2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Type-2 fuzzy LSTM for nonlinear system modeling

Francisco Vega
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico

Xiaoou Li
Departamento de Computación
CINVESTAV-IPN
Mexico City, 0736, Mexico

Brisbane Ovilla-Martínez
Departamento de Computación
CINVESTAV-IPN
Mexico City, 0736, Mexico

Wen Yu
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico

***Abstract*—Type-2 fuzzy systems have a great adoption in different branches of engineering, due to the fact that this type of fuzzy systems are very well suited to tasks related to nonlinear systems. Data driven models like neural networks and fuzzy systems have some disadvantages, such as the high and uncertain dimensions and complex learning process. In this paper, we show the advantages of type-2 fuzzy systems over type-1 fuzzy systems in modeling nonlinear systems. We combine Type-2 Takagi-Sugeno fuzzy model with the popular deep learning model, LSTM (long-short term memory), to overcome the disadvantages fuzzy model and neural network model. We propose a fast and stable learning algorithm for this model. Comparisons with others similar black-box and grey-box models are made, in order to show the advantages of the type-2 fuzzy LSTM neural networks.**

***Index Terms*—deep learning, LSTM, type-2 fuzzy systems, modeling**

I. Introduction

In recent years, intelligent systems have been widely recognized, mainly artificial neural networks (NNs), fuzzy systems (FSs) and the combination of both known as fuzzy neural networks (FNNs). These intelligent systems have proven to be very useful in tasks such as control and identification of systems in various areas of engineering. This is supported by a large body of work, as can be seen in [1]–[3]. A deep learning model, named long-short term memory (LSTM), has been developed [4]–[6]. It has a recurrent structure and is based on information management through gates, these gates measure the suitability of the data they receive as input data, the stored data by the LSTM and the data generated by the LSTM as result. LSTM networks overcome many disadvantages of RNNs and they converge relatively faster [7]–[10].

FSs and FNNs have had a really extensive development, since their structure is interpreted as a set of "IF-THEN" rules that are easy to understand. Their theory has arrived at the distinction of two main types of FSs, type-1 FSs (T1FSs) and type-2 FSs (T2FSs). T2FSs are considered extensions of the T1FSs, because the membership value of a type-2 fuzzy set is a type-1 fuzzy number. The membership functions (MFs) of T1FSs are well determined, while the MFs of T2FSs are fuzzy; there are infinitely type-1 MFs (T1MFs) contained in the uncertainty footprint characteristic of type-2 MFs (T2MFs). T2FSs have proven to be better at handling data with uncertainties and noise, as evidenced by the work discussed below. The structure of a FNN, which is known as a self-evolving interval type-2 fuzzy neural network (ST2FNN), is shown in [11]. The ST2FNN is constructed as a Takagi-Sugeno-Kang (TSK) FS with an adaptive structure, where the part of the antecedents in the fuzzy rules is defined with T2MFs. In addition, the ST2FNN was used in the modeling of nonlinear systems, adaptive noise cancelation and prediction of chaotic signals, obtaining good results. A study presented in [12] discusses the concept of type-2 fuzzy inference system (FIS) using type-2 fuzzy sets and FNN based refinement. A category of type-2 FNNs was developed based on type-2 fuzzy set constructions with these fuzzy sets forming a collection of IF-THEN rules.

Similar to the previous one, the functional-link based interval type-2 compensatory fuzzy neural network is described in [13]. It consists of six layers, which combines the compensatory fuzzy reasoning method, and the consequent part combines the proposed functional link NN with interval weights. For the identification of time-varying nonlinear systems, a mutually recurrent interval type-2 neural fuzzy system was proposed in [14]. This FS uses type-2 fuzzy sets to enhance noise tolerance in the identification process. The IF part of each fuzzy rule is defined using interval type-2 fuzzy sets, and the THEN part is of the TSK type with interval weights. To train T2FSs, a methodology based on sliding modes was proposed in [15]. With this methodology, the parameters of a type-2 FNN of the TSK form are adjusted, guaranteeing that this process will be done in a fast and robust way compared to other algorithms used in the learning of this type of IS.

An easy-to-interpret type-2 FNN was made in [16]. This network, known as self-organizing interval type-2 fuzzy-neural-network (ST2FNN), adjusts during its learning the size of its structure as well as the parameters specific to this process. In [17], a type-2 interval neural network based on long-short memory term is proposed. By means of the interval type-2 intuitionist set, the hesitation of reasoning is described

This work was supported in part by CONACYT under the grant CONACyT-A1-S-8216 and by CINVESTAV under the grants, SEP-CINVESTAV-62 and CNR-CINVESTAV

and involved to determine the fuzzy rules in the inference system, so the choice of fuzzy rules not only considers the membership value but also the degree of non-membership. Meanwhile, a type-2 FIS using genetic programming for time series forecasting applications was developed in [18]. This FS, called GPFIS-Forecast+, has been shown to be effective in noisy applications. To overcome the difficulties in data collection, the type-2 fuzzy time series forecasting model (T2FTS) was proposed in [19]. The T2FTS was used to exploit more information in time series forecasting. The concepts of sliding window method (SWM) and fuzzy rule-based systems (FRBS) were incorporated in the T2FTS to obtain forecast values.

