the proposed system and a conventional system. We could see that our system improved the system performance by 4 ~ 20% compared with the conventional method.

The developed system was commercialized in 1992 by an industrial company, and this product has a good reputation in the market. In this study, the area-weight was determined by the fuzzy approach. In a further study we will apply the same approach to other variables.

REFERENCES

[1] L. A. Zadeh, "Outline of new approach to the analysis complex systems and decision processes," *IEEE Trans. Syst. Man Cyber.*, vol. SMC-3, pp. 28–44, 1973.

[2] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," in *IEEE Proc. Control & Science*, vol. 121, no. 12, pp. 1285–1588, 1974.

[3] H. Chen, M. Mizumoto and Y.-fel Ling, "Automatic control of sewerage pumpstation by using fuzzy controls and neural networks," in *Proc. 2nd Fuzzy Logic & Neural Networks*, 1992, pp. 91–94.

[4] S. Yasunobu and T. Hasegawa, "Evaluation of an automatic container crane operation system based on predictive fuzzy control," in *Proc. 2nd IFSE*, vol. 121, no. 12, pp. 419–432, 1986.

[5] M. Sugeno and K. Murakami, "An experimental study on fuzzy parking control using a model car," *Industrial Applications of Fuzzy Control*, pp. 125–138, 1985.

[6] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cyber.*, vol. SMC-15, pp. 116–132, 1985.

[7] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 15–33, 1988.

[8] S. Araki, H. Nomura, I. Hayashi and N. Wakami, "A self-generating method of fuzzy inference rules," in *IFES '91 Proc.*, pp. 1047–1058, 1991.

[9] H. Nomura, I. Hayashi and N. Wakami, "A learning method of fuzzy inference rules by descent method," in *1992 IEEE*, pp. 203–210, 1992.

[10] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I," *IEEE Trans. Syst. Man Cyber.*, vol. 20, pp. 404–418, 1990.

[11] ——, "Fuzzy logic in control systems: Fuzzy logic controller—Part II," *IEEE Trans. Syst. Man Cyber.*, vol. 20, pp. 419–435, 1990.

[12] H.-J. Zimmermann, *Fuzzy Set Theory, and Its Applications*. New York: Kluwer Academic, 1991.

[13] H. Lee-Kwang and G.-R. Oh, *Fuzzy Theory and Application Volume I: Theory*. Hong-Lyung Science, 1991.

[14] ——, *Fuzzy Theory and Application vol II: Application*. Hong-Lyung Science, 1991.

[15] G. C. Barney and S. M. Dos Santos, *Elevator Traffic Analysis, Design, and Control*, 2nd. ed. Peter Peregrinus, 1985.

[16] G. C. Strakosch, *Vertical Transportation: Elevators and Escalators*, 2nd. ed. New York: Wiley, 1983.

[17] P. Chenais and K. Weinberger, "New approach in the development of elevator group control algorithms," in *Proc. ELEVCON '92*, vol. 4, pp. 48–57, 1992.

[18] Y.-S. Song, "A comparative study of fuzzy implication operators: An application to elevator systems," CSD Kaist, *M.S. Thesis*, 1992.

[19] J. B. Lee, "A study on the development of optimal model for determining configurations of the elevator systems," Hanyang University, M.S. thesis, 1988.

[20] Hitachi, Ltd., Japan, "Group supervisory system of elevator cars," United States Patent, 4,244,450, 1981.

[21] ——, "Elevator group control system," Patent Document 1273, Japan 87-555, 1987.

[22] ——, "Elevator group control system," Patent Document 208, Japan 90-6397, 1990.

Neural Network Based Fuzzy Identification and Its Application to Modeling and Control of Complex Systems

Yaochu Jin, Jingping Jiang, and Jing Zhu

***Abstract*—This paper proposes a novel fuzzy identification approach based on an updated version of pi-sigma neural network. The proposed method has the following characteristics: 1) The consequence function of each fuzzy rule can be a nonlinear function, which makes it capable to deal with the nonlinear systems more efficiently. 2) Not only each parameter of the consequence functions but also the membership function of each fuzzy subset can be modified easily on-line. In this way, the fuzzy identification algorithm is greatly simplified and therefore is suitable for real-time applications. Simulation results show that the new method is effective in modeling and controlling of a large class of complex systems.**

I. INTRODUCTION

Fuzzy model identification is developed based on the fuzzy set theory proposed by Zadeh [1] and has been widely investigated [2]–[5]. The main interest has been on building fuzzy relationship models that are expressed by a set of fuzzy linguistic propositions derived from the experience of the skilled operators or a group of observed input-output data. However, for some large complex systems, it is almost impossible to establish such a fuzzy relationship model due to the large amount of the fuzzy prepositions and the highly complicated multidimensional fuzzy relationship.

Takagi and Sugeno [5] proposed a new type of fuzzy model that has been proved to be effective in overcoming some of these difficulties. Their fuzzy model consists of fuzzy implications whose consequences are described by crisp linear input-output relation functions. Another significance of their fuzzy model we think is that, since every of its consequence parameter is identified by certain algorithms such as the least square method and therefore, the fuzzy model established is more systematic and objective. Unfortunately, the identification procedure is quite complicated and is carried out off-line (although Sugeno and Tanaka [6] suggested a successive identification algorithm, it still has difficulties for real-time implementation.), which makes it incompetent to deal with time-varying systems.

The theory of Artificial Neural Network (ANN) has been greatly developed in the recent years. Due to its strong nonlinear mapping and learning abilities, applications of ANN to control systems have been so successful that neurocontrol is no longer strange to those who work in the discipline of automatic control [7]. Mainly, there are two kinds of applications of neural networks to control systems, namely Neural-Network-Integrated Control (NNIC) and Neurocontrol or Neuromorphic Control (NC). By NNIC, we mean those control schemes that use neural networks to enhance the performances of some conventional control strategies, such as adaptive control, optimal control, internal model control and predictive control, as well as expert control and fuzzy control. NC, on the other hand, uses neural networks directly as the controller and no other conventional control means are involved. It is desirable to note that the marriage of neural networks with fuzzy set theory is showing special promise and is receiving more and more attentions. Use of neural networks to perform the adjustment of fuzzy membership functions and modification of fuzzy rules makes it practical to design adaptive fuzzy models and self-organizing fuzzy controllers.

