[9] ——, "Computational techniques for analysis of system dynamics models of social systems," *J. Socio-Economic Planning Sci.*, vol. 4, Oct. 1974.
[10] J. R. Burns, "Error analysis of nonlinear simulations: Applications to world dynamics," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-5, pp. 331–340, May 1975.
[11] ——, "A preliminary approach to automating the process of simulation model synthesis," *Proc. Seventh Pittsburgh Conf. Modeling and Simulation*, Apr. 1976.
[12] J. N. Warfield, "Binary matrices in system modeling," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 441–449, Sept. 1973.
[13] ——, "Developing subsystem matrices in structural modeling," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 74–80, Jan. 1974.
[14] ——, "Developing interconnection matrices in structural modeling," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 81–87, Jan. 1974.
[15] ——, "Toward interpretation of complex structural models," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 405–417, Sept. 1974.
[16] M. D. Mesarovic, D. Macko, and Y. Takahara, *Theory of Hierarchical, Multilevel, Systems.* New York: Academic, 1970.
[17] F. Harary, R. Norman, and D. Cartwright, *Structural Models: An Introduction to the Theory of Directed Graphs.* New York: Wiley, 1965.
[18] M. R. Goodman, *Study Notes in System Dynamics.* Cambridge, MA: Wright-Allen Press, 1974.
[19] J. R. Burns, "Toward a mathematically rigorous methodology for simulation of social processes," *Proc. Summer Computer Simulation Conf.*, pp. 1139–1147, July 1975.
[20] J. Kane, "A primer for a new cross-impact language KSIM," *Technology Forecasting and Social Change*, vol. 4, pp. 129–142, 1972.
[21] J. Kane *et al.*, "A methodology for interactive resource policy simulation," *J. Water Resources Res.*, vol. 9, no. 1, pp. 65–79, Feb. 1973.
[22] J. R. Burns *et al.*, "Causality: Its characterization by methodologies for modeling socioeconomic systems," to be published.
[23] G. J. Klir, *An Approach to General Systems Theory.* New York: Van Nostrand Reinhold, 1969.
[24] R. Fitz and D. Hornbach, "A participative methodology for designing dynamic models through structural models," *Proc. Seventh Annual Pittsburgh Conf. Modeling and Simulation*, Apr. 1976.

A Fuzzy Logic Controller for a Traffic Junction

C. P. PAPPIS AND EBRAHIM H. MAMDANI, MEMBER, IEEE

Abstract—**Work done on the implementation of a fuzzy logic controller in a single intersection of two one-way streets is presented. The model of the intersection is described and validated, and the use of the theory of fuzzy sets in constructing a controller based on linguistic control instructions is introduced. The results obtained from the implementation of the fuzzy logic controller are tabulated against those corresponding to a conventional effective vehicle-actuated controller. With the performance criterion being the average delay of vehicles, it is shown that the use of a fuzzy logic controller results in a better performance.**

INTRODUCTION

A CONSIDERABLE amount of work has been done on the problem of modeling and controlling traffic junctions. Although the major problem in cities concerns sets of intersections (not individual ones), any approach to this problem should also include a sufficient description of the events occurring in any individual intersection in the linked or disjoint system under study.

Zadeh's pioneering work on fuzzy sets, by which a conceptual framework is provided for dealing with problems of vagueness in the representation of complex processes, can be of great help to the task of constructing a controller for such an individual traffic intersection. Indeed, the strength of the theory of fuzzy sets lies in its capability of rendering a powerful conceptual basis for the modeling and analysis of such processes, to which the human approach is characterized by rough approximations. Note that, although stochastic and fuzzy logics can both be regarded as derived from a probability logic [1], a stochastic approach would be methodologically different from the fuzzy discipline which has been used here. It seems, therefore, that the fuzzy rather than the stochastic approach should be used as the domain for the implementation of heuristics.

Previous work reported in the literature (e.g. [2]–[5]) has shown the merits of the theory of fuzzy sets when applied to the design of controllers for real dynamic plants, industrial processes, etc. In this study, the system is a traffic junction and the problem of its control is considered as a classical example of nonprogrammed decisionmaking, i.e., decisionmaking characterized by the lack of well-specified analytical means for coping with a particular problem. Thus a linguistic control algorithm is synthesized, capable of dealing with a continuously reproduced decisionmaking situation. The starting point is an adequate (though qualitative) knowledge of the system and a protocol of control instructions used by a human operator. A fuzzy set theoretic representation of these instructions (which we call "a fuzzy logic controller") was tried as an answer to the control modeling problem, which gave very satisfactory results.

The work done on the construction of the model of the system and the implementation of the fuzzy logic controller is presented below. In order to validate the model a fixed-cycle controller was also simulated. The average delays of the vehicles resulted from the implementation of the fuzzy logic controller were compared to those caused by an

Manuscript received February 10, 1976; revised June 21, 1976, November 15, 1976, and April 21, 1977.
C. P. Pappis is with the Agricultural Bank of Greece, Athens, Greece.
E. H. Mamdani is with the Department of Electrical and Electronic Engineering, Queen Mary College, University of London, London E1 4NS, England.

