A Systematic Design of Automatic Fuzzy Rule Generation for Dynamic Systems

Hoon Kang*, Yong-Ho Kim**, Hong-Tae Jeon**

*Dept. of Control & Instrumentation Engineering
Dept. of Electronics Engineering
Chung-Ang University
221 Huksuk-dong, Dongjak-gu, Seoul 156-756, Korea
e-mail (bitnet): hkang@krcaucc1

ABSTRACT

We investigate a systematic design procedure of automatic rule generation of fuzzy logic based controllers for highly nonlinear dynamic systems such as an engine dynamic model. By "automatic rule generation" we mean autonomous clustering or collection of such meaningful transitional relations from one conditional subspace to another. During the design procedure, we also consider optimal control strategies such as minimum squared error, near minimum time, minimum energy or combined performance criteria. Fuzzy feedback control systems designed by our method have the properties of closed-loop stability, robustness under parameter variabtions, and a certain degree of optimality. Most of all, the main advantage of the proposed approach is that reliability can be potentially increased even if a large grain of uncertainty is involved within the control system under consideration. A numerical example is shown in which we apply our strategic fuzzy controller design to a highly nonlinear model of engine idling speed control.

1. Introduction

Fuzzy logic/linguistic control can be catagorized as a knowledge-based system or an expert control paradigm, the reason for which is that every control action derived by the fuzzy inference engine is based on some a priori knowledge source whether the inferring mechanism is the MAX-MIN method or the MAX-DOT operation, or something else[1]. However, the difficulties in constructing a rule base have prevented the fuzzy engineers from approaching to a generalized methodology for fuzzy rule based control systems[2]. The fuzzy logic control scheme is shown in Figure 1.

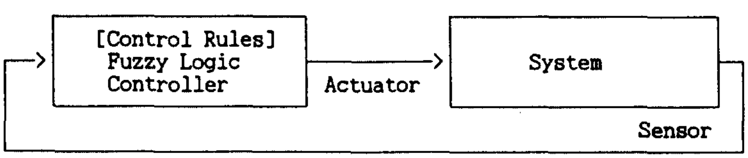


Figure 1. Block Diagram of Fuzzy Logic Control System

We propose a systematic design procedure of automatic rule generation for highly nonlinear dynamic processes. Fuzzy logic based feedback control is suitable for our physical target of automatic rule generation. Membership functions stored in a fuzzy logic controller can be easily modified and updated without repetitive tedious re-evaluation of different dynamic models. This procedure of generating the rules required in a fuzzy logic controller should guarantee stability of the closed-loop system and robustness under parameter variations. We utilize the cell-to-cell mapping theory originally introduced by Hsu[3] and later applied to fuzzy dynamic systems by Chen et al.[4] The key point in making a stable control rule base is that every stabilizable feedback system has a chain of state transitions from one cell-state to another. Consequential elements of such transitions are anticipated according to applied control action of each rule. Data required for this transitional set of rules can be collected via [5]

(A) A priori information such as experimental results,
(B) Numerical simulation runs based on dynamic models, and
(C) Expertise and heuristics.

Specifications, accuracy and precision, of the system tolerances can be arbitrarily adjusted and are a function of resolution of design parameter[6]. The next section deals with the step-by-step procedure as to the synthesis of a fuzzy logic control rule base based on the given input-state data pairs of a particular nonlinear dynamic model. These are the training data for approximate learning.

2. Automatic Rule Generation for Fuzzy Logic Controllers

Fuzzy Logic Control Based On Cell State Transitions: General fuzzy controllers have four components: fuzzifier, rule base, fuzzy inference engine, and defuzzifier. The control rules can be determined by using the cell-to-cell mapping theory[3] and the cell-state transitions. Comparing with the point-to-point mapping theory, this concept make use of the intervals and a finite number of cells in the cell-state. The dynamical characteristics are preserved as far as the resolution allows.

