Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Self-Evolving Interval Type-2 Fuzzy Neural Network for Nonlinear Systems Identification

Chia-Feng Juang*, Chun-Feng Lu, and Yu-Wei Tsao***

**Department of Electrical Engineering, National Chung Hsing University*
Taichung, Taiwan, R.O.C. (e-mail:cfjuang@dragon.nchu.edu.tw).
***Department of Electrical Engineering, Chung Chou Institute of Technology,*
Changhua, Taiwan, R.O.C.

Abstract: This paper proposes a Self-Evolving Interval Type-2 Fuzzy Neural Network (SEIT2FNN) for nonlinear systems identification. The SEIT2FNN has both on-line structure and parameter learning abilities. The antecedent parts in each fuzzy rule of the SEIT2FNN are interval type-2 fuzzy sets and the fuzzy rules are of the Takagi-Sugeno-Kang (TSK) type. An on-line clustering method is proposed to generate fuzzy rules which flexibly partition the input space. For parameter learning, the consequent part parameters are tuned by a rule-ordered Kalman filter algorithm for high accuracy learning performance. The antecedent part parameters are learned by gradient descent algorithms. Comparisons with identification using other type-1 and type-2 fuzzy systems verify the performance of the SEIT2FNN.

1. INTRODUCTION

Type-2 Fuzzy logic systems (FLS) are extensions of type-1 FLS, where the membership value of a type-2 fuzzy set is a type-1 fuzzy number (Karink et al, 1999, Mendel and John, 2002, Mendel 2001). Type-2 FLS appear to be a more promising method than their type-1 counterparts in handling problems with uncertainties such as noisy data and different word meanings. That is, type-2 fuzzy sets allow researchers to model and minimize the effects of uncertainties in rule-based systems. Type-2 FLS have been successfully applied to some control problems (Melin and Castillo, 2004, Hagras, 2004).

Fuzzy rule derivation is often difficult and time-consuming, and requires expert knowledge. This creates a common bottleneck in fuzzy system design. To overcome this disadvantage, many Fuzzy Neural Networks (FNNs) have been proposed (Jang, 1993, Lin and Lin, 1997, Juang and Lin, 1998, Kukolj and Levi, 2004). Most FNNs are proposed for the type-1 FLS design. Type-2 fuzzy rules are more complex than type-1 fuzzy rules because of their use of type-2 fuzzy sets in antecedent or consequent parts. Therefore, most type-2 FNN research is only concerned with interval type-2 fuzzy systems.

The theory of interval type-2 fuzzy systems was studied in (Liang and Mendel, 2000). In (Lee, et al, 2003), the authors proposed parameter learning of interval type-2 fuzzy system using a Gaussian primary membership functions with uncertain standard deviation as a type-2 fuzzy set. Studies (Wang, et al, 2004, Mendel, 2004, Hagras, 2006) proposed parameter learning algorithms of interval type-2 fuzzy systems using Gaussian primary membership functions with uncertain means as type-2 fuzzy sets.

In the aforementioned type-2 FNNs, only parameters are learned, and the structures are all fixed and must be assigned in advance. Fixed structures may not handle time-varying systems or systems with operating points changing with time. Therefore, the purpose of this paper is to develop a type-2 FNN with both structure and parameter evolution ability. The proposed FNN is called a Self-Evolving Interval Type-2 Fuzzy Neural Network (SEIT2FNN). Initially there are no fuzzy rules in a SEIT2FNN. All of the rules are generated on-line by proposed structure learning which not only helps automate rule generation, but also locates good initial rule positions for subsequent parameter learning. The SEIT2FNN's structure evolution ability makes it more suitable to handling time-varying systems than type-2 fuzzy systems which adjust their parameters based on pre-trained and fixed structures. The consequent parameters in the SEIT2FNN are learned by a rule-ordered Kalman filter algorithm. The antecedent part parameters are learned by gradient descent learning algorithms. Simulations on systems identification are conducted to verify SEIT2FNN performance. These simulations also compare other type-1 and type-2 systems.