TABLE I
AVERAGE DELAYS WITH FIXED-CYCLE CONTROLLER AND OPTIMUM SETTINGS

N - S traffic (veh/hr)	E - W traffic (veh/hr)	Delay (secs/veh) Model (d_m)	Formula (d_t)	Error %
360	360	7.2	7.4	- 2
360	720	7.4	7.9	- 6
360	1080	7.9	8.4	- 5
360	1440	8.4	9.0	- 7
360	1800	9.3	10.2	- 9
360	2160	12.3	12.9	- 5
360	2520	15.8	18.9	-17
720	720	9.7	10.0	- 3
720	1080	10.8	11.6	- 7
720	1440	12.7	13.8	- 8
720	1800	15.9	17.3	- 8
720	2160	21.8	24.9	-12
1080	1080	13.6	14.9	- 9
1080	1440	17.9	19.7	- 9
1080	1800	25.8	29.2	-11
1440	1440	27.3	30.7	-11

efficient vehicle-actuated one. The results obtained show that the performance of the system is better under the fuzzy logic controller.

THE MODEL

The major assumption concerning the model is that the arrival of vehicles at the intersection is considered as being random. More explicitly, arrival times are considered to be uniformly distributed. Note that this assumption affects not only the truthfulness of the model but the selection of the control policies as well. The cycle is divided into two periods of "effective red" and "effective green" for each phase; the first corresponding to the halted traffic and the second to the traffic having the right of way. A total lost time of 10 s/cycle is assumed. Vehicles leave the queue at a constant rate equal to the saturation flow during the effective green (see [6], [7] for definitions). The saturation flow equals 3600 vehicles/h at both arms. There is no turning traffic.

For each successive time unit (1 s) a pseudorandom number is generated and compared to some fixed quantity, which is equal to the mean rate of arrival (in vehicles per second). Thus the arrival of a vehicle is decided. Let

$$q_n = \begin{cases} 1, & \text{if a vehicle arrived during the } n\text{th unit interval} \\ 0, & \text{otherwise.} \end{cases}$$

If Q_G denotes the number of vehicles not cleared during the previous effective green period of a phase, then the queue Q_n at the nth time unit after the beginning of the effective red of that phase would be

$$Q_n = Q_G + \sum_{n_1=1}^{n} q_{n_1},$$

and the total waiting time of the vehicles in the queue would be

$$D_{n,R} = \sum_{n_2=1}^{n} \left(Q_G + \sum_{n_1=1}^{n_2} q_{n_1} \right).$$

Let s be the saturation flow, i.e., the rate at which vehicles are cleared during the effective green period. At the nth time unit after the beginning of the effective green, the number of vehicles not yet cleared would be

$$S_n = z \cdot \left(Q_R + \sum_{n_1=1}^{n} q_{n_1} - s \cdot n \right),$$

where Q_R is the queue which was built up during the previous effective red period of the phase, and z is equal to one when multiplied by a nonnegative quantity and zero otherwise.

These vehicles have been subjected to a delay

$$D_{n,G} = \sum_{n_1=1}^{n} z \cdot \left(Q_R + \sum_{n_2=1}^{n_1} q_{n_2} - s \cdot n_1 \right).$$

Thus during a cycle, the total delay experienced by vehicles traveling along one arm of the intersection would be

$$D = D_{R,R} + D_{G,G},$$

where $D_{R,R}, D_{G,G}$ are the delays during R and G, i.e., the whole effective red and green periods, respectively. Finally the average delay per vehicle would be

$$d_m = \frac{D}{\sum_{n=1}^{R+G} q_n}.$$

The model just described is quite simple in comparison to some more sophisticated ones (see, for example, [8], [9]), yet it suffices for the purpose of this work. A measure of its reliability was obtained by using a fixed-cycle controller which was implemented to the system. The system was subjected to a wide range of averages of random vehicle arrivals. Each time it was run for 7200 simulated seconds and the corresponding average delay per vehicle was calculated. Results of the calculations, together with the expected average delays, are given in Table I. The expected ones have been obtained from the following formula (see [6])

$$d_t = \frac{C(1-\lambda)^2}{2(1-\lambda X)} + \frac{X^2}{2_q(1-X)} - 0.65\left(\frac{C}{q^2}\right)^{1/3} X^{(2+5\lambda)},$$

where

- d_t average delay per vehicle on the particular arm,
- C cycle time,
- λ proportion of the cycle which is effective green for the phase under consideration (g/C),
- q flow,
- s saturation flow (i.e., the flow which would be obtained if there were an endless queue of vehicles and they were given a continuous green),
- X degree of saturation ($q/\lambda s$).

It can be verified that d_m and d_t yield comparable delays. Indeed, Table I shows a fair agreement between the calculated delays and those obtained from the above formula, thus providing a validation of the model.

The results of Table I actually correspond to optimum settings, i.e., optimum cycle and effective green times for the respective flow rates according to

$$C_0 = \frac{1.5L + 5}{1 - Y}$$

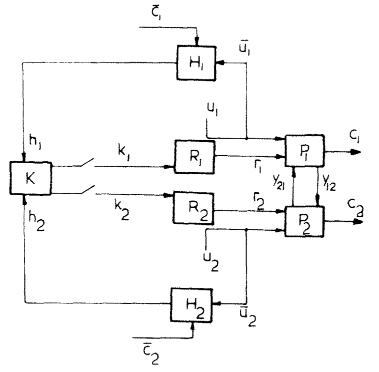
$$g_i = \frac{Y_i}{Y}(C_0 - L), \qquad i = 1,2,$$

where

- C_0 optimum cycle time,
- g_i effective green time, $i = 1,2$,
- Y_i $= q_i/s$, where $i = 1,2$,
- Y $= Y_1 + Y_2$,
- L total lost time per cycle (10 s).

