A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization

Rahib H. Abiyev\textsuperscript{a,*}, Okyay Kaynak\textsuperscript{b,1}, Tayseer Alshanableh\textsuperscript{a}, Fakhreddin Mamedov\textsuperscript{a}

\textsuperscript{a} Department of Computer Engineering, Near East University, P.O. Box 670, Lefkosia, TRNC, Mersin-10, Turkey
\textsuperscript{b} Department of Electrical and Electronic Engineering, Bogazici University, Bebek, 80815 Istanbul, Turkey

\textbf{ARTICLE INFO}

Article history:
Received 29 April 2009
Received in revised form 19 February 2010
Accepted 25 April 2010
Available online 2 May 2010

Keywords:
Type-2 fuzzy systems
Neuro-fuzzy network
Identification
Equalization of time-varying channel

\textbf{ABSTRACT}

The integration of fuzzy systems and neural networks has recently become a popular approach in engineering fields for modelling and control of uncertain systems. This paper presents the development of novel type-2 neuro-fuzzy system for identification of time-varying systems and equalization of time-varying channels using clustering and gradient algorithms. It combines the advantages of type-2 fuzzy systems and neural networks. The type-2 fuzzy system allows handling the uncertainties associated with information or data in the knowledge base of the process. The structure of the proposed type-2 TSK fuzzy neural system (FNS) is given and its parameter update rule is derived, based on fuzzy clustering and gradient learning algorithm. The proposed structure is used for identification and noise equalization of time-varying systems. The effectiveness of the proposed system is evaluated by comparing the results obtained by the use of models seen in the literature.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays fuzzy neural systems (FNS) are widely used to solve control, prediction, identification problems [2–5]. The structures of these systems are basically based on type-1 fuzzy systems, that is the membership functions used in the antecedent and/or the consequent parts of fuzzy rules are generally of type-1. The type-1 fuzzy systems cannot handle uncertainties associated with information in the knowledge base of the process. In such cases the type-2 fuzzy sets are applied to minimize the effects of the uncertainties in the rule base. The uncertainties in type-2 fuzzy set are arising from different sources [6,7]. One of them is related to the answers of experts to the same question in different manners. Second type of uncertainty is connected with the estimation of the membership function of the same linguistic value by different experts. Third type of uncertainty is connected with the noise of measurements that activate type-1 FLS. Last one is related to the noisy data that are used to tune the parameters of type-1 FLS.

Zadeh has introduced the type-2 fuzzy sets as the extension of the type-1 fuzzy sets [1]. The theoretical background of type-2 fuzzy system and its design principles are described in [6–11]. Type-2 fuzzy logic systems have found many diversified applications; such as for forecasting of time-series [10], equalization of nonlinear time-varying channels [11], pattern recognition [12,13], for control of speed of diesel engines [14], robot control [15] and DC converters [16]. The use of interval type-2 fuzzy logic for minimizing the effects of uncertainty produced by the instrumentation elements, environmental noise and other external disturbances is discussed in [17]. In [18] discrete interval valued type-2 fuzzy system models are generated by a learning parameter. In [19] the learning problems of type-2 fuzzy system and its use for controlling of Hot Strip Temperature are considered. In [20] the stability type-2 fuzzy control system is analysed. In [21] interval type-2 fuzzy sets are used to construct a type-2 support vector machines fusion fuzzy logic system.

Fuzzy systems are generally designed using either Mamdani or TSK type IF-THEN rules. In the former type, both the antecedent and the consequent parts utilize fuzzy values. The TSK type fuzzy rules utilize fuzzy values in the antecedent part, crisp values or often linear functions in the consequent part. In many research works, it has been shown that TSK type fuzzy neural systems can achieve a better performance than the Mamdani type fuzzy neural systems in learning accuracy [2,3]. The theory and design methodologies of Mamdani and TSK type-2 fuzzy systems are presented in [6–11].

The use of type-2 fuzzy neural systems for identification and channel equalization is relatively a recent topic and therefore one can find only a few structures proposed in the literature [22–25]. A recent work [22] describes the design of a type-2 fuzzy neural network (type-2 FNN). The optimal training algorithm derived in this
The paper is given in [25]. In [23] the design of a type-2 fuzzy neural network structure for modelling nonlinear systems is presented. This is further developed in [24] for nonlinear system identification and nonlinear time-varying channel equalization. In both [22, 23] (and also [24]), type-2 Gaussian membership functions with uncertain means and fixed standard deviations are used in the antecedent part. The consequent parts of these systems include weighting interval set with unity membership grade, i.e. an interval type-1 fuzzy set.

In this work, the design of a type-2 TSK based fuzzy neural system (FNS) structure for identification of dynamic time-varying plants and equalization of time-varying channel is presented. The time-varying nature of the plants may be interpreted as uncertainties. This interpretation suggests the use of type-2 fuzzy sets.

The paper is organized as follows. In the following section, the structure of the proposed system is presented. In Section 3, the parameter update rule of type-2 FNS based on fuzzy clustering and gradient learning algorithm is derived. In Section 4, the simulation studies are presented for the identification of a time-varying plant and the equalization of a time-varying channel.

2. Structure of type-2 fuzzy neural system

Type-2 fuzzy systems are characterized by fuzzy IF-THEN rules, the parameters in the antecedent and the consequent parts of the rules include type-2 fuzzy values. In Gaussian type-2 fuzzy sets uncertainties can be associated to the mean and the standard deviation (STD). In Fig. 1(a) and (b) Gaussian type-2 fuzzy sets with uncertain STD and uncertain mean are shown. The mathematical expression for the membership function is expressed as

$$\tilde{\mu}(x) = \exp\left(-\frac{1}{2} \frac{(x - c)^2}{\sigma^2}\right)$$

(1)

Here $c$ and $\sigma$ are the centre and widths of membership function, $x$ is the input vector. In this paper, we consider uncertain STD $\sigma \in [\sigma_1, \sigma_2]$ and uncertain mean $c \in [c_1, c_2]$.

