A Novel Approach to Generating an Interval Type-2 Fuzzy Neural Network Based on a Well-Behaving Type-1 Fuzzy TSK System

Junlong Gao, Ruyi Yuan, and Jianqiang Yi, *Senior Member, IEEE*
Institute of Automation,
Chinese Academy of Sciences
Beijing 10090, China
{junlong.gao, ruyi.yuan, jianqiang.yi}@ia.ac.cn

Hao Ying, *Fellow, IEEE*
Dept. of Electrical & Computer Engineering
Wayne State University
Detroit 48202, USA
hao.ying@wayne.edu

Chengdong Li, *Member, IEEE*
School of Information & Electrical Engineering
Shandong Jianzhu University
Jinan 250101, China
lichengdong@sdjzu.edu.cn

***Abstract*—This paper presents a novel approach to automatically creating an interval type-2 fuzzy neural network (IT2-FNN) from a type-1 fuzzy TSK system (T1-TSK). The IT2-FNN is constructed in such a way that it takes advantage of the well-behaving T1-TSK. Our approach makes designing the IT2-FNN more efficient and the resulting system is expected to perform better than the T1-TSK due to the footprint of uncertainty of the IT2 fuzzy sets, especially when the system is subject to heavy external or internal uncertainties. There are two automated procedures in the IT2-FNN formation: (1) antecedent structure construction, and (2) learning of the parameters in both the antecedent and consequent. The structure construction is based on antecedent structure of the T1-TSK and consists of three steps – IT2 fuzzy set creation, similarity categorization, and mergence. The IT2 fuzzy sets are directly initialized from the fuzzy sets of the T1-TSK. Then, the IT2 fuzzy sets are classified into different groups based on their similarities. Finally, the IT2 fuzzy sets in each group are merged to create a representative IT2 fuzzy set for each group. The parameter learning procedure uses a hybrid learning algorithm to attain the optimal values for all the parameters. The learning algorithm adopts a new adaptive steepest descent algorithm and a linear least-squares method to adjust the antecedent parameters and consequent parameters, respectively. One benchmark modelling problem is utilized to compare our approach with the T1-TSK systems in the literature under various scenarios. The comparison results show our IT2-FNN performs better than the T1-TSK systems, especially when there are strong uncertainties. In summary, the IT2-FNN can not only achieve better performance but its structure is simpler than that of the similar type-2 fuzzy neural networks in the literature.**

Keywords—fuzzy logic system; type transition; fuzzy set mergence; interval type-2 fuzzy neural network; adaptive steepest decent algorithm

This work is supported by NNSFC under grant No. 61273149, No. 61403381, No. 61421004, No. 61473176, by the NDBSR under grant No. B1320133020 and the Natural Science Foundation of Shandong Province for Outstanding Young Talents in Provincial Universities under grant ZR2015JL021 .

I. INTRODUCTION

In recent decades numerous achievements which use type-1 fuzzy logic, in data-driven modeling and prediction have been made as an important application branch with the property of universal approximation in fuzzy logic systems, e.g. [1-3] and so on. Since the type-2 fuzzy logic system (T2-FLS) was brought into practical applications[4, 5], it draws researchers' attention and becomes the hotspot of fuzzy society very rapidly for the advantages compared to type-1 fuzzy logic systems (T1-FLSs), i.e. footprint of uncertainty (FOU) which can bring additional design degree of freedom to make T2-FLS with more outstanding potential to overcome disturbances and to reduce the rule numbers. To date, simpler T2-FLS, i.e. interval type-2 fuzzy logic system (IT2-FLS), has already been applied into signal processing, control, pattern recognition, stock prediction and so forth [6]. However, between two existing IT2-FLSs which are IT2-Mamdani system (whose consequent part is Mamdani interval type-2 fuzzy sets (IT2-FSs)) and IT2-TSK system (whose consequent part is polynomial functions combined with input variables) respectively, the IT2-Mamdani system rules are hard to design in some complex systems which do not have figurative physical meanings whereas the IT2-TSK system rules have specific mathematical expressions which can be understood as special cases of piecewise approximations. Thus the IT2-TSK system provides an easier way to reduce rule-design difficulties at a certain degree. And either in study depth or breadth, the IT2-TSK system is in the trend to substitute the IT2-Mamdani system in recent years.

At present, most approaches in automate learning (i.e. self-organizing) IT2-TSK systems for system modelling or time series' prediction problems adopt structure of interval type-2 fuzzy neural networks (IT2-FNNs) or interval type-2 neural fuzzy systems (IT2-NFSs). In addition, there are two ways to construct an IT2-FLS, the first and the most common used way is to construct IT2-FLS structure directly through experts' opinions or through data-driven online/offline IT2-FNNs (or T2-NFSs) [7]. The second way, which this study adopts, is to transform the structure into IT2-FLSs from corresponding type-1 ones. The former one abandons numerous of well-behaving T1-FLSs which were developed for specific systems, whereas the latter one makes use of existing type-1 rules. Furthermore, the system needs to use optimization algorithms, e.g. genetic algorithm (GA), gradient descent algorithm, steepest descent algorithm and so forth, to adjust parameters to achieve the optimal behavior. To date, there is only a few related studies directly with the second way as aforementioned. One uses type-1 fuzzy TSK system (T1-TSK) to transform into a IT2-TSK system and parameters in antecedent and consequent are learned through GA [8]. However, it does not take the antecedent fuzzy sets' similarity categorization and mergence into consideration in structure construction procedure since the author considers the T1-TSK system generated from its IM-ENFS algorithm which is improved from ANFIS is optimal enough. In addition, the parameters to be optimized are scale factors bounding to centers and widths of its IT2-FSs membership functions (MFs) respectively, which results in the optimization algorithm being unable to optimize its IT2-FSs parameters independently. The other one presents constructing a type-2 neural fuzzy system through type-1 fuzzy rules (T2FNS-N1) which creates IT2-TSK system from T1-TSK and addresses the problems which the first one does not [9]. However, the second one uses T1-FSs similarity categorization and mergence before the IT2-FSs creation which cannot reflect the actual IT2-FSs condition and has the possibility to miss IT2-FSs which should be merged, and the gradient descent algorithm adopted to optimize parameters has fixed learning rates.