It should be noted that the delays of Table I correspond to random arrivals having fixed averages. In other words, should the average rates of arrival be different from those for which optimum settings have been found and used for the control of the intersection, the resulting delays would also be different from those of Table I. The use of timers, for tuning the controller in order to adjust its settings to the daily flow pattern, would not be an easy task. Consequently, the actual delays occurring in any intersection controlled by fixed-cycle controllers would be by far in excess of those shown in Table I, especially in cases of heavy traffic.

In the case of vehicle-actuated controllers, the results of Table I correspond to those that should be expected with an efficient vehicle-actuated installation [7]. That is, a vehicle-actuated controller with speed timing or with a low fixed extention operation would result in delays as those of Table I for the respective flow rates. These delays were the basis for the comparison between vehicle-actuated controllers and the fuzzy logic one (a fixed-cycle controller for a single intersection is scarcely worthy of comparison).



- P_i processing of vehicles within the junction along route i,
- R_i regulating function (signal setting),
- H_i data processing,
- K optimizing function (control algorithm),
- u_i vehicles on route i entering the junction,
- c_i vehicles on route i leaving the junction,
- y_{ij} subprocess's P_i interaction with subprocess P_j,
- $\bar{c}_i$ premeasured data concerning c_i (saturation flow),
- $\bar{u}_i$ information input concerning oncoming vehicles.

Fig. 1. System control process.

The system control process is shown in Fig. 1. The intervention of the controller takes place every 10 s during each phase's effective green period; the first intervention taking place just after the first 7 s of the period. At each intervention the length of the extension of the effective green time for the phase having the right of way is decided. Information concerning the flow pattern is collected by detection pads, which, it is assumed, have been installed before the traffic lights in both arms of the intersection. The role of the detection pads is very important, as will be made clear in the sequel. It was assumed for the calculations that the flow pattern, as detected at the pads, is preserved throughout the period after each intervention for the phase having the right of way. The distance between the pads and the stop lines is sufficient for the controller to be informed about the arrivals of vehicles in both arms of the intersection during the next $11\frac{1}{2}$ s, assuming that the effective green ends at the middle of the 3-s amber period.

Thus vehicle i passing over the detectors is registered in the following way. Its speed v_i is calculated. Assuming that its speed is preserved constant during its trip from the detectors to the junction, vehicle i will be at the "critical point" in ($l/v_i - 1.5$ s) time (see Fig. 2). The "critical point" is the point where, should the lights turn to amber, it would be possible for the vehicle just to pass. Let

$$N_i = l/v_i - 1.5 \text{ s}$$

be the number of seconds required for the vehicle to arrive at the critical point. N_i indicates the position of vehicle i in the flow pattern array for the next 10-s interval. The control input parameters are two continuously updated arrays

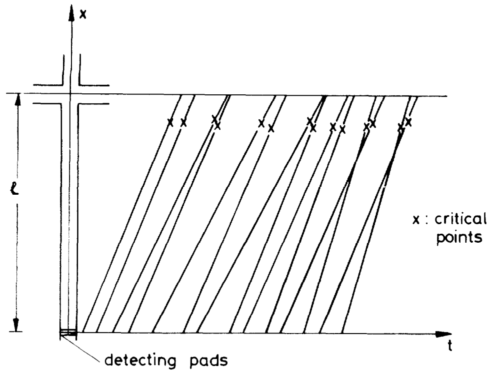


Fig. 2. Time-space diagram.

corresponding to the halted traffic and the traffic having the right of way.

The Fuzzy Logic Controller

In order to make our exposition self-contained, some of the basic definitions of the theory of fuzzy sets ([10], [11]) that were used to model the control algorithm are given below.

A fuzzy set F of a universe of discourse $U = \{x\}$ is defined as a mapping $\mu_F(x)$: $U \rightarrow [0,1]$ by which each x is assigned a number in $[0,1]$ indicating the extent to which x has the attribute F. Thus if x is the number of vehicles in a queue, then "small" may be considered as a particular value of the fuzzy variable "queue" and each x is assigned a number $\mu_{\text{small}}(x) \in [0,1]$ which indicates the extent to which that x is considered to be small.

Given the fuzzy sets A, B, or U, the basic operations on A,B are

i) the complement $\bar{A}$ of A, defined by

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x),$$

ii) the union $A \cup B$ of A and B, defined by

$$\mu_{A \cup B}(x) = \max \{\mu_A(x), \mu_B(x)\},$$

iii) the intersection $A \cap B$ of A and B, defined by

$$\mu_{A \cap B}(x) = \min \{\mu_A(x), \mu_B(x)\}.$$

A fuzzy relation R from $U = \{x\}$ to $V = \{y\}$ is a fuzzy set on the Cartesian product $U \times V$, characterized by a function $\mu_R(x,y)$, by which each pair (x,y) is assigned a number in $[0,1]$ indicating the extent to which the relation R is true for (x,y). There are several ways of constructing $\mu_R(x,y)$. The one used here will be seen later.

Finally given a fuzzy relation R from U to V and a fuzzy set A on U, a fuzzy set B on V is induced, given by the compositional rule of inference

$$B = A \circ R$$

or

$$\mu_B(y) = \max_x \{\min \{\mu_R(x,y), \mu_A(x)\}\}.$$

A heuristic approach to the control problem was employed, which resulted in a set of linguistic control statements. The above basic ideas of the theory of fuzzy sets were used for the quantitative interpretation of these instructions as well as the decisionmaking process.