The kernel of a type-2 fuzzy inference system is the fuzzy knowledge base. In a fuzzy knowledge base, the information that consists of input-output data points of the system is interpreted into linguistically interpretable fuzzy rules that have IF-THEN form. The rules used in this paper are such that the consequent parts are of TSK type. The $i$th input $n$ output type-2 TSK fuzzy rule has following form:

$$\text{IF } x_1 \text{ is } \tilde{A}_1^i \text{ and } \ldots \text{ and } x_l \text{ is } \tilde{A}_l^i \text{ THEN } y_j^i = \sum_{i=1}^{l} w_{ij} x_i + b_j$$

(2)

Here $x_1, x_2, \ldots, x_n$ are the input variables, $y_j (j = 1, \ldots, n)$ are the output variables which are linear functions, $\tilde{A}_j^i$ is the type-2 fuzzy membership function for the $i$th rule of the $j$th input which is defined as a Gaussian membership function, $w_{ij}$ and $b_j (i = 1, \ldots, l, j = 1, \ldots, n)$ are the parameters in the consequent part of rules.

In type-2 fuzzy rule both sides, i.e. the antecedent and consequent parts may be of type-2 or one of the sides may be of type-1. The design of fuzzy systems based on the Mamdani type-2 fuzzy rules is given in [6]. The design consists of the determination of the proper fuzzy values of the variables in the antecedent and consequent parts of the fuzzy system. A neural network type of structure is applied in order to design the system.

The structure of the multi-input-single output type-2 fuzzy neural system (FNS) used in this paper is given in Fig. 2. In this structure the input signals to the network are the external input signals $X = \{x_1, \ldots, x_n\}$. The kernel of the fuzzy inference system is the fuzzy knowledge base that consists of the input-output data points interpreted into fuzzy rules. The type-2 FNS is constructed using type-2 TSK fuzzy rules which are given by (2).

The development of the type-2 FNS includes the determination of the proper values of the unknown coefficients of the antecedent and the consequent parts of each rule. Let us consider the design of a type-2 FNS when the input membership functions are of Gaussian and given by (1). If both ($c$ and $\sigma$) parameters of the Gaussian function are considered to be uncertain (within certain intervals),
the parameter space of the system can become very large. In this paper, only one of these parameters is assumed to be uncertain, i.e. uncertain STD and fixed mean or fixed STD and uncertain mean. It is to be noted that the fixed values are also subject to parameter adjustment.

Let us first assume that the parameters of the membership functions are represented by fixed STD and uncertain mean as in Fig. 1(b). In the first layer of Fig. 2, the input signals are distributed. In the second layer each node corresponds to one linguistic term.

Due to the antecedent uncertainties, the output of the type-2 fuzzy rules will have uncertainties. In this paper, the interval type-2 sets are used in the antecedents. Each membership function of the antecedent part is represented by an upper and a lower membership function. They are denoted as $\bar{\mu}(x)$ and $\tilde{\mu}(x)$, or $A(x)$ and $\tilde{A}(x)$:

$$\mu_{\bar{x}}(x_k) = [\mu_{\tilde{x}}(x_k), \tilde{\mu}_{\tilde{x}}(x_k)] = [\tilde{\mu}_{\tilde{y}}, \bar{\mu} ]$$  \hspace{1cm} (3)

In the second layer, for each input signal entering the system, the membership degrees $\mu$ and $\bar{\mu}$ to which the input value belongs to a fuzzy set are determined.

In the inference engine, the common choices for the implication operator are “min” or “prod” t-norms. In this paper the latter is chosen to calculate the firing strengths as shown below, where $*$ is the protod operator:

$$f = \mu_{\bar{x}}(x_1) \ast \mu_{\bar{x}}(x_2) \ast \ldots \ast \mu_{\bar{x}}(x_n)$$

$$f = \tilde{\mu}_{\tilde{x}}(x_1) \ast \tilde{\mu}_{\tilde{x}}(x_2) \ast \ldots \ast \tilde{\mu}_{\tilde{x}}(x_n)$$  \hspace{1cm} (4)

Type reduction and defuzzification operations are next to be considered. First, using (4), the firing strengths of rules are determined. After determining the firing strengths of rules, the defuzzified output of the type-2 TSK fuzzy system is determined. The inference engine of type-2 TSK FNS is proposed in [11,26]. In this paper the inference engine given in [26] is used to determine the output of type-2 FMNN:

$$u = \frac{q \sum_{j=1}^{N} y_j}{ \sum_{j=1}^{N} f_j } + \left( 1 - q \right) \frac{1}{\sum_{j=1}^{N} f_j}$$

$$y_i = \sum_{j=1}^{l} x_j w_{ij} + b_i, \hspace{1cm} i = 1, \ldots, n, j = 1, \ldots, l.$$  \hspace{1cm} (5)

Here $N$ is number of active rules, $\bar{f}_j$ and $f_j$ are determined using (4). $y_i$ is determined using (6), $q$ is a design factor indicating the share of lower and upper values in the final output. The parameter $q$ enables to adjust the lower or the upper portions depending on the level of certainty of the system.

After the calculation of the output signal of the type-2 FNS the training of the network is started. The training includes the adjustment of the system certainty.

