Identification of Nonlinear Dynamic Systems Using Type-2 Fuzzy Neural Networks - A Novel Learning Algorithm and a Comparative Study

Erkan Kayacan, *Student Member, IEEE,* Erdal Kayacan, *Senior Member, IEEE* and Mojtaba Ahmadieh Khanesar, *Member, IEEE*

Abstract—In order to achieve faster and more robust convergence (especially under noisy working environments), a sliding mode theory-based learning algorithm has been proposed to tune both the premise and consequent parts of type-2 fuzzy neural networks in this paper. Differently from recent studies, where sliding mode control theory-based rules are proposed for only the consequent part of the network, the developed algorithm applies fully sliding mode parameter update rules for both the premise and consequent parts of the type-2 fuzzy neural networks. In addition, the responsible parameter for sharing the contributions of the lower and upper parts of the type-2 fuzzy membership functions is also tuned. Moreover, the learning rate of the network is updated during the online training. The stability of the proposed learning algorithm has been proved by using an appropriate Lyapunov function. Several comparisons have been realized and shown that the proposed algorithm has faster convergence speed than the existing methods such as gradient-based and swarm intelligence-based methods. Moreover, the proposed learning algorithm has a closed form, and it is easier to implement than the other existing methods.

Index Terms—Type-2 fuzzy neural networks, type-2 fuzzy logic systems, sliding mode learning algorithm, system identification.

I. INTRODUCTION

INSPIRED by the central nervous system of human, artificial neural networks (ANNs) are widely known computational tools with their representation capability, even in the case of highly nonlinear and complex structured systems. Fuzzy neural networks (FNNs) combine the capability of fuzzy reasoning to handle uncertain information and the capability of ANNs to learn from input-output data sets in modeling nonlinear dynamic systems. The FNNs have been used in many engineering areas and obtained successful results, such as the identification and control of dynamic systems [1], [2], temperature control [3], classification [4], energy conversion [5]. Another prominent feature of FNNs is that they have also been proven to be universal approximators similar to ANNs [6].

There are two different approaches to fuzzy logic system (FLS) design: Type-1 FLSs (T1FLSs) and type-2 FLSs (T2FLSs). The latter is proposed as an extension of the former with the intention of being able to model the uncertainties that invariably exist in the rule base of the system [7]. Whereas membership functions (MFs) are totally certain in type-1 fuzzy sets, they are themselves fuzzy in type-2 fuzzy sets. The latter case results in a fact that the antecedent and consequent parts of the rules are uncertain. As there are infinite type-1 fuzzy MFs in the footprint of uncertainty of a type-2 fuzzy MF, it is believed that the T2FLSs have the ability of modeling uncertainties in the rule base better than their type-1 counterparts. Therefore, T2FLSs appear to be a more promising method than their type-1 counterparts for handling uncertainties such as noisy data and variable working conditions both in modeling and control purposes [8]–[10].

The gradient descent (GD) algorithm is a well-known optimization method to tune the parameters of ANNs and FNNs. However, since the gradient-based algorithms (e.g. dynamic back propagation) include partial derivatives, the convergence speed may be slow especially when the search space is complex. What is more, with the repetitive algorithms, a number of numerical robustness issues may emerge when they are applied over long periods of time [11]. The selection of the learning rate in a GD algorithm is also challenging, because a large value for this parameter may result in instability in the system while a small value may increase the probability of entrapment in a local minima and lessen the speed of convergence. Especially, in FNNs, another issue for a GD algorithm is that obtaining the parameter update rules for the premise part of the network is very complex. In addition to these drawbacks, the tuning process can easily be trapped into a local minimum [12]. For a noisy input-output data set, the performance of gradient-based algorithms might be even worse. In order to improve the performance of these algorithms, several modifications are proposed such as the introduction of momentum term, adaptive learning rate and modification of the traditional GD method by using Levenberg-Marquardt (LM) algorithm [13]. As an alternative to GD methods, the use of evolutionary approaches have been suggested [14], [15]. However, the stability of such approaches is questionable and the optimal values for the stochastic operators are difficult to derive. Furthermore, the

Erkan Kayacan is with the Division of Mechatronics, Biostatistics and Sensors, Department of Biosystems, University of Leuven (KU Leuven), Kasteelpark Arenberg 30, B-3001 Leuven, Belgium. e-mail: erkan.kayacan@biw.kuleuven.be

Erdal Kayacan is with the School of Mechanical & Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. e-mail: erdal@ntu.edu.sg

M.A. Khanesar is with the Department of Electrical and Control Engineering, Semnan University, Semnan, Iran. e-mail: ahmadieh@semnan.ac.ir

computational burden can be very high. To overcome these issues, sliding mode control (SMC) theory-based algorithms are proposed for the parameter update rules of ANNs and type-1 FNNs (T1FNNs) as robust learning algorithms [16], [17]. SMC theory-based learning algorithms cannot only make the overall system more robust but also ensure faster convergence than the traditional learning techniques in online tuning of ANNs and FNNs [18], [19]. Moreover, the parameter update rules are much simpler when compared to other algorithms, such as GD methods. Motivated by the successful results of these learning algorithms on T1FNNs, the derivation of SMC theory-based learning algorithms for the training of type-2 FNNs (T2FNNs) are also proposed [20], [21].

In order to achieve faster and more robust convergence (especially under noisy working environments), an SMC theory-based learning algorithm has been proposed to tune both the premise and consequent parts of T2FNNs in this paper. Since the proposed learning algorithm does not include any partial derivatives or computationally expensive mathematical operations, e.g. inverse of matrices etc., it is believed that the proposed algorithm is more applicable especially for real-time systems.

In this paper, the major contributions to the T2FNNs are as follows: The first is the proposal of fully SMC theory-based learning rules. It is to be noted that all similar studies in literature consider SMC theory-based rules for only limited number of parameters. For instance, the parameter update rules for the center values of the type-2 fuzzy MFs in [20] do not have SMC theory-based rules. The second contribution of this paper is that the proposed algorithm tunes the sharing of the lower and upper MFs in a T2FNN which allows us to manage non-uform uncertainties in the rule base of T2FLSs. As the third contribution, the learning rate is also updated over the simulations.

