1

A Novel Fuzzy Modeling Structure – Decomposed Fuzzy System

Shun-Feng Su, *Fellow, IEEE*, Ming-Chang Chen, and Yao-Chu Hsueh

1 his work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 104-2221-E-011-090 -MY3

S.-F. Su is with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan.

(e-mail: sfsu@mail.ntust.edu.tw).

M.-C. Chen is with the National Chung-Shan Institute of Science and Technology, Taoyuan City 32546, Taiwan.

(e-mail: maximchen.chen@gmail.com).

Y.-C. Hsueh is with the Advantech Co., Ltd, Taipei, Taiwan.

(e-mail: D9307202@mail.ntust.edu.tw)

*Abstract*—Decomposed fuzzy system (DFS) is a fuzzy system with a novel structure. Due to its excellent learning performance, DFS is originally proposed for an online learning control scheme [1] and is shown to have effective learning performance. This study is about the use of DFS for modeling dynamic systems. Since the learning mechanism used in online learning control is not suitable for modeling tasks, a commonly-used back propagation learning algorithm is adapted for the use of DFS in modeling dynamic systems. The structure of DFS is to decompose each fuzzy variable into fuzzy sub-systems that are called component fuzzy systems. Owing to the independency among component fuzzy systems, the learning for those parameters is also independent among different component fuzzy systems and thus, the learning can become more efficient. From the simulation results, it is evident that the proposed DFS can have much faster convergent speed. In addition, the DFS has a smaller testing error than those of other fuzzy systems.

*Index Terms*—Decomposed fuzzy system, back propagation learning algorithm, identification.

# I. INTRODUCTION

A

s proved in the literature [2-5], neural networks or fuzzy systems are universal approximators. Thus, they have been pervasively employed to model dynamic systems [4-12]. These approaches are to construct systems directly from the input- output relationship without the use of any domain knowledge, and hence, they are often referred to as model-free estimators [13]. Neural fuzzy system (NFS) can be said to be a fuzzy system equipped with the learning capability of neural networks. Various studies on NFS have been reported in the literature [14-25]. In [14], by modifying the centers and the widths of Gaussian basis functions, which act as the fuzzy membership functions, a fuzzy radial basis function neural network (FRBFNN) is proposed to have a fast convergence learning process. In [15-18], recurrent fuzzy neural network structures are proposed to resolve temporal problems. The recurrent property is achieved by using the delayed output of each membership function feedback to itself and is referred to as delay feedback in [21]. In [19], an approach combining a fuzzy radial basis function neural network with a recurrent fuzzy neural network (termed as RFRBFNN) is proposed to obtain better function approximate capability and faster convergence speed. Basically, FRBFNN is a traditional fuzzy system with learning on both the premise and consequence parts. RFRBFNN is to have delay feedbacks to have more inputs. In this study, a novel structure of NFS is proposed. It is referred to as the back propagation decomposed fuzzy systems (BPDFS). The idea is completely different from previous approaches. BPDFS is to provide a structure that can have a simple learning process so that the learning can become efficient and then the performance will become better too.

BPDFS is to use the decomposed fuzzy system (DFS) structure for modeling dynamic systems. Originally, DFS is proposed for an online learning control scheme, adaptive fuzzy control, due to its effective learning [1]. DFS have been shown to have very effective learning for adaptive fuzzy control. In this study, whether DFS can also bring advantages to modeling tasks is considered. In the literature, various NSF structures have been proposed. Most of them are equipped with the back propagation learning algorithm. However, most approaches have complicated structures and then even may result in nice approximate capability, they may increase learning burden and cause ineffective learning. It can be found from our study that the structure of DFS together with the commonly-used back propagation learning algorithm not only has very effective learning but also shows better approximate capability in modeling dynamic systems compared to those existing fuzzy systems.