Manuscript received August 27, 1993; revised May 18, 1994. This work was supported by the National Science Foundation of China.

The authors are with the Department of Electrical Engineering, Zhejiang University, Hangzhou 310027, P.R. China.

IEEE Log Number 9410389.

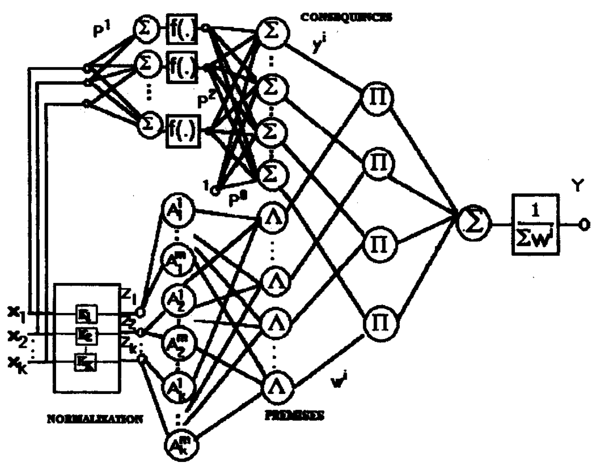


Fig. 1. Structure of the hybrid pi-sigma neural network.

This paper furthers the work of Tagaki and Sugeno [5] on the basis of an updated version of pi-sigma neural network. The neural network contains not only summing and product neurons, but also fuzzy neurons that perform minimum operation. To adjust the consequence parameters and the membership functions, the idea of error backpropagation is extended and hill-climbing searching technique is introduced. Through examples of short-term weather forecast, burden optimization, fuzzy control system design and dynamic control of robot manipulators, we demonstrate that our scheme is free of the shortcomings of the existing fuzzy identification methods.

II. Takagi and Sugeno's Fuzzy Model

In this section, we briefly describe the fuzzy model suggested by Takagi and Sugeno [5] and provide some discussions. Consider a fuzzy system with K inputs and single output

$$\begin{aligned} R^i\text{: If } x_1 \text{ is } A_1^i \text{ and } \cdots \text{ and } x_k \text{ is } A_k^i, \\ \text{Then } y^i = g^i(x_1, \ldots, x_k) \end{aligned} \tag{1}$$

where R^i $(i = 1, 2, \ldots, m)$ denotes the i-th implication, m is the number of the fuzzy implications of the fuzzy model, $x_1, \ldots, x_k$ are the premise variables, A_j^i $(j = 1, 2, \ldots, k)$ is the fuzzy subset whose membership function is continuous piecewise-polynomial function, y^i is the consequence of the i-th implication, which is a nonlinear or linear function of the premises. Given an input $(x_1^0, \ldots, x_k^0)$, the final output of the fuzzy model is expressed by

$$Y = \sum_{i=1}^{m} w^i y^i \Big/ \sum_{i=1}^{m} w^i \tag{2}$$

where w^i is the overall truth value of the premises of the i-th implication calculated as

$$w^i = \mathop{\Lambda}_{j=1}^{k} \mu_{A_j^i}(x_i^0) \tag{3}$$

In (3), $\mu A_j^i(x)$ is the membership function of the fuzzy subset A_j^i, which is supposed to be continuous piecewise-polynomial function, often of convex type straight lines.

The identification using input-output data consists of two parts: structure identification and parameter identification. The former decides which input variables affect the output, while the latter identifies the parameters in both the premises and consequences.

The above described method of designing fuzzy models is superior to the heuristic one probably in that all of its fuzzy rules (including the parameters of the membership of each premise and the parameters of the consequences) are identified and consequently seems more accurate and objective. However, this method is unavailable for wider application because the identification is too complicated and no systematic procedure has been established. Difficulty also arises in obtaining enough input-output data for off-line identification, since it is often the case that no experiment is allowed in an industrial process.

From the research history of the past decades, we believe that parameter identification cannot be effectively realized by the conventional fuzzy system methods. To cope this situation, it is necessary to develop a more sophisticated mathematical tool for fuzzy systems. One possible and perhaps the most prospective way is to combine the artificial network theory with fuzzy set theory. These hybrid systems seem to have the following features:

1) Fuzzy sets are used to create a relevant perception perspective, which possesses very clear physical meanings.
2) All the fuzzy rules are expressed by a group of weights of a neural network and can be adjusted in a more effective way.
3) The nonlinear characteristic of the neural network endows the fuzzy model greater abilities to describe a given complex system.

III. Hybrid Pi-Sigma Neural Network

A. Architecture of the Hybrid Pi-Sigma Neural Network

In order to deal with the fuzzy systems, the neural networks should contain not only summing and multiplying neurons, but also fuzzy neurons that are able to perform fundamental fuzzy operations such as the minimum or maximum operation. Therefore, we proposed a hybrid pi-sigma neural network as shown in Fig. 1. In Fig. 1, Σ denotes the summing neurons, Π denotes the product neurons and Λ represents the fuzzy neurons, which, in our case, perform minimum operation. Assume the consequent sub-network has n hidden neurons, then the output of the whole neural network is given by

$$Y = \sum_{i=1}^{m} w^i y^i \Big/ \sum_{i=1}^{m} w^i \tag{4}$$

$$y^i = g^i(x_1, \ldots, x_k) = P_{1i}^0 + \sum_{j=1}^{n} \left(P_{ji}^2 f\left(\sum_{l=1}^{k} P_{lj}^1 x_l \right) \right) \tag{5}$$

$$w^i = \mathop{\Lambda}_{l=1}^{k} A_l^i(z_l) \tag{6}$$

where $f(\cdot)$ is a sigmoid type nonlinear function, z_l is the normalized value of x_l. We will show later that the normalization of the premise variables fanned in the membership function helps to make the design simpler. Note that the membership grade $\mu_{A_l^i}(z_l)$ is written as $A_l^i(z_l)$ for short, which is a widely adopted simplification in fuzzy set theory. The truth value of the premise of each rule is determined by the minimum of the membership grades of all its premise variables, and will be updated indirectly by changing the form of the membership functions. The updating of the membership functions is of great necessity because in most cases, they are prescribed by heuristics and do not necessarily conform to the real world.