The rest of this paper is organized as follows. Section 2 introduces the SEIT2FNN structure. Section 3 introduces structure and parameter learning in a SEIT2FNN. Section 4 simulates nonlinear system identification problems. Finally, Section 5 draws conclusions.

2. SEIT2FNN STRUCTURE

This section introduces the structure of a Self-Evolving Interval Type-2 Fuzzy Neural Network (SEIT2FNN). Figure

1 shows the proposed network structure, which has a total of six layers. This six- layered network realizes an interval type-2 fuzzy system whose consequent part is a linear combination of input variables, i.e., Takagi-Sugeno-Kan (TSK)-type. Each SEIT2FNN rule has the following form

Rule i: IF x_1 is $\tilde{A}_1^i$ AND … AND x_n is $\tilde{A}_n^i$

$$\text{THEN } y \text{ is } \tilde{a}_0^i + \sum_{j=1}^{n} \tilde{a}_j^i x_j, \quad i=1, \ldots, M \quad (1)$$

where $\tilde{A}_j^i$, $j=1,\ldots,n$ and $\tilde{G}^i$ are interval type-2 fuzzy sets, M is the number of rules, $\tilde{a}_j^i$, $j=0,...,n$ are interval sets, and

$$\tilde{a}_j^i = [c_j^i - s_j^i, c_j^i + s_j^i], \; j=0,\ldots,n \quad (2)$$

Detailed mathematical functions of each layer are introduced as follows.

Layer 1 (Input layer): The inputs are crisp values. For input range unification, each node in this layer scales inputs x_i, $i=1,\ldots,n$, to within the range [–1, 1]. No weights to be adjusted in this layer.

Layer 2 (Fuzzification layer): This layer performs the fuzzification operation. Each node in this layer defines an interval type-2 membership function. For the i *th* fuzzy set $\tilde{A}_j^i$ in input variable x_j, a Gaussian primary membership function (MF) having a fixed standard deviation σ and an uncertain mean that takes on values in $[m_1, m_2]$ is used (Fig. 1), i.e.,

$$\mu_{\tilde{A}_j^i} = \exp\{-\frac{1}{2}(\frac{x_j - m_j^i}{\sigma_j^i})^2\} \equiv N(m_j^i, \sigma_j^i; x_j),$$

$$m_j^i \in [m_{j1}^i, \; m_{j2}^i] \quad (3)$$

The footprint of uncertainty (FOU) of this MF can be represented as a bounded interval in terms of upper MF, $\overline{\mu}_{\tilde{A}_j^i}$, and lower MF, $\underline{\mu}_{\tilde{A}_j^i}$, where

$$\overline{\mu}_{\tilde{A}_j^i}(x_j) = \begin{cases} N(m_{j1}^i, \sigma_j^i; x_j) & x_j < m_{j1}^i \\ 1 & m_{j1}^i \le x_j \le m_{j2}^i \\ N(m_{j2}^i, \sigma_j^i; x_j) & x_j > m_{j2}^i \end{cases} \quad (4)$$

and

$$\underline{\mu}_{\tilde{A}_j^i}(x_j) = \begin{cases} N(m_{j2}^i, \sigma_j^i; x_j) & x_j \le \dfrac{m_{j1}^i + m_{j2}^i}{2} \\ N(m_{j1}^i, \sigma_j^i; x_j) & x_j > \dfrac{m_{j1}^i + m_{j2}^i}{2} \end{cases} \quad (5)$$