DEFINITION 1: A "Cell" z is defined as an n-tuple of integers in the cell-state space Z such that n

$$z = (z_1, z_2, \ldots, z_n) = \sum_{i=1}^{n} z_i e_i \quad (1)$$

where ei is the unit vector in the direction of zi and the corresponding x is represented by

$$(z_i - 0.5)\, hi < x_i < (z_i + 0.5)\, h_i \quad (2)$$

h_i...interval size. z_i...integer representing x_i

DEFINITION 2: A "Cell-to-Cell Mapping" F is a relation between cells in the cell-state space. F: Z $-\rangle$ Z. and the function values have one-to-one correspondence with the point-to-point mapping f such that

$$x_{k+1} = f(x_k) \in z^* \langle - \rangle z^* = F(z_k) \quad (3)$$

DEFINITION 3: An "Equilibrium Cell" z^* (Invariant Cell) of F satisfies

$$z^* = F(z^*) \quad (4)$$

DEFINITION 4: A k-Period Motion Cell is the distinct k cells $z(1),\ldots,z(k)$ that satisfy

$$z(1) = F^k(z(1)) \text{ and } z(m+1) = F^m(z(1)).$$

As an example. four 4-period motion cells are represented in Figure 2.

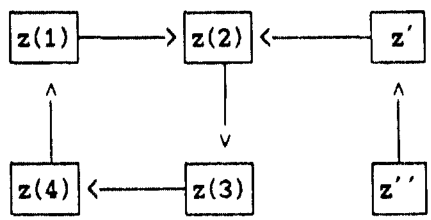


Figure 2. 4-Period Motion Cells z(1).z(2).z(3).z(4)

Systematic Procedure for Automatic Rule Generation:

STEP 1: Consider a two-state dynamic model given by

$$dx_1/dt = f_1(x_1, x_2, \delta, \theta) \quad (5a)$$
$$dx_2/dt = f_2(x_1, x_2, \delta, \theta) \quad (5b)$$

where x_1, x_2 are the states (or the errors); f_1, f_2 are the nonlinear mappings; and δ, θ are the control inputs. From the admissible controls, we select finite representative constant values of δ's and θ's and we call them δ_i's and θ_j's. These crisp numbers will be fuzzified later after the performance is satisfied. Moreover, we choose finite representative points in the state space to anticipate the trajectories from one subspace to another. These countless trajectories are called a 'manifold' and one set of data is collected by setting δ_i and θ_j constant. Repeated collections of such information are used next in order to obtain a rule base for feedback regulation of x_1 and x_2. i.e., x_1, $x_2 \rightarrow 0$ as t increases. We denote X_{1m}, X_{2n} the m-th and the n-th intervals of x_1 and x_2, respectively. Linguistically, we define

$$L_{mn} = X_{1m} \; X_{2n} \quad (6)$$

and then L_{mn} is a finite region in the state space. By applying fixed controls, δ_i and θ_j, one set of transitional relations is obtained for example,

Rules of Dynamical Behavior (Cell-State Transitions):

$$
\begin{aligned}
(\delta_1, \theta_1): z(L_{12}) &\rightarrow z(L_{34}) \\
(\delta_1, \theta_1): z(L_{23}) &\rightarrow z(L_{33}) \\
\cdots \quad &\quad \cdots \\
(\delta_1, \theta_1): z(L_{55}) &\rightarrow z(L_{73})
\end{aligned}
\tag{7}
$$

Table 1. Cell-State Transition Table in the 3-dim. Cell-State Space

Controls	Prev. State	Next State	Time t	Perf. Index
	{1,2,2}	{2,1,3}	0.5 sec	5.2
u = + 1	{1,1,1}	{3,2,1}	0.3 sec	7.7
	{2,1,3}*	{2,1,3}*	∞ sec	0.0
	{1,3,1}	{1,2,2}	0.4 sec	4.3
u = − 1	{2,1,2}	{3,2,1}	0.3 sec	11.2
	...	...	...	...