The body of the paper contains five sections: In Section II, the theoretical basics of T2FLSs are given. The novel SMC theory-based learning rules for the type-2 Gaussian MFs are introduced in Section III. In Section IV, the validation of the proposed novel parameter update rules for the identification of three nonlinear dynamic systems are given. The comparison of different learning techniques is given in Section V. Finally, some conclusions are drawn in Section VI.

II. TYPE-2 FUZZY LOGIC SYSTEMS (T2FLSS)

A. T2FLSs Overview

A first-order interval type-2 Takagi-Sugeno-Kang (TSK) fuzzy *if-then* rule base with r input variables is preferred in this investigation. Where the consequent parts are crisp numbers, the premise parts are type-2 fuzzy functions. The rth rule is as follows:

R_r : If x_1 is $\widetilde{A}_{1j}$ ... and x_i is $\widetilde{A}_{ik}$ and... and x_I is $\widetilde{A}_{Il}$ then

$$f_r = \sum_{i=1}^{I} a_{ri} x_i + b_r \qquad (1)$$

where $x_i (i = 1...I)$ are the inputs of the type-2 TSK model, $\widetilde{A}_{ik}$ is the kth type-2 fuzzy MF $(k = 1...K)$ corresponding to the input i^{th} variable, K is the number of MFs for the ith input which can be different for each input. The parameters a_r and b_r stand for the consequent part and $f_r (r = 1...N)$ is the output function.

The upper and lower type-2 fuzzy Gaussian MFs with an uncertain standard deviation (Fig. 1) can be represented as follows:

$$\overline{\mu}_{ik}(x_i) = exp\left(-\frac{1}{2} \frac{(x_i - c_{ik})^2}{\overline{\sigma}_{ik}^2} \right) \qquad (2)$$

$$\underline{\mu}_{ik}(x_i) = exp\left(-\frac{1}{2} \frac{(x_i - c_{ik})^2}{\underline{\sigma}_{ik}^2} \right) \qquad (3)$$

where c_{ik} is the center value of the kth type-2 fuzzy set for the ith input. The parameters $\overline{\sigma}_{ik}$ and $\underline{\sigma}_{ik}$ are standard deviations for the upper and lower MFs.
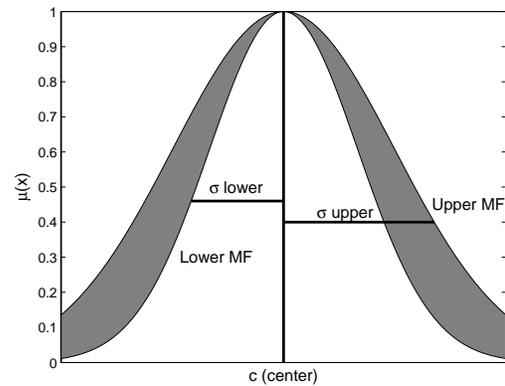

Fig. 1: Type-2 Gaussian fuzzy MF with uncertain standard deviation

B. Interval Type-2 A2-CO TSK Model

The structure used in this investigation is called A2-C0 fuzzy system [22]. In such a structure, first, the lower and upper membership degrees $\underline{\mu}$ and $\overline{\mu}$ are determined for each input signal being fed to the system. Next, the firing strengths of the rules using the *prod* t-norm operator are calculated as follows:

$$\underline{w}_r = \underline{\mu}_{\tilde{A}1}(x_1) * \underline{\mu}_{\tilde{A}2}(x_2) * \cdots \underline{\mu}_{\tilde{A}I}(x_I)$$
$$\overline{w}_r = \overline{\mu}_{\tilde{A}1}(x_1) * \overline{\mu}_{\tilde{A}2}(x_2) * \cdots \overline{\mu}_{\tilde{A}I}(x_I) \qquad (4)$$

The consequent part corresponding to each fuzzy rule is a linear combination of the inputs $x_1, x_2 ... x_I$. This linear function is called f_r and is defined as in (1). The output of the network is calculated as follows:

$$y_N = q \sum_{r=1}^{N} f_r \widetilde{\underline{w}}_r + (1-q) \sum_{r=1}^{N} f_r \widetilde{\overline{w}}_r \qquad (5)$$

where $\widetilde{\underline{w}_r}$ and $\widetilde{\overline{w}_r}$ are the normalized values of the lower and the upper output signals from the second hidden layer of the network as follows:

$$\widetilde{\underline{w}_r} = \frac{\underline{w}_r}{\sum_{i=1}^{N} \underline{w}_r} \text{ and } \widetilde{\overline{w}_r} = \frac{\overline{w}_r}{\sum_{i=1}^{N} \overline{w}_r} \qquad (6)$$

The design parameter, q, weights the sharing of the lower and the upper firing levels of each fired rule. This parameter

can be a constant (equal to 0.5 in most cases) or a time varying parameter. In this investigation, the latter is preferred. In other words, the parameter update rules and the proof of the stability of the learning process are given for the case of a time varying q.

The following vectors can be specified:
$\widetilde{\underline{W}}(t) = \left[\widetilde{\underline{w_1}}(t) \ \widetilde{\underline{w_2}}(t) ... \ \widetilde{\underline{w_N}}(t) \right]^T$,
$\widetilde{\overline{W}}(t) = \left[\widetilde{\overline{w_1}}(t) \ \widetilde{\overline{w_2}}(t) ... \ \widetilde{\overline{w_N}}(t) \right]^T$ and $F = [f_1 \ f_2 ... f_N]$

The following assumptions have been used in this investigation: The time derivative of both the input signals and output signal can be considered bounded:

$$|\dot{x}_i(t)| \leq B_{\dot{x}}, \ min(x_i^2(t)) = B_{x^2}, \ (i = 1 ... I) \ and \ |\dot{y}(t)| \leq B_{\dot{y}} \ \forall t \tag{7}$$

where $B_{\dot{x}}$, B_{x^2} and $B_{\dot{y}}$ are assumed to be some known positive constants.