In order to address the aforementioned shortages of the existing structure construction methods and make use of those well behaving T1-TSK rules to help researchers build effective rules in IT2-TSK systems faster, this research proposes a novel approach to creating an interval type-2 fuzzy neural network (IT2-FNN) from a type-1 fuzzy TSK system . It has two automated procedures in the IT2-FNN formation: antecedent structure construction and learning of the parameters in both the antecedent and consequent. The first procedure consists of three steps without deleting the number of rules. The IT2 fuzzy set creation step directly initializes IT2-FSs from T1-FSs in the T1-TSK. Then the similarity categorization step classifies the IT2-FSs into different groups based on their similarities. Finally, the mergence step merges the IT2-FSs in each group to create a representative IT2 fuzzy set for each group. Subsequently, the parameter learning procedure adopts a new adaptive steepest descent algorithm with adaptive descent directions and a linear least-squares method to adjust the antecedent parameters and consequent parameters, respectively.

The rest of this paper is organized as follows: Section 2 gives a brief introduction of the T1-TSK rule structure and the candidate method in generating the T1-TSK. This section continues by further describing the IT2-FNN structure which is adopted in this study; Section 3 provides the two automated procedures in the IT2-FNN formation with some algorithms which are key components in both procedures; Section 4 gives simulations and comparisons with several T1-TSK systems and IT2-TSK systems; Section 5 draws conclusions.

II. Structure of Fuzzy TSK Rules and Interval Type-2 Fuzzy Nueral Networks

A. *Type-1 Fuzzy TSK Rule Stucture and Generation*

1) *Rules of Type-1 Fuzzy TSK System*

In this paper, we adopt type-1 fuzzy TSK system (T1-TSK) which has Gaussian type-1 fuzzy sets in antecedent part and crisp linear polynomial in consequent part. In addition, the T1-TSK system has been proved as universal approximator [10].

The i^{th} rule of the T1-TSK system is given in the MIMO form as following:

Rule i: If x_1 is A_1^i and … and x_m is A_m^i, then (1)

$$y_i^k = \tilde{w}_{0,i}^k + \sum_{in=1}^{m} \tilde{w}_{in,i}^k x_{in} \quad i = 1, \dots, n;\ in = 1, \dots, m;\ k = 1, \dots, r$$

where A_{in}^i denotes type-1 fuzzy set (T1-FS) which corresponds to in^{th} input variable x_{in} . Moreover, in the function of k^{th} output y_i^k , $\tilde{w}_{0,i}^k$ stands for crisp value together with $\tilde{w}_{in,i}^k$ which denotes crisp consequent parameter of the in^{th} input variable. n is the number of rules, m is the number of input variables, r is the number of outputs.

2) *Generating Rules of Type-1 Fuzzy TSK System*

In this research, we use generalized dynamic fuzzy neural networks (GD-FNN) [11] as the candidate to generate T1-TSK rule structures for specific plants.

B. *Structure of Interval Type-2 Fuzzy Neural Networks*

1) *Rules of Interval Type-2 Fuzzy TSK System*

The structure of interval type-2 fuzzy neural networks (IT2-FNN) is mainly based on interval type-2 fuzzy TSK system (IT2-TSK), which can be distinguished into 2 types: A2-C0 and A2-C1 [12].

In this study, we adopt A2-C0 IT2-TSK system to construct the IT2-FNN, since the A2-C0 IT2-TSK system has already been proved as universal approximator [13]. Without losing generality, the i^{th} rule of A2-C0 IT2-TSK system is given in MIMO form as well:

Rule i: If x_1 is $\tilde{A}_1^i$ and … and x_m is $\tilde{A}_m^i$, then (2)

$$y_i^k = \tilde{w}_{0,i}^k + \sum_{in=1}^{m} \tilde{w}_{in,i}^k x_{in} \quad i = 1, \dots, n;\ in = 1, \dots, m;\ k = 1, \dots, r$$

where $\tilde{A}_{in}^i$ denotes Gaussian interval type-2 fuzzy set (IT2-FS) which corresponds to in^{th} input variable x_{in}. Other definitions of consequent part are the same as the previous ones stated in the type-1 rules.