The fuzzy control instructions (see Appendix for a complete set) are of the form

if T = medium
and $A = mt$(medium)
and $Q = lt$(small)
then E = medium

else

if T = long
and $A = mt$(many)
and $Q = lt$(medium)
then E = long

else, etc. where

- T the fuzzy variable "time," which is assigned values like "very short," "short," "medium," etc.,
- A the fuzzy variable "arrivals," i.e., the number of vehicles arriving at the arm having the right of way, which may be assigned values like "many," "more than a few," etc.,
- Q the fuzzy variable "queue," which is assigned values like "any," "less than small," etc.,
- E the fuzzy variable "extension," which has values identical to "time."

The terms "medium," "more than medium," "less than small," etc., are labels of fuzzy sets defined on the relevant universes of discourse T,A,Q,E. Tables II–IV show the fuzzy sets used in this application. Further to the above basic operations, in this application we have introduced the operators mt and lt, standing for "more than" and "less than," respectively. These are defined as follows. If A is a fuzzy set defined on the real line $R^1 = \{x_i\}$, $\mu_A(x_i)$ is its grade of membership function and x_0 is the element of R^1 for which $\mu_A(x_i)$ is maximum, then $lt(A)$ and $mt(A)$ are fuzzy sets defined on U such that

$$\mu_{lt(A)}(x_i) = \begin{cases} 0, & \text{for } x_i \geq x_0 \\ 1 - \mu_A(x_i), & \text{for } x_i < x_0 \end{cases}$$

$$\mu_{mt(A)}(x_i) = \begin{cases} 0, & \text{for } x_i \leq x_0 \\ 1 - \mu_A(x_i), & \text{for } x_i > x_0. \end{cases}$$

The result of these operations on the fuzzy sets of Tables III and IV above is shown in Tables V and VI. Obviously

$$lt(A) \quad \text{or} \quad mt(A) = \text{not } (A)$$
$$lt(A) \quad \text{and} \quad mt(A) = 0$$

or

$$\max \{\mu_{lt(A)}(x_i), \mu_{mt(A)}(x_i)\} = 1 - \mu_A(x_i)$$
$$\min \{\mu_{lt(A)}(x_i), \mu_{mt(A)}(x_i)\} = 0.$$

Note that if a fuzzy assignment like "A = small" is characterized by the poor content of the information conveyed, a fuzzy assignment like "A = less than small" is

TABLE II
FUZZY SETS DEFINED ON TIME (OR EXTENSION)

Fuzzy sets \ time (secs)	1	2	3	4	5	6	7	8	9	10
very short	1	.5	0	0	0	0	0	0	0	0
short	0	.5	1	.5	0	0	0	0	0	0
medium	0	0	0	.5	1	.5	0	0	0	0
long	0	0	0	0	0	.5	1	.5	0	0
very long	0	0	0	0	0	0	0	.5	1	1

TABLE III
FUZZY SETS DEFINED ON ARRIVALS

F. sets \ Arrivals (veh)	1	2	3	4	5	6	7	8	9	10
none	.5	.2	.1	0	0	0	0	0	0	0
a few	1	.5	.2	.1	0	0	0	0	0	0
few	.5	1	.5	.2	.1	0	0	0	0	0
medium	.2	.5	1	.5	.2	.1	0	0	0	0
many	.1	.2	.5	1	.5	.2	.1	0	0	0
too many	0	.1	.2	.5	1	.5	.2	.1	0	0

TABLE IV
FUZZY SETS DEFINED ON QUEUES

f.sets \ queue (veh)	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
very small	0	.5	.7	.9	.1	.9	.7	.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
small	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
small plus	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0	0	0	0	0	0	0	0	0
medium	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0	0	0	0	0
long	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0
very long	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0

TABLE V
"MORE THAN"-OPERATION ON FUZZY SETS OF TABLE III

fuzzy sets \ Arrivals (vehicles)	1	2	3	4	5	6	7	8	9	10
mt(none)	.5	.8	.9	1	1	1	1	1	1	1
mt(a few)	0	.5	.8	.9	1	1	1	1	1	1
mt(few)	0	0	.5	.8	.9	1	1	1	1	1
mt(medium)	0	0	0	.5	.8	.9	1	1	1	1
mt(many)	0	0	0	0	.5	.8	.9	1	1	1
mt(too many)	0	0	0	0	0	.5	.8	.9	1	1

TABLE VI
"LESS THAN"-OPERATION ON FUZZY SETS OF TABLE IV

f. sets \ queue (veh)	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
lt(very small)	1	.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lt(small)	1	1	1	1	1	.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lt(small plus)	1	1	1	1	1	1	1	1	1	.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lt(medium)	1	1	1	1	1	1	1	1	1	1	1	1	1	.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0
lt(long)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	.5	.3	.1	0	0	0	0	0	0	0	0	0
lt(very long)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	.5	.3	.1	0	0	0	0	0

conveying even less information. In other words, fuzzy assignments like "A = less than small" are used whenever the grade of fuzziness is high.

"Any" is considered as a fuzzy set with all the elements of its universe of discourse assigned a grade of membership equal to one.