This hybrid neural network has very clear physical meaning. Obviously, it is equivalent to the fuzzy system defined by (1)–(3). However, the membership function is of Gaussian type rather than continuous piecewise polynomial so that the neural network can learn

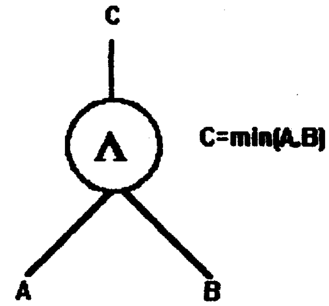


Fig. 2. Illustration of a fuzzy neuron.

more effectively

$$A_l^i(z_l) = \exp(-(z_l - a_l^i)^2/b_l^i), \quad b_l^i > 0 \tag{7}$$

where a_l^i and b_l^i are two parameters that describe the form of the function. In the conventional design, a_l^i and b_l^i have to be carefully evaluated so that satisfying results can be obtained. But up to now, we still have no concrete rules to choose proper values for them. What we can do is to select some "best" values depending on either statistical data or past experiences. As a matter of fact, it is impossible to acquire best values for them beforehand because they are dependent on the distribution of the input data. In our case, this difficult situation is avoided. We just need to preset some initial values, which will automatically be adjusted to their optimal values when the on-line learning is implemented.

B. Learning Algorithm

To adjust the consequence parameters and the parameters of the membership function, the error backpropagation algorithm should be slightly extended. Suppose the desired output of the pi-sigma network is Y^d, which can be obtained from heuristics or other approaches, we form the error function as follows:

$$E = \frac{1}{2}(Y - Y^d)^2 \tag{8}$$

According to the principle of error backpropagation, the generalized error of the final output node Σ is $(Y^d - Y)/\Sigma w^i$. Since the consequence y^i and the overall truth value of the premises w^i of the i-th implication are multiplied in the multiplication node Π, the error can not be directly back propagated. However, if we think w^i is the 'weight' connecting the consequence node Σ and the final output node Σ, the product node can be 'eliminated'. In this way, the generalized error of the i-th consequence node Σ is obtained approximately as

$$\delta_1^i = (Y^d - Y)w^i \Big/ \sum_{i=1}^{m} w^i. \tag{9}$$

Similarly, the generalized error of each fuzzy node is

$$\delta_2^i = (Y^d - Y)y^i \Big/ \sum_{i=1}^{m} w^i. \tag{10}$$

Therefore, the consequence parameters are adjusted in the following form:

$$\Delta P_{li}^0 = \eta \delta_1^i \tag{11}$$

$$\Delta P_{lj}^1 = \eta x_l f'\left(\sum_{l=1}^{k} P_{lj}^1 x_l\right)\delta_1^i \tag{12}$$

$$\Delta P_{ji}^2 = \eta x_l f\left(\sum_{l=1}^{k} P_{lj}^1 x_l\right)\delta_1^i \tag{13}$$

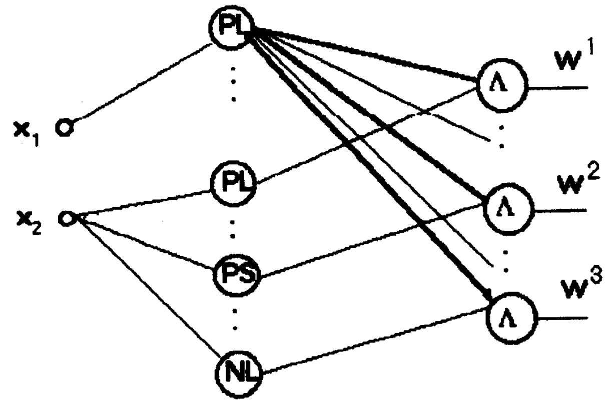


Fig. 3. Membership function updating. At most 6 rules are involved in updating the membership function $\mu_{\rm PL}(x)$ of premise x_1. This blacken lines denote the selected node by the first searching algorithm.

where η is the learning rate and other parameters are all defined as before.

Now, we discuss the updating of the membership functions. For the sake of simplicity, we consider the fuzzy node that implements minimum operation as in Fig. 2. From Fig. 2, we have

$$C = \min(A, B). \tag{14}$$

If $A = 0.8$ and $B = 0.9$, then $C = \min(0.8, 0.9) = 0.8$. Suppose the desired output of C is 0, and hence the error $e = -0.8$. In order to back propagate this error, which input node is to blame? Although the value of the error e in this case is determined by node A, this does not necessarily mean that the error is caused by node A. It may be the case that the true value of node B is 0 and the error is fully caused by node B. To cope this situation, we introduce the hill-climbing search method, which is a counterpart of the gradient method and does not require the differentiability of the cost function. Thus, if the error of the node C is e, we update node A and B simultaneously and then compare these two alternatives to see which modification is better. For example, if the error before modification is -0.8, and after updating node A, the new error is -0.4 while after updating node B, the error becomes -0.2, then we think that updating node B is better than updating node A. Consequently, node B is actually modified.

There is another problem remained to solve. In real applications, if the fuzzy system has m fuzzy implications, not every premise variable is necessarily divided into m subspaces. Without loss of generality, we assume that there exists a fuzzy model whose two input variables are divided into six subspaces, namely PL, PM, PS, NS, NM, NL, where 'P' means positive, 'N' stands for negative and 'L', 'M', 'S' denotes large, medium and small respectively. Thus the fuzzy model has 36 fuzzy rules:

R^1: If x_1 is PL and x_2 is PL, then $y^1 = g^1(x_1, x_2)$
R^2: If x_1 is PL and x_2 is PM, then $y^2 = g^2(x_1, x_2)$
: :
R^{36}: If x_1 is NL and x_2 is NL, then $y^{36} = g^{36}(x_1, x_2)$.