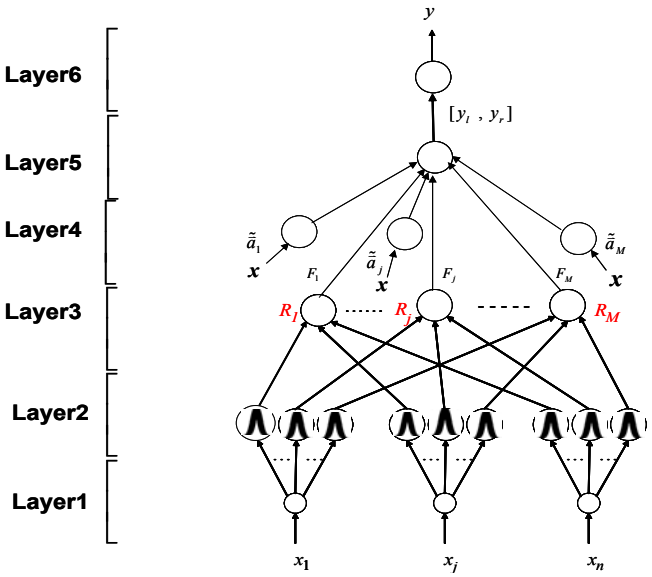

Fig. 1. Structure of the Self-Evolving Interval Type-2 Fuzzy Neural Network (SEIT2FNN)

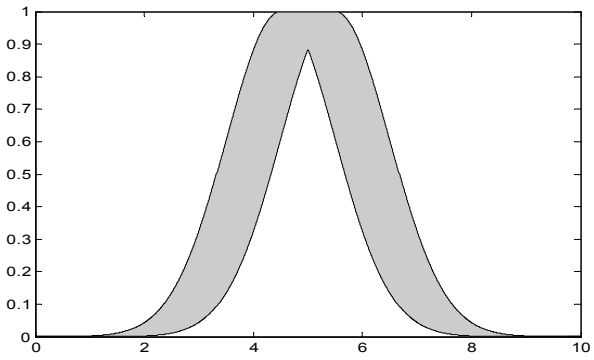

Fig. 2. An interval type-2 fuzzy set with an uncertain mean.

That is, the output of each node can be represented as an interval $[\underline{\mu}_{\tilde{A}_j^i}, \overline{\mu}_{\tilde{A}_j^i}]$.

Layer 3 (Meet layer): Each node in this layer is a rule node, and performs the fuzzy meet operation using an algebraic product operation. The output of a rule node is a firing strength, F^i, which is an interval type-1 fuzzy set. The firing strength is computed as follows (Liang and Mendel, 2000)

$$F^i = [\underline{f}^i, \overline{f}^i] \quad (6)$$

where

$$\overline{f}_i = \prod_{j=1}^{n} \overline{\mu}_{\tilde{A}_j^i} \text{ and } \underline{f}_i = \prod_{j=1}^{n} \underline{\mu}_{\tilde{A}_j^i} \quad (7)$$

Layer 4 (Consequent layer): Each node in this layer is called a consequent node. Each rule node in Layer 3 has its own corresponding consequent node in Layer 4. The output of

each node is an interval type-1 fuzzy set , denoted by $[w_l^i\ ,\ w_r^i]$. According to Eq. (1) and (2), the node output is

$$[w_l^i\ ,\ w_r^i] = [c_0^i - s_0^i,\ c_0^i + s_0^i] + \sum_{j=1}^{n} [c_j^i - s_j^i, c_j^i + s_j^i] \cdot x_j \tag{8}$$

That is,

$$w_l^i = \sum_{j=0}^{n} c_j^i x_j - \sum_{j=0}^{n} \left| x_j \right| s_j^i \tag{9}$$

where $x_0 \triangleq 1$, and

$$w_r^i = \sum_{j=0}^{n} c_j^i x_j + \sum_{j=0}^{n} \left| x_j \right| s_j^i \tag{10}$$

Layer 5 (Output processing layer): The extended output is an interval type 1 set $[y_l, y_r]$. Each node in this layer computes this interval output. The outputs y_l and y_r can be computed using the Karnik-Mendel iterative procedure (Mendel, 2001). In that procedure, the consequent parameters are re-ordered in ascending order. Let $\boldsymbol{w_l} = (w_l^1, \ldots, w_l^M)$ and $\boldsymbol{w_r} = (w_r^1, \ldots, w_r^M)$ denote the original rule-ordered consequent values, and let $\boldsymbol{y_l} = (y_l^1, \ldots, y_l^M)$ and $\boldsymbol{y_r} = (y_r^1, \ldots, y_r^M)$ denote the re-ordered sequence, where $y_l^1 \le y_l^2 \le \cdots \le y_l^M$ and $y_r^1 \le y_r^2 \le \cdots \le y_r^M$. According to (Mendel, 2004), the relationship between $\boldsymbol{w_l}$, $\boldsymbol{w_r}$, $\boldsymbol{y_l}$, and $\boldsymbol{y_r}$ is