Accurate forecasting of renewable electricity generation (REG) is essential in energy planning. In [20], the adaptive neuro-fuzzy inference system (ANFIS) with fuzzy c-means and T2MFs, and the NN of long-short term memory (LSTM), were applied to perform short-term day-ahead REG forecasts. Following the same theme, a nonlinear consequent part recurrent type-2 recurrent fuzzy system (NCPRT2FS) for modeling renewable energy systems is presented in [21]. The FNN is used for identifying and predicting the behavior of an experimental array of solar cells and a wind turbine. Also, optimization techniques were used, such as pruning of the fuzzy rules and initial tuning of the MFs. A complex problem such as the COVID-19 pandemic is characterized by high levels of uncertainty in its behavior. A deep interval type-2 fuzzy LSTM model for the prediction of COVID-19 incidence, including new cases, recovery cases and mortality rate in short and long time series is presented in [22]. This IS was evaluated on real data sets produced by the WHO on major high-risk countries.

The application of type-2 fuzzy neural networks (T2FNNs) is reviewed in [23]. Different T2FNNs that have been used for system identification are discussed, their disadvantages and advantages are described, as well as their effectiveness in different applications. The previous works show the advantages of using T2FSs over T1FSs, as well as the value of the former in the estimation of nonlinear models. Motivated by the above, in this paper we modified the FNN shown in [24] and [25] to boost its performance. The FNN is based on a T1FS for the estimation of nonlinear systems, obtaining favorable results with it, so adapting its structure to a T2FS will improve its performance. In the Fig. 1, we show a Gaussian T2MF with uncertain standard deviation, which is upper bounded by a T1MF (UMF) and lower bounded by another T1MF (LMF) and the gray area is the footprint of uncertainty (FOU). The T2FSs cope with the uncertainties of a system from the fuzzy rules that define it. Unlike the T1MFs present in the T1FSs, the T2MFs of the T2FSs are themselves fuzzy because they are defined in their respective FOU. There are an infinite T1MFs in a FOU, which is why T2FSs have the ability to work with data that have uncertainties and noise in a more efficient way.

We combine the FNN structure known as a type-2 Takagi-Sugeno fuzzy neural network with long-short memory term cells, and propose a new model Type-2 fuzzy LSTM (T2LSTM) and a training algorithm for it. Then, we compare it with the type-1 Takagi-Sugeno fuzzy neural network with long-short memory term cells (T1LSTM). T1LSTM and T2LSTM were tested in the identification and control of nonlinear systems, in order to observe the advantages of the T2FS implementation.

II. T2LSTM STRUCTURE

The T2LSTM is obtained from the NARMA model of a nonlinear system:

$$\begin{aligned} y\left(k\right) &= \varphi\left[U_r\left(k\right)\right] \\ U_r\left(k\right) &= \left[\left(k-1\right), \cdots, y\left(k-n_y\right), u\left(k\right), \cdots \right. \\ &\quad \left. \cdots, u\left(k-n_u\right)\right]^T = \left[u_{r_1} \cdots u_{r_m}\right]^T \end{aligned} \tag{1}$$

where $y(k)$ is the variable of interest of the system under study (output signal), $\varphi\left(\cdot\right)$ is an unknown nonlinear difference equation, $U_r\left(k\right)$ is the state vector with $u\left(k\right)$ and $y\left(k\right)$ with the former as the input signal for the system; n_y indicates the number of the delayed output signal, n_u indicates the number of the delayed input signal, and m indicates the number of elements u_{r_m} in $U_r\left(k\right)$.

To model (1), we used fuzzy IF-THEN rules. For the p-th rule it has:

$$\begin{aligned} &R_p: \text{IF } u_{r_1}\left(k\right) \text{ IS } A_{1p} \ \& \ u_{r_2}\left(k\right) \text{ IS } A_{2p} \ \& \ \cdots \\ &\cdots \ \& \ u_{r_m}\left(k\right) \text{ IS } A_{jp}, \text{ THEN } h_p\left(k\right) = \varrho_p\left(k\right) \end{aligned} \tag{2}$$

where $h_p\left(k\right)$ is an estimation to the function $\varrho_p(k)$ that represents the consequent part of each fuzzy rule. The sets A_{jp}, with $j = 1 \dots \kappa$, are the fuzzy sets for the fuzzification (using κ fuzzy sets) of each u_{r_m} in (2).