3. Parameter update rules

The design of type-2 FNS (Fig. 2) includes determination of the unknown parameters that are the parameters of the antecedent and the consequent parts of the fuzzy if-then rules (2). In the antecedent parts, the input space is divided into a set of fuzzy regions, and in the consequent parts the system behaviour in those regions is described. Recently a number of different approaches have been used for designing fuzzy if-then rules based on clustering [27–32], table look-up scheme [32,33], least-squares method (LSM) [2,11,19,26,28], gradient algorithms [2,4,5,24,25,34,35,37] and genetic algorithms [3,35,36,38]. In [26] the use LSM is proposed for updating parameters of the consequent parts, and fuzzy clustering for the premise parts of the fuzzy systems. In the paper, the fuzzy clustering is applied to design the antecedent (premise) parts, and the gradient algorithm is applied to design the consequent parts of the fuzzy rules. Fuzzy clustering is an efficient technique for constructing the antecedent structures. The aim of clustering methods is to identify a certain group of data from a large data set, such that a concise representation of the behaviour of the system is produced. Each cluster centre can be translated into a fuzzy rule for identifying the class. Clustering has been well used for type-1 fuzzy systems [27]. For type-2 fuzzy systems, subtractive clustering and fuzzy clustering have been developed recently [12,13,27,29,39]. Subtractive clustering [29] is an extension of the grid based mountain clustering [30]. It is unsupervised clustering, in which the number of clusters for input data points is determined by the clustering algorithm. Sometimes we need to control the number of clusters in an input space. In these cases, the supervised clustering algorithms are of primary concern. Fuzzy c-means clustering is one of them. It can efficiently be used for type-1 fuzzy systems [27] with simple structure and sufficient accuracy. However, it is well known that with many clustering algorithms, imprecise information may create imperfect representations of data sets. Therefore various types of uncertainties may have to be taken into account. To this end, the use of interval type-2 FCM is proposed for pattern recognition [12,13,39] and signal classification problems [40]. In this paper, type-2 FCM is used to select the cluster centres of the membership functions in the antecedent part of fuzzy rules of the type-2 FNS, using the input data set of the plant.

The fuzzy membership in type-1 FCM is determined by computing the relative distance among the patterns and the cluster prototypes [27]. The memberships and cluster centres are obtained as

$$u_j(x_k) = \left[ \sum_{k=1}^{c} \left( \frac{d_{ji}(x_k)}{d_{kj}} \right)^{2/(m-1)} \right]^{-1} : \hspace{0.5cm} c_j = \frac{\sum_{i=1}^{l} (u_j(x_i))^{m}}{\sum_{i=1}^{l} (u_j(x_i))^m}.$$  \hspace{1cm} (7)

where $i$ is the number of input patterns (data items), $u_j(x_i)$ is the membership value of pattern $x_i$ for cluster $j$, $d_{ji}(d_{kj})$ denotes the distance between cluster prototypes $c_j(c_k)$ and data point $x_i$ and $m$ is fuzzifier that controls the amount of fuzziness in fuzzy classification. Several studies have shown that reasonable clustering results can be obtained using $m = 2$ [12,13,27]. Such a choice is suitable for a pattern set that consist clusters that are of similar volume and density. Rhee and his co-authors show that, if there is a difference in density among clusters in a pattern set, then the clustering result of FCM may significantly differ depending on the choice of $m$ [12,13]. The establishment of a maximum fuzzy region can result in desirable clustering results. However, this region cannot be represented by FCM since the choice of $m$ affects all clusters equally. To overcome this problem, [12,13,39] propose the interval type-2 FCM, where the maximum fuzzy boundary is controlled by the two values of the fuzzifiers $m_1$ and $m_2$ and the input data set is extended into interval type-2 fuzzy sets. For an input data point $x_i$, the highest and the lowest memberships are defined by using
different fuzzy degrees $m_1$ and $m_2$ and the footprint of uncertainty is thus created. The primary memberships that extend data point $x_i$ by interval type-2 fuzzy sets are determined as [12,13]:

$$\pi_c(X_c) = \begin{cases} 
\frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_1-1}}, & \text{if } \frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_1-1}} > \frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_2-1}} \\
\frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_2-1}}, & \text{otherwise}
\end{cases}$$

$$\psi_c(X_c) = \begin{cases} 
\frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_1-1}}, & \text{if } \frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_1-1}} \leq \frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_2-1}} \\
\frac{1}{\sum_{k=1}^C (d_{jk}/d_{k})^{2/m_2-1}}, & \text{otherwise}
\end{cases}$$

where $d_{jk}/d_{k}$ denotes the distance between the cluster prototypes $c_j(c_k)$ and the data point $x_i$. $C$ is the number of clusters and $\pi_c(X_c)$ and $\psi_c(X_c)$ are the lower and the upper memberships of the data point. Updating of the cluster centres are performed by the extension of interval type-2 fuzzy sets while executing fuzzy c-means algorithm.

The use of the fuzzifiers $m_1$ and $m_2$ result in different objective functions to be minimized:

$$J_{m_1} = \sum_{i=1}^N \sum_{j=1}^C w_{ij}^m d_{jk}^2$$

and

$$J_{m_2} = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m d_{jk}^2,$$

where $1 \leq m_1 \leq m_2 \leq \infty$.

Fuzzy partitioning is carried out through an iterative optimization of the objective functions (9), with the update of membership function and the cluster centres.

The centroids of the type-2 fuzzy sets are determined by the use of the extension principle. Since the secondary membership function of an interval type-2 fuzzy set is equal to one, the following can be obtained [12,13] where $m$ is the degree of fuzziness:

$$c_x = [c_L, c_R] = \sum_{u(x_i)=j} \sum_{x_i \in d_{k}} 1/\left( \sum_{i=1}^I x_i u(x_i)^m / \sum_{i=1}^I u(x_i)^m \right)$$

(10)

The interval type-1 fuzzy set for the cluster centres can be represented as $c_j = 1/\{c_L, c_R\}$. Here $c_L \leq c_j \leq c_R$. The crisp centre can be determined by defuzzification as

$$c_j = \frac{c_L + c_R}{2}$$

The iterative algorithm given in [12,13] is used to find the maximum $c_R$ and minimum $c_L$ value of centres. For a detailed analysis, the reader can refer to [12,13].