$$\boldsymbol{y}_l = Q_l \boldsymbol{w_l} \text{ and } \boldsymbol{y_r} = Q_r \boldsymbol{w_r} \tag{11}$$

where Q_l and Q_r are $M \times M$ permutation matrices. Re-order the f^i accordingly and call them g^i. The outputs y_l and y_r can be computed as follows

$$y_l = \frac{\sum_{i=1}^{L} \overline{g}^i y_l^i + \sum_{i=L+1}^{M} \underline{g}^i y_l^i}{\sum_{i=1}^{L} \overline{g}^i + \sum_{i=L+1}^{M} \underline{g}^i} \tag{12}$$

and

$$y_r = \frac{\sum_{i=1}^{R} \underline{g}^i y_r^i + \sum_{i=R+1}^{M} \overline{g}^i y_r^i}{\sum_{i=1}^{R} \underline{g}^i + \sum_{i=R+1}^{M} \overline{g}^i} \tag{13}$$

Layer 6 (Output layer): Each output node corresponds to one output linguistic variable. Nodes in this layer compute the output linguistic variable y using a defuzzification operation. Because the output of Layer 5 is an interval set, nodes in Layer 6 defuzzify it by computing the average of y_l and y_r. Hence, the defuzzified output is

$$y = \frac{y_l + y_r}{2} \tag{14}$$

3. SEIT2FNN LEARNING

The SEIT2FNN simultaneously uses two types of learning to evolve: structure and parameter learning. The following sections introduce detailed structure and parameter learning algorithms.

3.1. Structure Learning

The first task in structure learning is determining when to generate a new rule. Geometrically, a rule corresponds to a cluster in the input space, and a rule firing strength can be regarded as the degree to which an input data belongs to cluster. Based on this concept, a previous study (Juang and Lin, 1998) used the rule firing strength as a criterion for type-1 fuzzy rule generation. This idea is extended to type-2 fuzzy rule generation criteria in a SEIT2FNN. Since the firing strength in the SEIT2FNN is an interval (see Eq. (6)), the center of the interval is computed

$$f_c^i = \frac{1}{2}(\overline{f}^i + \underline{f}_i) \tag{15}$$

The firing strength center then serves as a rule generation criterion. That is, for each piece of incoming data $\vec{x} = (x_1, \ldots, x_n)$ find

$$I = \arg \max_{1 \le i \le M(t)} f_c^i(\vec{x}) \tag{16}$$

where $M(t)$ is the number of existing rules at time t. If $f_c^I(\vec{x}) \le \phi_{th}$, then a new rule is generated, where $\phi_{th} \in (0,1)$ is a pre-specified threshold. The threshold decides the number of input clusters generated in a SEIT2FNN. A smaller ϕ_{th} value generates a smaller number of rules. Once a new rule is generated, a corresponding fuzzy set in each input variable is generated.

Once a new fuzzy set is generated, it should be assigned with an initial shape. According to the notation in Eq. (3), the initial uncertain mean m_j^i and standard deviation σ_j^i for the new interval type-2 fuzzy set in input variable x_j are

$$m_j^i \in [x_j - 0.1,\ x_j + 0.1] \tag{17}$$

$$\sigma_j^i = 0.2 \tag{18}$$

Repeating the above process for every incoming piece of training data generates new rules, one after another, until the complete IT2FNN is finally constructed.

3.2. Parameter Learning

The parameter learning phase occurs concurrently with the structure learning phase. For each piece of incoming data, all free SEIT2FNN parameters are tuned, whether the rules are newly generated or originally existent. To clarify, consider the single-output case, where the objective is to minimize the error function