A total of 25 rules were used (5 for each intervention). Every rule is a fuzzy relation between the inputs T, A, Q, and the output E. The connectives "and" and "else" are interpreted as the operators "min" and "max" respectively. Thus

T = very short
and $A = mt$(none)
and Q = any

is a fuzzy phrase P (see [12]) defined on the universe of discourse $T \times A \times Q$ with grades of membership function

$$\mu_P(t,a,q) = \min\{\mu_{\text{v.short}}(t), \mu_{mt(\text{none})}(a), \mu_{\text{any}}(q)\}.$$

The fuzzy implication "if P then E = very short" is also a fuzzy phrase R defined on $T \times A \times Q \times E$ with grades of membership function

$$\mu_R(t,a,q,e) = \min\{\mu_{\text{v.short}}(t), \mu_{mt(\text{none})}(a)\mu_{\text{any}}(q), \mu_{\text{v.short}}(e)\}.$$

Finally two or more fuzzy implications $R, S, \cdots$, connected by "else" form a fuzzy clause C defined on $T \times A \times Q \times E$ with grades of membership function

$$\mu_C(t,a,q,e) = \max\{\mu_R(t,a,q,e), \mu_S(t,a,q,e), \cdots\}.$$

In this application, since each fuzzy rule is represented by a four-dimensional array, the fuzzy algorithm employed at each intervention for deciding the control action is represented by the union of five such arrays, as five rules operate at each intervention. 25(5 × 5) rules given in the Appendix thus provide for a maximum of five interventions (each consisting of five rules) taking place at 7th, 17th, 27th, 37th, and 47th s. Thus the maximum possible effective green time is 57 s. At each intervention the five rules are invoked in the manner described below ten times (i.e., for each of the next 10 s). Note that, as the detecting pads are sufficiently far away from the junction, at each intervention, data is available for each of the next 10 s. Consider now the (t_i,a_j,a_k,e_1) entry of the array C_2, corresponding to the algorithm used at the second intervention of the controller, where

$t_i = 8$ i.e., we consider the next 8 s,

$a_j = 4$ i.e., 4 vehicles will cross the critical point if no change of the current state of the system occurs during the next 8 s,

$q_k = 5$ i.e., five vehicle queue will build up if no change of the current state of the system occurs during the next 8 s,

$e_1 = 8$ i.e., the extension given to the present state of the system is 8 s.

The first control statement R_1 for the second intervention (see Appendix) is

if T = very short
and $A = mt$(none)
and Q = any
then E = very short.

From Tables II, V, VI we have

$$\mu_{\text{v.short}}(8) = 0.0$$
$$\mu_{mt(\text{none})}(4) = 1.0$$
$$\mu_{\text{any}}(5) = 1.0.$$

Thus

$$\mu_{R_1}(8,4,5,8) = \min\{\mu_{\text{v.short}}(8),\mu_{mt(\text{none})}(4),\mu_{\text{any}}(5),\mu_{\text{v.short}}(8)\} = \min\{0,1.0,1.0,0\} = 0.$$

Similarly we find

$$\mu_{R_2}(8,4,5,8) = \min\{\mu_{\text{short}}(8),\mu_{mt(\text{a few})}(4),\mu_{lt(\text{v.short})}(5),\mu_{\text{short}}(8)\} = \min\{0,0.9,0.5,0\} = 0$$

$$\mu_{R_3}(8,4,5,8) = \min\{\mu_{\text{medium}}(8),\mu_{mt(\text{few})}(4),\mu_{lt(\text{v.short})}(5),\mu_{\text{medium}}(8)\} = \min\{0,0.8,0.5,0\} = 0$$

$$\mu_{R_4}(8,4,5,8) = \min\{\mu_{\text{long}}(8),\mu_{mt(\text{medium})}(4),\mu_{lt(\text{v.short})}(5),\mu_{\text{long}}(8)\} = \min\{0.5,0.5,0.5,0.5\} = 0.5$$

$$\mu_{R_5}(8,4,5,8) = \min\{\mu_{\text{v.long}}(8),\mu_{mt(\text{many})}(4),\mu_{lt(\text{short})}(5),\mu_{\text{v.long}}(8)\} = \min\{0.5,0,1,0.5\} = 0.$$

Thus the (t_i,a_j,q_k,e_1) entry of matrix C_2 is

$$\mu_{C_2}(t_i,a_j,q_k,e_1) = \max\{0,0,0,0.5,0\} = 0.5.$$

The Procedure for Deciding the Control Action

Having determined the entries of the array corresponding to the algorithm for each intervention, the process of inferring the control action is carried out as follows.

For each successive time unit (1 s) for the next 10 s, data concerning vehicles crossing the critical point and vehicles added to the queue are used as input to the algorithm array in use. The corresponding entry of the array is thus determined. This entry is a measure of the confidence with which the algorithm may be applied, for the corresponding data. Obviously, that extension will be selected which corresponds to the maximum degree of confidence. In other words, fuzzy predictive decisionmaking implies that, that action is selected *that minimizes fuzziness*. Thus given a set of fuzzy rules, choose the one which is provided for coping with conditions as similar to the actual ones as possible. And, given a set of alternative actual conditions, consider those which are as similar to the conditions, for which the algorithm provides, as possible.