The T2MFs associated with each A_{jp} are of the form shown in Fig. 1 and are described as follows, first the UMF associate to each A_{jp}:

$$\overline{\mu}_{A_{jp},u_{r_m}}\left(k\right) = \exp\left(-\frac{\left(u_{r_m}\left(k\right)-\varsigma_{jp}\right)^2}{2\overline{\nu}_{jp}}\right) \tag{3}$$

and the LMF:

$$\underline{\mu}_{A_{jp},u_{r_m}}\left(k\right) = \exp\left(-\frac{\left(u_{r_m}\left(k\right)-\varsigma_{jp}\right)^2}{2\underline{\nu}_{jp}}\right) \tag{4}$$

in these Gaussian functions, the center is $\varsigma_{jp} \in \mathbb{R}$ and the widths are $\overline{\nu}_{jp}, \underline{\nu}_{jp} \in \mathbb{R}^+$. For the estimation of (1), the

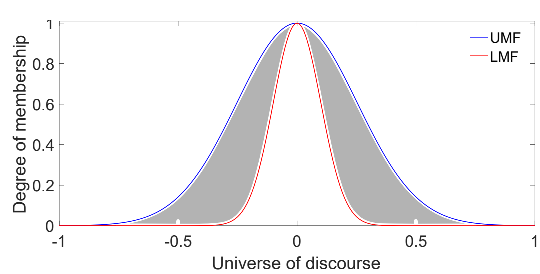

Fig. 1: Type-2 Gaussian MF, bounded by UMF and LMF, with uncertain standard deviation.

contribution of each input element to the premise part of a fuzzy rule in (2) is obtained by the T-norm:

$$\overline{z}_p(k) = \prod_{j=1}^{\kappa} \overline{\mu}_{A_{jp},u_{r_m}}(k)$$
$$\underline{z}_p(k) = \prod_{j=1}^{\kappa} \underline{\mu}_{A_{jp},u_{r_m}}(k)$$

assuming $j = m$.

The vectorial representation of the value of each element of (3) and (4) in each fuzzy set is:

$$\overline{\zeta}_j = \exp\left[(U_r(k) - \chi_j)^2 \otimes \left(-\frac{1}{2}\overline{\Upsilon}_j\right)\right] \tag{5}$$
$$\underline{\zeta}_j = \exp\left[(U_r(k) - \chi_j)^2 \otimes \left(-\frac{1}{2}\underline{\Upsilon}_j\right)\right] \tag{6}$$

with $\chi_j, \overline{\Upsilon}_j \underline{\Upsilon}_j \in \mathbb{R}^m$ as the center and width vectors for $\overline{\zeta}_j, \underline{\zeta}_j \in \mathbb{R}^m$, respectively.

The vectors $\overline{\zeta}_j, \underline{\zeta}_j$ represent the value of each element of $u_{r_m}(k)$ in a fuzzy set A_j, and $\otimes$ is the operator for the element to element product in vectors. This representation will be useful for the adjustment of the parameters of the FNN.

The estimation of (1) is obtained by the defuzzification of the FS (2) with p rules:

$$\begin{aligned}\hat{y}(k) &= \frac{\sum_{n=1}^{p} \overline{z}_n h_n(k)}{\sum_{n=1}^{p} \overline{z}_n} + (1-\beta)\frac{\sum_{n=1}^{p} \underline{z}_n h_n(k)}{\sum_{n=1}^{p} \underline{z}_n} \\ &= \sum_{n=1}^{p} \overline{\hat{z}}_n h_n(k) + (1-\beta)\sum_{n=1}^{p} \underline{\hat{z}}_n h_n(k)\end{aligned} \tag{7}$$

where:

$$\overline{\hat{z}}_p = \overline{z}_p / (\overline{z}_1 + \overline{z}_2 + \cdots + \overline{z}_n)$$
$$\underline{\hat{z}}_p = \underline{z}_p / (\underline{z}_1 + \underline{z}_2 + \cdots + \underline{z}_n)$$

In (7), the premise part can be represented in a vectorial way as $\overline{Z}_F, \underline{Z}_F \in \mathbb{R}^p$, where all the elements of this new vector are the organized multiplications as was explained in (3) and (4). Also, each element of $\overline{Z}_F$ and $\underline{Z}_F$ is normalized. Acording with the theory of T2FSs, the design parameter, β, weights the sharing of the lower and the upper firing levels of each fired rule, this parameter can be a constant and in this paper we take $\beta = 0.5$.

For multiple estimations, the elements of $\overline{Z}_F$ and $\underline{Z}_F$ can be organized in such a way that the premise parts repeats for every estimation, hence the consequent parts are the only ones that are different for several estimation in a same system.