The structure of DFS is to decompose each fuzzy variable into a fuzzy sub-system called component fuzzy system. Conceptually, each component fuzzy system can be viewed an extension of a fuzzy rule in a traditional fuzzy system. We shall give more detailed introduction in the following section. It can be found that the learning mechanism used in adaptive fuzzy control is based on the Lyapunov theorem and is not suitable for modeling tasks. A commonly-used back propagation learning algorithm is adapted for the use of decomposed fuzzy systems in modeling dynamic systems. Owing to the independency among component fuzzy systems, the learning for those parameters is also independent among different component fuzzy systems and thus, the learning can become more efficient. In addition, it can be found from our experiments that the generalization capability can also become better when DFS is used.

This paper is divided into five sections. After this introduction section, Section II outlines the structure of the DFS. The learning process of the proposed BPDFS is introduced in Section III. Section IV presents a number of examples to verify the effectiveness of the proposed BPDFS. Finally, conclusions are given in Section V.

# II. Decomposed Fuzzy Systems

In this section, a novel fuzzy system structure called the decomposed fuzzy system is introduced for the use in modeling dynamical systems. DFS is originally proposed for adaptive fuzzy control and has been shown to have nice learning performance [1]. Basically, the layer number is similar to the fuzzy partition number in traditional fuzzy systems. In our study, the selection of the layer number is to have one component fuzzy system corresponding to one fuzzy rule in a traditional fuzzy system so that the learning effects of DFS can be viewed as the variation effects of membership functions in traditional fuzzy systems. Normally, the layer number can be selected based on the required resolution.

Consider a fuzzy partition as shown in the lower part of Fig. 1. The fuzzy variable  is partitioned into  fuzzy sets where. To construct the partition of DFS, the fuzzy input variable is decomposed into  layers as shown in Fig. 2. Each layer (for *i*=1, …,) consists of one fuzzy set () corresponding to the original fuzzy set and two complement fuzzy sets ( and ). When there are *n* input fuzzy variables and each variable has layers, those layers can form  fuzzy sub-systems called the component fuzzy systems.

Now, the *l-*th fuzzy IF-THEN rule of the *d*-th component system is written as

, (1)

where  is the input variable vector,  and  are the corresponding input and output fuzzy sets, respectively.  is the output variable, and  are those  in Fig. 2. Note that ’s are fuzzy singletons and are parameters to be tuned in the learning process. According to the center-average defuzzification and the product inference, the output of the *d*-th component fuzzy system is defined as

 , (2)

where  is the membership function of the fuzzy set  and  is the rule number in one component fuzzy system.  is the adjustable parameter vector and  is a fuzzy basis vector, where  is defined as

. (3)



Fig. 1. Uniform fuzzy partition of a fuzzy variable 



Fig. 2. Antecedent-part design of DFS

Hence, the average of the outputs of all component fuzzy systems can be expressed as

, (4)

where  is the number of component fuzzy systems. The general term of (4) can be represented as

, (5)

where .  can be viewed as the weighting factor for the corresponding component fuzzy system. If there exist some performance indices that can be used to evaluate the related importance of component fuzzy systems, it is possible to define the meaningful values for those**’s.** However, there is no way of evaluating which component fuzzy systems may be more important than the others. Thus, in this study, those component fuzzy systems are equally treated and then will have the same weights; that is,  as in (4). With such a simple idea, the performance is promising as shown in our experiments.

# III. Back Propagation Decomposed Fuzzy Systems

As mentioned, DFS is originally proposed for an online learning control scheme, adaptive fuzzy control [1]. The learning mechanism used in adaptive fuzzy control is based on the Lyapunov theorem and is not suitable for modeling tasks. Thus, in this study, the back propagation learning algorithm is adapted. This system is referred to as the Back Propagation Decomposed Fuzzy Systems (BPDFS). In this section, the detailed algorithm is introduced.

The configuration of the proposed BPDFS is shown in Fig. 3. There are four layers, which are described respectively in the following.

Layer 1: In this layer,  is input vector, and each node represents an input linguistic variable , .

. (6)

Layer 2: In this layer, each node performs a membership function mapping. In our study, the partition is selected to have one component fuzzy system corresponding to one fuzzy rule in a traditional fuzzy system. For each layer, it consists of the original fuzzy set and two complement as Fig. 2. For the jth fuzzy set  on the input variable, the membership function is defined as

. (7)

Layer 3: In this layer, each node performs a fuzzy *t*-norm operation (algebraic product) on inputs that is received from layer 2 as

 (8)

Layer 4: This layer performs the defuzzification operation. The outputs standing for the values of the coefficient vector are expressed as

, (9)

where  is the adjustable parameter vector and  is the rule number.