III. SLIDING MODE CONTROL THEORY-BASED LEARNING ALGORITHM

The zero value of the learning error coordinate can be defined as a time-varying sliding surface in (8). The condition defined in (8) guarantees that when the system is on the sliding surface, the output of the network, $y_N(t)$, will perfectly follow the desired output signal, $y(t)$, for all time $t > t_h$. The time instant t_h is defined to be the hitting time for being $e(t) = 0$.

$$S(e(t)) = e(t) = y_N(t) - y(t) = 0 \tag{8}$$

Definition: A sliding motion will appear on the sliding manifold $S(e(t)) = e(t) = 0$ after a time t_h, if the condition $S(t)\dot{S}(t) < 0$ is satisfied for all t in some nontrivial semi-open subinterval of time of the form $[t, t_h) \subset (0, t_h)$.

It is desired to devise a dynamical feedback adaptation mechanism, or an online learning algorithm for the parameters of the T2FNN considered, such that the sliding mode condition of the above definition is enforced.

A. The Proposed Parameter Update Rules for the T2FNN

The parameter update rules for the T2FNN proposed in this paper are given by the following theorem.

Theorem 1: If the adaptation laws for the parameters of the considered T2FNN are chosen as:

$$\dot{c}_{ik} = \dot{x}_i + (x_i - c_{ik})\alpha_1 sgn(e) \tag{9}$$

$$\dot{\underline{\sigma}}_{ik} = -\left(\underline{\sigma}_{ik} + \frac{(\underline{\sigma}_{ik})^3}{(x_i - c_{ik})^2} \right) \alpha_1 sgn(e) \tag{10}$$

$$\dot{\overline{\sigma}}_{ik} = -\left(\overline{\sigma}_{ik} + \frac{(\overline{\sigma}_{ik})^3}{(x_i - c_{ik})^2} \right) \alpha_1 sgn(e) \tag{11}$$

$$\dot{a}_{ri} = -x_i \frac{q\widetilde{\underline{w}}_r + (1-q)\widetilde{\overline{w}}_r}{(q\widetilde{\underline{w}}_r + (1-q)\widetilde{\overline{w}}_r)^T(q\widetilde{\underline{w}}_r + (1-q)\widetilde{\overline{w}}_r)} \alpha sgn(e) \tag{12}$$

$$\dot{b}_r = -\frac{q\widetilde{\underline{w}}_r + (1-q)\widetilde{\overline{w}}_r}{(q\widetilde{\underline{w}}_r + (1-q)\widetilde{\overline{w}}_r)^T(q\widetilde{\underline{w}}_r + (1-q)\widetilde{\overline{w}}_r)} \alpha sgn(e) \tag{13}$$

$$\dot{q} = -\frac{1}{F(\widetilde{\underline{W}} - \widetilde{\overline{W}})^T} \alpha sgn(e) \tag{14}$$

where α is an adaptive learning rate with an adaptation law as follows:

$$\dot{\alpha} = \gamma(I+2)|e| - \nu\gamma\alpha, \ 0 < \gamma, \nu \tag{15}$$

Then, given an arbitrary initial condition $e(0)$, the learning error $e(t)$ will converge to zero within a finite time t_h.

Proof: The reader is referred to Appendix A.

Remark 1. It is to be noted that in (15), the parameter γ has a small positive real value which is interpreted as the learning rate for the adaptive learning rate. Moreover, the first term of the adaptation law of (15) is always positive which may result in unnecessarily large value for α. In order to avoid bursting in the parameter α, the second term introduces a reset mechanism which avoids a possible parameter bursting in α. The value of ν should be selected very small to avoid it from interrupting the adaptation mechanism.

Remark 2. As can be seen from (15), the learning rate considered in this paper is itself adaptive and does not need to be known as a priori. This is an obvious superiority of the current approach with respect to previous approaches in which the upper bounds of the states of the system should be known as a priori in order to choose an appropriate value for the learning rate.

In order to avoid division by zero in the adaptation laws of (9)-(14) an instruction is included in the algorithm to make the denominator equal to 0.001 when its calculated value is smaller than this threshold.

It is well-known that sliding mode control suffers from high-frequency oscillations in the control input, which are called *chattering*. The following are the two common methods used to eliminate chattering [23]:

1) Using a saturation function to replace the signum function.
2) Inserting a boundary layer so that an equivalent control replaces the corrective one when the system is inside this layer.

In order to reduce the chattering effect, the following function is used in this paper with $\delta_s = 0.05$ instead of the signum function in the dynamic strategy:

$$sgn(e) := \frac{e}{|e| + \delta_s} \tag{16}$$

Remark 3. Since the output of the T2FNN is quite sensitive to the changes in the parameters of the antecedent parts, different values for the learning rates of the antecedent and consequent part parameters are used. In other words, a smaller value (α_1) is chosen for the antecedent parts.

IV. SIMULATION STUDIES

The identification problem involves the finding of the input-output relations of the system. In Fig. 2 the structure of the identification scheme is shown. The inputs to the T2FNN based identifier are the external input signals, its one-, two, ..., d_i- step delayed values and the one-, two-, ..., d_o- step delayed outputs of the plant. The problem is defined as to find such values of the parameters of the T2FNN 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 $u(k)$.
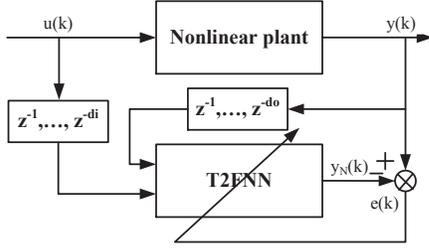
Fig. 2: Identification scheme

The number of the parameters to be updated in the T2FNN structure is the summation of the following parameters:

- The number of the parameters for the premise parts of the rules (the center and sigma values of the Gaussian MFs)
- The number of the parameters for the consequent parts of the rules (a_{ri} and b_r matrices in (1))
- The parameter q (in (5))
- The learning rate α

In order to evaluate the identification performance of the proposed learning algorithm, number of simulation studies are carried out. The systems are taken from literature in order to be able to make a fair comparison with the existing algorithms.

