



Computational Intelligence in Electrical Engineering
Vol. 12, No. 4, 2022
Research Paper

Lyapunov stability analysis in the training of type 2 Neuro-Fuzzy Identifier with a swarm-based hybrid intelligent algorithm

Mohammad Mahdi Zabihi Shesh Poli¹, Mahdi Aliyari Shoorehdeli*², Ali Moarefianpour³

¹ Department of Electrical Engineering, Nowshahr Branch, Islamic Azad University, Nowshahr, Iran

mohammad_mahdi_zabihi@yahoo.com

² Department of Mechatronics Engineering, K. N. Toosi University of Technology, Tehran, Iran
aliyari@kntu.ac.ir

³ Department of Mechanical, Electrical, and computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
moarefian@srbiau.ac.ir

Abstract :

Training stability of a model in an identification process has been one of the primitive requirements in recent control researches. This paper aims at analyzing the training stability of the interval type 2 adaptive Neuro-Fuzzy inference system (IT2ANFIS) with a swarm-based hybrid algorithm. The antecedent and the consequent parts of the model are trained by particle swarm optimization (PSO) and Kalman filter (KF) algorithms, respectively (PSO+KF). The Lyapunov stability theorem with a newly found Lyapunov function is employed to assess the stability conditions. These conditions led to adaptive stabilizing boundaries in the adjustable parameters of the algorithms (APAs), such as the covariance matrix in KF, inertia factor, and maximum gain in PSO. The selection of APAs within these boundaries guaranteed the stability of the training process. The analytical approach of this study resulted in finding new and broader stabilizing boundaries for the APAs. Implementation of the theorem to the training and predicting the future values of the Mackey-Glass chaotic time series and a stochastic non-linear system revealed the superiority of the theorem in terms of their root mean square errors (RMSEs), simulation times, and their entrapment in the local minimums.

Keywords: Identification, Lyapunov stability, IT2ANFIS, Kalman filter, particle swarm optimization.

1. Introduction

Computational intelligence (CI) is the knowledge

for the emulation of human-mind reasoning or instinct of living creatures by a computer program. Learning is one of the foremost cognitive phenomena of CI. The phenomenon employs acquired data along with training algorithms to generate functional information. CI refers to the methodologies for solving or modeling complex problems with uncertainties where the traditional modeling approaches are ineffective. Two main divisions of CI are Fuzzy

¹ Submission date: 06, 07, 2021

Acceptance date: 15, 12, 2021

Corresponding author: Mahdi Aliyari Shoorehdeli,
Department of Mechatronics Engineering, K. N. Toosi
University of Technology, Tehran, Iran



reasoning and the neural network. The fusion of Fuzzy reasoning as linguistic rules and network structure of the neural network with learning capability, namely ANFIS has the potential to possess the benefits of both in a unified framework. So that in recent studies its usage has been cited in many engineering problems such as control, system identification, prediction, etc.

Zadeh (1975), proposed type 2 Fuzzy systems that are undoubtedly prominent because of confronting uncertainty and dealing with noisy data. [1]. For this reason, T2FS and T2ANFIS seem to score superiority compared with their type 1 family. Unlike T1FS, where their membership grades are precise, the membership grades of T2FS are Fuzzy [2].

System identification requires a suitable model and an algorithm (or algorithms) to update its parameters. In this way, the model's output should have the lowest difference with the output of the identified system. Training algorithms of ANFIS and T2ANFIS as a model in system identification are divided into three types; derivative-based, free derivative, and hybrid algorithms. Hybrid algorithms are made by merging of prior mentioned algorithms.

Among derivative-based algorithms, the gradient descent (GD) algorithm is a traditional algorithm for optimizing the parameters of ANN, ANFIS, and T2ANFIS. The learning rate has a crucial role in the convergence speed of GD. Selecting a high value of learning rate to acquire a faster convergence may result in instability of the training process. By contrast, selecting a small value for the learning rate reduces the convergence rate, and hence the algorithm may stagnate in the local minimum [3, 4]. Levenberg Marquardt (LM) algorithm evolved from the interpolation of GD and Gauss-Newton (GN). Its momentum coefficient determines how much it is close to GD or GN. Although it has a faster convergence rate compared with GD, it seems that its convergence rate slows whenever the dimension of the parameters is large [5]. Kalman filter families, like KF [6], EKF [7], and UKF [8] are the other cases of derivative-based algorithms. Convenient consistency and good estimation in the Kalman filter family, even with the existence of process noise or uncertainty, are the main reasons for the preference of the KF family compared with other derivative-based algorithms

in recent studies.

More probability of stagnation in the local minimum is the main problem of derivative-based algorithms [9]. However, stochastic or adaptive selection of the learning parameters (e.g. learning rate, and momentum coefficient) may tackle this problem to a certain extent [10-12].