Therefore, there are at most six rules (not always six because of the hill-climbing search algorithm) that may be used to update some certain membership function(see Fig. 3). Two possible solutions are at hand. The first is, rather directly, to use the hill-climbing technique for the second time to decide which rule should be adopted for updating. This makes the algorithm a little complicated. The other method is to choose a rule randomly among those which have been selected

TABLE I
HISTORICAL DATA FOR TRAINING

Year	x1	x2	Rainfall (mm)	Year	x1	x2	Rainfall (mm)
1952	0.73	-5.28	283	1967	0.20	-5.43	431
1953	-2.08	5.18	647	1968	3.46	-19.85	179
1954	-5.53	10.23	731	1969	0.08	8.59	615
1955	-3.31	4.21	561	1970	1.46	7.26	433
1956	0.53	-2.46	467	1971	0.24	-1.10	401
1957	2.33	7.32	399	1972	0.89	-16.94	206
1958	-0.32	-10.81	315	1973	-0.50	10.46	639
1959	-2.35	3.85	521	1974	2.15	-10.06	418
1960	-0.95	2.74	472	1975	-0.89	12.11	570
1961	-0.64	6.00	536	1976	1.40	-6.26	415
1962	0.92	0.65	385	1977	-0.59	7.15	796
1963	2.98	-11.83	259	1978	0.49	22.63	
1964	-0.85	-2.30	657	1979	-0.09	7.32	
1965	0.46	-14.68	348	1980	0.04	-6.92	
1966	-2.31	-1.36	644	1981	0.66	2.37	

by the first searching algorithm. It has proved through practice that either alternative will do. Finally, the parameters of the membership functions are modified in the following form:

$$\Delta a_l^i = \beta z_l \delta_2^i 2(z_l - a_l^i)/b_l^i \tag{15}$$

$$\Delta b_l^i = \beta z_l \delta_2^i (z_l - a_l^i)^2/(b_l^i)^2 \tag{16}$$

where β is the learning rate and the value of l is decided by the hill-climbing search algorithm.

Up to now, we have developed a new approach to identify a fuzzy model via the theory of artificial neural networks. We argue that the proposed method is quite general because almost no prerequisite is posed. Neither do we need to specify membership functions for every premise, nor do we have to spend much time for premise and consequence structure identifications. In addition, the consequence parameters and the parameters of each membership function are adjusted on-line. In this situation, the consequence parameters are randomly initialized and the fuzzy subspaces are coarsely divided before operation. In Section IV, we will show three examples of fuzzy modeling. In Section V, we applied the proposed algorithm to the dynamic control of a rigid robot manipulator. All the examples demonstrate that our algorithm is effective in modeling and control of a wide class of complex systems.

IV. FUZZY MODELING APPLICATIONS

In the following examples, all the premises are divided into 6 fuzzy subspaces, that is PL, PM, PS, NS, NM and NL, whose membership functions are all initially taken as

$$\mu_{\mathrm{PL}}(z) = \begin{cases} 1, & z > 1 \\ \exp(-(z-1)^2/0.3), & 0 < z \le 1 \end{cases}$$
$$\mu_{\mathrm{PM}}(z) = \exp(-(z-0.5)^2/0.125), \quad z > 0$$
$$\mu_{\mathrm{PS}}(z) = \exp(-z^2/0.125), \quad z \ge 0$$
$$\mu_{\mathrm{NL}}(z) = \mu_{\mathrm{PL}}(-z), \quad \mu_{\mathrm{NM}}(z) = \mu_{\mathrm{PM}}(-z)$$
$$\mu_{\mathrm{NS}}(z) = \mu_{\mathrm{PS}}(-z) \tag{17}$$

For simplicity, all the consequence functions take the linear form.

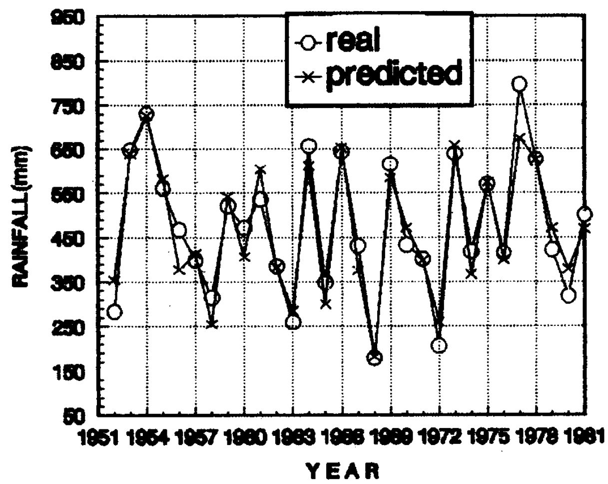


Fig. 4. Results of rainfall forecast.

A. Short-Term Rainfall Forecast

From the historical data during the year of 1952 and 1977 recorded by Tianjin meteorological observatory [9], we select the western Pacific ocean temperature anomaly (x_1) from Jan. the year before to Feb. of the next year and the Eurasia 500 mb height anomaly (x_2) as two input variables of the rainfall forecast fuzzy model. The processed data and the amount of the rainfall are listed in Table I. According to the assumed division of the fuzzy subspaces of each premise, we have the following 36 fuzzy rules:

R^1: If x_1 is PL and x_2 is PL,
then $y^1 = P_0^1 + P_1^1 x_1 + P_2^1 x_2$;
R^2: If x_1 is PL and x_2 is PM,
then $y^2 = P_0^2 + P_1^2 x_1 + P_2^2 x_2$;
: :
R^{36}: If x_1 is NL and x_2 is NL,
then $y^{36} = P_0^{36} + P_1^{36} x_1 + P_2^{36} x_2$.