As a performance criterion, the root-mean-square-error (RMSE) given in (17) is used:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{K}(y(k) - y_N(k))^2}{K}} \tag{17}$$

where K is the number of samples.

The identification performance of the proposed learning algorithm has been compared with GD, particle swarm optimization (PSO), SMC theory-based online learning for T1FNNs, and the extended sliding mode on-line algorithm for T2FNN presented in [24]. It is to be noted that even if the network structure is the same with the one in [24], the proposed learning rules in this investigation are completely novel and fully sliding mode. In all the examples in this section, the network is designed with three inputs and one output. The inputs are the input signal to the plant, the two delayed signals from the plant output with a discretization period T_o of 1ms. Each input is fuzzified by using three Gaussian type-2 fuzzy MFs with a fixed center and uncertain standard deviation. To be able to make a fair comparison, each experiment has been realized for ten times with a random initialization of the network parameters, and the average numbers are given in the paper.

A. Example 1: Identification of a non-BIBO nonlinear plant

The proposed identification procedure is applied to a non-bounded-input-bounded-output (non-BIBO) nonlinear plant model [25] described by the following equation:

$$
\begin{aligned}
y(k+1) &= 0.2y^2(k) + 0.2y(k-1) \\
&+ 0.4sin\Big(0.5\big(y(k) + y(k-1)\big)\Big) \\
&\times cos\Big(0.5\big(y(k) + y(k-1)\big)\Big) \\
&+ 1.2u(k) \tag{18}
\end{aligned}
$$

The plant output may diverge for a sequence of uniformly bounded input signals. For instance, when a step input $u(k) = 0.83$ is applied to the system, the output of the system diverges. On the other hand, when a step input $u(k) < 0.83$ is applied to the system, the possible maximum output of the system is approximately equal to 2.26. Thus, the input signal in this investigation has the following form [24]:

$$u(k) = 0.5e^{-0.1kT_o}sin(5kT_o) \tag{19}$$

Whereas Fig. 3(a) demonstrates the output of the model and the real-time system, Fig. 3(b) shows the RMSE values versus epoch number which indicates a stable learning with the proposed learning algorithm. As can be seen from Fig. 3(a), the T2FNN gives accurate modeling results. Thanks to the novel fully sliding mode parameter update rules in this paper, the presented results are significantly better when compared to the ones in [24]. In Fig. 3(c), the adaptation of the learning rate is presented. Thanks to the reset mechanism presented in (15) the learning rate does not go to infinity; it converges to an appropriate value. In Fig. 3(d), the adaptation of the parameter q is presented which is also learnt by the proposed algorithm. By doing so, the contributions of the upper and lower MFs are also tuned during the simulations.
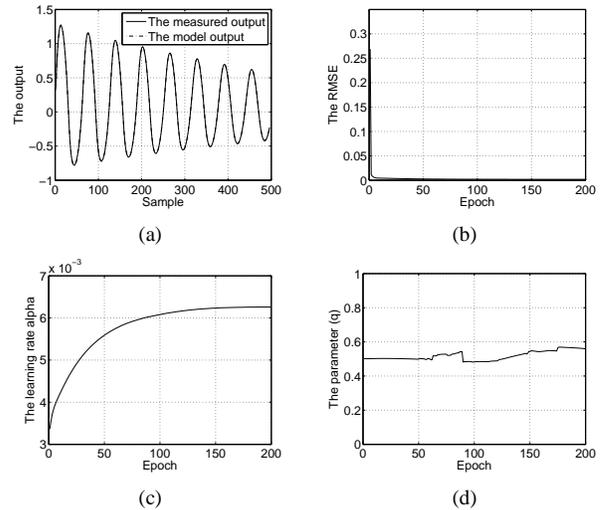

Fig. 3: The output of the model and the T2FNN system (a) RMSE versus epoch number (b) The adaptation of the learning rate (c) The adaptation of the parameter q (d)

B. Example 2: Identification of a second-order nonlinear time-varying plant

In the second example, the proposed identification procedure is applied to a second-order nonlinear time-varying plant

[26] described by the following equation:

$$y(k) = \frac{x_1 x_2 + x_3}{x_4} \qquad (20)$$

where $x_1 = y(k-1)y(k-2)y(k-3)u(k-1)$, $x_2 = y(k-3) - b(k)$, $x_3 = c(k)u(k)$ and $x_4 = a(k) + y(k-2)^2 + y(k-3)^2$.

The time-varying parameters a, b and c in (20) are given by the following equation:

$$\begin{aligned} a(k) &= 1.2 - 0.2cos(2\pi k/T) \\ b(k) &= 1 - 0.4sin(2\pi k/T) \\ c(k) &= 1 + 0.4sin(2\pi k/T) \end{aligned} \qquad (21)$$

where $T = 1000$ is the time span of the test.

There exist number of papers in the literature that claim that the performance of T2FNNs is better than their type-1 counterparts under uncertain working conditions and the claim is tried to be justified by simulation studies only for some specific systems. However, this claim is justified numerically in a general way in [27]. Since the system described in (20) is a nonlinear and time-varying system which has to operate under uncertain conditions. Therefore, this system is considered to be an appropriate system to show the performance of the T2FNN which is trained with the novel parameter update rules in this paper.

Similar to Fig. 3, whereas Fig. 4(a) demonstrates the output of the model and the real-time system, Fig. 4(b) shows the RMSE values versus epoch number which indicates a stable learning with the proposed learning algorithm. Thanks to the novel fully sliding mode parameter update rules in this paper, the presented results are quite similar to the ones in [26]. However, it is to be noted that the paper [26] uses more complex fuzzy logic system structure and more complex parameter adaptation rules when compared to the ones in this investigation. Moreover, the adaptation laws in [26] do not have closed (explicit) forms. In Fig. 4(c), the adaptation of the learning rate is presented. Thanks to the reset mechanism presented in (15) the learning rate does not go to infinity; it converges to an appropriate value. In Fig. 4(d), the adaptation of the parameter q is presented which is also tuned by the proposed algorithm. By doing so, the contributions of the upper and lower MFs are also tuned during the simulations.