Fig. 3. Configuration of the proposed BPDFS

The task of learning is to minimize the error function

, (10)

where  is the training pattern number,  represents the DFS output for pattern , and  is the desired output for pattern . Let  be the parameter vector to be tuned in the training process at time *t*. The basic idea is to use the gradient descent like algorithm to tune those parameters. In other words,

, (11)

where  is the learning rate, which is usually selected by user and is dependent on the training data considered. It can be seen that by using the traditional error function as (10), the update algorithm as (11) needs to consider all training patterns for one update. The main difference of the back propagation learning algorithm from gradient descend is that the upload algorithm is for one datum only. Thus, the gradient used becomes

, (12)

where  is the error function. It should be noted that the idea of using DFS is to have fixed parameters in the antecedent part. Thus, the back propagation learning algorithm is applied to tuning the parameters of the consequence part only. Then, the weight update rule of BPDFS is

, (13)

where  is the basis function obtained from (3). In performance evaluation, the root-mean-square error (RMSE) is considered and is defined as

 . (14)

Finally, in order to illustrate the algorithm, the overall scheme of the proposed system is shown in Fig. 4.



Fig. 4. Overall scheme of the back propagation decomposed fuzzy systems

# IV. Simulation Results

In this section, several different identification models used in the literature [26-31] are employed to demonstrate the nice performance of the proposed BPDFS. These models include two dynamic systems and one chaotic series. In order to show the superiority of the proposed approach, several approaches used in the literature for modeling systems are also considered here. Those approaches are the back propagation fuzzy system (BPFS) [26], the fuzzy radial basis function neural network (FRBFNN) [14], the recurrent fuzzy radial basis function neural network (RFRBFNN) [15], and the proposed back propagation decomposed fuzzy system (BPDFS). BPFS is a traditional fuzzy system and the learning part is similar to BPDFS. In other words, only the consequence parts are tuned by back propagation learning algorithm. FRBFNN is similar to BPFS but the antecedent part is also tuned by the back propagation learning algorithm simultaneously. RFRBFNN is similar to FRBFNN, but with an extra information by using delay feedbacks to have one more order of input [21]. Finally, the proposed method is also compared to fuzzy CMAC techniques. The CMAC network is a connectionist-based model developed to capture and emulate the learning mechanism and function approximating capabilities of the cerebellum. In fact, CMAC can simply be viewed as a table-look-up learning mechanism with overlapping layers to facilitate generation capability. Thus, CMAC is usually used for applications requiring fast learning [27]. Since there is no optimization process used in the CMAC learning algorithm, the modeling accuracy of CMAC is usually worse than tradition fuzzy systems or radial basis function networks. Also, due to fast learning characteristics, CMAC may also suffer from serious degradation when there are outliers [28]. That is why there is little direct comparison between CMAC and fuzzy modeling systems in the literature. Another drawback of CMAC is state-based modeling accuracy, which means CMAC cannot interpret within one cell. In order to solve the problem, the fuzzy CMAC was implemented in [20-22, 28]. It can easy to check fuzzy CMAC can achieve much better modeling accuracy than traditional CMAC. From the simulation results, it is evident that the BPDFS can obtain a good function approximation capability and much better convergent behavior than those CMAC approaches.

**Example 1:** First, the discrete-time Henon chaotic sequence of dynamic system with one delay and two sensitive parameters is considered in our study. The Henon chaotic sequence is generated by the following equation [29]:

. (15)

Eq.(15) produces a chaotic strange attractor with  and . In this example, the inputs are  and . The output is . The initial states  . The learninig rate is 0.2. Same as that used in [26], there are 2000 patterns with 1000 patterns used for training and the remaining 1000 patterns used for testing. The termination condition is the error variation less than 1e-005. The used fuzzy systems are with =3.