Free-derivative algorithms are distinguished from derivative-based algorithms in many aspects. One of the most brilliant advantages of the free-derivative algorithms such as genetic algorithm (GA; [13]), simulated annealing (SA; [14]), artificial bee colony (ABC; [15]), ant colony optimization (ACO; [16]), search group algorithm (SGA; [17]), water cycle algorithm (WCA; [18]), sliding mode control (SMC; [19]) and PSO [20] is that they do not require derivative of the objective function for MF parameters in the update process. Besides, they have more robustness in finding a global minimum. Another noteworthy point is that they are implementable to any membership function, mainly in the antecedent part of T2ANFIS (with any shape and structure) [21]. But the main drawback of these algorithms is their high computational burden that makes their usage inconceivable in online identification and control issues. Therefore, Hybrid algorithms have been introduced, which can integrate the favorable features of derivative-based and free-derivative algorithms. This integration provides a synergy that improves the unfavorable characteristics of both algorithms as well.

Training stability, as one of the most significant issues in the identification process, has attracted the attention of many researchers in this field. Generally, Lyapunov stability analysis is the most commonly used analytical approach for this purpose. Lyapunov stability theorem determines adaptive boundaries for the APAs so that these boundaries guarantee the stability of the training process. Because of the probabilistic nature of some algorithms like SA or the existence of mutation in GA, the application of the Lyapunov stability theorem on derivative-based algorithms seems to be more implementable than free-derivative or hybrid algorithms. However, some researchers reported the implementation of the Lyapunov stability theorem on free-derivative algorithms like PSO [22] and SMC [19].

Authors of [23] presented stability analysis for a type 2 fuzzy Neural network using LM. They

used the first-order expansion of the Taylor series for all of the model parameters separately. Despite providing suitable degrees of freedom by this issue which can result in a proper performance, the LM has an intrinsically high computational burden. The authors in [22] obtained stable adaptive boundaries for different algorithms. They provided the details of mathematical proofs to attain the stabilizing boundaries of GD and then LM in training both the antecedent and the consequent parameters of the ANFIS model. They also obtained adaptive lower and upper boundaries of a hybrid algorithm in which PSO and EKF were used for training the antecedent and consequent parameters of the model (PSO+EKF), respectively. Different from [23], they divided the model parameters into two distinct forms, vector and matrix form parameters. The same analysis was provided by the authors of [24] and [25] for obtaining the learning rate boundary. A type 2 Fuzzy wavelet Neural network model was presented in [24] where its parameters were optimized by GD. In addition to defining a stabilizing boundary for learning rate, they started learning with a small learning rate. During learning, the proposed algorithm changed the learning rate to guarantee stability and speed up the learning. The stabilizing boundaries of their work were similar to that in [22]. In [25], a self-compensatory interval type-2 FNN (TSCIT2FNN) was utilized as a model. The Antecedent and the consequent parameters were optimized by GD and variable EKF, respectively. To guarantee the learning stability, they examined the convergence of GD and did not provide any analysis for the convergence of the consequent parameters. A similar approach was reported in [26] and [27]. The authors of [26] examined the training stability and presented two Fuzzy models. They also provided their first-order derivatives in full detail. In [27], the input data were clustered to reduce the computational complexity. In their work, the algorithm updated the weights by pseudo-inverse matrix instead of calculating them iteratively. Moreover, the authors provided a stability analysis similar to those proved in [24], [25], and [26].

In most of the above references or other related publications, the authors' focus has been on how to extend Lyapunov functions in terms of model parameters to obtain stabilizing boundaries. They usually used the Lyapunov function in the form of the squared error between the model output and the output of the real system (the system to be identified). Zabihi shesh Poli, Aliyari Shoorehdeli,

and Moarefianpour (2019), proposed a novel Lyapunov function in the form of squared of error difference for the first time [28]. They analyzed the stability of the training process in the identification of an Interval type 2 adaptive Neuro-Fuzzy inference system (IT2ANFIS) at first by GD and then by a hybrid algorithm based on GD and FRLS. They provided new adaptive laws for the learning rate in GD and the covariance matrix in FRLS. The same authors in [29] studied the stability of the prior-mentioned system by considering the GD and KF for training antecedent and consequent parameters, respectively. They showed, when KF was substituted instead of FRLS, higher degrees of freedom were attained. In addition, they reached a relatively better performance compared with GD+FFRS. This paper utilizes the same Lyapunov function used in [28] and [29] to obtain stability conditions of a swarm-based algorithm. In doing so, stability analysis is conducted based on the identification of the IT2ANFIS model when its antecedent and consequent parameters are trained by PSO and KF, respectively (PSO+KF).