The $h_n(k)$ elements in (7) are calculate by LSTM cells. The cells that we used in the T2LSTM have several parts, which are describe as follows:

$$F(k) = \sigma\left(W^f U_r(k) + V^f H(k-1)\right) \tag{8}$$
$$I(k) = \sigma\left(W^i U_r(k) + V^i H(k-1)\right) \tag{9}$$
$$S(k) = \psi\left(W^s U_r(k) + V^s H(k-1)\right) \tag{10}$$
$$C(k) = F(k) \otimes C(k-1) + I(k) \otimes S(k) \tag{11}$$
$$O(k) = \sigma\left(W^o U_r(k) + V^o H(k-1)\right) \tag{12}$$
$$H(k) = O(k) \otimes \psi(C(k)) \tag{13}$$

where $F(k)$, $I(k)$, $S(k)$, $C(k)$, $O(k)$ and $H(k) \in \mathbb{R}^p$ are the fitness of the internal state, the fitness of the internal input, the internal input, the internal state, the fitness of the output, and he output of the cells, respectively. The synaptic weights are: W^f, W^i, W^s and $W^o \in \mathbb{R}^{p\times m}$; V^f, V^i, V^s and $V^o \in \mathbb{R}^{p\times p}$ as diagonal matrices or V^f, V^i, V^s and $V^o \in \mathbb{R}^p$ as vectors, according to the need. The functions $\sigma(\cdot)$ and $\psi(\cdot)$ are the sigmoid and hyperbolic tangent functions, respectively, $U_r(k) \in \mathbb{R}^m$ is the input in (1).

From (7), the output of the FS is

$$\hat{y}(k) = \left[\overline{Z}_F + (1-\beta)\underline{Z}_F\right] H(k) \tag{14}$$

where $H(k) = [h_1(k) \cdots h_p(k)]^T$ corresponds to the consequent parts and $\overline{Z}_F, \underline{Z}_F \in \mathbb{R}^n$ are the elements of the premise parts of the rules described by (2).

The cells, as well as the number of rules, are defined as $p = \kappa^m$ in the case of one estimation; for several estimations it has $p = l(\kappa^m)$ where l is the number of estimations, thus $\hat{y} \in \mathbb{R}^l$, as was described for (7). The complete structure of the T2LSTM is contained in (5)-(14).

By applying the function approximation theories of FSs, (1) can be represented as:

$$\begin{aligned}y(k) = &\left[\overline{Z}_F(W^*) + \cdots\right. \\ &\left.\cdots + (1-\beta)\underline{Z}_F(W^*)\right] H(W^*) + \mu(k)\end{aligned} \tag{15}$$

where W^* corresponds to the unknown weights which can minimize the unmodeled dynamic $\mu(k)$.

We assume that (1) is bounded-input and bounded-output (BIBO) stable, *i.e.*, $y(k)$ and $U_r(k)$ in (1) are bounded. By the bound of the membership functions (3) and (4), $\mu(k)$ in (15) is bounded.

Remark 1: It can be noted that the main difference between the T2LSTM and the T1LSTM lies in (14), since the latter equation is defined for a T1LSTM as follows:

$$\hat{y}(k) = Z_F H(k)$$

for the T1FS, the premise part of the fuzzy rules, Z_F, is simpler than in the T2FS case. This largely due to the use of T2MFs instead of T1MFs, which as explained above, offers higher robustness. But on the other hand, it increases the computational complexity as shown later in the paper.

III. Training of the T2LSTM

To train the FNN we employed a variation of the back propagation through time (BPTT) algorithm, this training can be used online and offline. We utilized a narrow "window" of data to apply the BPTT, which considers the values generated by the FNN in the current iteration and its immediately preceding iteration, values generated by the FNN in older iterations are forgotten. The algorithm is defined by:

$$W(k+1) = W(k) + \eta_W \Delta W(k) + \alpha_W \Delta W(k-1) \quad (16)$$

where W is any synaptic parameter array of the FNN, ΔW is the weight adjustment, $\eta_W \in (0,1]$ is the learning rate, $\alpha_W \in (0,1]$ is the momentum term for the training, and $\eta_W > \alpha_W$.

In (16), η_W determines the amount that each weight changes, while α_W helps to stabilize the modification considering the past weight adjustment.

Also, for the training is necessary to consider the modeling error between (15) and (14):

$$e(k) = \hat{y}(k) - y(k) \quad (17)$$

and:

$$\begin{aligned} \xi(k) &= \tfrac{1}{2} e^T(k) e(k) \\ E(k) &= \tfrac{1}{N} \textstyle\sum_{k=1}^{N} \xi(k) \end{aligned} \quad (18)$$

where $\xi(k)$ is the instant error energy, $E(k)$ is the total energy during the whole processes, and N is the total number of iterations.

