

Comments on "Use of Stochastic Automata for Parameter Self-Optimization with Multimodal Performance Criteria"

IAN H. WITTEN

***Abstract*—In the above paper,[1] an optimal method of self-optimization of certain system parameters using noisy binary-valued performance feedback is extended, without losing optimality, to situations with many-valued performance feedback. The effect of time-varying feedback mechanisms is briefly considered.**

In the above paper[1] Shapiro and Narendra considered the problem of on-line self-optimization of a set of parameters $\{\alpha\}$ contained in a given system, the performance of which was only available in a form corrupted by noise. Thus measurements $g(\alpha,z)$, where z is a random quantity, are available, and the aim is to find a value for α which maximizes $I(\alpha) = E[g(\alpha,z)]$, the expected value of g. They called a self-optimization algorithm *optimal* if it eventually chose, with probability 1, the optimum value for the parameter set, and presented an optimal self-optimization algorithm for the case where measurements of the system's performance were limited to the values 0 and 1—a penalty/nonpenalty situation. It is clearly of interest to extend this algorithm to systems with more general performance functions, and although Shapiro and Narendra claim to have accomplished this without sacrificing optimality (see the appendix[1]), this correspondence will show that their extension is not, in fact, optimal, but that such an optimal extension can be made by introducing an additional random element. Some comments will also be made on the use of the optimal algorithm in situations where the performance evaluation function I varies with time.

The set of parameters $\{\alpha\}$ is assumed to have r possible values α_i. Restricting attention for the moment to the penalty/nonpenalty situation, let

$$C_i = \Pr[\text{action } \alpha_i \text{ causes a penalty response}] = \Pr[g(\alpha_i,z) = 0].$$

We assume that the C_i completely characterize the performance evaluation mechanism, so that successive performance measures are statistically independent. The "linear reinforcement scheme" of Shapiro and Narendra chooses the parameter value α_i with probability p_i and updates the p_j according to the consequent performance measure g, in accordance with the following:

1) if $g = 0$, do not change any p_j;
2) if $g = 1$, then

$$p_i \leftarrow \gamma p_i + 1 - \gamma$$
$$p_j \leftarrow \gamma p_j, \qquad \text{for all } j \neq i$$

where α_i is the α value last chosen.

Here, $\gamma \in (0,1)$ is a constant controlling the rate of adaptation and the vulnerability of the p_j to noise. This self-optimization algorithm is optimal, as was elegantly shown by Shapiro and Narendra, provided only that initial values for the p_j are chosen in the open interval (0,1).

In order to extend the algorithm to situations where $g(\alpha_i,z)$ may take on any values, Shapiro and Narendra propose storing, for each α_i, the mean value g_i^* of $g(\alpha_i,z)$ observed so far. Then, if α_i is chosen and the consequent performance measure is g, a secondary measure g' is computed by

$$g' = \begin{cases} 1, & \text{if } g \geq g_i^* \\ 0, & \text{otherwise.} \end{cases}$$

This new measure is used as the performance measure in the preceding algorithm. (The ith mean estimator g_i^* is, of course, updated by g using a conventional mean estimation procedure.) Thus this extension of the algorithm acts as a preprocessor which transforms the many-valued performance measures g into binary-valued ones g'.

It is possible to produce an example which shows that this extended self-optimization algorithm is not optimal in the sense described earlier. Suppose that α has only two possible values, α_1 and α_2. Let $g(\alpha,z)$ have the following form:

$$g(\alpha_1) = \begin{cases} 1, & \text{with probability } 1/4 \\ 7/15, & \text{with probability } 3/4 \end{cases}$$

$$g(\alpha_2) = \begin{cases} 2/3, & \text{with probability } 3/4 \\ 0, & \text{with probability } 1/4. \end{cases}$$

Then the performance evaluation function $I(\alpha) = E[g(\alpha,z)]$ is

$$I(\alpha_1) = 0.6 \qquad I(\alpha_2) = 0.5$$

and thus an optimal algorithm will eventually choose α_1 with probability 1. However, consider the preceding algorithm. Suppose that the process has been going on long enough for accurate estimations g^* of the mean performance measure to have been made

$$g_1^* = 0.6 \qquad g_2^* = 0.5$$

and these estimates are not significantly changed by future experience (this can obviously be made more precise using the law of large numbers). Then

$$E[g'(\alpha_1,z)] = \Pr[g'(\alpha_1,z) = 1] = \Pr[g(\alpha_1,z) > g_1^*] = 1/4$$

$$E[g'(\alpha_2,z)] = \Pr[g'(\alpha_2,z) = 1] = \Pr[g(\alpha_2,z) > g_2^*] = 3/4.$$

Thus the binary-valued performance function g' favors the parameter value α_2 instead of the value α_1 favored by the original function g. Hence the extended algorithm will converge to the wrong α value.

Consider now the following system for computing a binary-valued performance measure g'' from a many-valued one g. The g' are assumed to be normalized to lie in the interval [0,1]. (Normalization is easy if the original g are known to lie within certain limits. We may dispense with this restriction by observing that the only conditions necessary on the normalization function are a) monotonicity and b) that the image of the function is contained in [0,1]. Thus a function such as $x \rightarrow e^x/(1 + e^x)$, which maps the entire real line monotonically into [0,1], will do.) The new performance measure g'' is computed by

$$g'' = \begin{cases} 1, & \text{with probability } g \\ 0, & \text{with probability } 1 - g. \end{cases}$$

Manuscript received August 9, 1971.
The author is with the Department of Electrical Engineering Science, University of Essex, Colchester, Essex, England.
[1] I. J. Shapiro and K. S. Narendra, *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-5, pp. 352–360, Oct. 1969.