2) *Structure of the Interval Type-2 Fuzzy Neural Networks*

Based on the aforementioned IT2-TSK system, the IT2-FNNs consists of a seven layer structure which is shown in Fig.1. There is a new rule-ordered fuzzification topological relationship layer (e.g. layer 3) inside the structure. This layer can help the IT2-FNN with clearer relationship expressions between the IT2-FSs in each input variables' universe discourse and the rules. Moreover, the structure can help researchers design steepest descent related optimization algorithms with more convenient mathematical expressions.

a) First Layer (Input Layer):

In this layer, each neuron accepts corresponding input variables and transfers the values into the second layer.

b) Second Layer (Original-ordered Fuzzification Layer)

In this layer, the neurons, which stand for Gaussian IT2-FSs with uncertain centers in each input variable discouse, use inputs x_{in} to generate interval fuzzified values:

$$\left[\underline{\mu}_{in}^{t\_in}(x_{in}),\overline{\mu}_{in}^{t\_in}(x_{in})\right]_{t\_in=1,2,\ldots,T_{in}} \tag{3}$$

where $\overline{\mu}_{\tilde{A}_{in}^{t\_in}}(x_{in})$ and $\underline{\mu}_{\tilde{A}_{in}^{t\_in}}(x_{in})$ are upper membership function (UMF) and lower membership function (LMF) respectively, $T_{in} \le n$ is the number of the antecedent IT2-FSs and t_in stands for the $(t\_in)^{th}$ IT2-FS of input x_{in}. The detailed expression of (3) is shown in (4)-(5):

$$\overline{\mu}_{in}^{t\_in}(x_{in})=\begin{cases} Ga\left(c_{Lin}^{t\_in},\sigma_{in}^{t\_in};x_{in}\right) & x_{in} < c_{Lin}^{t\_in} \\ 1 & c_{Lin}^{t\_in} \le x_{in} \le c_{Rin}^{t\_in} \\ Ga\left(c_{Rin}^{t\_in},\sigma_{in}^{t\_in};x_{in}\right) & x_{in} > c_{Rin}^{t\_in} \end{cases} \tag{4}$$

$$\underline{\mu}_{in}^{t\_in}(x_{in})=\begin{cases} Ga\left(c_{Rin}^{t\_in},\sigma_{in}^{t\_in};x_{in}\right) & x_{in} \le \left(c_{Lin}^{t\_in}+c_{Rin}^{t\_in}\right)/2 \\ Ga\left(c_{Lin}^{t\_in},\sigma_{in}^{t\_in};x_{in}\right) & x_{in} > \left(c_{Lin}^{t\_in}+c_{Rin}^{t\_in}\right)/2 \end{cases} \tag{5}$$

where the membership function $Ga(c,\sigma;x)$ stands for Gaussian function with center c, width σ and input x as $\exp[-(x-c)^2/(2\sigma^2)]$. L and R stand for left and right, respectively.

c) Third Layer (Rule-ordered Fuzzification Topological Relationship Layer):

However, all those values and arrangements in the second layer cannot be delivered into the fourth layer directly since the fourth layer needs rule-ordered fuzzified values.

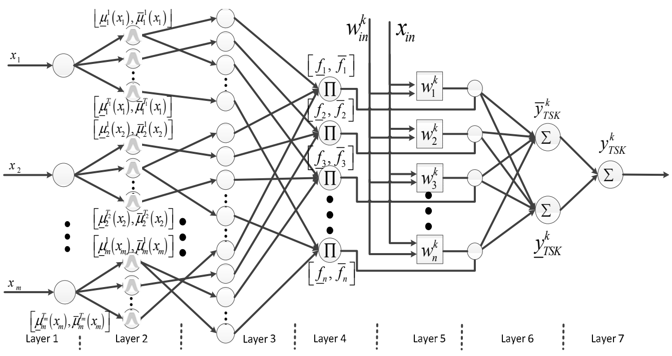

Fig. 1 Structure of the 7-layer IT2-TSK-FNN

Therefore, this layer will process the topological relationship between the rule-ordered fuzzification information and variable-discourse-ordered fuzzification information. Thus the UMF and LMF in the previous layer are subsequently reorganized into the rule-ordered fuzzified interval UMF $\overline{\mu}_{in}^{i}(x_{in})$ and LMF $\underline{\mu}_{in}^{i}(x_{in})$, which are short for $\overline{\mu}_{\tilde{A}_{in}^{i}}(x_{in})$ and $\underline{\mu}_{\tilde{A}_{in}^{i}}(x_{in})$ separately.

The topological relationships can be expressed as (6)-(7):

$$\frac{\partial\overline{\mu}_{in}^{i}}{\partial\overline{\mu}_{in}^{t\_in}}=\begin{cases} 1 & original\ t\_in^{th}\ FS\ is\ fired\ in\ i^{th}\ rule \\ 0 & others \end{cases} \tag{6}$$

$$\frac{\partial\underline{\mu}_{in}^{i}}{\partial\underline{\mu}_{in}^{t\_in}}=\begin{cases} 1 & original\ t\_in^{th}\ FS\ is\ fired\ in\ i^{th}\ rule \\ 0 & others \end{cases} \tag{7}$$

d) Fourth Layer (Firing Layer or Inference Layer):

In this layer, inputs $\overline{\mu}_{in}^{i}(x_{in})$ and $\underline{\mu}_{in}^{i}(x_{in})$ are collected to calculate the i^{th} rule's upper firing value $\overline{f}_i$ and lower firing value by prod t-norms in (8). Then the $\overline{f}_i$ and $\underline{f}_i$ values are transferred into the fifth layer:

$$\overline{f}_i=\prod_{in=1}^{m}\overline{\mu}_{in}^{i}(x_{in}),\ \ \underline{f}_i=\prod_{in=1}^{m}\underline{\mu}_{in}^{i}(x_{in}) \tag{8}$$

e) Fifthth Layer (Consequent Layer)

This layer uses input variables and consequent parameters to calculate components y_i^k which are substituted by w_{in}^k into the sixth layer (9):

$$w_i^k=y_i^k=\tilde{w}_{0,i}^k+\sum\nolimits_{in=1}^{m}\tilde{w}_{in,i}^k x_{in} \tag{9}$$

f) Sixth Layer (Type Reduction Layer)