The objective of the T2LSTM always will be $\min_{W(k)} \xi(k)$. Then, the adjustment of each element of ΔW is defined as follows:

$$\Delta w_{ij}(k) = \frac{\partial \xi(k)}{\partial w_{ij}(k)} \quad (19)$$

The modification (19) can be obtained by the application of the the chain rule and the signal flow of the FNN and it can be organized into an array like in (16). By the considerations made before, the adjustment of the parameters of the FNN described in (5)-(14) can be easy to obtain.

For example, if it is considered a FNN with $m = 1$, $l = 1$ and $\kappa > 1$, the gradient (19) for each element of W^i in the consequent part is:

$$\Delta w_p^i = \frac{\partial \xi(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial h_p(k)} \cdot \frac{\partial h_p(k)}{\partial \varepsilon_1} \cdots$$
$$\cdots \cdot \frac{\partial \varepsilon_1}{\partial c_p(k)} \cdot \frac{\partial c_p}{\partial i_p(k)} \cdot \frac{\partial i_p(k)}{\partial w_p^i(k)}$$

with $\varepsilon_1 = \psi(c_p(k))$. This in a vectorial form:

$$\begin{aligned} &\Delta W^i(k) = (\dot{\sigma}(W^i U_r(k) + V^i H(k-1)) \otimes D_i) U_r(k) \\ &D_i = S(k) \otimes \dot{\psi}(C(k)) \otimes \Big(\overline{Z}_F^T + \cdots \\ &\cdots + (1+\beta) \underline{Z}_F^T\Big) e(k) \otimes O(k) \end{aligned}$$

On the other hand, for the premise part, the fit of each element of χ_j in (5) and (6) is:

$$\Delta \chi_j = \frac{\partial \xi(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \overline{z}_{Fj}} \cdot \frac{\partial \overline{z}_{Fj}}{\partial \overline{\zeta}_j(k)} \cdot \frac{\partial \overline{\zeta}_j(k)}{\partial \chi_j}$$
$$+ \frac{\partial \xi(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \underline{z}_{Fj}} \cdot \frac{\partial \underline{z}_{Fj}}{\partial \underline{\zeta}_j(k)} \cdot \frac{\partial \underline{\zeta}_j(k)}{\partial \chi_j}$$

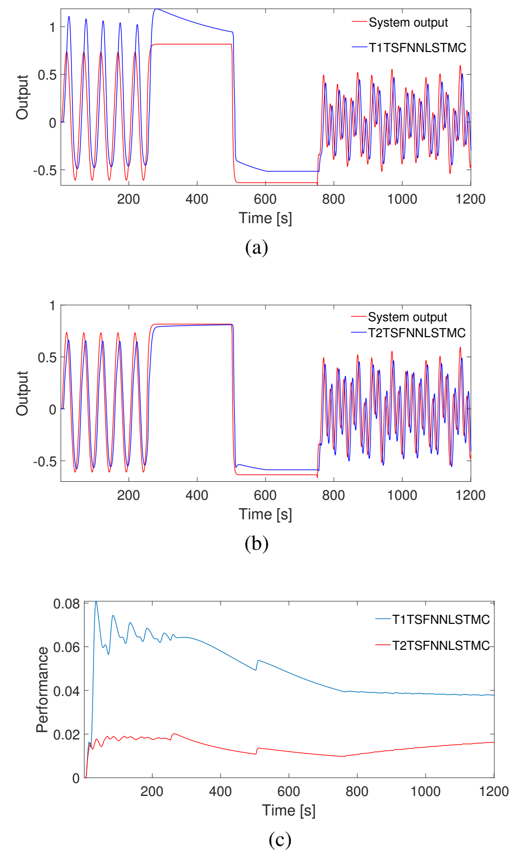

Fig. 2: Nonlinear system identification with tne FNNs: (a) T1TSFNNLSTMC, (b) T2TSFNNLSTMC, (c) Performance comparison.

and in a vectorial form:

$$\begin{aligned} &\Delta \chi_j = (U_r(k) - \chi_j) \otimes \overline{\Upsilon}_j \otimes \overline{D}_\chi \otimes e(k) H(k) + \cdots \\ &\cdots + \tfrac{1}{1+\beta} (U_r(k) - \chi_j) \otimes \underline{\Upsilon}_j \otimes \underline{D}_\chi \otimes e(k) H(k) \\ &\overline{D}_\chi = \exp\left[(U_r(k) - \chi_j)^2 \otimes \left(-\tfrac{1}{2} \overline{\Upsilon}_j\right) \right] \\ &\underline{D}_\chi = \exp\left[(U_r(k) - \chi_j)^2 \otimes \left(-\tfrac{1}{2} \underline{\Upsilon}_j\right) \right] \end{aligned}$$

Similar calculations can be done for the adjustment of the other parameters of the FNN.