2. Methodology

2.1. The structure of the Type 2 fuzzy system (T2FS) of this study

From the idea of blurring the points on the type 1 fuzzy membership function (MF), a type 2 MF can be attained, which defines a type 2 fuzzy set (T2FS) [30]. Unlike type 1 fuzzy sets, the membership degree in T2FS is a fuzzy set in [0, 1]. This evolution provides a three-dimensional, bivariate MFs represented as $\mu_A(x, u)$ in which for a general T2FS, $0 \leq \mu_A(x, u) \leq 1$. A T2FS is represented as

$$A = \int_{x \in X} \int_{u \in [0,1]} \mu_A(x, u) / (x, u) \quad (1)$$

$$\forall x \in X, \forall u \in [0,1]$$

The variables x and u are the primary and the secondary variables respectively. However, restricting $\mu_A(x, u)$ to 1, forms interval T2FS (IT2FS) that trivialize the third dimension that can be expressed as follows:

$$\square A = \int_{x \in X} \int_{u \in [0,1]} 1 / (x, u) \quad (2)$$

The Fuzzy system of this study is a specific Takagi- Sugeno- Kang (TSK) type IT2FS. This

structure employs interval type 2 MFs (T2MF) with Gaussian spread in the antecedent and summation of crisp multiples of inputs in the consequent. A Gaussian T2MF with the uncertain mean and an identical standard deviation is shown in Fig. 1. Similar to type 1 TSK Fuzzy systems, the K^{th} rule governing the Fuzzy system is represented as below [2]:

R^k :

if x_1 is $A_{k,1}$ and x_2 is $A_{k,2}$ and $\dots x_i$ is $A_{k,i}$ (3)

then $y_k = r_k + p_k x_1 + q_k x_2 + \dots + z_k x_i$

Where $A_{k,i}$ is a T2MF corresponds to the K^{th} rule, and i^{th} input in the antecedent, x_i is the i^{th} input, and $r_k, p_k, q_k, \dots, z_k$ are crisp coefficients in the consequent.

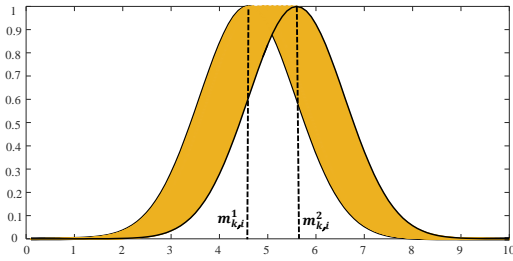


Fig. 1. A Gaussian T2MF with the uncertain mean and an

2.2. IT2ANFIS and its output calculation

Representation of IT2FS in a network structure with learning capability creates a model which is called IT2ANFIS. This model is shown in Fig. 2. The i^{th} input to the model is fuzzified as

$$\mu_{k,i}^1(x_i, [\sigma_{k,i}, m_{k,i}^1]) = e^{-0.5 \left(\frac{x_i - m_{k,i}^1}{\sigma_{k,i}} \right)^2} \quad (4)$$

$$\mu_{k,i}^2(x_i, [\sigma_{k,i}, m_{k,i}^2]) = e^{-0.5 \left(\frac{x_i - m_{k,i}^2}{\sigma_{k,i}} \right)^2} \quad (5)$$

In equations (4) and (5), $m_{k,i}^1, m_{k,i}^2$ are the lower and upper mean values correspond to the k^{th} rule and i^{th} input respectively. As defined in above equations, $\sigma_{k,i}$ is the standard deviation. The lower and upper membership grades i.e. $\underline{\mu}_{k,i}$ and $\bar{\mu}_{k,i}$ are determined as

$$\bar{\mu}_{k,i}(x_i) = \begin{cases} \mu_{k,i}^1(x_i, [\sigma_{k,i}, m_{k,i}^1]), & x_i < m_{k,i}^1 \\ 1, & m_{k,i}^1 \leq x_i \leq m_{k,i}^2 \\ \mu_{k,i}^2(x_i, [\sigma_{k,i}, m_{k,i}^2]), & x_i > m_{k,i}^2 \end{cases} \quad (6)$$

$$\underline{\mu}_{k,i}(x_i) = \begin{cases} \mu_{k,i}^2(x_i, [\sigma_{k,i}, m_{k,i}^2]), & x_i \leq \frac{m_{k,i}^1 + m_{k,i}^2}{2} \\ \mu_{k,i}^1(x_i, [\sigma_{k,i}, m_{k,i}^1]), & x_i \geq \frac{m_{k,i}^1 + m_{k,i}^2}{2} \end{cases} \quad (7)$$