The values of $m_1$ and $m_2$ are selected by increasing each value by 1 in the interval $[1, 1.1, 1.0]$. A number of simulation studies are carried out for their possible combinations and the values of the objective function are estimated for these combinations. On the basis of the results obtained, the values of $m_1$ and $m_2$ are decided upon.

As the result of the clustering algorithm described above, each cluster centre $c_j$ is determined, taking into account the maximum of the centre $c_R$, and also the minimum of the centre $c_L$, where $c \leq c_L \leq c_L$. In the simulation studies presented in the paper, we use type-2 membership functions with uncertain means for each input (see Fig. 1(b)). The centres of these membership functions are obtained by spreading the cluster centres by $\pm 3\sigma_j$, that is to say $c_1 = c_j - 3\sigma_j$ and $c_2 = c_j + 3\sigma_j$, where $c_1 \leq c_j \leq c_2$. However, a decrease in the number of clusters leads to a decrease in the membership functions used, and this affects the accuracy of the model, leading to an increase in the output error. In the paper, the selection of the number of clusters is done through simulations to provide required accuracy.

After the design of the antecedents parts by fuzzy clustering, the gradient descent algorithm is applied to design the consequent parts of the fuzzy rules. In what follows, the parameter update rules are derived for a type-2 TSK FNS with fixed mean and uncertain STD (it should not be forgotten that the “fixed” mean is also updated).

At the first step, the output error is calculated:

$$E = \frac{1}{2} \sum_{i=1}^N (u_i^d - u_i)^2$$

(11)

Here $O$ is number of output signals of the network (in the given case $O = 1$), $u_i^d$ and $u_i$ are the desired and the current output values of the network, respectively. The parameters $w_{ij}$, $b_i (i = 1, \ldots, l, j = 1, \ldots, n)$ and $c_{1j}, c_{2j}$ and $\sigma_{ij} (i = 1, \ldots, l, j = 1, \ldots, n)$ are adjusted using the following formulas:

$$w_{ij}(t+1) = w_{ij}(t) - \gamma \frac{\partial E}{\partial w_{ij}}; \quad b_j(t+1) = b_j(t) - \gamma \frac{\partial E}{\partial b_j}$$

(12)

$$c_{1j}(t+1) = c_{1j}(t) - \gamma \frac{\partial E}{\partial c_{1j}}; \quad c_{2j}(t+1) = c_{2j}(t) - \gamma \frac{\partial E}{\partial c_{2j}}$$

(13)

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \gamma \frac{\partial E}{\partial \sigma_{ij}}$$

(14)

Here $\gamma$ is the learning rate, $l$ is the number of input signals of the network (input neurons) and $n$ is the number of rules (hidden neurons), $i = 1, \ldots, l, j = 1, \ldots, n$. The derivatives in (12) are determined by the following formulas:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial u_i} \frac{\partial u_i}{\partial w_{ij}} = \frac{\partial E}{\partial u_i} (u(t) - u^d(t)) \times \frac{q \times f_j}{\sum_{j=1}^n f_j} \times x_i,$$

$$j = 1, \ldots, n$$

(15)

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial u_i} \frac{\partial u_i}{\partial b_j} = \frac{\partial E}{\partial u_i} (u(t) - u^d(t)) \times \frac{q \times f_j}{\sum_{j=1}^n f_j},$$

$$j = 1, \ldots, n$$

(16)

The derivatives in (13) and (14) are determined by the following formulas:

$$\frac{\partial E}{\partial \sigma_{ij}} = \sum_j \left( \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial f_j} \frac{\partial f_j}{\partial \sigma_{ij}} + \frac{\partial E}{\partial u_1} \frac{\partial u_1}{\partial f_1} \frac{\partial f_1}{\partial \sigma_{ij}} \right)$$

(17)
Upper and lower membership functions between ith input and jth hidden neurons of layer 3 can be written as follows (see Fig. 1(b)):

\[
\begin{align*}
\mu^u_{ij}(x_i) &= \begin{cases} 
G(c_{2ij}, \sigma_j, x_i), & x_i \leq \frac{c_{1ij} + c_{2ij}}{2} \\
G(c_{1ij}, \sigma_j, x_i), & x_i > \frac{c_{1ij} + c_{2ij}}{2} 
\end{cases} \\
\mu^l_{ij}(x_i) &= \begin{cases} 
G(c_{1ij}, \sigma_j, x_i), & x_i < c_{1ij} \\
1, & c_{1ij} \leq x_i \leq c_{2ij} \\
G(c_{2ij}, \sigma_j, x_i), & x_i > c_{2ij} 
\end{cases}
\end{align*}
\]

Here \(G(c_{ij}, \sigma_j, x_i)\) is determined as

\[
G(c_{ij}, \sigma_j, x_i) = \exp\left(-\frac{1}{2} \frac{(x_i - c_{ij})^2}{\sigma_j^2}\right)
\]