Fig. 5 shows the training data of the chaotic sequence prediction of dynamic system and the output of BPDFS, where the reference data are shown by the blue spots and the BPDFS results are the red spots. Fig. 6 shows the results for the testing data. Fig. 7 shows the RMSE histories of the proposed BPDFS and of the other fuzzy systems used. From the learning histories, it is clearly evident that the proposed BPDFS has a much faster convergence speed than the others do. Tables 1 shows the numbers of rules, training epochs when the learning is finished, the training RMSE, and the testing RMSE for all four systems employed. It can be found that not only BPDFS can have the fastest learning process, but also it can have the lowest testing error. It should be noted that BPDFS and BPFS only tune the parameters in the consequence part and the other two systems have their antecedent and consequence parts tuned in the learning process simultaneously. Besides, RFRBFNN include extra information into the system, but still does not get a better performance. It is because as mentioned in [21], the input variables are sufficient now and the use of an extra order cannot bring any advantage.

In order to have the same number of rules, 9 fuzzy sets for all variables are considered for the other fuzzy systems. The results are shown in Table 2. It can be found that all systems have the same number of rules. Nevertheless, BPDFS still perform better than the others do no matter in the learning efficiency or in the testing error. Next, in order to show the learning efficiency of BPDFS, the rule number is changed to =5. The results are given in Table 3, which also include the results of =3 for comparison. Although the fuzzy rules are increased from 81 to 225, the number of epochs only increases slightly. Furthemorem, the training and testing errors are both improved when =5. Finally, the proposed method is also compared to fuzzy CMAC techniques [22] in the case of the discrete-time Henon chaotic sequence of dynamic system. After 100 training epochs, the simulation results are shown in Table 4. From the results, it can be found as expected that the BPDFS not only use less parameter, but also obtain a much better performance.



Training data

Fig. 5. Results of the training for Example 1



Testing data

Fig. 6. Results of the testing for Example 1



Iterations

Fig. 7. RMSE histories of all fuzzy systems for Example 1

Table 1.

Performances of all fuzzy systems used for chaotic sequence prediction in Example 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | BPFS | FRBFNN | RFRBFNN | BPDFS |
| Rules | 25 | 25 | 25 | 81 |
| Epoch | 438 | 1623 | 1675 | 122 |
| Training RMSE | 3.11e-002 | 3.05e-002 | 3.22e-002 | 2.75e-002 |
| Testing RMSE | 3.23e-002 | 3.07e-002 | 3.23e-002 | 3.06e-002 |

Table 2.

Performances of all fuzzy systems used for chaotic sequence prediction with the same number of rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | BPFS | FRBFNN | RFRBFNN | BPDFS |
| Rules | 81 | 81 | 81 | 81 |
| Epoch | 174 | 337 | 1634 | 122 |
| Training RMSE | 3.11e-002 | 3.34e-002 | 2.77e-002 | 2.75e-002 |
| Testing RMSE | 3.5 e-002 | 3.26e-002 | 3.95e-002 | 3.06e-002 |

Table 3.

Performances of BPDFS for Example 1 by using *N**d* =3 and *N**d* =5

|  |  |  |
| --- | --- | --- |
| Models | BPDFS (*Nd* = 3) | BPDFS(*Nd* =5) |
| Rules | 81 | 225 |
| Epoch | 122 | 160 |
| Training RMSE | 2.75e-002 | 2.16e-002 |
| Testing RMSE | 3.06e-002 | 2.34e-002 |

Table 4.

Performances of BPDFS and FCMAC used for chaotic sequence prediction

|  |  |  |
| --- | --- | --- |
| Models | FCMAC [22] | BPDFS |
| Number of parameters | 112 | 81 |
| Epoch | 100 | 100 |
| Average MSE | 0.0141 | 0.0075 |

**Example 2:** The second example is a nonlinear model used in [30, 31]. The model is

.(16)

Same as used in [30, 31], the input signal in the training process is

. (17)

and the input signal in the testing process is

. (18)

The learninig rate is 0.2. The initial value of  is 0. In the example, 1000 training patterns and 1000 testing patterns are generated. The termination condition is the error variation less than 1e-005. The used fuzzy systems are with =3.