IV. Comparisons

We employed two exercises to compare the new T2LSTM with a similar T1LSTM, taking into account the remark 1. We talk about the performance of the T2FS, the exercises exhibit the advantages of the T2FS over the T1FS. Both exercises were handled as applications in real time under the same conditions for the algorithms.

A. System identification

The first exercise consist on a model generation for a nonlinear system, which is defined as:

$$
\begin{aligned}
y(k+1) &= 0.72y(k) + 0.025y(k-1)u(k-1) + \\
&+ 0.01u^2(k-2) + 0.2u(k-3)
\end{aligned} \tag{20}
$$

$$
u(k) = \begin{cases} \sin\left(\dfrac{\pi k}{25}\right), \quad k \leq 250 \\ 1, \quad 250 < k \leq 500 \\ -1, \quad 500 < k \leq 750 \\ 0.3\sin\left(\dfrac{\pi k}{25}\right) + 0.1\sin\left(\dfrac{\pi k}{32}\right) + \\ +0.6\sin\left(\dfrac{\pi k}{10}\right), \quad 750 < k \end{cases}
$$

with $y(0) = 0$, $u(0) = 0$, and a sampling time of $T = 1$s.

So, (20) was solved for $1,200$s and the system was sampled, creating the vector $y(k)$ with $k = 1, \ldots, 1,201$. We took the values of $y(k)$ to define $U_r(k) = [y(k-4), y(-8)]^T$, which was used to make the $\hat{y}(k)$ estimate. We employed the first 601 iterations to train the FNNs, meanwhile the rest data were used for testing the algorithms.

We set $p = 9$ fuzzy rules for the FNNs ($m = 2$, $\kappa = 3$, $l = 1$). Here, the modeling error $E(k)$ at the end of each phase is defined as in (18) and it represents the performance of the algorithms, a low value indicates a better performance. We used (17) for the training of the FNNs because is the error that we want to minimize.

The comparison results are shown in the Table I and the Fig.2, the latter has three parts: (a) shows the system output and the output of the T1LSTM, (b) shows the system output and the output of the T2LSTM, and (c) shows a performance comparison between the T1LSTM and the T2LSTM.

It can be seen that both FNNs have similar behavior, however, the T2LSTM learned the system dynamics faster compared to the T1LSTM, which is why the former has a smaller modeling error or better performance during the whole process.

TABLE I: Performance of the FNNs during the identification ($\times 10^{-2}$)

System	Training	Testing
T1TSFNNLSTMC	4.80	3.78
T2TSFNNLSTMC	1.19	1.62

B. System control

For the second exercise we chose the control of a nonlinear system, which is defined as:

$$
y((k+1)T) = \frac{a(kT)y(kT)}{1 + y^2(kT)} + \theta(kT) + u(kT) \tag{21}
$$

$$
y(0) = 0, \quad u(kT) = r(kT) - \hat{y}(kT)
$$

$$
a(k) = \begin{cases} 1, \quad kT \leq 250 \\ \sin\left(\dfrac{kT}{10}\right), \quad 250 < kT \end{cases}
$$

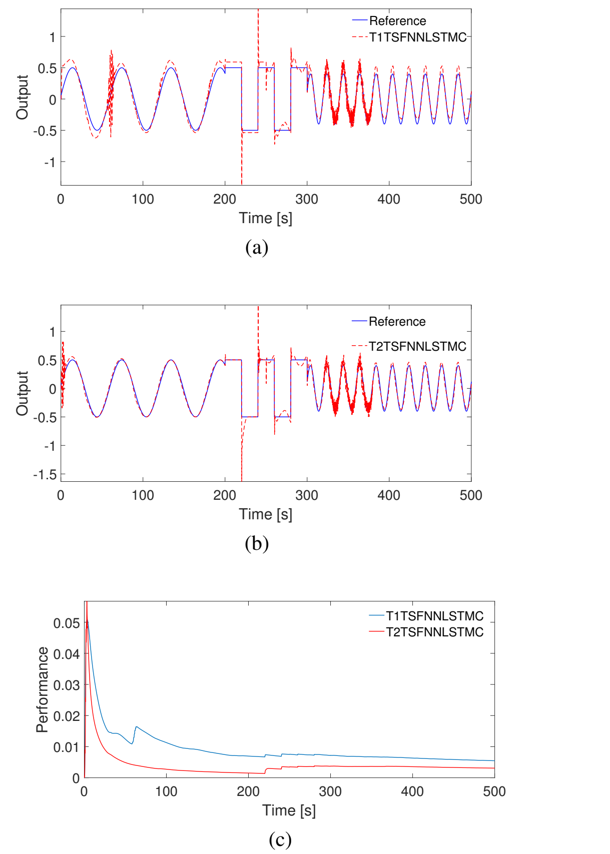

Fig. 3: Nonlinear system control with tne FNNs: (a) T1TSFNNLSTMC, (b) T2TSFNNLSTMC, (c) Performance comparison.