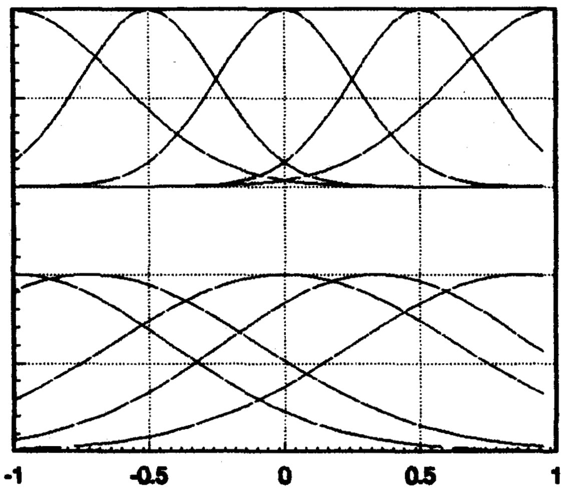
The overall output of the fuzzy model is expressed by

$$Y = \sum_{i=1}^{m} w^i y^i \Big/ \sum_{i=1}^{m} w^i. \tag{18}$$

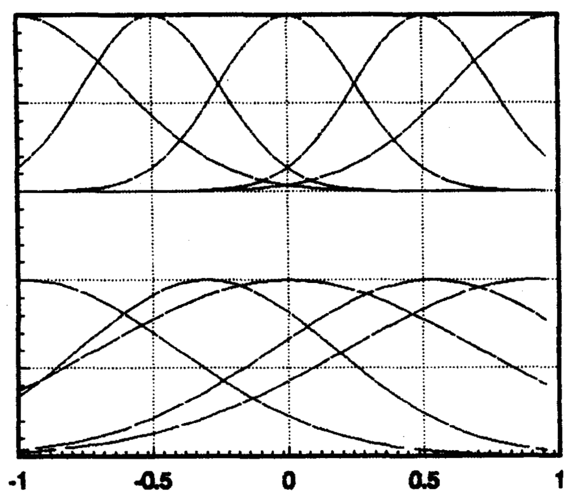
Training the hybrid neural network with twenty-six group of data from the year of 1952 to 1977 given in Table I for about 1000 times, the neural network converges and the learning process completes. First, we verify the rainfall learned by the trained fuzzy model. Except for the year of 1977, which is an over-raining year of Tianjin, the outputs of the fuzzy model are quite consistent with the real values (see Fig. 4). Then we use the fuzzy model to predict the rainfall from the year of 1978 to 1981. The results are rather satisfying (again see Fig. 4). The membership functions of the two premises before and after learning are shown in Fig. 5. From Fig. 5, we show that the distribution of the fuzzy membership functions is more agreeable to the distribution of the training data.

B. Burden Optimization Model

Suppose some material is composed of three components A, B and C, where component A and component B forms 100 percent, C is an extra adding component. Table II gives 14 experimental burden



(a)



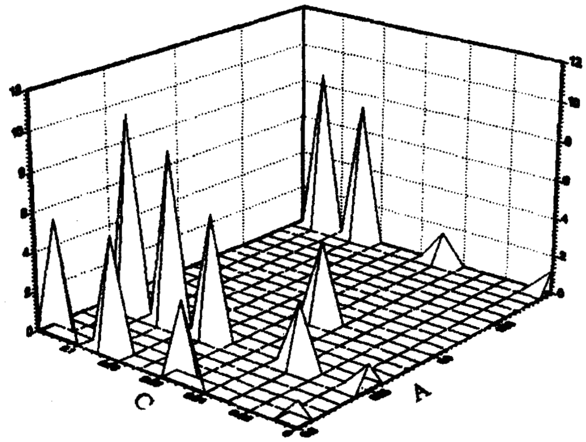
(b)

Fig. 5. Membership modification through learning. (a) Premise x_1; (b) premise x_2.

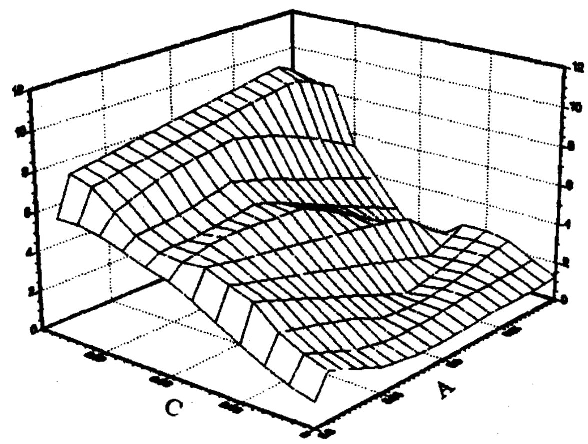
TABLE II
DIFFERENT FORMULATIONS AND THEIR PERFORMANCE INDICES

A(%)	B(%)	C(%)	Index	A(%)	B(%)	C(%)	Index
100	0	0	1.23	85	15	7	6.13
85	15	0	1.27	85	15	9	8.72
80	20	0	1.32	85	15	11	9.97
100	0	5	1.58	80	20	11	6.01
80	20	5	4.12	81	19	9	5.61
90	10	5	4.06	100	0	9	7.30
85	15	3	3.20	100	0	11	8.51

methods and their performance indices. We are asked to acquire the relationship between different burden methods and their performance indices and the optimal burden point. Using the given 14 groups of data to train the neural network for about 12000 times, the testing mean error is less than 1%. Fig. 6(a) shows the given data and Fig. 6(b) is the associated data.



(a)



(b)

Fig. 6. Burden optimization. (a) Given data; (b) associated data.

C. Fuzzy Control System

In some cases, we only have a set of input-output data without knowing anything of the internal knowledge of a system (sometimes we call it black box). In order to control such systems properly, we have to obtain the characteristic relationship between the inputs and outputs of the system. Twenty sets of training data are listed in the left side of Table III. The concerned system has three input variables (x_1, x_2, x_3) and one output variable (y) and y^* denotes the desired output. In the right side of Table III, twenty groups of testing data are listed. Despite that there eleven input-output patterns are new to the fuzzy model, the output of the fuzzy model is acceptable.