Fig. 8 shows the training data (blue line) and the outputs generated by BPDFS (red spots). Fig. 9 shows the result for testing data. Fig. 10 shows the RMSE histories of all systems used. Tables 5 lists the number of rules, the training epochs, the training RMSE, and the testing RMSE of those four fuzzy systems. Also, Table 6 shows the results of using the same number of rules for all systems. It can be found again that not only BPDFS can have the fastest learning process, but also it can have the lowest testing error. When all systems have the same number of rules, BPDFS still perform better. In Table 7, the case of =5 (together with that of =3) are shown. The the number of fuzzy rules increases from 729 to 3375. It is well known that a more complex fuzzy partition may lead to a better function approximation capability, but the learning of such a fuzzy approximator may become inefficient or even degrade the performance. Nevertheless, from our simulation results, it can be found that the phenomenon does not exist in the BPDFS. Moreover, the required learning epochs for BPDFS nearly the same as that for =3 even when the parameters of the fuzzy system increase dramatically.



Training data

Fig. 8. Results of the training for Example 2



Test data

Fig. 9. Results of the testing for Example 2



Iterations

Fig. 10. RMSE histories of all fuzzy systems for Example 2

Table 5.

Performances of all fuzzy systems used for Example 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | BPFS | FRBFNN | RFRBFNN | BPDFS |
| Rules | 125 | 125 | 125 | 729 |
| Epoch | 482 | 193 | 563 | 89 |
| Training RMSE | 3.48e-002 | 3.83e-002 | 3.31e-002 | 3.01e-002 |
| Testing RMSE | 1.79e-002 | 2.01e-002 | 1.34e-002 | 1.07e-002 |

Table 6.

Performances of all fuzzy systems used forExample 2 with the same number of rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | BPFS | FRBFNN | RFRBFNN | BPDFS |
| Rules | 729 | 729 | 729 | 729 |
| Epoch | 162 | 324 | 245 | 89 |
| Training RMSE | 3.28e-002 | 3.53e-002 | 3.84e-002 | 3.01e-002 |
| Testing RMSE | 2.2e-002 | 1.72e-002 | 2.06e-002 | 1.07e-002 |

Table 7.

Performances of BPDFS for Example 2 by using *N**d* =3 and *N**d* =5

|  |  |  |
| --- | --- | --- |
| Models | BPDFS (*Nd* =3) | BPDFS(*Nd* =5) |
| Rules | 729 | 3375 |
| Epoch | 89 | 87 |
| Training RMSE | 3.01e-002 | 3.07e-002 |
| Testing RMSE | 1.07e-002 | 1.11e-002 |

# V. Conclusions

This paper reports our study on the use of a novel fuzzy structure, Decomposed Fuzzy Systems (DFS) for modeling dynamic systems. A commonly-used back propagation learning algorithm is adapted. The DFS is composed of multiple component fuzzy systems, and each component fuzzy system is based on one traditional fuzzy rule. Owing to the independency among component fuzzy systems, the learning for those parameters is also independent among different component fuzzy systems and thus, the learning can become more efficient. From the simulation results, it is evident that BPDFS can not only have a faster learning process, but also achieve a lower testing error compared to those existing fuzzy systems. We believe that the proposed structure is a very good idea and can provide a great impact on the use of fuzzy systems.

References

[1] Y.-C. Hsueh, S.-F. Su, and M.-C. Chen, “Decomposed fuzzy systems and their application in direct adaptive fuzzy control,” *IEEE Trans. on Cybernetics*, vol. 40, no. 10, pp. 1772-1783, Oct. 2014.

[2] K. Hornik, M. Stinchcombe and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol.2, no.5, pp. 359–366, 1989.

[3] L.-X. Wang and J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least square learning,” *IEEE Trans. on Neural Networks*, vol.3, no.5, pp. 807–814, 1992.

[4] Y. Li, and S. Tong, “Fuzzy Adaptive Backstepping Decentralized Control for Switched Nonlinear Large-Scale Systems with Switching Jumps,” *International Journal of Fuzzy Systems*, vol. 17, no. 1, pp. 12 - 21, March 2015.

[5] B. Kosko, “Fuzzy systems as universal approximators,” *in* *Proc. of the IEEE Int’l Conf. On Fuzzy Systems,* pp. 1153–1162, 1992.