So, the task consists of generating a model $\hat{y}(kT)$ that compensates the system such that the output of the system $y(kT)$ is equal to the reference $r(kT)$. This reference is defined as follows:

$$
r(kT) = \begin{cases} 0.5\sin\left(\frac{2\pi kT}{60}\right), \quad kT \leq 200 \\ 0.5, \quad 200 < kT \leq 220 \\ -0.5, \quad 220 < kT \leq 240 \\ 0.5, \quad 240 < kT \leq 260 \\ -0.5, \quad 260 < kT \leq 280 \\ 0.5, \quad 280 < kT \leq 300 \\ 0.4\sin\left(\frac{2\pi kT}{20}\right), \quad kT > 300 \end{cases} \tag{22}
$$

Similar to the past exercise, (21) was solved for 500s and the system was sampled considering a sample period of time $T = 0.1$s, obtaining $k = 1, \ldots, 5,001$. The input vector for the FNNs was defined as $U_r(k) = [e_c(k-1)T, e_c(k-5)T]^T$, where:

$$
e_c(kT) = r(kT) - y(kT) \tag{23}
$$

To simulate perturbations in (21), random values in the range of $[-0.1, 0.1]$ were added at $320 < kT \leq 380$ to the process and these perturbations were represented by $\theta(k)$ in (21).

We established $p = 9$ fuzzy rules for the FNNs ($m = 2$, $\kappa = 3$, $l = 1$). The control error (23) is defined as in (18) and it represents the performance of the algorithms, a low value indicates a better performance. Also, we use (23) for the training of the FNNs because is the error that we want to minimize.

We work the system in the following way: we train the algorithms to compensate (21) with (23) during 100s, obtaining $1,001$ iterations for the training process, a testing is made immediately after the training with the same input signal during 400s (4,001 more iterations).

In the Table II are shown the control errors, according to (18), obtained for several events during the whole process. The Fig. 3 has three parts: (a) shows the system output and the output of the T1LSTM, (b) shows the system output and the output of the T2LSTM, and (c) shows a performance comparison between the T1LSTM and the T2LSTM.

It can be seen that both FNNs have similar behavior, buth the T2LSTM compensate the system faster than the T1LSTM, which is why the former has a better performance during the whole process. In addition, the T2LSTM allows (21) to have better tracking of (22), since this FS offer more robust results.

TABLE II: Performance of the FNNs during the control ($\times 10^{-2}$)

Event	T1TSFNNLSTMC	T2TSFNNLSTMC	
End of training	1.11	0.27	
Change in $a(k)$	0.75	0.34	
Start of $\theta(k)$	0.68	0.36	
End of $\theta(k)$	0.65	0.36	
End of the process	0.54	0.31	

As shown in the figures and tables above, the new T2LSTM offers very good results for the identification and control of nonlinear systems. In addition, it is more robust and adaptable than the previous T1LSTM.

V. Conclusions

In this paper, we use the advantages of type-2 fuzzy system and LSTM neural networks to design a novel model for nonlinear system modeling. The main advantage is this model can reduce the modeling error compared with the other fuzzy neural networks. However, the cost of the computational complexity of the algorithm is more than the others. This disadvantage can be reduced by current computational technology, which allows the easy application of this model.

References

[1] P. Singh, A brief review of modeling approaches based on fuzzy time series, International Journal of Machine Learning and Cybernetics, vol. 8, no. 2, pp. 397-420, Feb. 2015.

[2] J. Qiu, H. Gao, and S. X. Ding, Recent Advances on Fuzzy-Model-Based Nonlinear Networked Control Systems: A Survey, IEEE Transactions on Industrial Electronics, vol. 63, no. 2, pp. 1207-1217, Feb. 2016.

[3] Z. Pezeshki and S. M. Mazinani, Comparison of artificial neural networks, fuzzy logic and neuro fuzzy for predicting optimization of building thermal consumption: a survey, Artificial Intelligence Review, vol. 52, no. 1, pp. 495-525, Apr. 2018.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling , arXiv preprint arXiv:1412.3555, 2014.

[5] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, Recent advances in recurrent neural networks , arXiv preprint arXiv:1801.01078, 2017.