Regular type reduction algorithms (or inference format), e.g. KMA, are not easy to perform the reduction formulas with UMF and LMF separately. Hence, in this layer, we use Begian-Melek-Mendel type-reduction method [12] to calculate the k^{th} type reduced output $\underline{y}_{TSK}^k$ and $\overline{y}_{TSK}^k$ by using the outputs from the fourth layer and fifth layer as (10):

$$\underline{y}_{TSK}^k=\frac{0.5\sum_{i=1}^{m}\underline{f}^i w_i^k}{\sum_{i=1}^{m}\underline{f}^i},\ \overline{y}_{TSK}^k=\frac{0.5\sum_{i=1}^{m}\overline{f}^i w_i^k}{\sum_{i=1}^{m}\overline{f}^i} \tag{10}$$

g) Seventh Layer (Output Layer):

The k^{th} output of the IT2-FNN is the sum of the sixth layer's outputs as (11):

$$y_{TSK}^k=\underline{y}_{TSK}^k+\overline{y}_{TSK}^k \tag{11}$$

III. The Automated Procedures in Generating The Interval Type-2 Fuzzy Neural Networks

In this section, we will give detailed automated procedures for antecedent structure construction and parameter learning. The antecedent structure construction procedure consists of interval type-2 fuzzy sets' (IT2-FSs) creation, similarity categorization and mergence operation. The input variables are normalized into [-1, 1], thus can help transform type-1 fuzzy sets (T1-FSs) into the initialized IT2-FSs with uncertain centers more reasonable. Subsequently, the similarity categorization step includes similarity measurements of adjacent IT2-FSs and to categorize them into different groups. Then they are merged through our improved mergence algorithm (which takes more comprehensive geometric information of the whole groups into consideration and has an adjustable merge-expansion index to control the merged sets' width than previous methods from literature [9]) to a representative IT2-FS, respectively. The parameter learning procedure adopts a new adaptive steepest descent algorithm with adaptive descent directions and a linear least-squares (LLS) method to separately adjust the antecedent parameters and consequent parameters.

Comments: In order to keep the effectiveness of the proposed IT2-FNN, this work only involves mergence of the initialized IT2-FSs without takes account of deleting rule numbers based on consequent parts in the original type-1 fuzzy rules.

A. Antecedent Structure Construction from Type-1 Fuzzy TSK Rules

1) Interval Type-2 Fuzzy Sets Creation

Assuming antecedent part of the type-1 fuzzy TSK system (T1-TSK) has the form of original T1-FSs in each input discourse. The T1-FSs consist of the information vectors of widths $\Theta_{in}^{t\_in}$ and centers $c_{in}^{t\_in}$ respectively, where $in = 1,...,m$, $t\_in = 1,2,...,T_{in}$. The input variables are normalized, which means each input will be restricted into a certain range of [-1, 1]. Thus we assign the T1-FSs' information vectors into the initial IT2-FSs' information vectors of widths in (12) and expand centers into left and right center vectors shown in (13):

$$\sigma_{in}^{t\_in} = \Theta_{in}^{t\_in} \tag{12}$$

$$\begin{bmatrix} c_{Lin}^{t\_in} & c_{Rin}^{t\_in} \end{bmatrix} = \begin{bmatrix} c_{in}^{t\_in} - 0.1, c_{in}^{t\_in} + 0.1 \end{bmatrix} \tag{13}$$

2) Similarity Categorization of Interval Type-2 Fuzzy Sets

Noticing that there may be existing high-overlapped possibilities in the original T1-FSs or the initialized IT2-FSs, and these redundancy FSs will increase the computational complexity and barely do efficient contributions to the IT2-FNN. Therefore, those adjacent FSs with highly similarities need to be merged into a representative FS in each input variables in order to improve the system's performance. One related research applied T1-FSs similarity measurements to detect and merge high overlapping T1-FSs and then transform those merged T1-FSs into initialized IT2-FSs [9]. However, this method firstly may not reflect the real similarity degree between initialized IT2-FSs since it measures T1-FSs' instead. Secondly, its mergence operation just combines geometrical information of T1-FSs into a representative IT2-FS without taking distributed conditions of the merged IT2-FSs into consideration.

We adopt the IT2-FSs similarity measurement method [14], which is an extension of Jaccard similarity measurements of T1-FSs. Moreover, this method has been proved to be more reliable than other existing methods. The similarity degree S_J of two adjacent initialized IT2-FSs $\tilde{A}_a, \tilde{A}_{a+1}$ is compute as follows:

$$\mathrm{S}_J(\tilde{A}_a, \tilde{A}_{a+1}) = \frac{\sum_{tp=1}^{2} \int_X \min\left(\mu_{\tilde{A}_a}^{(tp)}(x), \mu_{\tilde{A}_{a+1}}^{(tp)}(x)\right) dx}{\sum_{tp=1}^{2} \int_X \max\left(\mu_{\tilde{A}_a}^{(tp)}(x), \mu_{\tilde{A}_{a+1}}^{(tp)}(x)\right) dx} \tag{14}$$

where $\mu^{(1)}$ and $\mu^{(2)}$ represent $\overline{\mu}$ and $\underline{\mu}$ respectively. If the similarity degree $\mathrm{S}_J(\tilde{A}_a, \tilde{A}_{a+1}) \geq S_{sim}$, where S_{sim} is the similarity threshold, then the IT2-FSs $\tilde{A}_a$ and $\tilde{A}_{a+1}$ are highly overlapped and should be merged into one single IT2-FS. More broadly, if sequential adjacent IT2-FSs' similarity degrees are all exceeding the similarity threshold, e.g. $\tilde{A}_a$ to $\tilde{A}_{a+h}$, then these IT2-FSs are categorized into one group, e.g. group j .