The explicit description of the procedure for deciding the control action is given below, by means of an example. Thus we consider the controller's second intervention. Arm N–S has the right of way. There are five vehicles queued at E–W arm. Data, concerning number of vehicles crossing the critical point (N–S traffic) and queued (E–W traffic) at each successive time unit during the next 10 s, is summarized in arrays α and α' respectively

$$\alpha = (0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$$
$$\alpha' = (0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0).$$

From α,α' arrays β,β' are constructed

$$\beta = (0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 5 \quad 5 \quad 6)$$
$$\beta' = (5 \quad 6 \quad 6 \quad 6 \quad 7 \quad 7 \quad 7 \quad 8 \quad 8 \quad 8)$$

as follows. If the ith elements of α and β are a_i,b_i, respectively, it is

$$b_i = \sum_{j=1}^{i} a_j,$$

and if the ith elements of α' and β' are a'_i,b'_i, respectively, it is

$$b'_i = Q + \sum_{j=1}^{i} a'_j,$$

where Q is the present queue at E–W arm. For example, if no change of the current state of the system occurs during the next 6 s, it is seen (from β and β') that four vehicles will cross the critical point and there will be a total of seven vehicles in the queue at E–W arm after 6 s from now.

The ith elements of arrays $\beta\beta'$, $i = 1, \cdots, 10$, determine the appropriate entry of the algorithm matrix C_2, which indicates the applicability of the algorithm to the situation described by these elements of the arrays. Thus for $t = 1$ s (i.e., considering an extension of 1 s) we have that no vehicle will cross the critical point (first element of array β) and that the queue will remain the same (five vehicles, first element of

TABLE VII
DECISION TABLE FOR CONTROL ACTION

Fuzzy Control Statement			Time (seconds)									
time	passing vehicles	queue	1	2	3	4	5	6	7	8	9	10
very short	mt(none)	any	0	.5	0	0	0	0	0	0	0	0
short	mt(a few)	lt(very short)	0	0	0	.3	0	0	0	0	0	0
medium	mt(few)	lt(very short)	0	0	0	0	.1	.1	0	0	0	0
long	mt(medium)	lt(very short)	0	0	0	0	0	.1	.1	0	0	0
very long	mt(many)	lt(short)	0	0	0	0	0	0	0	.5	.5	.8
Fuzzy Algorithm			0	.5	0	.3	.1	.1	.1	.5	.5	.8

array β'). It is easy to show that the rules of the algorithm are assigned the grade 0, and, consequently, the algorithm is assigned the grade 0 for $t = 1$ s. The results have been summarized in Table VII. Obviously, the controller will select the extension of 10 s. Thus the state of the system will remain the same for the next 10 s and the above procedure will be repeated (with new β and β') at the end of the 10-s period. If the extension given to the present state of the system were less than 10 s, the state of the system would change at the end of the extension period.

Note that if all the entries of the last row of Table VII were less than 0.5, no extension would be given and the state of the system (i.e., the phase) would be immediately changed.

Finally, if the maximum of the entries of the last row were not unique, i.e., if two or more alternative extension periods were indicated, then the maximum of these alternative extension periods would be selected. Of course, some other rule might be used instead (e.g., one giving the minimum extension period or the median or one randomly selected among these alternatives). For this, however, another control algorithm would be required in the place of the one used in this implication, which was based on the rule giving the maximum extension period.

RESULTS AND FINAL REMARKS

Because of the random nature of the arrivals assumed, many runs of the model were needed in order to get reliable results. The simulation work was carried out on the ICL-1900 general purpose computer.

The results have been summarized in Table VIII, whence it is clear that the system's performance is best under the fuzzy logic controller for all possible combinations of flow rates. Note that the effectiveness of the controller, as indicated by the percentage improvement in delays, is not seriously affected by the total volume of traffic through the junction.

These results have been obtained after several modifications of the control instructions initially set. The trial and error method was used in order to obtain an effective set of rules (or what is termed "satisficing control" in management science). In other words, a learning procedure was employed, by which the human performance in a similar real life situation of controlling a traffic junction is derived.

It is interesting to note that, having defined (in terms of fuzzy sets) what was considered to be a "small queue" or "few arrivals," it was the rules rather than the fuzzy sets which were modified. It is quite apparent that if the parameters describing the membership functions were introduced as additional input to the decision concerning the control algorithm, the dimensionality of the problem would radically increase, thus imposing severe difficulties in the obtainment of its solution.

The question of stability has been part of the whole problem of obtaining what is termed "effective set of rules." In this particular application, the stability of the system is defined as the condition of the system not getting saturated if subjected to a wide range of flow rates.

As far as fuzzy set theory is concerned, its basic concept, "fuzziness," characterizes only a state of knowledge. It exists neither in the system nor in the controller but in the human mind. Although the controller which was actually designed is termed "a fuzzy logic controller," it actually acts deterministically. That is, the algorithm by means of which the decision is taken, although conceived in fuzzy linguistic terms, is not fuzzy after the actual design is completed, i.e., after the fuzzy sets and implications are established [13].

Some interesting questions arise from the application reported here. These are related to the advantage gained by using the theory of fuzzy sets in modeling a controller in general and a traffic one in particular. They are also related to the feasibility of the construction of the fuzzy logic controller, as well as cost considerations concerning its construction.