[6] S. Hochreiter and J. Schmidhuber, Long short-term memory , Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[7] O. Ogunmolu, X. Gu, S. Jiang, and N. Gans, Nonlinear systems identification using deep dynamic neural networks , arXiv preprint arXiv:1610.01439, 2016.

[8] Y. Wang, A new concept using lstm neural networks for dynamic system identification , in 2017 American Control Conference (ACC). IEEE, 2017, pp. 5324 -5329.

[9] F. Nicola, Y. Fujimoto, and R. Oboe, A lstm neural network applied to mobile robots path planning , in 2018 IEEE 16th International Conference on Industrial Informatics (INDIN). IEEE, 2018, pp.349 - 354.

[10] Y. Liu, Y. Zhou, and X. Li, Attitude estimation of unmanned aerial vehicle based on lstm neural network , in 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018, pp. 1-6.

[11] Chia-Feng Juang and Yu-Wei Tsao, A Self-Evolving Interval Type-2 Fuzzy Neural Network With Online Structure and Parameter Learning, EEE Transactions on Fuzzy Systems, vol. 16, no. 6, pp. 1411-1424, Dec. 2008.

[12] R. A. Aliev et al., Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization, nformation Sciences, vol. 181, no. 9, pp. 1591-1608, May 2011.

[13] J.-Y. Chang, Y.-Y. Lin, M.-F. Han, and C.-T. Lin, A functional-link based interval type-2 compensatory fuzzy neural network for nonlinear system modeling, 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Jun. 2011.

[14] Yang-Yin Lin, Jyh-Yeong Chang, N. R. Pal, and Chin-Teng Lin, A Mutually Recurrent Interval Type-2 Neural Fuzzy System (MRIT2NFS) With Self-Evolving Structure and Parameters, IEEE Transactions on Fuzzy Systems, vol. 21, no. 3, pp. 492-509, Jun. 2013.

[15] E. Kayacan, E. Kayacan, and M. Ahmadieh Khanesar, Identification of Nonlinear Dynamic Systems Using Type-2 Fuzzy Neural Networks: A Novel Learning Algorithm and a Comparative Study, IEEE Transactions on Industrial Electronics, vol. 62, no. 3, pp. 1716-1724, Mar. 2015.

[16] H.-G. Han, Z.-Y. Chen, H.-X. Liu, and J.-F. Qiao, A self-organizing interval Type-2 fuzzy-neural-network for modeling nonlinear systems, Neurocomputing, vol. 290, pp. 196-207, May 2018.

[17] W. Yuan and L. Chao, Online Evolving Interval Type-2 Intuitionistic Fuzzy LSTM-Neural Networks for Regression Problems, IEEE Access, vol. 7, pp. 35544-35555, 2019.

[18] M. A. Cunha Ferreira, R. Tanscheit, and M. Vellasco, Automatic Generation of a Type-2 Fuzzy System for Time Series Forecast based on Genetic Programming, Proceedings of the 2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT 2019), 2019.

[19] N. F. Rahim, M. Othman, R. Sokkalingam, and E. Abdul Kadir, Type 2 Fuzzy Inference-Based Time Series Model, Symmetry, vol. 11, no. 11, p. 1340, Oct. 2019.

[20] M. Bilgili, A. Yildirim, A. Ozbek, K. Celebi, and F. Ekinci, Long short-term memory (LSTM) neural network and adaptive neuro-fuzzy inference system (ANFIS) approach in modeling renewable electricity generation forecasting, International Journal of Green Energy, vol. 18, no. 6, pp. 578-594, Dec. 2020.

[21] J. Tavoosi, A. A. Suratgar, M. B. Menhaj, A. Mosavi, A. Mohammadzadeh, and E. Ranjbar, Modeling Renewable Energy Systems by a Self-Evolving Nonlinear Consequent Part Recurrent Type-2 Fuzzy System for Power Prediction, Sustainability, vol. 13, no. 6, p. 3301, Mar. 2021.

[22] A. Safari, R. Hosseini, and M. Mazinani, A novel deep interval type-2 fuzzy LSTM (DIT2FLSTM) model applied to COVID-19 pandemic time-series prediction, Journal of Biomedical Informatics, vol. 123, p. 103920, Nov. 2021.

[23] J. Tavoosi, A. Mohammadzadeh, and K. Jermsittiparsert, A review on type-2 fuzzy neural networks for system identification, Soft Computing, vol. 25, no. 10, pp. 7197-7212, Mar. 2021.

[24] W. Yu and F. Vega, Nonlinear system modeling using the takagi-sugeno fuzzy model and long-short term memory cells, Journal of Intelligent and Fuzzy Systems, vol. 39, no. 3, pp. 4547-4556, Oct. 2020.

[25] F.Vega, W.Yu, Fuzzy Modeling Using LSTM Cells for Nonlinear Systems, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, UK, July 19-24, 2020