3) Mergence of Interval Type-2 Fuzzy Sets

A new mergence algorithm is given, which takes more comprehensive geometric information of the whole group j into consideration rather than simply calculating width and centers average values of the group j 's very left and right IT2-FS in [15].

The mergence algorithm is given in (15)-(17), which is used for calculating the merged IT2-FS $\tilde{A}_j$ with left / right centers c_{jL} / c_{jR} and width σ_j to represent the group j .

$$\begin{cases} c_{jL} = \frac{4}{5}\min\left(c_{aL},...,c_{(a+h)L}\right) + \frac{1}{5}\max\left(c_{aR},...,c_{(a+h)R}\right) \\ c_{jR} = \frac{1}{5}\min\left(c_{aL},...,c_{(a+h)L}\right) + \frac{4}{5}\max\left(c_{aR},...,c_{(a+h)R}\right) \\ \sigma_j = \max\left(\sigma_{jL}, \sigma_{jR}\right) \end{cases} \tag{15}$$

$$\sigma_{jL} = \frac{1}{\sqrt{-\ln\varepsilon}}\left|\frac{1}{5}\left(c_{aL} - c_{(a+h)R}\right) - \sigma_a\sqrt{-\ln\varepsilon}\right| \tag{16}$$

$$\sigma_{jR} = \frac{1}{\sqrt{-\ln\varepsilon}}\left|\frac{1}{5}\left(c_{(a+h)R} - c_{aL}\right) + \sigma_{(a+h)}\sqrt{-\ln\varepsilon}\right| \tag{17}$$

where $c_{aL/R}, c_{(a+h)L/R}$ are the left and right centers of IT2-FSs $\tilde{A}_a$ and $\tilde{A}_{a+h}$, σ_a and $\sigma_{(a+h)}$ are width values of $\tilde{A}_a$ and $\tilde{A}_{a+h}$ respectively. σ_{jL} and σ_{jR} are the merged leftmost and rightmost width values where ε is the merge-expansion index (which is with range of (0, 1)) to determine expansion strength compared to the group j .

B. Parameter Learning of the IT2-TSK Rules

After the structure construction procedure as previous stated, the antecedent part of the IT2 -FNN is set up. In addition, since the antecedent part is changed in both type and total number of fuzzy sets, the consequence part of the IT2-FNN still need to be confirmed and the parameters both in the consequent and antecedent parts will be optimized by a hybrid data-driven learning algorithm.

Supposing there are S input-output training pairs $\left(X^T(s), Y^T(s)\right)$ where s=1,…,S, $X(s)=\left(x_1(s) \quad \dots \quad x_m(s)\right)^T$ and $Y(s)=\left(y_d^1(s) \quad \dots \quad y_d^r(s)\right)^T$. System evaluation function is chosen as a squared error function as:

$$E=\frac{1}{2}\sum_{k=1}^{r}\sum_{s=1}^{S}\left(y_{TSK}^k(s)-y_d^k(s)\right)^2 \tag{18}$$

1) Consequent Parameter Learning

In order to simplify (10)-(11) in the IT2-FNN into matrix-compute form, we rewrite (11) as:

$$y_{TSK}^k=\mathbf{\Phi}\cdot \boldsymbol{w}^k=\left[\overline{\phi}/2 \quad \underline{\phi}/2\right]\cdot \boldsymbol{w}^k \tag{19}$$

where $\boldsymbol{w}^k=\left[\boldsymbol{\varpi}^k \quad \boldsymbol{\varpi}^k\right]^T$,

$$\boldsymbol{\varpi}^k=\left[\tilde{w}_{0,1}^k \quad \tilde{w}_{0,2}^k \quad \dots \quad \tilde{w}_{0,n}^k \quad \dots \quad \tilde{w}_{m,1}^k \quad \dots \quad \tilde{w}_{m,n}^k\right]_{1\times[(m+1)\times n]}$$

$$\overline{\phi}=\left[\overbrace{\overline{f}_1 \quad \dots \quad \overline{f}_n \quad \overline{f}_1x_1 \quad \dots \quad \overline{f}_nx_1 \quad \dots \quad \overline{f}_1x_m \quad \dots \quad \overline{f}_nx_m}^{(m+1)\times n}\right]/\sum_{i=1}^{n}\overline{f}_i$$

$$\underline{\phi}=\left[\overbrace{\underline{f}_1 \quad \dots \quad \underline{f}_n \quad \underline{f}_1x_1 \quad \dots \quad \underline{f}_nx_1 \quad \dots \quad \underline{f}_1x_m \quad \dots \quad \underline{f}_nx_m}^{(m+1)\times n}\right]/\sum_{i=1}^{n}\underline{f}_i$$

Substituting (19) into (18), we can get:

$$E=\frac{1}{2}\sum_{k=1}^{r}\left(\mathbf{\Phi}\cdot\boldsymbol{w}-Y_d^k\right)^T\left(\mathbf{\Phi}\cdot\boldsymbol{w}-Y_d^k\right)=\frac{1}{2}\sum_{k=1}^{r}\left\|\mathbf{\Phi}\cdot\boldsymbol{w}-Y_d^k\right\|_2^2 \tag{20}$$

Then (20) can be rewritten to meet the objective of the hybrid learning algorithm in tuning the consequent part which applies LLS algorithm to solve the following optimization problem:

$$\min_{w}\left\|\mathbf{\Phi}\cdot\boldsymbol{w}-Y_d^k\right\|_2^2 \tag{21}$$

In detail, assuming that the optimal consequent parameter w^* can be formulated as a linear minimizing problem of (21), then we can get w^* by pseudoinverse technique as：

$$\boldsymbol{w}^{k*}=\mathbf{\Phi}^+Y_d^k \tag{22}$$

where $\mathbf{\Phi}^+$ is a Moore-Penrose generalized inverse matrix of $\mathbf{\Phi}$.