Then

\[
\begin{align*}
\frac{\partial \mu^u_{ij}(x_i)}{\partial x_i} &= \begin{cases} 
\frac{G(c_{1ij}, \sigma_j, (x_i - c_{1ij}))}{\sigma_j^2}, & x_i < c_{1ij} \\
0, & c_{1ij} \leq x_i \leq c_{2ij} \\
0, & x_i > c_{2ij}
\end{cases} \\
\frac{\partial \mu^l_{ij}(x_i)}{\partial x_i} &= \begin{cases} 
0, & x_i < c_{1ij} \\
0, & c_{1ij} \leq x_i \leq c_{2ij} \\
\frac{G(c_{2ij}, \sigma_j, (x_i - c_{2ij}))}{\sigma_j^2}, & x_i > c_{2ij}
\end{cases}
\end{align*}
\]

The parameters of the type-2 FNS can thus be updated using (12)–(14) together with (15)–(25). It should be stressed here that in the simulations examples given in this paper, Eqs. (13), (14) and (17)–(25) are not used since the parameters of the antecedent part are determined by the use clustering techniques. The equations are driven and given above to enable the reader to use them in online applications where the use of clustering can be difficult. In such cases, clustering can be used at the beginning on some training data to arrive at some initial values and then the parameters of the type-2 membership functions can be updated in an online manner using the update rules given above.

During learning, the value of \(q\) is optimized from an initial value of 0.5 using

\[
q(t + 1) = q(t) - \gamma \frac{\partial E}{\partial q} = (u - u^d) \left( \frac{f_j - \tilde{f}_j}{\sum_{j=1}^{N} f_j^2 - \sum_{j=1}^{N} \tilde{f}_j^2} \right)
\]

One important problem in learning algorithms is convergence. The convergence of the gradient descent method depends on the selection of the initial values of the learning rate. Usually, these values are selected in the interval [0–1]. A large value of the learning rate may lead to unstable learning, a small value of the learning rate results in a slow learning speed. In this paper, an adaptive approach is used for updating these parameters. That is, the learning of the type-2 FNS parameters is started with a small value of the learning rate \(\gamma\). During learning, \(\gamma\) is increased if the value of change of error \(\Delta E = E(t) - E(t+1)\) is positive, and decreased if negative. This strategy ensures a stable learning for the type-2 FNS, guarantees the convergence and speeds up the learning. However, the optimal value of the learning rate for each time instance can be obtained using a Lyapunov function [24,41].

Let \(\gamma(t)\) be the learning rate for the weights \(W = [w_{ij}, \sigma_{ij}, b_{ij}, c_{1ij}, c_{2ij}, \sigma_j, q]\) of the type-2 TSK FNS, trained using (12)–(14). The convergence is guaranteed if the following condition is satisfied:

\[
0 < \gamma(t) < \frac{2}{\left(\max_t ||\partial u(t)/\partial W||\right)^2}
\]

The derivation of this condition is given in Appendix A.

4. Simulation studies

In industry many dynamic plants are susceptible to internal and external disturbances. These disturbances cause variations in the parameters of the plants. The time-varying nature of the plant may
be interpreted as the uncertainties in the plant coefficients, which can be described using type-2 fuzzy sets. In this section, the type-2 FNS structure described in the previous section is used as an adaptive system for the identification of time-varying plants. In what follows, a number of applications are described and the simulation results are presented and compared with those obtained using other approaches proposed in the literature.

4.1. Identification of a dynamic time-varying plant

The identification problem involves the finding of the relation between the input and the output of the system. In Fig. 3, the structure of the identification scheme is shown. The inputs to the type-2 FNS based identifier are the external input signals, its one-, two-,...-\textit{d}_t-step delayed values and the one-, two-,...-\textit{d}_o-step delayed outputs of the plant. Here the problem is to find such values of the parameters of the type-2 FNS structure that the difference between the plant output \( y(k) \) and the identifier output \( y_n(k) \) will be minimum for all input values of \( u(k) \).

Example 1. As a first example to identification, a Bounded-Input, Bounded-Output (BIBO) nonlinear plant which has become almost a benchmark in the literature [42] is considered. The process is described by the following difference equation:

\[
y(k) = u(k)^3 + \frac{y(k - 1)}{1 + y(k - 1)^2} \quad (28)
\]

where \( y(k) \) and \( y(k - 1) \) are the current and the one step delayed outputs of the plant, \( u(k) \) is the current input for the plant.

The current output of plant depends on the previous input and output signals. The one step delayed state of the system and the control signal are used as the input signals for identifier and fed into the type-2 FNS. The excitation signal for the plant (28) is an independent and identically distributed (iid) uniform sequence over \([-1, 1]\) for about 1/4 of the 400 time steps and the sinusoid \( \sin(\pi k/45) \) for the remaining time. The unknown parameters of type-2 TSK FNS are the parameters of the membership functions of the second layer \( (c_1, c_2, \text{and } c_i) \) and the parameters of the linear function \( (w \text{ and } b) \).

First, the fuzzy classification described in Section 3 is applied to the input data points of the plant in order to determine the cluster centres. In identification, the clustering involves the determination of clusters in data space for the input signals \( y(k - 1) \) and \( u(k) \), and the translation of these clusters into fuzzy rules such that the model obtained is close to the identified system. The obtained cluster centres are then used in order to organize the premise parts of the fuzzy rules. It is to be noted that with such an approach, the number of parameters to be determined for the antecedent part are reduced significantly. That is to say, with three cluster centres for each input, a total of six type-2 membership functions exist with 3 parameters each and thus having 18 parameters in total. Using these six membership functions, three for each input signal, nine rules could be constructed. On the other hand, if the generation of the membership functions is done completely randomly, then 18 membership functions are needed for the 9 rules and this would require the determination of 81 parameters.