V. DYNAMIC CONTROL OF RIGID ROBOT MANIPULATORS

One problem we meet when we apply the fuzzy model to system control is that we have to identify large number of consequence parameters. For example, for a rigid robot with N-degree of freedoms, there are $3N$ input variables (link position, velocity and acceleration), and if each input is divided into 6 fuzzy subspaces, there will be 6^{3N} fuzzy rules. Since every rule consists of $N \times (3N+1)$ consequence parameters (the linear case), we have to identify $N(3N+1)6^{3N}$ parameters. If $N = 6$, the number is about 1.16×10^{16}, which is very huge and makes it impossible for real-time implementation. Therefore, when we apply the fuzzy model to robot control, we first

TABLE III
TRAINING DATA AND ASSOCIATED DATA OF A FUZZY CONTROL SYSTEM

No.	x1	x2	x3	y*	y	No.	x1	x2	x3	y*	y
1	1	3	1	11.4	11.3	21	1	1	5	9.54	9.30
2	1	5	2	6.21	6.21	22	1	3	4	6.04	6.04
3	1	1	3	10.2	10.2	23	1	5	3	5.72	5.72
4	1	3	4	6.04	6.04	24	1	1	2	11.3	10.9
5	1	5	5	5.24	5.24	25	1	3	1	11.1	11.4
6	5	1	4	19.0	19.0	26	5	5	2	14.4	14.4
7	5	3	3	14.2	14.1	27	5	1	3	19.6	18.9
8	5	5	2	14.4	14.4	28	5	3	3	14.2	14.1
9	5	1	1	27.4	27.3	29	5	5	5	12.4	11.9
10	5	3	2	15.4	15.5	30	5	1	4	19.0	19.0
11	1	5	3	5.72	5.72	31	1	3	3	6.38	6.10
12	1	1	4	9.76	9.76	32	1	5	2	6.21	6.21
13	1	3	5	5.87	5.87	33	1	1	1	16.0	14.8
14	1	5	4	5.40	5.40	34	1	3	2	7.22	7.93
15	1	1	3	10.2	10.2	35	1	5	3	5.72	5.72
16	5	3	2	15.4	15.4	36	5	1	4	19.0	19.0
17	5	5	1	19.7	19.6	37	5	3	5	13.4	13.3
18	5	1	2	21.0	21.1	38	5	5	4	12.7	12.8
19	5	3	3	14.2	14.1	39	5	1	3	19.6	18.9
20	5	5	4	12.7	12.8	40	5	3	2	15.4	15.4

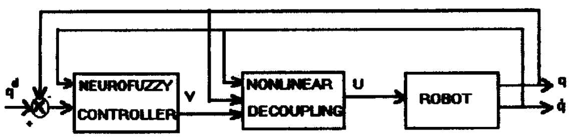


Fig. 7. Structure of the fuzzy controller for robot system.

have to decouple the robot dynamics. After properly selecting the decoupled dynamical model, the fuzzy controller for each link has only 2 input variables. In this case, we need only $N\times 6^2$ fuzzy rules, each consisting 3 consequence parameters. When $N=6$, the number reduces to 648.

A. Decoupling of the Robot Dynamics

Consider an N-degree-of-freedom rigid robot, its dynamics is expressed by

$$\tau = H(q)\ddot{q} + M(q,\dot{q}) + G(q) \tag{19}$$

where τ is $N\times 1$ torque vector, $H(q)$ is $N\times N$ inertia matrix, $M(q,q)$ is $N\times 1$ coriolis and centrifugal force vector, $G(q)$ is $N\times 1$ gravity vector and q, q and q is $N\times 1$ angular, velocity and acceleration vector respectively. Let $x_i = q_i$, $x_{N+1} = \dot{q}_i\,(i = 1,2,\ldots,N)$, then (19) becomes

$$\dot{X} = A(X) + B(X)U \tag{20}$$

$$Y = C(X) \tag{21}$$

where $X = [X_1\ X_2]^T = [x_1,\ldots,x_N\ x_{N+1},\ldots,x_{2N}]^T$, $U = [\tau_1,\ldots,\tau_N]^T$, $C(X) = X_1$,

$$A(X) = \begin{bmatrix} X_2 \\ -H^{-1}(M+G)\end{bmatrix}, \quad B(X) = \begin{bmatrix} 0 \\ -H^{-1}\end{bmatrix}.$$

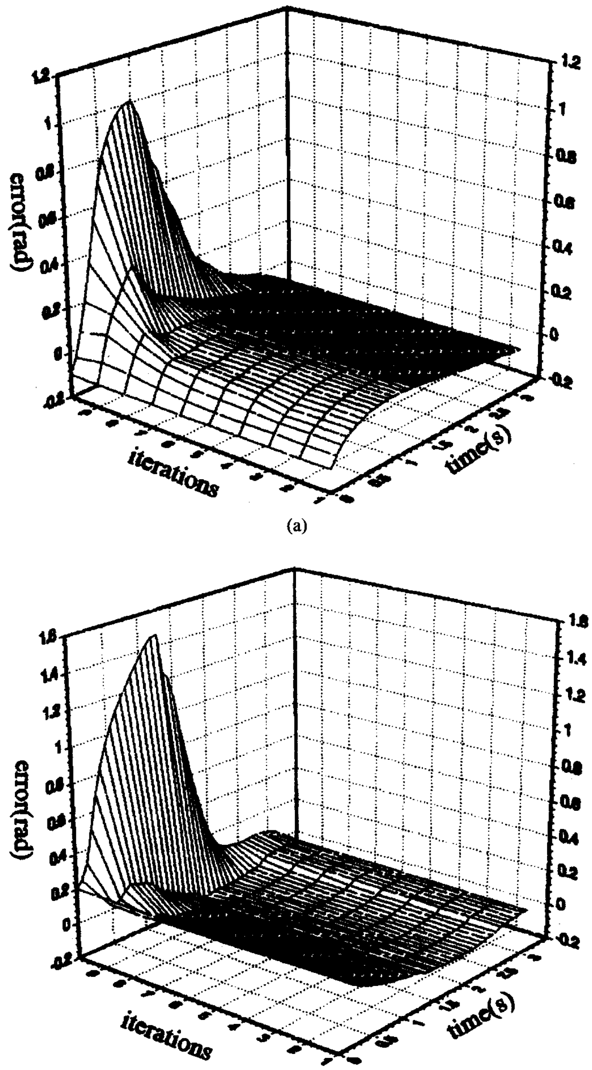
Define the following nonlinear operator [10]:

$$N_A^jC_i(X) = [\partial/\partial X(N_A^{j-1})C_i(X)]A(X), \quad j = 1,2,\ldots,N-1 \tag{22}$$

$$N_A^0C_i(X) = C_i(X) \tag{23}$$

where $C_i(X)$ is i-th row of $C(X)$. Define the relative degree of the system

$$d_i = \min\{j : [\partial/\partial X(N_A^{j-1})C_i(X)]B(X) \neq 0, \quad j = 1,2,\ldots,N\} \tag{24}$$