We continue to change (δ_i, θ_j) to get other sets of transitional rules and exhaustively gather finite number of transitional relations. When we store the above information, we add time required during transitions, and other optimal performance indices such as energy, squared errors, etc. These transition relations are stored in the table which we call a "cell-state transition table". An example of cell-state transition table in the 3 dimensional cell-state space is shown in Table 1. Changing (δ_i, θ_j), we may have the same L_{mn} for the source and the destination and this is called an 'invariant cell' or an 'invariant manifold'. For an invariant cell, there should be a design limit in the transition time since it is an indefinite stay in that L_{mn}. It is emphasized that the target L_{mn} (the specified goal) must have an invariant manifold for some fixed controls (δ_i, θ_j) for convergence and asymptotic stability. This is equivalent to the 'reachability condition' in the classical control theory. The target is denoted as L^*.

STEP 2: From the collected data, we generate an N-ary tree that connects from one node L_{mn} to another. The root of the tree is L^* and we avoid any looping structures. We proceed with a search technique of artificial intelligence (AI) and the search procedure is initiated by finding all possible dynamic transitions to the target region L^*. There may be multiple paths from one region to another and we eliminate multiplicity and extract only one transition by considering the following concepts:

1. Redundancy in Controls, Transitions
2. Optimal Strategies – Minimum Energy, Minimum Time, Minimum Squared Errors, or Combinations
3. Minimum Euclidean Distance

The elements of the finalized tree constitute a set of control rule base. These rules are automatically generated on the basis of optimal performance criteria such as minimum time or minimum energy concepts. Simply, the transitional relations that forces the trajectories from any points in the state space to the desired goal within the prescribed tolerances are themselves the control rules for feedback regulation. We store the membership functions for the transition relations in matrices P_i and C_j, in which rows the numerical values in [0,1] are the chosen membership functions.

STEP 3: The fuzzification procedure undertakes the rule generation so that the crisp transitions between the regions in the state space can be smoothed out. To each L_{mn} is assigned as many elements as accuracy and precision can allow. In a practical sense, five to seven elements are suitable for fuzzification of L_{mn} in the state space. Membership functions may be triangular or of simple functional type. The trade-off's between the number of quantization in Lmn and the transition smoothness, the total numbers of X_{1m}, X_{2n} and the performance are important and these issues are related to heuristics. For each rule, numbers between 0 and 1 are stored for each vector array of one membership function. In our example, a two-input two-output fuzzy controller has two vector arrays for the conditional parts and two vector arrays for the action parts for each rule. The fuzzy sets for control inputs δ_i and θ_j are denoted as Δ_i and Θ_j, respectively.

3. Fuzzy Inferencing Using Decomposition of Fuzzy Hypercubes

For practical purpose, a discrete version of fuzzy controllers is needed and it is convenient if we utilize a decomposed fuzzy hypercube [7] which is suitable for implementing a fuzzy logic controller with the cell-state mapping concept. Each rule numerically stored in a fuzzy hypercube corresponds with each cell in the cell-state. For each rule, one membership function in X_{1m} is stored in the premise matrix no. 1, P_1, and so is another in the premise matrix no. 2, P_2, and so on. Each row in P_1 or P_2 is the membership function obtained in the stepwise procedure stated earlier. The same is true for the consequence matrices, C_1 and C_2, representing Δ_i and Θ_j for each rule. Let the max-min product is denoted as " ∘ ", then for given fuzzy sets x_1 in X_{1m} and x_2 in X_{2n}, the control input fuzzy sets, Δ and Θ, are obtained as

$$\Delta = C_1^T \circ \{(P_1\ x_1) \times (P_2 \circ x_2)\} \tag{8a}$$

$$\Theta = C_2^T \circ \{(P_1\ x_1) \times (P_2 \circ x_2)\} \tag{8b}$$

where C_i^T is the transpose of the matrix Ci and "×" is the element-wise minimum operator. The crisp results of Δ and Θ are

$$\delta = \text{DEFUZZIFIER}(\Delta) \tag{9a}$$

$$\theta = \text{DEFUZZIFIER}(\Theta) \tag{9b}$$

where DEFUZZIFIER(.) is a defuzzification operator chosen among the maximum criterion method, the mean of maxima procedure, and the centroid algorithm.