$$E = \frac{1}{2}[y(t) - y_d(t)]^2 \tag{19}$$

Here, $y(t)$ and $y_d(t)$ denote real and desired outputs, respectively. The Karnik-Mendel iterative procedure for computing y_l and y_r in Eq. (12) and (13) has the premise that w_l^i and w_r^i are re-arranged in ascending order. During the parameter learning process, the w_l^i and w_r^i values change, and their orders and corresponding rule orders in computing Eq. (12) and (13) should change accordingly. To update parameters, it is necessary to know exactly where specific antecedent and consequent parameters are located, and this is very difficult to ascertain when y_l and y_r are not in a rule-ordered format. This problem was addressed in (Mendel, 2004). The SEIT2FNN approach considers this problem in its derived rule-ordered Kalman filtering algorithm for consequent parameter learning. Let $\underline{\boldsymbol{f}} = (\underline{f}^1, \underline{f}^2, ..., \underline{f}^M)^T$ and $\overline{\boldsymbol{f}} = (\overline{f}^1, \overline{f}^2, ..., \overline{f}^M)^T$ where the firing strengths are expressed according to the original rule order. According to (Mendel, 2004), Eq. (12) can be re-expressed in the following rule-ordered form

$$y_l = \frac{\overline{\boldsymbol{f}}^T Q_l^T E_1^T E_1 Q_l \boldsymbol{w_l} + \underline{\boldsymbol{f}}^T Q_l^T E_2^T E_2 Q_l \boldsymbol{w_l}}{\sum_{i=1}^{L} (Q_l \overline{\boldsymbol{f}})_i + \sum_{i=L+1}^{M} (Q_l \underline{\boldsymbol{f}})_i} \tag{20}$$

where

$$E_1 = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_L, \boldsymbol{0}, ..., \boldsymbol{0}) \in \Re^{L \times M}$$

$$\text{and } E_2 = (\boldsymbol{0}, ..., \boldsymbol{0}, \boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, ..., \boldsymbol{\varepsilon}_{M-L}) \in \Re^{(M-L) \times M}, \tag{21}$$

and where $\boldsymbol{e}_i \in \Re^{L \times 1}$ and $\boldsymbol{\varepsilon}_i \in \Re^{M-L}$ are elementary vectors. The output y_r can be re-expressed in same way. In (Mendel, 2004), consequent parameters are tuned by a gradient descent algorithm. In a SEIT2FNN, a rule-ordered Kalman filtering algorithm tunes consequent parameters. Output y_l and y_r can be expressed as

$$y_l = \boldsymbol{\phi}_l^T \boldsymbol{w}_l, \ \boldsymbol{\phi}_l \in \Re^{1 \times M} \tag{22}$$

and

$$y_r = \boldsymbol{\phi}_r^T \boldsymbol{w}_r, \ \boldsymbol{\phi}_r \in \Re^{1 \times M} \tag{23}$$

respectively. Thus, the output y in Eq. (14) can be re-expressed as

$$y = \frac{1}{2}(y_l + y_r) = \frac{1}{2}(\boldsymbol{\phi}_l^T \boldsymbol{w}_l + \boldsymbol{\phi}_r^T \boldsymbol{w}_r) = \overline{\boldsymbol{\phi}}_{TSK}' \boldsymbol{w}_{TSK} \tag{24}$$

The consequent parameter vector $\boldsymbol{w}_{TSK}$ is updated by executing the following rule-ordered Kalamn filtering algorithm

$$\begin{aligned} \boldsymbol{w}_{TSK}(t+1) &= \boldsymbol{w}_{TSK}(t) + S(t+1)\overline{\boldsymbol{\phi}}_{TSK}(t+1)(y^d(t+1) - \overline{\boldsymbol{\phi}}_{TSK}^T(t+1)\boldsymbol{w}_{TSK}(t)) \\ S(t+1) &= \frac{1}{\lambda}[S(t) - \frac{S(t)\overline{\boldsymbol{\phi}}_{TSK}(\tau+1)\overline{\boldsymbol{\phi}}_{TSK}^T(t+1)S(t)}{\lambda + \overline{\boldsymbol{\phi}}_{TSK}^T(t+1)S(t)\overline{\boldsymbol{\phi}}_{TSK}}] \end{aligned} \tag{25}$$

where $0 < \lambda \leq 1$ is a forgetting factor (λ =0.9995 in this paper). Here, $S_0 = q \cdot I \in \Re^{2M(n+1) \times 2M(n+1)}$, and q is a large positive constant. SEIT2FNN antecedent parameters are tuned by a gradient descent algorithm. That is,