2) Antecendent Parameter Learning

After the consequent parameter tuning phase, we apply a new adaptive steepest descent algorithm which has adaptive descent directions to tune the widths, left centers and right centers of the IT2-FSs in layer 2 of the IT2-FNN.

a) Steepest Descent Algorithm for the IT2-FNN

The antecedent parameters in the IT2-FNN tuned by the steepest descent algorithm are the width $\sigma_{in}^{t\_in}$, left center $c_{Lin}^{t\_in}$ and right center $c_{Rin}^{t\_in}$ of the IT2-FSs in layer 2 respectively. Thus the steepest descent algorithm is shown in (23):

$$O_{in}^{t\_in}(s+1)=O_{in}^{t\_in}(s)-\eta_O d_{O_{in}^{t\_in}(s)}\left(\partial E/\partial O_{in}^{t\_in}(s)\right) \tag{23}$$

where O stands for σ, c_L and c_R respectively, and η_O is the step size of each of the components. The derivatives in (23) are as following expressions:

$$\frac{\partial E}{\partial O_{in}^{t\_in}}=\sum_k\sum_i\frac{\partial E}{\partial y^k}\left(\begin{aligned}&\frac{\partial y^k}{\partial \underline{f}_i}\frac{\partial \underline{f}_i}{\partial \underline{\mu}_{in}^i}\frac{\partial \underline{\mu}_{in}^i}{\partial \sigma_{in}^i}\frac{\partial \sigma_{in}^i}{\partial O_{in}^{t\_in}}\\&+\frac{\partial y^k}{\partial \overline{f}_i}\frac{\partial \overline{f}_i}{\partial \overline{\mu}_{in}^i}\frac{\partial \overline{\mu}_{in}^i}{\partial \sigma_{in}^i}\frac{\partial \sigma_{in}^i}{\partial O_{in}^{t\_in}}\end{aligned}\right) \tag{24}$$

where $\partial\sigma_{in}^i/\partial O_{in}^{t\_in}$ stands for detailed topological components in (6)-(7).

b) Adaptive Steepest Descent Directions

The steepest descent algorithm is a classical learning algorithm used in IT2-FNN parameter learnings and applied in several works [16-19]. However, the shortcoming of traditional steepest descent algorithm is the descent directions which are fixed and thus makes the algorithm lack adaptiveness in computing optimal solutions. A new adaptive steepest descent directions is given below:

Assuming $d^*(s)\in R^{t\_in}$ is an adaptive steepest descent direction vector，then we can treat $d^*(s)$ to be the optimal solution of the optimization problem in (25):

$$\max\left\{\left|\nabla E(\sigma,c_L,c_R)^T d(s)\right| \mid s.t.\left\|d(s)\right\|_p\le 1\right\} \tag{25}$$

Thus we can use the Holder inequality as (26) to deal with the components in (25) and to help us calculate $d^*(s)$ in (27):

$$\left|\nabla E^T d(s)\right|\le\left\|\nabla E\right\|_q\left\|d(s)\right\|_p\le\left\|\nabla E\right\|_q \tag{26}$$

$$\left|\nabla E(\sigma,c_L,c_R)^T d^*(s)\right|=\left\|\nabla E(\sigma,c_L,c_R)\right\|_q \tag{27}$$

where q stands for the norm type in the denominator, $q=p/(p-1)$, $p>1$. The components of $d^*(s)$ in (27) are:

$$d_{O_{in}^{t\_in}(s)}^*=\left|\partial E(\sigma,c_L,c_R)/\partial O_{in}^{t\_in}(s)\right|^{q-1}/\left(\left\|\nabla E(\sigma,c_L,c_R)\right\|_q\right)^{\frac{q}{p}} \tag{28}$$

IV. Simulation Examples

In practical applications, all the dynamic systems are with structural or parameter uncertainties more or less. The uncertainties come from plants' internal or external disturbances, e.g. the internal disturbances from airframe flexible coupling and the external disturbances from unpredictable varying aerodynamic in Hypersonic Vehicle [20] and so forth. This paper uses a multi-input single-output (MISO) benchmark system which adopts the proposed IT2-FNN for system modelling. Two types of the IT2-FNN with different antecedent learning methods are studied (i.e. IT2-FNN A is with the proposed adaptive steepest descent algorithm while IT2-FNN B is with fixed learning rate BP algorithm). In what follows, an example with three cases are presented. Case 1 is system modelling of the system without noise. Case 2 and case 3 are additional studies of the case 1 with external disturbances and internal disturbances respectively. In these examples, similarity threshold S_{sim}=0.6, merge-expansion strength index ε=0.6 , step size

$\eta_\sigma=0.5\times10^{-3}$, $\eta_{C_L}=0.75\times10^{-3}$ and $\eta_{C_R}=0.75\times10^{-3}$ for the steepest algorithm structure may feedback unexpected large sequential-derivative values which may lead the algorithm into freak results if step size values are too big. The cases 1 to 2 are compared with existing similar interval type-2 fuzzy TSK systems (IT2-TSK) or similar IT2-FNN approaches proposed in literatures. In addition, the behaviors of the T1-TSK whose rules are used in antecedent structure construction procedure in section III are also taken into the comparison (i.e. the performances of the GD-FNN method). However, since there is no current literature studies in internal disturbances of the system, the case 3 is only compared with the IT2-FNN A/B and the GD-FNN. Performances both in training and test phases are evaluated using root-mean-square-error (RMSE). Moreover, each case has 20 Monte Carlo realizations both in training and test processes, and all the RMSEs are averaged in both processes.