4. Application : Design of An Engine Idling Speed Fuzzy Controller

Simulation Model : The well-known model for engine idling speed control has been rigorously studied in [8,9]. For simulation input-state training pairs, we collect the data exhaustively for the given fixed control values from the following highly nonlinear model:

$x_1 = N$ (Engine Rotor Speed), $x_2 = P$ (Manifold Pressure)

[Rotating Dynamics] $dx_1/dt = K_n\,(T_i - T_L)$, (10a)

[Manifold Dynamics] $dx_2/dt = K_p\,(m'_{ai} - m'_{ao})$ (10b)

where $m'_{ai} = (1 + 0.907\theta + 0.0998\theta^2)\, g(x_2)$

$$g(x_2) = \begin{cases} 1 & x_2 < 50.66 \\ 0.0197\,(101.325\,x_2 - x_2^2)^{1/2} & x_2 \geq 50.66 \end{cases}$$

$m'_{ao} = -0.0005968\,x_1 - 0.1336\,x_2 + 0.0005341\,x_1x_2 + 0.000001757\,x_1x_2^2$

$T_i = -39.22 + 325024.0\,m_{ao} - 0.0112\,\delta^2 + 0.000675\,\delta x_1\,(2\pi/60) + 0.635\,\delta$
$\quad + 0.0216\,x_1\,(2\pi/60) - 0.000102\,x_1^2\,(2\pi/60)^2$

$T_L = (x_1/263.17)^2 + T_d$ (T_L: Load Torque, T_d: Disturbance Torque)

$m_{ao} = m'_{ao}(t-\tau)/120x_1$, $\tau = 120/4x_1$ (Induction Power Delay), K_p, K_n: constants

Equations (10) are highly nonlinear two state engine model for idle speed control. We will obtain cell-state transitions from the above model in order to derive fuzzy logic control rules that stabilize and regulate the state trajectories toward the goal state, and in our case, the goal is $N = x_1 = 750$ (rpm), $P = x_2 = 35.0$ (kPascal).

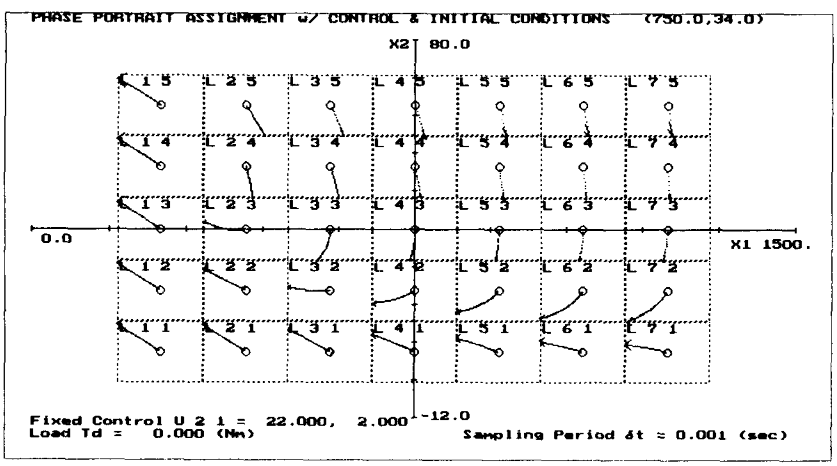


Figure 3. State Trajectories for ($\delta_2 = 22.0$, $\theta_1 = 2.0$) $= U_{21}$