First, it is clear that the improvement of the particular traffic junction of this study should be attributed to a)

TABLE VIII
COMPARISON BETWEEN DELAYS CAUSED BY EFFICIENT VEHICLE-ACTUATED CONTROLLER AND FUZZY LOGIC ONE

N - S traffic (veh/hr)	E - W traffic (veh/hr)	Average overall delay (secs/veh)		Improvement %
		Vehicle-actuated controller	Fuzzy-logic controller	
360	360	7.2	5.7	+21
360	720	7.4	6.1	+18
360	1080	7.9	6.6	+17
360	1440	8.4	7.3	+13
360	1800	9.3	8.4	+10
360	2160	12.3	10.0	+19
360	2520	15.8	13.6	+14
720	720	9.7	7.4	+21
720	1080	10.8	8.8	+19
720	1440	12.7	10.9	+14
720	1800	15.9	14.1	+11
720	2160	21.8	18.5	+15
1080	1080	13.6	12.0	+12
1080	1440	17.9	15.4	+14
1080	1800	25.8	21.6	+16
1440	1440	27.3	22.9	+16

placing the detecting pads at a considerable distance from the stop lines, b) detecting the traffic flow at both arms of the intersection simultaneously, and c) implementing the set of rules which resulted as the final product of the trial and error procedure used. However, it is very important to note that fuzzy set theory is credited with the convenience it offers in modeling linguistic rules. Thus in view of the design of a controller for a complex process, it is easier for the designer to synthesize a control algorithm in linguistic terms and subsequently implement it to the process, after interpreting them in terms of the theory. This advantage of the theory of fuzzy sets was fully appreciated in this application. Furthermore, modification of the initial set of linguistic rules was carried out in terms of computer instructions in a natural manner, by considering labels of fuzzy sets rather than values of the variables.

Second, hardware implementation of the fuzzy logic controller is obviously a straightforward computer exercise exploiting present day LSI technology, the decision being taken by table look-up. This was demonstrated earlier, by means of examples. Furthermore, although cost analysis was not a major concern of this study, it seems fair to state that investment for the implementation of the controller would pay for itself. This is justified in view of the fact that considerable reduction in delay times, and consequently in the associated cost, is caused by this controller, in respect to conventional vehicle-actuated ones.

It is hoped that further work will be done on the problem of controlling traffic by use of the theory of fuzzy sets. It must however be kept in mind that the fuzzy logic controller was designed for the purpose of controlling traffic characterized by randomness. In a linked system the traffic would be modulated. This should be taken into account when considering a controller for an intersection being part of a whole network, forming an integrated control system. On the other hand, special problems would arise in this case owing to the hierarchical structure of the system and consequently the control policy itself. It is thought also that in the case of integrated traffic control systems the theory of fuzzy sets would show its merits much more so than in the present simple case of an individual intersection.

APPENDIX
THE FUZZY ALGORITHM

Intervention: 7th second

if $T =$ very short
and $A = mt$(none)
and $Q =$ any
then $E =$ very short
else
if $T =$ short
and $A = mt$(a few)
and $Q = lt$(very small)
then $E =$ short
else
if $T =$ medium
and $A = mt$(few)
and $Q = lt$(very small)
then $E =$ medium
else
if $T =$ long
and $A = mt$(medium)
and $Q = lt$(very small)
then $E =$ long
else
if $T =$ very long
and $A = mt$(many)
and $Q = lt$(very small)
then $E =$ very long.

Intervention: 17th second

if $T =$ very short
and $A = mt$(none)
and $Q =$ any
then $E =$ very short
else
if $T =$ short
and $A = mt$(a few)
and $Q = lt$(very small)
then $E =$ short
else
if $T =$ medium
and $A = mt$(few)
and $Q = lt$(very small)
then $E =$ medium
else
if $T =$ long
and $A = mt$(medium)
and $Q = lt$(very small)
then $E =$ long
else
if $T =$ very long
and $A = mt$(many)
and $Q = lt$(small)
then $E =$ very long.

Intervention: 27th second

if $T =$ very short
and $A = mt$(none)
and $Q =$ any
then $E =$ very short
else
if $T =$ short
and $A = mt$(a few)
and $Q = lt$(very small)
then $E =$ short
else
if $T =$ medium
and $A = mt$(few)
and $Q = lt$(very small)
then $E =$ medium
else
if $T =$ long
and $A = mt$(medium)
and $Q = lt$(very small)
then $E =$ long
else
if $T =$ very long
and $A = mt$(many)
and $Q = lt$(small)
then $E =$ very long.

Intervention: 37th second

if $T =$ very short
and $A = mt$(none)
and $Q =$ any
then $E =$ very short
else
if $T =$ short
and $A = mt$(a few)
and $Q = lt$(small plus)
then $E =$ short
else
if $T =$ medium
and $A = mt$(medium)
and $Q = lt$(small plus)
then $E =$ medium
else
if $T =$ long
and $A = mt$(many)
and $Q = lt$(medium)
then $E =$ long
else
if $T =$ very long
and $A = mt$(too many)
and $Q = lt$(long)
then $E =$ very long.

Intervention: 47th second

if $T =$ very short
and $A = mt$(none)
and $Q =$ any
then $E =$ very short
else
if $T =$ short
and $A = mt$(a few)
and $Q = lt$(long)
then $E =$ short
else
if $T =$ medium
and $A = mt$(medium)
and $Q = lt$(long)
then $E =$ medium
else
if $T =$ long
and $A = mt$(too many)
and $Q = lt$(very long)
then $E =$ long
else
if $T =$ very long
and $A = mt$(too many)
and $Q = lt$(very long)
then $E =$ very long.

References

[1] B. R. Gaines, "Stochastic and fuzzy logics," *Electr. Letters*, vol. 11, pp. 188–189, 1975.