(b)

Fig. 8. Convergence of the tracking error vs. time and iterations. (a) Link 1; (b) link 2.

then we have the following control which decouples the robot dynamics

$$U = F(X) + G(X)V \tag{25}$$

where V is the new control vector of the decoupled linear system, and

$$F(X) = -(D^*)^{-1}(X)(F_1^*(X) + F_2^*(X)) \tag{26}$$

$$G(X) = -(D^*)^{-1}(X)\Lambda \tag{27}$$

$$D_i^*(X) = [\partial/\partial X(N_A^{di})C_i(X)]B(X) \tag{28}$$

$$F_{1i}^*(X) = N_A^{d_i}C_i(X) \tag{29}$$

$$F_{2i}^*(X) = \sum_{k=1}^{d_i-1} a_{k,i}N_A^kC_i(X) \tag{30}$$

$$\Lambda = \mathrm{diag}[\lambda_1,\ldots,\lambda_N] \tag{31}$$

where $D_i^*(X), F_{1i}^*(X)$ and $F_{2i}^*(X)$ is the ith row of matrix $D^*(X)$, $F_1^*(X)$ and $F_2^*(X)$ respectively, $a_{k,i}$ are some constants to be determined. For the robot system given in (19), since

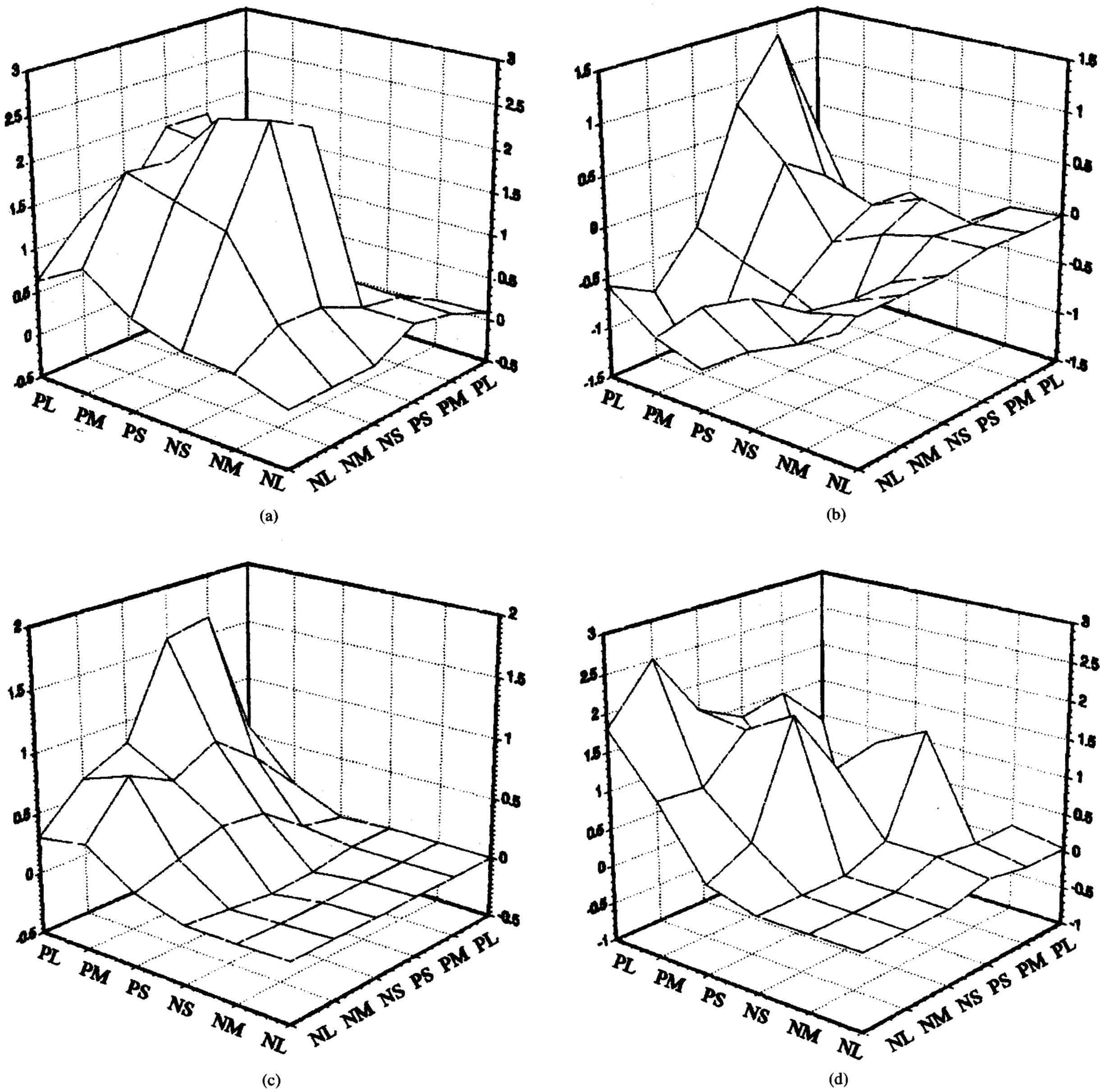


Fig. 9. Distribution of the consequence parameters. (a) P_0, (b) P_1 of link 1. (c) P_2 of Link 1 and (d) P_0 of link 2.