V. ANALYSIS AND DISCUSSION

In Table I, the comparison of different learning techniques is given with respect to their identification performance and computation time. It can be seen that the identification performances of extended Kalman filter and the proposed SMC theory-based learning algorithm are similar to each other, and they seem to be the best when compared to other techniques. However, the computation time of the proposed SMC theory-based learning algorithm is significantly lower than the other methods. This conclusion results in a fact that the proposed method in this paper is more practical in real-time applications.

The reason for having the largest computation time for the extended Kalman filter is that this algorithm includes the manipulation of some high order matrices. A large amount of memory is used by these high dimensional matrices which
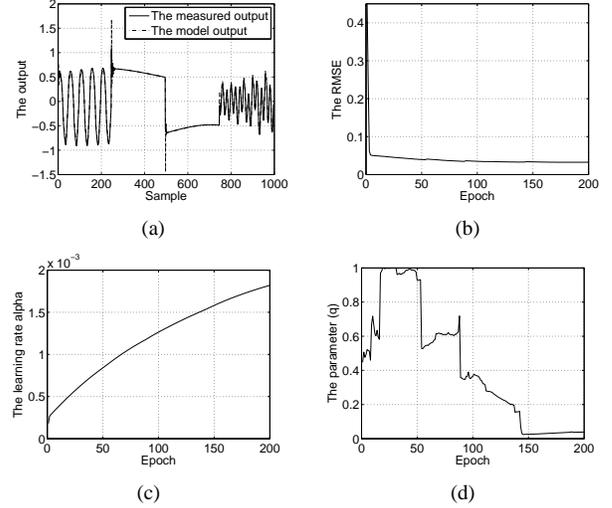

Fig. 4: The output of the model and the T2FNN system (a) RMSE versus epoch number (b) The adaptation of the learning rate (c) The adaptation of the parameter q (d)

makes the algorithm difficult to implement in most of the real time applications. On the other hand, there are no matrix manipulations in the proposed SMC theory-based rules. Moreover, GD, LM, EKF and other gradient-based training methods include the calculations of the partial derivatives of the output with respect to the parameters which is very difficult, and do not have any closed form. On the other hand, as can be seen from the adaptation laws proposed in this paper, the parameter update algorithm has closed form.

In order to have a better comparison, the proposed method is compared with an algorithm which is composed of particle swarm optimization (PSO) and GD. This algorithm has been previously used to train recurrent type-1 fuzzy neural networks [28] and interval T2FNNs [15]. The reader may refer to these references for a complete description about the training algorithm. The basic idea behind of this algorithm is that the parts which appear nonlinearly in the output can be trained by using PSO while the parameters which appear linearly in the output can be trained using GD. This is because its computational back-bone which makes it more logical than random optimization methods. This method is also implemented on A2-C0 fuzzy model which is considered in this paper and the results are summarized in Table I. As can be seen from the table, the results obtained using this method are quite close to those of SMC theory-based training algorithm while the computational time is much higher. The reason behind this computational burden is that the PSO+GD method necessitates too many feed-forward computation of T2FNN which is very complex and time consuming task.

One issue that should be taken into account is that the adaptation laws proposed in this paper are continuous. However, for the simulation of the method in a computer, an optimal sampling time should be chosen. The choice of the optimal sampling time may be a problem, because a very large value for the sampling time may cause instability in the system.

TABLE I: Comparison of different learning techniques

	Performance		
	Training	Testing	Computation time (s)
SMC-based learning	0.0342	0.0280	85.3026
GD	0.0636	0.0919	120.5814
LM	0.0463	0.0567	296.3332
EKF	0.0276	0.0266	218.7358
PSO+GD	0.0307	0.0525	7536.400

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

A novel fully sliding mode parameter update rules have been proposed for the training of interval T2FNNs for the identification of nonlinear and time-varying dynamic systems. The performance of the learning algorithm has been tested on the identification of two nonlinear systems which are taken from literature to be able to make a fair comparison. The simulation results indicate the potential of the proposed structure in real time systems since the computation time of the proposed algorithm is significantly lower than the other methods as well as it gives high identification accuracy. It is to be noted that these parameter update rules can also be used for the control purposes in which the computation time is prominent.

One issue which should be taken into account is that the adaptation laws proposed in this paper are continous. However, for the simulation of the method in a computer, an optimal sampling time should be chosen. The choice of the optimal sampling time may be a problem, because a very large value for the sampling time may cause an instability in the system.

B. Future Work

The proposed method with its stability analysis is valid for only the type-2 Gaussian membership functions with uncertain standard deviation. As a future work, the extension of the proposed method to other types of membership functions like Gaussian type-2 membership functions with uncertain centers, Elliptic type-2 MFs and so on are some interesting topics to be investigated.

APPENDIX A
PROOF OF THEOREM 1

The time derivative of (6) is calculated as follows:

$$\dot{\underline{\tilde{w}}}_r = -\underline{\tilde{w}}_r \underline{K}_r + \underline{\tilde{w}}_r \sum_{r=1}^{N} \underline{\tilde{w}}_r \underline{K}_r; \quad \dot{\overline{\tilde{w}}}_r = -\overline{\tilde{w}}_r \overline{K}_r + \overline{\tilde{w}}_r \sum_{r=1}^{N} \overline{\tilde{w}}_r \overline{K}_r \quad (22)$$

where

$$\underline{A}_{ik} = \frac{x_i - c_{ik}}{\underline{\sigma}_{ik}} \text{ and } \overline{A}_{ik} = \frac{x_i - c_{ik}}{\overline{\sigma}_{ik}}$$

$$\underline{K}_r = \sum_{i=1}^{I} \underline{A}_{ik} \dot{\underline{A}}_{ik} \text{ and } \overline{K}_r = \sum_{i=1}^{I} \overline{A}_{ik} \dot{\overline{A}}_{ik}$$