[2] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Intern. J. Man-Machine Studies*, vol. 7, pp. 1–13, 1975.

[3] E. H. Mamdani, "Application of fuzzy algorithms for the control of a dynamic plant," *Proc. IEE*, vol. 121, no. 12, pp. 1585–1588, 1974.

[4] P. J. King and E. H. Mamdani, "The application of fuzzy control systems to industrial processes," Round Table on Fuzzy Automata and Decision Processes, Proc. Sixth Triennial IFAC World Congress, Cambridge/Boston, MA, August 24–30, 1975.

[5] E. H. Mamdani and N. Baaklini, "Prescriptive method for deriving control policy in fuzzy-logic controller," *Electr. Letters*, vol. 11, p. 625, Dec. 1975.
[6] F. V. Webster, "Traffic signal settings," Technical Paper 39, Road Research Laboratory, 1958.
[7] F. V. Webster and B. M. Cobbe, "Traffic signals," Technical Paper 56, Road Research Laboratory, 1966.
[8] H. H. Goode, C. H. Pollmar, and J. B. Wright, "The use of a digital computer to model a signalized intersection," *Proc. High. Res. Ed.*, vol. 35, pp. 548–577, 1956.
[9] W. D. Ashton, *The Theory of Road Traffic Flow*. London: Methuen, 1966.
[10] L. A. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, pp. 338–353, 1965.
[11] ——, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28–44, 1973.
[12] R. C. T. Lee and C. L. Chang, "Some properties of fuzzy logic," *Inform. Contr.*, vol. 19, pp. 417–431, 1971.
[13] P. N. Marinos, "Fuzzy logic and its application to switching systems," *IEEE Trans. Comput.*, vol. C-18, pp. 343–348, 1969.

Machine Recognition of Abnormal Behavior in Nuclear Reactors

R. C. GONZALEZ, MEMBER, IEEE, AND L. C. HOWINGTON, MEMBER, IEEE

Abstract—**A multivariate statistical pattern recognition system for reactor noise analysis is presented. The basis of the system is a transformation for decoupling correlated variables and algorithms for inferring probability density functions. The system is adaptable to a variety of statistical properties of the data, and it has learning, tracking, updating, and dimensionality reduction capabilities. System design emphasizes control of the false-alarm rate. Its abilities to learn normal patterns and to recognize deviations from these patterns were evaluated by experiments at the Oak Ridge National Laboratory (ORNL) High-Flux Isotope Reactor. Power perturbations of less than 0.1 percent of the mean value in selected frequency ranges were readily detected by the pattern recognition system.**

I. Introduction

INTEREST in the application of pattern recognition techniques to perform automatic monitoring functions in nuclear power plants stems from a need to provide assistance to plant operators in assimilating quickly the large number of interrelated signals that provide insight into the operational status of the plant.

The recognition problem in this field can be broadly divided into two categories: surveillance and diagnosis. The surveillance problem is one of classifying the status of the components being monitored into two classes: normal or abnormal. The diagnosis problem is concerned with identifying the source and degree of a detected abnormality. At the present time most of the work in this area is being focused on the surveillance problem.

Manuscript received August 20, 1976; revised May 4, 1977. This work was supported by the Energy Research and Development Administration under contract with the Union Carbide Corporation. A shorter version of this paper was presented at the Third International Joint Conference on Pattern Recognition, Coronado, CA, November 8–11, 1976.

R. C. Gonzalez is with the Department of Electrical Engineering, University of Tennessee, Knoxville, TN 37916, and the Instrumentation and Controls Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830.

L. C. Howington was with the Department of Electrical Engineering, University of Tennessee, Knoxville, TN 37916, and the Oak Ridge National Laboratory, Oak Ridge, TN 37830. He is now with the Development Division, Y-12 Plant, Union Carbide Corporation, Oak Ridge, TN.

Unlike many pattern recognition applications where representative patterns from each class of interest are available, it is difficult to obtain patterns characterizing abnormal behavior in the components of a nuclear plant. Simulation of abnormal behavior by planned failure of components in a plant is usually not a feasible alternative for generating the desired abnormal patterns because of governmental regulations and implementation costs. The design of a pattern recognition system for surveillance is thus reduced to characterizing normal behavior and establishing limits beyond which the operation of a component is classified as abnormal. This approach is based on the assumption that the data which are labeled as normal are derived from plant components whose operation is within design specifications.

One of the principal features of a pattern recognition system designed to aid plant operators is that the system output should contain information beyond a simple "normal" or "abnormal" message. This information can be in the form of graphs, tables, or numerical data, but, in the presence of an abnormality, it must contain reference to the characteristics and limits of normal behavior. This information aids the operator in detecting trends and also in evaluating the relative importance of an alarm.

The system described in this paper is oriented in this direction. It utilizes the Hotelling transformation to decouple the multivariate input data and thus allows simplified interpretation of one-dimensional variables. In addition, this transformation is used for dimensionality reduction to compress archival storage of a plant's operational history. The system also utilizes the Mahalanobis distance as a global measure of performance in the original pattern space. Following detection of an abnormal pattern, the system makes available to the operator a set of options to help him analyze and interpret the pattern.

The performance of the pattern recognition system was evaluated with noise data from the Oak Ridge National Laboratory (ORNL) High-Flux Isotope Reactor (HFIR). The extraction and use of information from noise signals