$$m_{j1}^i(t+1) = m_{j1}^i(t) - \eta \frac{\partial E}{\partial m_{j1}^i} \tag{26}$$

$$m_{j2}^i(t+1) = m_{j2}^i(t) - \eta \frac{\partial E}{\partial m_{j2}^i} \tag{27}$$

$$\sigma_j^i(t+1) = \sigma_j^i(t) - \eta \frac{\partial E}{\partial \sigma_j^i} \tag{28}$$

where η is a learning coefficient. Details of the learning algorithm can be found in (Mendel, 2004), where the authors had explicitly derived gradient calculations considering the rules re-ordering problem.

4. SIMULATIONS

Example-System Identification.

This example uses the SEIT2FNN to identify a nonlinear system. The plant to be identified is guided by the difference equation (Lin and Lin, 1997, Juang and Lin, 1998)

$$y_d(t+1) = \frac{y_d(t)}{1 + y_d^2(t)} + u^3(t) \tag{29}$$

The training patterns are generated with $u(t) = \sin(2\pi t/100)$, t =1, … , 200. The SEIT2FNN inputs are $y_d(t)$ and $u(t)$, and the desired output is $y_d(t+1)$. Performance is evaluated using the root-mean-squared error (RMSE)

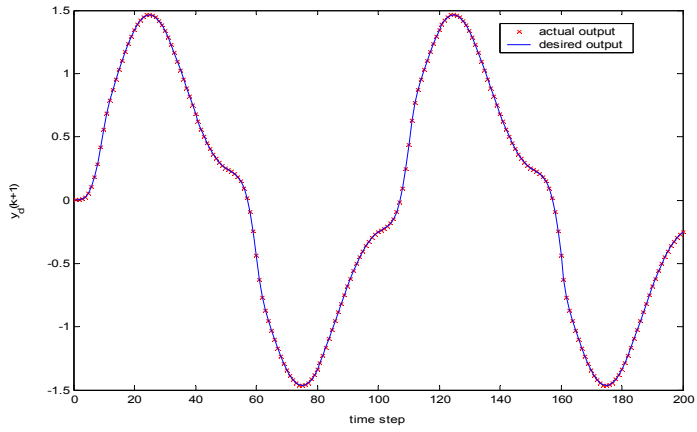

Fig. 3. Identification results of the SEIT2FNN using ten rules.

$$RMSE = \sqrt{\frac{1}{200}\sum_{k=1}^{200}[y(t+1) - y^d(t+1)]^2} \qquad (30)$$

where $y(t)$ is the SEIT2FNN output. The learning coefficient η is set at 0.05. The structure learning threshold ϕ_{th} determines the number of generated fuzzy rules. Three rules are generated when ϕ_{th} is set at 0.01. Since the number of fuzzy sets and the number of rules is small, the fuzzy set reduction operation in structure learning is not used, i.e., ρ is set to be zero. Training is performed for 500 iterations. Table 1 shows the performance of the SEIT2FNN.

This example also tests SEIT2FNN performance with a greater number of rules, caused by assigning a larger value of ϕ_{th}. Table II shows the performance of the SEIT2FNN with a greater number of rules: 10 rules are generated when ϕ_{th} = 0.15. The results in Tables I and II indicate that with the same consequent type, a higher number of ϕ_{th} generates a higher number of rules and generally improves SEIT2FNN performance.

The interval type 2 fuzzy logic system presented in (Mendel, 2004) is compared to the SEIT2FNN, and denoted as T2FLS hereafter. The T2FLS has no structure learning, and the number of rules in T2FLS is assigned *a priori*. T2FLS parameters are tuned by the gradient descent algorithm in (Mendel, 2004). Tables I and II show T2FLS performance with TSK-type consequents, respectively. The results show that with the same number of rules and consequent type, the RMSE of SEIT2FNN is much smaller than the RMSE of T2FLS.