If (9)-(11) are inserted into the equations above, (23) can be obtained:

$$\overline{K}_r = \underline{K}_r = \sum_{i=1}^{I} \overline{A}_{ik} \dot{\overline{A}}_{ik} = \sum_{i=1}^{I} \underline{A}_{ik} \dot{\underline{A}}_{ik} = I\alpha \text{sgn}(e) \quad (23)$$

By using the following Lyapunov function, the stability condition is checked as follows:

$$V = \frac{1}{2}e^2 + \frac{1}{2\gamma}(\alpha - \alpha^*)^2, \ 0 < \gamma \quad (24)$$

The time derivative of (24) can be calculated as follows:

$$\dot{V} = \dot{e}e + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*) = e(\dot{y}_N - \dot{y}) + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*) \quad (25)$$

Differentiating (5), the following term can be obtained:

$$\dot{y}_N = \dot{q}\sum_{r=1}^{N} f_r \underline{\tilde{w}}_r + q\sum_{r=1}^{N}(\dot{f}_r \underline{\tilde{w}}_r + f_r \dot{\underline{\tilde{w}}}_r) - \dot{q}\sum_{r=1}^{N} f_r \overline{\tilde{w}}_r$$

$$+(1-q)\sum_{r=1}^{N}(\dot{f}_r \overline{\tilde{w}}_r + f_r \dot{\overline{\tilde{w}}}_r) \quad (26)$$

By using (22), (23) and (26), the following term can be obtained:

$$\dot{y}_N = \dot{q}\sum_{r=1}^{N} f_r \underline{\tilde{w}}_r + q\sum_{r=1}^{N}\left(\dot{f}_r \underline{\tilde{w}}_r + f_r(-\underline{\tilde{w}}_r \underline{K}_r + \underline{\tilde{w}}_r \sum_{r=1}^{N} \underline{\tilde{w}}_r \underline{K}_r)\right)$$

$$-\dot{q}\sum_{r=1}^{N} f_r \overline{\tilde{w}}_r$$

$$+(1-q)\sum_{r=1}^{N}\left(\dot{f}_r \overline{\tilde{w}}_r + f_r(-\overline{\tilde{w}}_r \overline{K}_r + \overline{\tilde{w}}_r \sum_{r=1}^{N} \overline{\tilde{w}}_r \overline{K}_r)\right)$$

$$= \dot{q}\sum_{r=1}^{N} f_r \underline{\tilde{w}}_r + q\sum_{r=1}^{N}\left(\dot{f}_r \underline{\tilde{w}}_r - I\alpha \text{sgn}(e) f_r(\underline{\tilde{w}}_r - \underline{\tilde{w}}_r \sum_{r=1}^{N} \underline{\tilde{w}}_r)\right)$$

$$-\dot{q}\sum_{r=1}^{N} f_r \overline{\tilde{w}}_r$$

$$+(1-q)\sum_{r=1}^{N}\left(\dot{f}_r \overline{\tilde{w}}_r - I\alpha \text{sgn}(e) f_r(\overline{\tilde{w}}_r - \overline{\tilde{w}}_r \sum_{r=1}^{N} \overline{\tilde{w}}_r)\right) (27)$$

The equation (28) is correct by definition:

$$\sum_{r=1}^{N} \underline{\tilde{w}}_r = 1 \text{ and } \sum_{r=1}^{N} \overline{\tilde{w}}_r = 1 \quad (28)$$

By using (12), (13), (14) and (28), the following function can be achieved:

$$\dot{y}_N = -\frac{1}{F(\underline{\tilde{W}} - \overline{\tilde{W}})^T}\alpha \text{sgn}(e)\sum_{r=1}^{N} f_r(\underline{\tilde{w}}_r - \overline{\tilde{w}}_r)$$

$$+\sum_{r=1}^{N} \dot{f}_r(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)$$

$$= -\alpha \text{sgn}(e)$$

$$+\sum_{r=1}^{N}\left[\left(\sum_{i=1}^{I}(\dot{a}_{ri}x_i + a_{ri}\dot{x}_i) + \dot{b}_r\right)\right.$$

$$\left.(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right] \quad (29)$$

If (29) is inserted into the candidate Lyapunov function, (30) can be obtained:

$$\dot{V} = \dot{e}e + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*) = e(\dot{y}_N - \dot{y}) + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$= e\left[-\alpha \text{sgn}(e) + \sum_{r=1}^{N}\left[\left(\sum_{i=1}^{I}(\dot{a}_{ri}x_i + a_{ri}\dot{x}_i) + \dot{b}_r\right)\right.\right.$$
$$\left.\left.(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right] - \dot{y}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

(30)