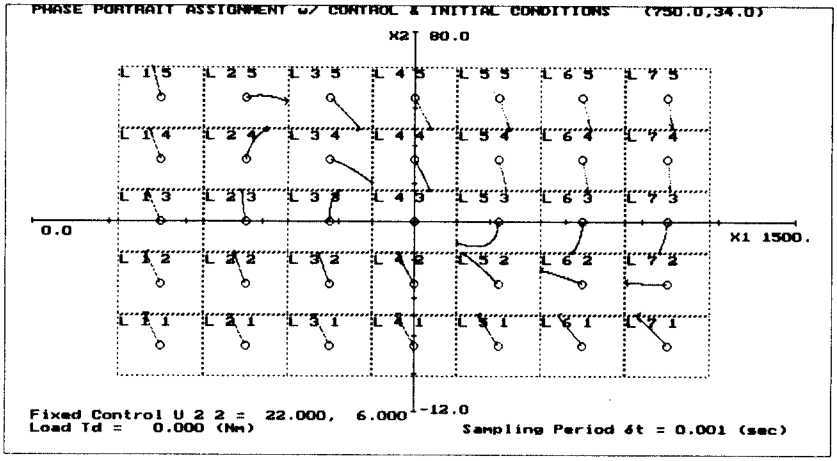


Figure 4. State Trajectories for $(\delta_2 = 22.0, \theta_2 = 6.0) = U_{22}$

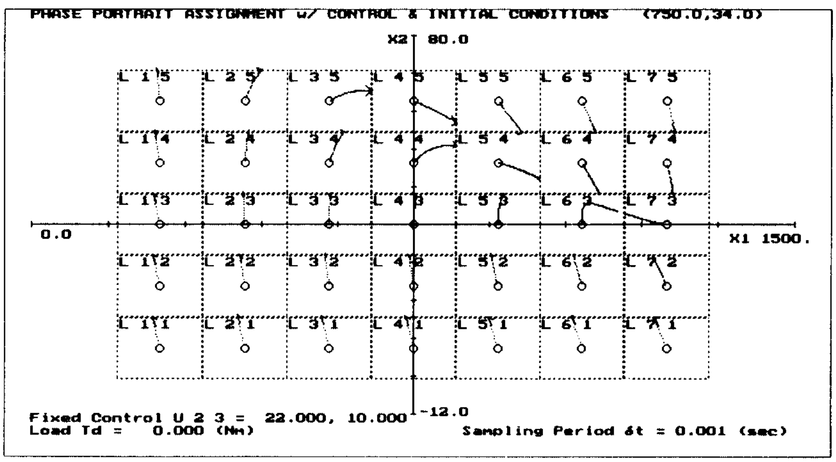


Figure 5. State Trajectories for $(\delta_2 = 22.0, \theta_3 = 10.0) = U_{23}$

Cell- State Transitions : As in Step 1, we gathered 9 kinds of the complete state trajectories by using 9 fixed controls from (δ_1, θ_1) to (δ_3, θ_3). Among them, the state trajectories of every initial conditions for (δ_2, θ_1), (δ_2, θ_2), and (δ_2, θ_3), are shown in Figures 3-5. Initial states start from the representative positions in the cell-state space. In Figure 4, we can find an equilibrium cell with fixed $(\delta_2, \theta_2) = U_{22}$.

The next step (Step 2) is to find a chain of connections among the cells with the assigned controls according to the chosen optimal strategy. This procedure is the most important one in the design of fuzzy logic controllers. The finalized control rules are determined by using the back tracking algorithm as shown in Figure 6. In our example, 5 elements are assigned to each cell. Figure 6 represents the results of the performance criterion for the minimum squared error control and the simulation results are shown in Figure 7.

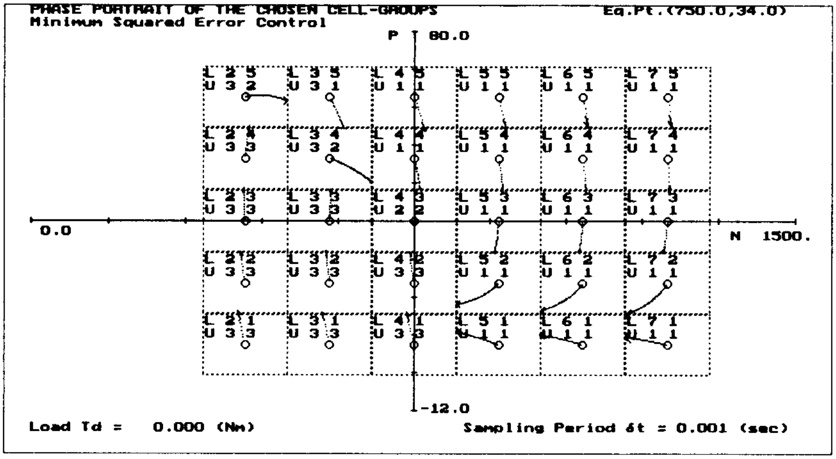


Figure 6. Rule Base Based On The Minimum Squared Error Strategy (MSE)