$[\partial/\partial X(N_A^0)C_i(X)]B(X) = 0$ and $[\partial/\partial X(N_A^1)C_i(X)]B(X) \neq 0$, then $d_i = 2\,(i = 1, \ldots, N)$. Then we have the following decoupled model:

$$\begin{bmatrix} \dot{x}_1 \\ : \\ \dot{x}_N \\ \dot{x}_{N+1} \\ : \\ \dot{x}_{2N} \end{bmatrix} = \begin{bmatrix} x_{N+1} \\ : \\ x_{2N} \\ -a_{0,1}x_1 - a_{1,1}x_N \\ : \\ -a_{0,N}x_N - a_{1,N}x_{2,N} \end{bmatrix} + \begin{bmatrix} & 0 & \\ \lambda_1 & & 0 \\ & : & \\ 0 & & \lambda_N \end{bmatrix} \begin{bmatrix} v_1 \\ : \\ v_N \end{bmatrix} \tag{32}$$

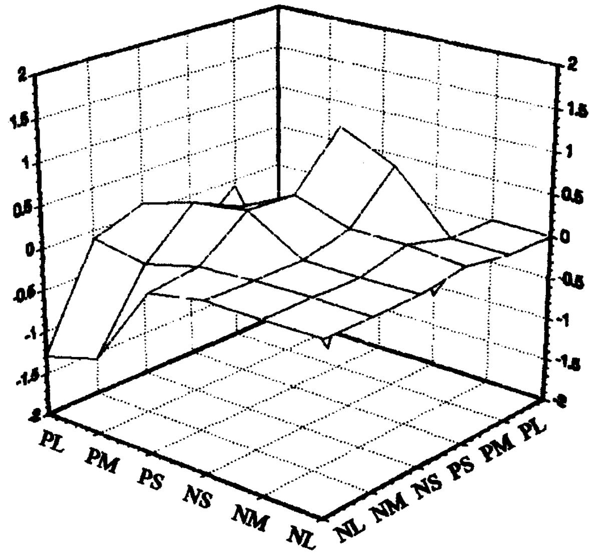
Obviously, the decoupled subsystem for each link is a linear system, which has two inputs and one outputs. Parameter a and λ should be properly chosen so that the linear subsystem is stable.

B. Simulation Study of Adaptive Fuzzy Control for a Two-Degree-of-Freedom Robot

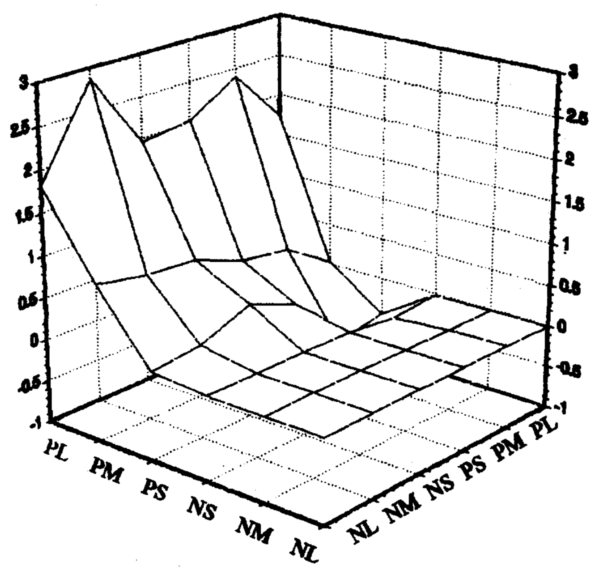
Without loss of generality, we study with a two-link simple robot. The decoupled model for each link is set by

$$0.1\ddot{q}_i + \dot{q}_i = V_i \quad (i = 1, 2). \tag{33}$$

Choose $x_1^i = q_i$, $x_2^i = q_i^d - q_i$ and $y^i = V_i$ ($i = 1, 2$), where x_1^i, x_2^i are two input variables of the fuzzy controller and y^i is the output variable. When we train the fuzzy model on-line, the desired output of the neural network is calculated from (33). Fig. 7 shows the structure of the developed control system. The training procedure is given in Fig. 8. It demonstrates that after 10 times of training, the performance of the system is satisfying. Fig. 9 presents the distribution of each consequence parameter in the fuzzy space. In the simulation, we



(e)



(f)

Fig. 9. (*Continued*) Distribution of the consequence parameters. (e) v P_1 and (f) P_2 of link 2.

assume the real robot model is $(m_1, m_2, l_1, l_2) = (2.0, 1.0, 0.223, 0.2)$, where m_i and l_i are the matter and length of each link. Suppose there exists a matter error of 50% and 40% for each link respectively and the desired trajectory for link 1 and link 2 is taken respectively as

$$q_1^d(t) = \exp(0.5t)_{(\mathrm{rad})} \quad q_2^d(t) = 1.5\exp(1.5t)_{(\mathrm{rad})} \tag{34}$$

$$q_1(0) = 1.1_{(\mathrm{rad})} \qquad q_2(0) = 1.3_{(\mathrm{rad})}. \tag{35}$$

VI. Conclusion

In this paper, we have extended the fuzzy identification algorithm proposed in [5]. The new algorithm is based on a hybrid neural network structure. We believe that our method makes it easier to design a fuzzy model for a complex system. The performances of the novel fuzzy model are also improved.

However, as the number of the input variable and the number of fuzzy subspaces increase, the consequence parameters to identify increase rapidly, which makes the identification much time-consuming. Therefore, it is desirable to develop a more suitable form of consequence function so as to decrease the number of the consequence parameters without deteriorating the performance of the system.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–352, 1965.

[2] E. Czogala and W. Pedrycz, "On identification in fuzzy systems and its applications in control problem," *Fuzzy Sets and Systems*, vol. 10, pp. 73–83, 1981.

[3] W. Pedrycz, "An identification algorithm in fuzzy relational systems," *Fuzzy Sets and Systems*, vol. 13, pp. 153–167, 1984.

[4] M. Higashi and G. J. Klir, "Identification of fuzzy relation systems," *IEEE Trans. Syst., Man, Cyber.*, vol. SMC-14, pp. 349–355, 1984.

[5] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cyber.*, vol. SMC-15, pp. 116–132, 1985.

[6] M. Sugeno and K. Tanaka, "Successive identification of a fuzzy model and its application to prediction of a complex system," *Fuzzy Sets and Systems*, vol. 42, pp. 315–324, 1991.

[7] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Trans. Ind. Electron.* , vol. 39, pp. 472–489, 1992.

[8] R. Durbin and D. E. Rumelhart, "Product units: A computationally powerful and biologically plausible extension to backpropagation networks," *Neural Computation*, vol. 1, pp. 133–142, 1989.

[9] W. X. Zhang and Y. K. Kong, "Fuzzy control weather forecast model," *Chinese J. Fuzzy Sets*, vol. 3, pp. 32–38, 1982.

[10] E. Freud, "The structure of decoupled nonlinear system," *Int. J. Contr.*, vol. 21, pp. 443–450, 1975.