A. *Example (System Modelling)*

In this example, the IT2-FNN is used to model a nonlinear system presented in [8, 9]:

$$y(t+1)=y(t)/\left(1+y^2(t)\right)+v^3(t) \tag{29}$$

Each data pair consists of two inputs $y(t)$, $v(t)$ and one output $y(t+1)$, where $v(t)=\sin(2\pi t)/100$, t=1,2,...,400 and $y(1)=0$. Then, all the raw inputs are normalized into range from -1 to 1 whereas the output remains unchanged. Moreover, we pick 200 data pairs which are uniformly randomly selected as training data while the other 200 are adopted as test data, the comparison methods also use the same data size in the training phase.

1) Case 1: System without noise

Table I presents the rule numbers, total fuzzy sets among all inputs and RMSE information both in the training and test phases of the IT2-FNN and five comparison methods where T1NFS, GD-FNN are T1-TSK methods and T2NFS-T1, T2FLS-G, SEIT2FNN are other IT2-FNN methods based on IT2-TSK systems. The comparison results validate the performances of the IT2-FNN are better than the compared transition method T2NFS-T1, and the IT2-FNN performs even better than the type-2 self-organizing methods with less rules and total number of fuzzy sets. More specifically, comparing to the original T1-FLS performance of the GD-FNN in the training phase, our method can dramatically reduces the RMSE and the IT2-FNN A which uses the adaptive steepest descent algorithm has nearly the same but better performance than the IT2-FNN B.

TABLE I. PERFORMANCES OF THE IT2-FNN AND GD-FNN WITHOUT DISTURBANCES IN CASE 1

Models	Rule Number	No. Fuzzy Set	RMSE Train	RMSE Test
T1NFS[9]	3	6	null	0.0123
T2NFS-T1[9]	6	11	null	0.00763
T2FLS-G[9]	6	12	null	0.0224
SEIT2FNN[9]	6	12	null	0.00596
GD-FNN[11]	4	6	0.0592	0.0065
IT2-FNN B*	4	6	0.0076	0.0041
IT2-FNN A*	4	6	0.0072	0.0039

2) Case 2: System with external uncertainty

In this case, two different amplitude characteristics of white noise are added into the system as external disturbances respectively, where the training data pairs' inputs remain unchanged and the output is added with the noisy signals. On the other hand, the original clean data pairs are used as test data. Table II presents the rule numbers, total fuzzy sets among all inputs and RMSE information both in the training and test phases of the IT2-FNN. Moreover, three comparison methods where the GD-FNN is T1-TSK method and T2FONFS, DIT2NFS-IP are IT2-TSK self-organizing methods. Since the training RMSE values stand for the differences between system performances and the noise-polluted training data. That is the reason why the IT2-FNN method performed poorly under bigger external disturbances in the training phase. Apart from the training phase, the test results are more capable to evaluate the precision degrees in modeling the original system. The comparison results show the performances of the IT2-FNN are better than the corresponding T1-TSK (i.e. GD-FNN). And IT2-FNN A/B perform better than T2SONFS and DIT2NFS-IP with more concise rule structures (i.e. the rule numbers and total number fuzzy sets).

TABLE II. PERFORMANCES OF THE IT2-FNN AND GD-FNN UNDER DIFFERENT EXTERNAL DISTURBANCES IN CASE 2

Models	Noise Amplitude	Rule Number	No. Fuzzy Set	RMSE Train	RMSE Test
T2SONFS[7]	0.1	6	12	null	0.034
	0.5	6	12	null	0.178
DIT2NFS-IP[7]	0.1	6	12	null	0.034
	0.5	6	12	null	0.124
GD-FNN[11]	0.1	4	7	0.0747	0.0268
	0.5	7	10	0.2438	0.1328
IT2-FNN B*	0.1	4	6	0.055	0.0179
	0.5	7	7	0.2568	0.0907
IT2-FNN A*	0.1	4	6	0.0525	0.0178
	0.5	7	7	0.2567	0.0905

3) Case 3: System with internal uncertainty

In this case, two different amplitude characteristics of white noise are added into the system as internal disturbances respectively, where the training data pairs' output remains unchanged and the inputs are added with the same strength noise signals simultaneously. In addition, the original clean data pairs are used as test data. Table III presents the rule numbers, total fuzzy sets among all inputs and RMSE information both in the training and test phases of the IT2-FNN and the GD-FNN. The comparison in these two methods shows the superiority of the IT2-FNN compared to the GD-FNN. However, the performances in Table III are over three times bigger than those in Table II, which indicate the IT2-FNN may not have enough ability in dealing with internal disturbances. Thus we may need to find other ways to identify internal uncertainties in order to enhance the system performances in future works.