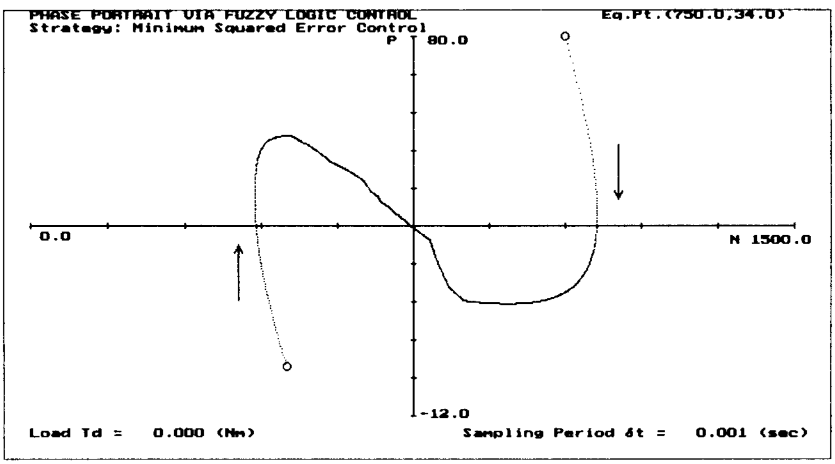


Figure 7. Simulation Results for The Minimum Squared Error Strategy (MSE)

In Figure 8, different optimal strategy has been chosen to compare the results and this case is the minimum energy/control effort criterion. Moreover, Figure 9 represents the simulation runs on different initial conditions.

In Figure 10, the responses of the minimum squared error (MSE) and the minimum control effort (MCE) control results are shown with the inferred control actions together. Figure 10 (a) and (b) are the idle speed control results for MSE and MCE, respectively, while (c) and (d) represents the throttle angles for MSE and MCE. The membership functions for each fuzzy subsets in the minimum energy based strategic control rule base are shown in Figure 11 where the premise part and the consequence part of fuzzy implications are represented. In Figure 11, each cell represents 5 elements in the universe of discourse for fuzzification.

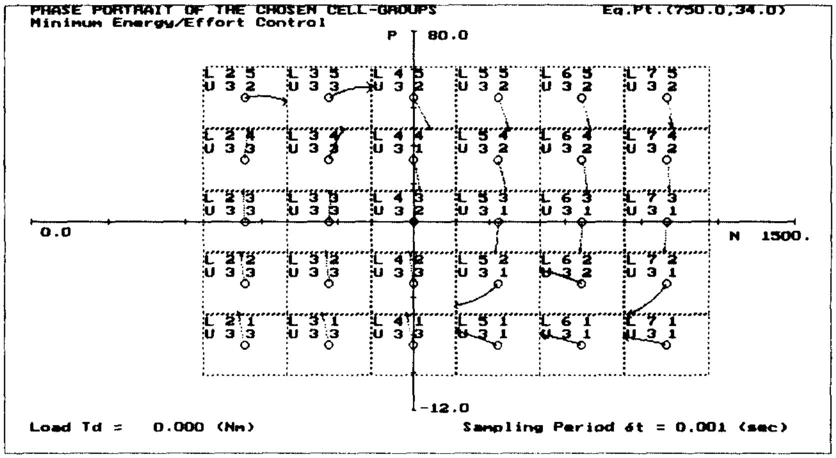


Figure 8. Rule Base Based On The Minimum Energy/Control Effort Strategy (MCE)