$$= e\left[-\alpha \text{sgn}(e) + \sum_{r=1}^{N}\left[\left(\sum_{i=1}^{I}\left(-(x_i\alpha \text{sgn}(e)\right.\right.\right.\right.$$
$$\left.\frac{(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)}{(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)^T(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)}\right)x_i + a_{ri}\dot{x}_i\right)$$
$$\left.-\alpha \text{sgn}(e)\frac{(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)}{(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)^T(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)}\right)$$
$$\left.(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right] - \dot{y}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$= e\left[-\alpha \text{sgn}(e)\right.$$
$$\left.+\sum_{r=1}^{N}\left[\sum_{i=1}^{I}\left(-\alpha \text{sgn}(e)x_i^2 + a_{ri}\dot{x}_i(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right)\right.\right.$$
$$\left.\left.-\alpha \text{sgn}(e)\right] - \dot{y}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$= e\left[-2\alpha \text{sgn}(e) + \sum_{r=1}^{N}\left[\sum_{i=1}^{I}\left(-\alpha \text{sgn}(e)x_i^2\right.\right.\right.$$
$$\left.\left.\left.+ a_{ri}\dot{x}_i(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right)\right] - \dot{y}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$= -|e|2\alpha + e\left[\sum_{r=1}^{N}\left[\sum_{i=1}^{I}\left(-\alpha \text{sgn}(e)x_i^2\right.\right.\right.$$
$$\left.\left.\left.+ a_{ri}\dot{x}_i(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right)\right] - \dot{y}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$\dot{V} < -|e|2\alpha + |e|\left[\sum_{r=1}^{N}\left[\sum_{i=1}^{I}\left(-\alpha B_{x^2}\right.\right.\right.$$
$$\left.\left.\left.+ B_aB_{\dot{x}}(q\underline{\tilde{w}}_r + (1-q)\overline{\tilde{w}}_r)\right)\right] + B_{\dot{y}}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$< -|e|2\alpha + |e|\left[\sum_{r=1}^{N}\left[-I\alpha B_{x^2} + IB_aB_{\dot{x}}(q\underline{\tilde{w}}_r\right.\right.$$
$$\left.\left.+ (1-q)\overline{\tilde{w}}_r)\right] + B_{\dot{y}}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$< -|e|2\alpha + |e|\left[-I\alpha B_{x^2} + IB_aB_{\dot{x}}(q\sum_{r=1}^{N}\underline{\tilde{w}}_r\right.$$
$$\left.+ (1-q)\sum_{r=1}^{N}\overline{\tilde{w}}_r) + B_{\dot{y}}\right] + \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$< -|e|2\alpha + |e|\left[-I\alpha B_{x^2} + IB_aB_{\dot{x}} + B_{\dot{y}}\right]$$
$$+ \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

$$< -|e|\left(2\alpha + I\alpha B_{x^2}\right) + |e|\left[IB_aB_{\dot{x}} + B_{\dot{y}}\right]$$
$$+ \frac{1}{\gamma}\dot{\alpha}(\alpha - \alpha^*)$$

where

$$\alpha^* \geq \frac{2(IB_aB_{\dot{x}} + B_{\dot{y}})}{2 + IB_{x^2}}$$

and α^* is considered to be an unknown parameter which is determined during the adaptation of the learning rate. This is an obvious superiority of the current approach over past approaches in which the upper bounds of the states of the system should be known a priori.

$$\dot{V} \leq -|e|(2\alpha + I\alpha B_{x^2}) + |e|(IB_aB_{\dot{x}} + B_{\dot{y}})$$
$$+ (2 + IB_{x^2})\alpha^*|e| - (2 + IB_{x^2})\alpha^*|e|$$
$$+ \frac{1}{\gamma}(\alpha - \alpha^*)\dot{\alpha} \quad (31)$$

$$\dot{V} \leq |e|(IB_aB_{\dot{x}} + B_{\dot{y}}) - (2 + IB_{x^2})\alpha^*|e|$$
$$- (2 + IB_{x^2})(\alpha^* - \alpha)|e| + \frac{1}{\gamma}(\alpha - \alpha^*)\dot{\alpha} \quad (32)$$

and further:

$$\dot{V} \leq |e|(IB_aB_{\dot{x}} + B_{\dot{y}}) - (2 + IB_{x^2})\alpha^*|e|$$
$$+ (\alpha^* - \alpha)\left[(2 + IB_{x^2})|e| - \frac{1}{\gamma}\dot{\alpha}\right] \quad (33)$$

using the adaptation law for the adaptive learning rate (α) as:

$$\dot{\alpha} = (2 + IB_{x^2})\gamma|e| - \nu\gamma\alpha \quad (34)$$

in which ν has a small real value. Using this adaptation law, the time derivative of the Lyapunov function can be rewritten as:

$$\dot{V} \leq |e|(IB_aB_{\dot{x}} + B_{\dot{y}}) \quad (35)$$
$$- (2 + IB_{x^2})\alpha^*|e| + (\alpha^* - \alpha)\nu\alpha$$

so that:

$$\dot{V} \leq |e|(IB_aB_{\dot{x}} + B_{\dot{y}}) \quad (36)$$
$$- (2 + IB_{x^2})\alpha^*|e| - \nu(\alpha - \frac{\alpha^*}{2})^2 + \frac{\nu\alpha^{*2}}{4}$$

considering the fact that $\alpha^* \geq \frac{2(IB_aB_{\dot{x}} + B_{\dot{y}})}{2 + IB_{x^2}}$ we have: $|e|$ $(IB_aB_{\dot{x}} + B_{\dot{y}}) - \frac{\alpha^*}{2}(2 + IB_{x^2})|e| \leq 0$ and consequently:

$$\dot{V} \leq -\frac{\alpha^*}{2}(2 + IB_{x^2})|e| + \frac{\nu\alpha^{*2}}{4} \quad (37)$$

So that error converges to a very small region around zero in which $|e| \leq \frac{\alpha^*\nu}{2(2 + IB_{x^2})}$ and it remains there. It should also be noted that ν is a small user defined positive number which can be selected as small as desired to make this neighborhood as narrow as requested by the user.

REFERENCES

[1] R.-J. Wai and P.-C. Chen, "Intelligent tracking control for robot manipulator including actuator dynamics via tsk-type fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 552–560, 2004.

[2] C.-H. Lu, "Wavelet fuzzy neural networks for identification and predictive control of dynamic systems," *IEEE Trans. Ind. Electron.*, vol. 58, no. 7, pp. 3046–3058, 2011.

[3] Y.-X. Liao, J. hua She, and M. Wu, "Integrated hybrid-pso and fuzzy-nn decoupling control for temperature of reheating furnace," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2704–2714, 2009.

[4] C.-T. Lin, C.-M. Yeh, S.-F. Liang, J.-F. Chung, and N. Kumar, "Support-vector-based fuzzy neural network for pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 31–41, 2006.

[5] I.-H. Li, W.-Y. Wang, S.-F. Su, and Y.-S. Lee, "A merged fuzzy neural network and its applications in battery state-of-charge estimation," *IEEE Trans. Energy Convers.*, vol. 22, no. 3, pp. 697–708, 2007.

[6] L. C.-T. and L. C. S. G., *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ: Prentice Hall, 1996.