As a result of clustering, three clusters are obtained for each input signal and 9 fuzzy rules are constructed. These are then used in order to organize the premise parts of the fuzzy rules, i.e. the 2nd layer of the type-2 FNS. Then the gradient algorithm is applied for the learning of the parameters in the consequent part. These are the \( w \) and \( b \) parameters of the linear functions of the layer 4. The initial values of \( w \) and \( b \) are selected randomly, in the interval \([-1, 1]\). Using the parameter update rules derived above, they are updated for the given excitation signals. The training of type-2 FNS is carried out for 400 time steps; training epoch being equal to one. Total number of parameters to be determined are 18 (premise part) + 27(consequent part) = 45. As a performance criterion the root-mean-square-error, given in [25], is used with \( K = 400 \).

In Fig. 4 RMSE values of type-2 FNS with 9 fuzzy rules is shown for 10 training epochs. After training, the following test signal is used to see identified results:

\[
u(k) = \begin{cases} 
-0.7 + \frac{\text{mod}(k, 50)}{40}, & k \leq 80 \\
rands(1.1), & 80 < k \leq 130 \\
0.7 - \frac{\text{mod}(k, 180)}{180}, & 130 < k \leq 250 \\
0.6 \cos(\pi k/50), & k > 250
\end{cases} \quad (29)
\]

The RMSE value in one epoch was obtained as 0.002415 and in ten epochs as 0.001746. Fig. 5 shows identification results of the type-2 FNS. Here the solid line is the plant output and the dashed line is the type-2 FNS identifier output. This performance is com-

Fig. 3. Identification scheme.

Fig. 4. RMSE values obtained during learning (with nine rules).

Fig. 5. Simulation results of identification, where the solid line denotes the output of the plant and the dashed line denotes the type-2 TSK FNS output.
pared in Table 1 with the performance of the type-1 Fuzzy Neural Network (FNN) and the type-2 FNN structures proposed in [23,24].

Table 1

<table>
<thead>
<tr>
<th>Models</th>
<th>Network parameters</th>
<th>Epochs</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-2 FNN [23]</td>
<td>8 rules</td>
<td>1</td>
<td>0.0032047</td>
</tr>
<tr>
<td>Type-1 FNN [23]</td>
<td>8 rules</td>
<td>1</td>
<td>0.0069245</td>
</tr>
<tr>
<td>Type-2 TSK FNS</td>
<td>9 rules</td>
<td>1</td>
<td>0.002415</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.001746</td>
</tr>
<tr>
<td>Type-2 TSK FNS</td>
<td>4 rules</td>
<td>1</td>
<td>0.003272</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.002701</td>
</tr>
</tbody>
</table>

When the number of rules in a type-1 and a type-2 FNS are the same, then the number of design parameters in the latter is more than the one in the former. The larger number of parameters increases the approximation capability of the type-2 FNS. An increase in the computational load (as compared to type-1) may not be the case, as a type-2 FNS may be able to describe the process, using less number of fuzzy rules than a type-1 fuzzy system. In order to show this, the performance of a type-1 FNS with 8 rules was obtained during learning is shown, the number of epochs being 10. As shown in Fig. 6, the RMSE value in one epoch was obtained as 0.003272 and ten epochs as 0.002701. The simulation results obtained for different number of rules shows that the performance of type-2 FNS is better than type-1 FNS. Type-2 FNS has a better capability to describe the process using less number of fuzzy rules.

Example 2. In the second example, a second order nonlinear time-varying plant [43] is considered. The process is described by the following difference equation:

\[ y(k) = \frac{y(k-1)y(k-2)u(k-1)(y(k-3) - b(k)) + c(k)u(k)}{a(k) + y(k-2)^2 + y(k-3)^2} \] (30)

Here \( y(k-1), y(k-2), y(k-3) \) are one-, two- and three-step delayed outputs of the plant, \( u(k) \) and \( u(k-1) \) are the current and the one step delayed input of the plant. The time-varying parameters \( a(k), b(k) \) and \( c(k) \) are given by the following formulas:

\[ a(k) = 1.2 - 0.2 \cos \left( \frac{2\pi k}{T} \right) \]
\[ b(k) = 1 - 0.4 \sin \left( \frac{2\pi k}{T} \right) \]
\[ c(k) = 1 + 0.4 \sin \left( \frac{2\pi k}{T} \right) \] (31)

Here \( T \) is the time-span of the test. The plots of the time-varying coefficients over the time span are given in Fig. 7. This plant is a special case of the plant that is used in [42]. When \( a = 1, b = 1 \) and \( c = 1 \), it reduces to it. The training data are the same as those used in Example 1.

The above described type-2 FNS structure with 4 fuzzy rules is used for the identification of the time-varying plant (30). As can be seen, the current output of plant depends on the previous input and the output signals. During identification, in order to reduce the number of parameters, only the one delayed state of system and the control signal are fed into the type-2 FNS as inputs.

As before, as the first step, type-2 fuzzy classification is applied in order to select the parameters of the premise parts, that is the parameters of Gaussian membership functions used in the second layer of type-2 FNS. As in the previous example, type-2 fuzzy c-means clustering is used for the input space with 2 clusters for each input and thus 4 fuzzy rules are constructed. After all this the gradient algorithm is applied for updating the parameters of the consequent parts, i.e. the layer 4 of type-2 FNS. The initial values of the linear parameters of layer 4 are randomly generated in the interval \([-1,1]\] and, using the parameter update rules derived above, they are updated for the given excitation signals. The training is continued for 100 epochs with 1000 time steps in each epoch. As a performance criterion RMSE is used with \( K = 1000 \).