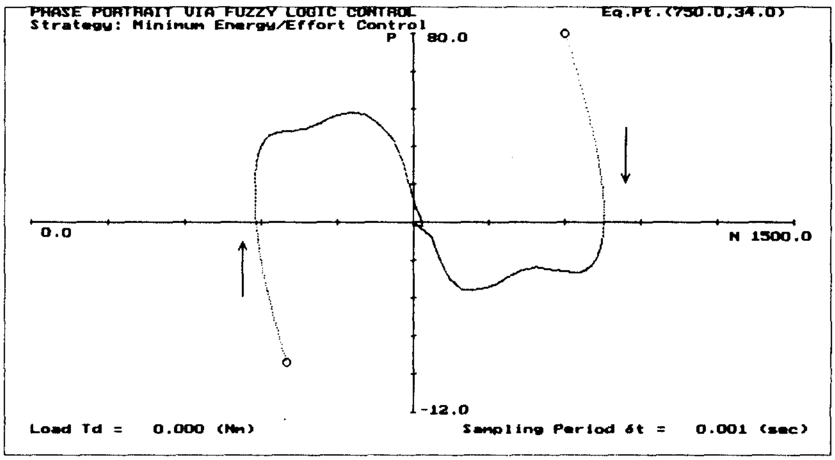


Figure 9. Simulation Results for The Minimum Energy/Control Effort Strategy (MCE)

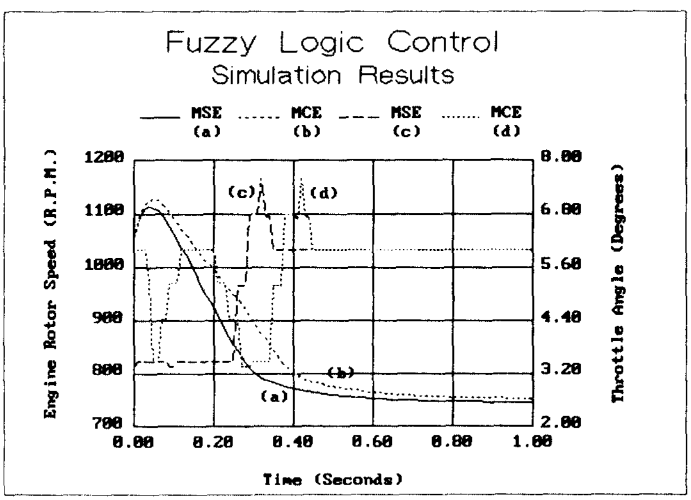
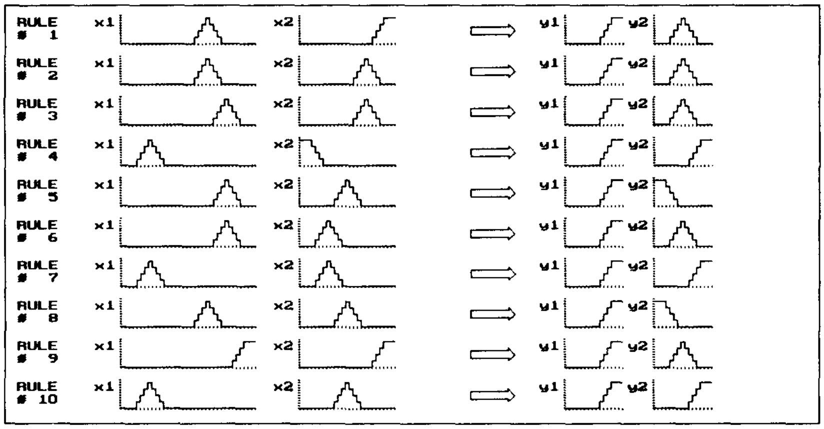
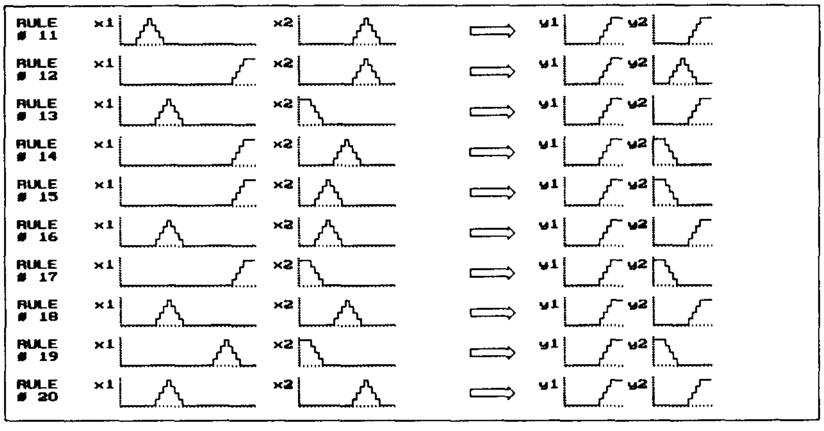