[7] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic System: Introduction and New Directions*. Prentice Hall, Upper Saddle River, 2001.

[8] C.-F. Juang and C.-H. Hsu, "Reinforcement interval type-2 fuzzy controller design by online rule generation and q-value-aided ant colony optimization," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1528 –1542, dec. 2009.

[9] ——, "Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3931 –3940, oct. 2009.

[10] Y.-Y. Lin, J.-Y. Chang, and C.-T. Lin, "A tsk-type-based self-evolving compensatory interval type-2 fuzzy neural network (tscit2fnn) and its applications," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 447–459, Jan 2014.

[11] K. Astrom and B. Wittenmark, *Adaptive Control*. Addison-Wesley, 1995.

[12] A. Topalov and O. Kaynak, "Online learning in adaptive neurocontrol schemes with a sliding mode algorithm," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 445–450, december 2001.

[13] M. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Levenberg marquardt algorithm for the training of type-2 fuzzy neuro systems with a novel type-2 fuzzy membership function," in *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2011 IEEE Symposium on*, 2011, pp. 88–93.

[14] A. Topalov, K.-C. Kim, J.-H. Kim, and B.-K. Lee, "Fast genetic on-line learning algorithm for neural network and its application to temperature control," in *IEEE Int. Conf. Evolutionary Computation, Nagoya, Japan*, May 1996, pp. 649 –654.

[15] M. Khanesar, M. Teshnehlab, E. Kayacan, and O. Kaynak, "A novel type-2 fuzzy membership function: application to the prediction of noisy data," in *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2010 IEEE International Conference on*, 2010, pp. 128–133.

[16] G. Parma, B. Menezes, and A. Braga, "Sliding mode algorithm for training multilayer artificial neural networks," *Electronics Letters*, vol. 34, no. 1, pp. 97 –98, 1998.

[17] Y. Shuanghe, Y. Xinghuo, and M. Zhihong, "A fuzzy neural network approximator with fast terminal sliding mode and its applications," *Fuzzy Sets and Systems*, vol. 148, no. 3, pp. 469–486, 11 2004.

[18] G. Cascella, F. Cupertino, A. Topalov, O. Kaynak, and V. Giordano, "Adaptive control of electric drives using sliding-mode learning neural networks," in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, vol. 1, 2005, pp. 125 – 130.

[19] M. O. Efe, O. Kaynak, and X. Yu, "Sliding mode control of a three degrees of freedom anthropoid robot by driving the controller parameters to an equivalent regime," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 122, no. 4, pp. 632 –640, December 2000.

[20] E. Kayacan, O. Cigdem, and O. Kaynak, "Sliding mode control approach for online learning as applied to type-2 fuzzy neural networks and its experimental evaluation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 9, pp. 3510–3520, 2012.

[21] E. Kayacan and O. Kaynak, "Sliding mode control theory-based algorithm for online learning in type-2 fuzzy neural networks: application to velocity control of an electro hydraulic servo system," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 7, pp. 645–659, 2012.

[22] M. Begian, W. Melek, and J. Mendel, "Parametric design of stable type-2 tsk fuzzy systems," in *Fuzzy Information Processing Society, 2008.*

NAFIPS 2008. Annual Meeting of the North American, May 2008, pp. 1–6.

[23] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.

[24] S. Ahmed, N. Shakev, A. Topalov, K. Shiev, and O. Kaynak, "Sliding mode incremental learning algorithm for interval type-2 takagi-sugeno-kang fuzzy neural networks," *Evolving Systems*, vol. 3, no. 3, pp. 179 –188, September 2012.

[25] C.-C. Ku and K. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 144–156, 1995.

[26] R. Abiyev and O. Kaynak, "Type 2 fuzzy neural structure for identification and control of time-varying plants," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4147–4159, Dec 2010.

[27] M. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Analysis of the noise reduction property of type-2 fuzzy logic systems using a novel type-2 membership function," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 41, no. 5, pp. 1395–1406, Oct 2011.

[28] M. Khanesar, M. Shoorehdeli, and M. Teshnehlab, "Hybrid training of recurrent fuzzy neural network model," in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. IEEE, 2007, pp. 2598–2603.

Erkan Kayacan (S'12) was born in Istanbul, Turkey, on April 17, 1985. He received the B.Sc. and the M.Sc. degrees in mechanical engineering from Istanbul Technical University, Istanbul, in 2008 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Division of Mechatronics, Biostatistics and Sensors (MeBioS), KU Leuven, Leuven, Belgium.

He is a Research Assistant at the University of Leuven. His current research interests include model predictive control, moving horizon estimation, decentralized and distributed control, intelligent control and mechatronics.

Erdal Kayacan (S'06-SM'12) was born in Istanbul, Turkey on January 7, 1980. He received a B.Sc. degree in electrical engineering from in 2003 from Istanbul Technical University in Istanbul, Turkey as well as a M.Sc. degree in systems and control engineering in 2006 from Bogazici University in Istanbul, Turkey. In September 2011, he received a Ph.D. degree in electrical and electronic engineering at Bogazici University in Istanbul, Turkey. After finishing his post-doctoral research in KU Leuven at the division of mechatronics, biostatistics and sensors (MeBioS), he is currently pursuing his research in Nanyang Technological University at the School of Mechanical and Aerospace Engineering as an assistant professor. His research areas are unmanned aerial vehicles, robotics, mechatronics, soft computing methods and model predictive control.

Mojtaba Ahmadieh Khanesar (S'07-M'12) received the M.S. and Ph.D. degree in Control Engineering from K. N. Toosi University of Technology, Tehran, Iran, in 2007 and 2012 respectively. In 2010, he has held a 9 months visiting student position at Bogazici University, Istanbul, Turkey. He is currently an assistant professor in the Control and Electrical Engineering Department, Semnan University, Semnan, Iran. His current research interests are identification, mechatronics, sliding mode control, adaptive controller design, fuzzy systems, intelligent optimization and model predictive control.

Dr. Khanesar is a member of IEEE technical committee on soft computing and IEEE technical committee on networked control systems.