As a result of training, the parameters of the type-2 FNS are determined. Fig. 8 shows RMSE values obtained during learning. After training the following signal is used for the test:

\[ u(k) = \begin{cases} 
\sin(\pi k/25), & k < 250 \\
1.0, & 250 \leq k < 500 \\
-1.0, & 500 \leq k < 750 \\
0.3 \sin(\pi k/25) + 0.1 \sin(\pi k/32) + 0.6 \sin(\pi k/10), & 750 \leq k < 1000 
\end{cases} \] (32)

The RMSE obtained after training for 4 fuzzy rules (24 parameters) was 0.0284, and the RMSE for test data was 0.0601. Fig. 9 compares the actual plant output with that of the type-2 FNS identifier using test signal (32). For 9 fuzzy rules RMSE after training was 0.0253 and for the test was 0.0424. In order to have a basis for comparison, the simulation of the control system with the plant (30) is performed using a type-1 TSK FNS too. After training, 9 fuzzy rules...
Fig. 9. Results of identification, where the solid line denotes the output of the plant and dashed line type-2 TSK FNS output.

(63 parameters) are generated. The RMSE value after training was 0.0282 and for the test data was 0.0598. The comparison is done using the same initial conditions. This again indicates that the use of type-2 FNS, instead of type-1 FNS does not necessarily mean an increase in the parameter space, the former can achieve similar (or better, as is the case in this example) performance with much less number of rules.

4.2. Equalization of time-varying channel

Signals transmitted through a channel are corrupted in a random manner by a variety of possible mechanisms, such as additive thermal noise generated by electronic devices, man-made noise and atmospheric noise. Interference from other users of the channel is another form of additive noise that often arises in both wireless and wire line communication systems [44]. All these interference and noise cause distortion of the transmitted signal. Channel equalization is considered to eliminate these distortions between a transmitter and a receiver. It can greatly improve the quality of transmission, which in turn leads to more efficient communication.

Equalization can be divided into two types: sequence estimation, and symbol detection [45,46]. The first one needs channel estimation, and it is computationally complex. In this paper adaptive channel equalization that realizes symbol detection technique is considered. This is a classification problem and here the aim is the separation of symbols in the output signal space whose optimal decision region boundaries are nonlinear.

Various equalizers have been applied to equalize these distortions and recover the original transmitted signal [44–50]. Linear equalizers cannot reconstruct the transmitted signal when channels have significant nonlinear distortion [44–49]. Neural networks based on multilayer perceptron (MLP) and radial basis functions (RBF) are widely used for the equalization of nonlinear channel distortion [46–50]. The MLP equalizers require long time for training and are sensitive to the initial choice of network parameters [46,48,49]. The RBF equalizers are simple and require less time for training, but usually require a large number of centres, which increase the complexity of computation [47,50].

One of the effective ways for the development of adaptive equalizers for nonlinear channels is the use of fuzzy technology. This type of adaptive equalizer can process numerical data and linguistic information in natural form [11,51–55]. Using input-output data pairs of the channel the fuzzy IF-THEN rules are determined. These rules are used to construct the filter for the nonlinear channel. In these systems, the incorporation of linguistic and numerical information improves the adaptation speed and the bit error rate (BER) [52]. However, the type of uncertainties described above and the linguistic and numeric uncertainties are very often poorly described by type-1 fuzzy equalizers. The presence of uncertainty changes and sometimes increases the degree of nonlinearity of the decision boundary of the channel. The use of a type-2 FNS equalizer can then become necessary to achieve a better performance. For example Liang and Mendel [11] have used a type-2 fuzzy adaptive filter for a time-varying channel. In this section, the type-2 TSK FNS structure discussed above for identification purposes is used for equalization of a time-varying channel.

The architecture of the type-2 TSK FNS based equalization system is shown in Fig. 10. The random binary input signals $s(k)$ are transmitted through the communication channel. Channel medium includes the effects of the transmitter filter, transmission medium, receiver filter and other components. Input signals can be distorted by noise and intersymbol interference. The transmitted signals $\bar{s}(k)$ are known input samples with an equal probability of being $-1$ and $1$. These signals are corrupted by additive noise $n(k)$. These corrupted signals are inputs for the equalizer. In channel equalization, the problem is the classification of incoming input signal of equalizer onto a feature space which is divided into two decision regions. A correct decision of equalizer occurs if $\bar{s}(k) = s(k)$. Here

Fig. 10. The architecture of the type-2 TSK FNS based equalization system.

Fig. 11. (a) Channel states (noise free) of time invariant channel, where $\times$ denotes the category $x(k)$ is 1, and $o$ denotes the category $x(k)$ is $-1$ and (b) optimal decision region of time invariant channel.
s(k) is transmitted signal, i.e. channel input, $\tilde{s}(k)$ is the output of equalizer.

For simulation studies, the following nonlinear channel model was used [11,46,52]:

$$
x(k) = a_1(k)s(k) + a_2(k)s(k-1) - 0.9 \times (a_1(k)s(k) + a_2(k)s(k-1))^3 + n(k)
$$

(33)

In above $x(k)$ is the output of the channel, $s(k-1)$ is the time delay introduced by the channel, and $a_1(k)$ and $a_2(k)$ are time-varying channel coefficients with initial values $a_1(0) = 1$ and $a_2(0) = 0.5$. When channel coefficients are constant, that is $a_1(k) = 1$ and $a_2(k) = 0.5$, for all values of $k$, then the channel is a time invariant one. The states of time invariant channel are plotted in Fig. 11(a), the optimal decision region of this channel is shown in Fig. 11(b). As can be seen, the input space is not linearly separable, the boundary is nonlinear. Because a linear equalizer can only generate a linear decision boundary, its bit error rate will be considerably large. For this reason, a nonlinear equalizer needs to be used for this channel. For a time-varying channel, the coefficients $a_1(k)$ and $a_2(k)$ are uncertain. For the simulation studies presented in this paper, these are generated by using second-order Markov model in which white Gaussian noise source drives a second-order Butterworth low-pass filter [11] with a cut-off frequency 0.1. As time-varying coefficients $a_i$, colored Gaussian sequences are generated with a standard deviation of $\beta = 0.1$. The curves representing the time variation of the channel coefficients are depicted in Fig. 12. During equalizer design, the sequence of transmitted signals is given to the channel input. The first 200 symbols are used for training and the remaining 800 signals are used for testing. They are assumed to be an independent sequence taking values from $\{-1,1\}$ with equal probability. The additive Gaussian white noise $n(k)$ with noise variance $\sigma^2$ and normal distribution is added to the channel. The states of the time-varying channel for noise free (a) and noisy cases (b) are plotted in Fig. 13. The channel states are eight clusters instead of eight individual points. As shown from these clusters the channel output states are uncertain and $s(k)$ determines which cluster channel states belongs to. In the output of the equalization system, the deviation of original transmitted signal from the current equalizer output is determined. This error $\varepsilon(k)$ is used to adjust network parameters. Training is continued until the value of the error for all training sequence of signals is acceptably low.