Figure 10. Comparison of Simulation Results between MSE and MCE

5. Conclusions

Even though we presented only a two-input two-output multivariable fuzzy logic control scheme for the automated design of a fuzzy controller rule base, we can easily generalize the systematic procedure for a m-input n-output multivariable fuzzy control system. The automated production design of fuzzy logic control rule bases for different optimal control strategies and the associated simulation results ensure versatility and flexibility of the proposed cell-state transition method. Emphasis is placed upon the fact that, for given arbitrary systems, we can make fuzzy logic based control rule bases that stabilize the closed-loop feedback control systems, and that the design procedure is totally automated. Furthermore, the rules are determined according to the chosen optimal strategy.





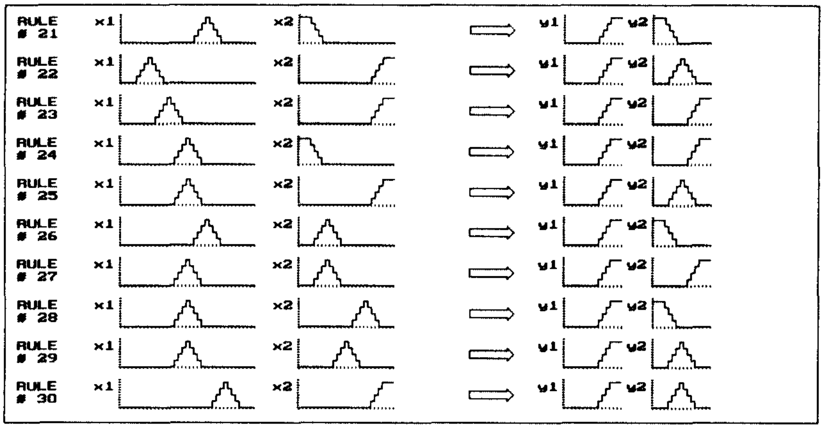


Figure 11. Fuzzy Logic Control Rule Base (30 Rules) of the Minimum Energy Strategy

References

1. H. -J. Zimmermann, Fuzzy Set Theory and Its Applications, 2nd ed. , Kluwer-Nijhoff, Boston, 1985
2. Togai InfraLogic, Inc. , TIL Shell & Fuzzy C-Development System User's Manuals, 1988
3. C. S. Hsu, "A Theory of Cell-to-Cell Mapping Dynamical Systems," Trans. of ASME, Journal of Applied Mechanics, vol. 47, pp. 931-939, Dec 1980
4. Y. Y. Chen and T. C. Tsao, "A Description of the Dynamical Behavior of Fuzzy Systems," IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-19, no. 4, pp. 745-755, Jul/Aug 1989
5. H. Kang and G. Vachtsevanos, "Nonlinear Fuzzy Control Based On The Vector Fields of The Phase Portrait Assignment Algorithm," Proc. American Control Conference, San Diego CA, pp. 1479-1484, May 1990
6. M. Braae and D. A. Rutherford, "Selection of Parameters for a Fuzzy Logic Controllers," Fuzzy Sets and Systems, vol. 2, pp. 185-199, 1979
7. H. Kang and G. Vachtsevanos, "Fuzzy Hypercubes: A Possibilistic Inferencing Paradigm," Proc. IEEE Conf. on Fuzzy Systems, San Diego CA, pp. 553-560, Mar 1992
8. A. W. Olbrot and B. K. Powell, "Robust Design and Analysis of Third and Fourth Order Time Delay Systems with Application to Automotive Idle Speed Control," Proc. American Control Conference, vol. 2, pp. 1029-1039, May 1989
9. B. K. Powell and A. W. Olbrot, "Robust Analysis of Automotive Idle Speed Control System," FORD, Technical Report No. SR-90-09, Jan 1990