[6] C.-H. Wang, W.-Y. Wang, T.-T. Lee and P.-S. Tseng, “Fuzzy B-spline membership function (BMF) and its applications in fuzzy-neural control,” *IEEE Trans. on Systems,* *Man and Cybernetics*, vol.25, no.5, pp. 841–851, 1995.

[7] C.-L. Hwang and L.-J. Chang, “Fuzzy neural-based control for nonlinear time-varying delay systems,” *IEEE Trans. on System Man and Cybernetics-Part B: Cybernetics*, vol.37, no.6, pp. 1471–1485, 2007.

[8] C.-F. Hsu, C.-M. Lin and T.-T. Lee, “Wavelet adaptive backstepping control for a class of nonlinear systems,” *IEEE Trans. on Neural Networks*, vol. 17, no. 5, pp. 1175–1183, 2006.

[9] L. Zadeh, “Outline of a new approach to the analysis of complex systems and decision process,” *IEEE Trans. on Systems,* *Man,* *and Cybernetics,* vol. 14, no. 1, pp. 28–44, 1973.

[10] W. Pedrycz, “Structured fuzzy models,” *Cybernetics and Systems,* vol. 16, pp. 103–117, 1986.

[11] L.-X. Wang and J. M. Mendel, “Generating fuzzy rules by learning from examples,” *IEEE Trans. on Systems,* *Man,* *Cybernetics,* vol. 22, no. 6, pp. 1414–1427, 1992.

[12] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control,* John Wiley & Sons, 1994.

[13] B. Kosko, *Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, 1992.

[14] S. B. Roh, S. K. Oh, and W. Pedrycz, “Design of fuzzy radial basis function-based polynomial neural networks,” *Fuzzy Sets and Systems,* vol. 185, no. 1, pp. 15–37, Dec. 2011.

[15] P. A. Mastorocostas and J. B. Theocharis, “A recurrent fuzzy-neural model for dynamic system identification,”*IEEE Trans. on Systems,* *Man,* *and Cybernetics*, vol. 32, no. 2, pp. 176–190, Apr. 2002.

[16] J.-S. Wang and Y.-P. Chen, “A hammerstein recurrent neurofuzzy network with an online minimal realization learning algorithm,” *IEEE Trans. on Fuzzy Syst.* vo. 16, no. 6, pp. 1597–1612, Dec. 2008.

[17] J. Hu and J. Wang, “Global stability of complex-valued recurrent neural networks with time-delays,” *IEEE Trans. on Neural Netw. Learning Syst.,* vol. 23, no.6, pp. 853–865, Jun. 2012.

[18] C.-T. Lin, C.-L. Chang, and W.-C. Cheng, “A recurrent fuzzy cellular neural network system with automatic structure and template learning,” *IEEE Trans. on Circuits and Systems-I,* vol. 51, no. 5, pp. 1024-1035, May 2004.

[19] F.-J. Lin, P.-H. Shen, S.-L. Yang, and P. H. Chou, “Recurrent radial basis function network-based fuzzy neural network control for permanent-magnet linear synchronous motor servo drive,” *IEEE Trans. on Magnetics*, vol. 42, no. 11, Nov. 2006.

[20] C. H. Lee, F. Y. Chang, and C. M. Lin, “An Efficient Interval Type-2 Fuzzy CMAC for Chaos Time-Series Prediction and Synchronization,” *IEEE Trans. on Cybernetics*, vol. 44, no. 3, pp. 329 - 341, 2014.

[21] S.-F. Su and F.-Y. P. Yang, “On the dynamical modeling with neural fuzzy networks” *IEEE Trans. on Neural Networks*, vol. 13, no. 6, pp. 1548–1553, 2002.

[22] J. Sim, W. L. Tung, and C. Qeuk, “FCMAC-Yager: A novel yagerinference-scheme-based fuzzy CMAC,” *IEEE Trans. on Neural Networks*, vol. 17, no. 6, pp. 1394-1410, Nov. 2006.