The performance of Type 1 fuzzy neural networks (FNNs) that were applied to the same identification problem is also compared. These type 1 FNNs include the fuzzy adaptive learning control network (FALCON) (Lin and Lin, 1997) and self-constructing neural fuzzy inference network (SONFIN) (Juang and Lin, 1998), both of which also have structure learning abilities. Table I lists the performance of the

TABLE 1. PERFORMANCE OF SEIT2FNN AND OTHER TYPE 1 AND TYPE 2 FUZZY STSTEMS.

Methods	Type 1		Type 2	
	FALCON	SONFIN	T2FLS(TSK)	SEIT2FNN
Rule number	6	7	3	3
parameter number	54	35	36	36
Iterations	60000	500	500	500
RMSE	0.02	0.008	0.039	0.0069

FALCON and the SONFIN, indicating that the SEIT2FNN has a smaller RMSE and number of rules than these two type 1 FNNs. The results also show that SONFIN performance, which has structure learning ability, is better than the T2FLS when a nearly identical number of parameters are used in both networks.

5. CONCLUSIONS

This paper proposes a SEIT2FNN. In contrast to existing type-2 fuzzy systems, there is no need to determine SEIT2FNN structure in advance because the proposed structure learning ability enables the SEIT2FNN to evolve its structure on-line. The proposed rule-ordered Kalman filter algorithm helps tune the consequent parameters on-line and improves learning accuracy. Simulation results show that, compared with type-1 fuzzy systems that also have on-line structure learning ability, the SEIT2FNN achieves higher learning accuracy with a smaller number of rules for both clean and noisy data. Future studies will examine practical applications of the SEIT2FNN to control and signal processing problems with noise or uncertainty.

REFERENCES

Hagras, H (2004). "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Trans. Fuzzy Systems*, vol. 12, no. 524-539.

Hagras, H (2006). **"**Comments on dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," *IEEE Trans. Syst., Man and Cyber. - Part B: Cybernetics,* vol. 36, no. 5, pp. 1206-1209.

Jang, J. S (1993). "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665-685.

Juang, C. F and C. T. Lin (1998). "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Systems*, vol. 6. no. 1, pp. 12-32.

Karnik, N. N, J. M. Mendel, and Q. Liang (1999). "Type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 6, pp. 643-658.

Kukolj, D and E. Levi (2004). "Identification of complex systems based on neural and Takagi-Sugeno fuzzy model," *IEEE Trans. Sys., Man, Cybern., Part B: Cybernetics*, vol. 34, no. 1, pp. 272-282.

Lee, C. H, Y. C. Lin, and W. Y. Lai (2003). "Systems identification using type-2 fuzzy neural network (Type-2 FNN) systems," *Proc. IEEE Int. Symp.*

Computational Intelligence in Robotics and Automation, vol. 3, pp. 1264-1269.

Liang, Q and J. M. Mendel (2000). "Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. On Fuzzy Systems*, vol. 8, no. 5, pp. 535-550.

Lin, C. J. Lin and C. T. Lin (1997). "An ART-based fuzzy adaptive learning control network," *IEEE Trans. Fuzzy Systems*, vol. 5, no. 4, pp. 477-496.

Melin, P and O. Castillo (2004). "Intelligent control of non-linear dynamic plants using type-2 fuzzy logic and neural networks," *Proc. IEEE Int. Conf. Fuzzy Systems*, Budapest, Hungary.

Mendel, J. M. (2001). Uncertain Rule-Based Fuzzy Logic System: Introduction and New Directions, Prentice Hall, Upper Saddle River.

Mendel, J. M. and R. I. John (2002). "Type-2 fuzzy sets made simple," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 2, pp. 117-127.

Mendel J. M (2004). "Computing derivatives in interval type-2 fuzzy logic system," *IEEE Trans. On Fuzzy Systems*, vol. 12. no. 1, pp. 84-98.

Wang, C. H, C. S. Cheng, and T. T. Lee (2004). "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," *IEEE Trans. on Syst., , Man, and Cyber. - Part B: Cybernetics*, vol. 34, no. 3, pp. 1462-1477.

Appendix. Acknowledgement

This was supported by the National Science Council, Taiwan, Republic of China, under Grant number NSC 96-2628-E-005-087-MY3.