TABLE III. PERFORMANCES OF THE IT2-FNN AND GD-FNN UNDER DIFFERENT INTERNAL DISTURBANCES IN CASE 3

Models	Noise Amplitude	Rule Number	No. Fuzzy Set	RMSE Train	RMSE Test
GD-FNN[11]	0.1	4	7	0.0915	0.0471
	0.5	7	10	0.3843	0.2828
IT2-FNN B*	0.1	4	6	0.0607	0.0447
	0.5	7	7	0.2820	0.2304
IT2-FNN A*	0.1	4	6	0.0601	0.0446
	0.5	7	7	0.2787	0.2285

V. CONCLUSIONS

This paper presents a novel approach to creating an interval type-2 fuzzy neural network (IT2-FNN) from a type-1 fuzzy TSK system (T1-TSK). The novelty in the IT2-FNN is with a new seven-layer IT2-FNN structure, an improved mergence algorithm and a new adaptive steepest learning algorithm. The IT2-FNN structure has a rule-ordered fuzzification topological transition layer which helps the structure deal with internal relationships more efficiently and intuitively. There are two automated procedures to construct the IT2-FNN, which are antecedent structure construction and parameter learning in both the antecedent and consequent. The first procedure is fulfilled based on antecedent structure of the T1-TSK and the steps of IT2 fuzzy sets creation, similarity categorization and mergence respectively. The second procedure uses a hybrid learning algorithm to attain the optimal values for all the parameters where the adaptive steepest descent algorithm and the linear least-square method are adopted to adjust the antecedent parameters and consequent parameters, respectively. One benchmark example is utilized with three cases under different disturbance scenarios to compare our approach with the T1-TSK systems, the similar interval type-2 fuzzy neural networks and other type-2 fuzzy self-organizing methods in literatures. The comparisons validate the improved effectiveness of our IT2-FNN for system modelling as compared to T1-TSK systems, especially when there are strong uncertainties. Moreover, our IT2-FNN also shows superior behaviors than similar interval type-2 fuzzy neural networks and interval type-2 fuzzy self-organizing methods. Thus our approach gives an easy and feasible way to take advantage of existing well-behaving T1-TSK systems to generating an IT2-FNN.

ACKNOWLEDGMENT

The authors would like to express our thanks to the China Scholarship Council for supporting the first author's oversea study during this work being done at Wayne State University and also voice appreciation for the comments from professor Shuning Wang, who is with Tsinghua University, China.

REFERENCES

[1] T.-P. Hong and C.-Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy sets and Systems,* vol. 84, pp. 33-47, 1996.

[2] R. Thawonmas and S. Abe, "Function approximation based on fuzzy rules extracted from partitioned numerical data," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 29, pp. 525-534, 1999.

[3] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 22, pp. 1414-1427, 1992.

[4] N. N. Karnik and J. M. Mendel, "Type-2 fuzzy logic systems: type-reduction," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, 1998, pp. 2046-2051.

[5] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *Fuzzy Systems, IEEE Transactions on,* vol. 7, pp. 643-658, 1999.

[6] J. Mendel, H. Hagras, W.-W. Tan, W. W. Melek, and H. Ying, *Introduction to type-2 fuzzy logic control: theory and applications*: John Wiley & Sons, 2014.

[7] C.-F. Juang and C.-Y. Chen, "Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability," *Cybernetics, IEEE Transactions on,* vol. 43, pp. 1781-1795, 2013.

[8] S. D. Nguyen, S.-B. Choi, and Q. H. Nguyen, "An optimal design of interval type-2 fuzzy logic system with various experiments including magnetorheological fluid damper," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* vol. 228, pp. 3090-3106, 2014.

[9] C.-F. Juang and W.-S. Jang, "A type-2 neural fuzzy system learned through type-1 fuzzy rules and its FPGA-based hardware implementation," *Applied Soft Computing,* vol. 18, pp. 302-313, 2014.

[10] H. Ying, "General Takagi-Sugeno fuzzy systems are universal approximators," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 819-823.

[11] S. Wu, M. J. Er, and Y. Gao, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *Fuzzy Systems, IEEE Transactions on,* vol. 9, pp. 578-594, 2001.

[12] M. Biglarbegian, W. Melek, and J. M. Mendel, "On the stability of interval type-2 TSK fuzzy logic control systems," *IEEE Transactions on System, Man and Cybernetics: Part B Cybernetics,* vol. 41, pp. 798-818, 2010.

[13] H. Ying, "Interval type-2 Takagi-Sugeno fuzzy systems with linear rule consequent are universal approximators," in *Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American*, 2009, pp. 1-5.

[14] D. Wu and J. M. Mendel, "A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets," *Information Sciences,* vol. 179, pp. 1169-1192, 2009.

[15] S. W. Tung, C. Quek, and C. Guan, "eT2FIS: an evolving type-2 neural fuzzy inference system," *Information Sciences,* vol. 220, pp. 124-148, 2013.

[16] O. A. Malik, S. Arosha Senanayake, and D. Zaheer, "An adaptive interval type-2 fuzzy logic framework for classification of gait patterns of anterior cruciate ligament reconstructed subjects," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, 2014, pp. 1068-1075.

[17] P. Meesad, "An interval type-2 fuzzy system with hybrid intelligent learning," in *Information and Communication Technologies (WICT), 2014 Fourth World Congress on*, 2014, pp. 263-268.

[18] G. M. Mendez and M. Hernandez, "Intelligent hybrid-learning mechanism for IT2 TSK NSFLS2 composed by REFIL-BP methods," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, 2013, pp. 1119-1124.

[19] G. M. MéNdez and M. De Los Angeles HernáNdez, "Hybrid learning mechanism for interval A2-C1 type-2 non-singleton type-2 Takagi–Sugeno–Kang fuzzy logic systems," *Information Sciences,* vol. 220, pp. 149-169, 2013.

[20] J. Gao, R. Yuan, J. Yi, and C. Li, "Adaptive interval type-2 fuzzy sliding mode controller design for flexible air-breathing hypersonic vehicles," in *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, 2015, pp. 1-6.