The simulations studies are carried out on the proposed type-2 TSK FNS, and, for comparison purposes, on a well known ANFIS structure with 16 rules. The order of the equalizer was chosen to be 2. The input signals to the equalizer are the outputs of channel $x(k)$ and $x(k-1)$. Fuzzy clustering described above is applied to these input signals to determine the cluster centres, which are then used as the centres of Gaussians in the second layer of the structure. As in the identification examples, the parameters of linear parts are updated using the gradient algorithm. Nine type-2 fuzzy rules are used in the hidden layer of type-2 FNS. The final decision region of the type-2 FNS equalizer for the time-varying channel with Gaussian white noise with the variance $\sigma^2 = 0.2$ is given in Fig. 14. Fig. 15(a) and (b), the convergence curves of the type-2 FNS based equalizer over 200 learning iterations for two different values of $\beta (0.1$ and $0.3$) are given. The curves are obtained for SNR = 20 dB. A number of simulation studies were carried out for both (type-2 TSK FNS and ANFIS) equalizers to determine the Bit error rate versus signal to noise ratio for different noise variances and also bit error rate versus noise variances for different signal to noise ratios. Fig. 16 illustrates the BER performance of the type-2 TSK FNS and ANFIS equalizers for the time-varying channel (33), averaged over 10 independent trials. As shown in figure performance of the type-2 TSK FNS based equalizer is better. Here the factor of time-varying coefficient is selected as $\beta = 0.1$. In Fig. 17 the change of BER is plotted for six different values of $\beta$, keeping SNR fixed at 20 dB. There again, the performance of the type-2 TSK FNS equalizer is better than the ANFIS equalizer.
5. Conclusions

A type-2 TSK FNS is designed for identification and equalization of time-varying systems. The usage of type-2 fuzzy sets enables the system to cope with uncertainties and to handle uncertain information effectively. The structure of the type-2 TSK FNS is presented and the parameter update rules of the structure are derived, based on the gradient descent algorithm. In order to evaluate the performance of the system, a number of simulation studies are carried out and some results are presented for identification of dynamic plants and equalization of a time-varying channel. The plant and channel models are taken from the literature as much as possible to enable a direct performance comparison. The simulation results indicate the potential of the proposed structure. In both the identification and the equalization cases, the performance of the type-2 TSK FNS is much better, resulting in smaller RMSE values. For the same number of rules a type-2 FNS has more parameters than a type-1 FNS. However, it is also shown in the paper that the type-2 FNS can model a complex process using less number of rules than type-1 FNS. This means that the number of parameters to be updated may not necessarily need to be more but in fact can be less.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful suggestions in the improvement of this manuscript. The second author acknowledges the TUBITAK Grant No: 107E284 and Bogazici University Project Grant No: 08A204. The authors also thank to Near East University for support of this project.

Appendix A.

As mentioned earlier, the learning rate used in this paper is time varying. Let us derive the optimal learning rate using a Lyapunov function [41].

Let $\gamma(t)$ be the learning rate at discreet time $t$ in the weight update formulas (12)–(14) for the weights $W = [w_i, a_{ij}, b_{ij}, c_{1ij}, c_{2ij}, o_{ij}, q]$. The convergence is guaranteed if the condition (27) is satisfied.

The statement (27) can be proved by choosing a Lyapunov function. Let us define Lyapunov function as

$$V(t + 1) = \frac{1}{2} e^2(t + 1) - e^2(t)$$

Here $e(t) = (u_i^d(t) - u(t))$ represents the error function calculated during the learning process:

$$\Delta V(t) = V(t + 1) - V(t) = \frac{1}{2} e^2(t + 1) - e^2(t)$$

The error difference is determined as

$$\Delta e(t) = \frac{\partial e(t)}{\partial W} \Delta W$$

From (35) $\Delta V(t)$ is represented as
\[
\Delta V(t) = \frac{1}{2} \alpha \Delta e(t) (t) \Delta e(t) - \frac{1}{2} \left( \frac{\partial u(t)}{\partial W} \right)^T \gamma(t) e(t) \frac{\partial u(t)}{\partial W} \\
\times \left( 2 \gamma(t) \left( \frac{\partial u(t)}{\partial W} \right)^T \gamma(t) e(t) \frac{\partial u(t)}{\partial W} \right)^2 - 2 \right)
\]

From here

\[
\Delta V(t) = \frac{1}{2} \gamma(t) \left( \frac{\partial u(t)}{\partial W} \right)^T \gamma(t) e(t) \frac{\partial u(t)}{\partial W}^2 - 2 \right)
\]

From the Lyapunov stability theorem, asymptotic stability is guaranteed under the following sufficient condition:

\[
0 < \gamma(t) < \frac{2}{(\max[\partial u(t)/\partial W])^2}
\]

References