[23] P. Liu and H. Li, “Efficient learning algorithms for three-layer regular feedforward fuzzy neural networks,” *IEEE Trans. on Neural Networks*, vol. 15, no. 3, pp. 545 - 558, 2004.

[24] G.-C. Liao and T.-P. Tsao, “Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting,” *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 3, pp. 330-340, 2006.

[25] W. Pedrycz and K.-C. Kwak, “Linguistic models as a framework of user-centric system modeling,” *IEEE Trans. on Systems,* *Man,* *and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 4, pp. 727-745, 2006.

[26] S. Stoeva and A. Nikov, “A fuzzy backpropagation algorithm,” *Fuzzy Sets and Systems*, vol. 112, pp. 27–39, 2000.

[27] S.-F. Su, T. Tao, and T.-H. Hung, “Credit assigned CMAC and its application to online learning robust controllers,” *IEEE Trans. on Systems,* *Man,* *and Cybernetics,* *Part B: Cybernetics,* vol. 33, no. 2, pp. 202-213, 2003

[28] S.-F. Su, Z.-J. Lee, and Y.-P. Wang, “Robust and fast learning for fuzzy Cerebellar model articulation controllers,” *IEEE Trans. on Systems,* *Man,* *and Cybernetics,* *Part B: Cybernetics*, vol. 36, no. 1, pp. 203-208, 2006.

[29] C.-F. Juang, Y.-Y. Lin, and C.-C. Tu, “A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing,” *Fuzzy Sets and Systems*, vol. 161, pp. 2552–2568, 2010.

[30] C.-F. Juang, “A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms,” *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 115–170, 2002.

[31] P. S. Sastry, G. Santhara, and K. P. Unnikrishnan, “Memory Neural Networks for Identification and Control of Dynamical Systems,” *IEEE Trans. on Neural Networks*, vo5. 1, no. 2, pp. 306–319, 1994.

**Shun-Feng Su** (S’89–M’91–SM’05–F’10) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1991, respectively. He is currently a Chair Professor of the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei. He has published more than 160 refereed journal and conference papers in the areas of robotics, intelligent control, fuzzy systems, neural networks, and non-derivative optimization. His current research interests include computational intelligence, machine learning, virtual reality simulation, intelligent transportation systems, smart home, robotics, and intelligent control. Dr. Su is very active in various international/domestic professional societies. He is now the President of the Taiwan Fuzzy Systems Association and is the President-Elect of International Fuzzy Systems Association. He also serves in the Boards of Governors of the Chinese Automatic Control Society, the Taiwan Society of Robotics, and the Taiwan Association of System Science and Engineering. He also acted as the Program Chair, Program Co-Chair, or PC Members for various international and domestic conferences. He currently serves as an Associate editor of the IEEE Transactions on Systems, Man, Cybernetics, Part B, and the IEEE Transactions on Fuzzy Systems, and is the publisher and Area Editor for *International Journal of Fuzzy Systems*. He is a fellow of the CACS.

**Ming-Chang Chen** received the M.S. degree in electronic engineering form Fu-Jen Catholic University, Taipei, Taiwan, and the Ph.D. degree in electrical engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2006 and 2011, respectively. From 2012 to 2016, he was a Post-Doctoral Fellow with the Department of Electrical Engineering, National Taiwan University of Science and Technology. He is currently a researcher at the National Chung-Shan Institute of Science and Technology. His current research interests include anti-lock braking system, fuzzy logic system, active suspension system, adaptive control, and intelligent control.

**Yao-Chu Hsueh** received the A.S. degree in electrical engineering from Oriental Institute of Technology, Taipei, Taiwan, in 2000, the B.S. degree in electrical engineering from Ta Hwa Institute of Technology, Hsinchu, Taiwan, in 2002, the M.S. degree in electronic engineering from China Institute of Technology, Hsinchu, in 2004, and the Ph.D. degree in electrical engineering from National Taiwan University of Science and Technology, Taipei, in 2009.

From 2009 to 2011, he was a Post-Doctoral Fellow with the Communication and Electromagnetic Technology Center, National Taiwan University of Science and Technology. His current research interests include intelligent systems, FPGA-based factory automation, adaptive controls, and robust controls.