Using (6) and (7), the lower and upper firing strengths are defined as follows:

$$\underline{f}^k = \prod_{i=1}^n \underline{\mu}_{k,i}; \quad \bar{f}^k = \prod_{i=1}^n \bar{\mu}_{k,i} \quad (8)$$

Different from T1FS, defuzzification, and output calculation in T2FS require type reduction. Iterative type reduction mechanisms were presented in some pieces of literatures [31– 34]. But for performing stability analysis, the output of the model requires a closed-form. On that account, this paper utilizes a closed solution provided by [35] that calculates the model output as follows:

$$Y = Y_{TSK} = \frac{\sum_{k=1}^M \underline{f}^k y_k}{\sum_{k=1}^M \underline{f}^k + \sum_{k=1}^M \bar{f}^k} + \frac{\sum_{k=1}^M \bar{f}^k y_k}{\sum_{k=1}^M \underline{f}^k + \sum_{k=1}^M \bar{f}^k} \quad (9)$$

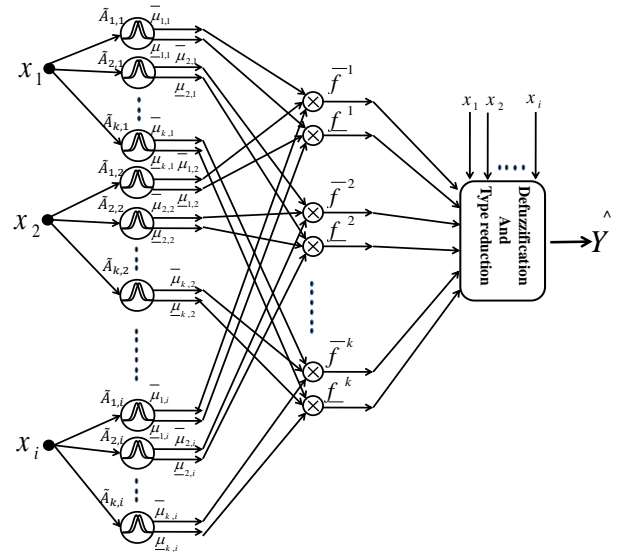


Fig. 2. The IT2ANFIS model [28]

2.3. Stability analysis

Generally, in an identification process, the algorithms tune the model parameters so that the model output best fits the actual output of an identified system. Selecting suitable values of APAs (such as covariance matrix in the KF or inertia factor and maximum gain in PSO) is of high priority. As explained previously, to escape

from local minimums, as well as to have a consistent convergence, these parameters must be selected adaptively. For determining APAs adaptively and finding their stabilizing boundaries, the Lyapunov stability theorem requires a suitable positive definite Lyapunov function. These boundaries are calculated in such a way to make the difference of the Lyapunov function negative. The boundaries guarantee the stability of the identification process. Although outside of these boundaries, the algorithm may be stable, this stability is not guaranteed. The boundaries of the APAs acquired from the Lyapunov stability theorem are applied to the algorithms at each data entry and iteration to maintain the training stability of the model (i. e. IT2ANFIS). The main contribution of this study is the exploitation of a newly found Lyapunov function in finding the stabilizing boundaries for the APAs of the KF and PSO based on a hybrid combination, namely PSO+KF.

2.3.1. Proposed Lyapunov function

Nearly in most studies related to this field, the objective function is the square of the error between the model and the actual system to be identified. Moreover, the references that selected the Lyapunov function in the same way as this objective function did not consider the error rate. For this reason, the Lyapunov function proposed by the authors of [28] and [29] is utilized. This Lyapunov function is the square of the error difference between the model and the actual system. The proposed Lyapunov function can be interpreted as the kinetic energy of the system that considers the error rate. The error itself is minimized because the objective function is the square of this error. Therefore, the Lyapunov function of this study is represented as

$$V(k) = \frac{1}{2}(\Delta e(k))^2 > 0 \quad (10)$$

And,

$$e(k) = Y_d(k) - Y(k) \quad (11)$$

Error $e(k)$ is the difference between the real output of the system to be identified $Y_d(k)$, and the output of the IT2ANFIS model $\hat{Y}(k)$. According to (10), the change of Lyapunov function $\Delta V(k)$ can be written as:

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) = \\ &= \frac{1}{2}[(\Delta e(k+1))^2 - (\Delta e(k))^2] = \\ &= \frac{1}{2}[\Delta e(k+1) + \Delta e(k)][\Delta e(k+1) - \Delta e(k)] = \\ &= \frac{1}{2}[\Delta e(k+1) + \Delta e(k) + \Delta e(k) - \Delta e(k)] \times \\ &\quad \times [\Delta e(k+1) - \Delta e(k)] = \\ &= \frac{1}{2}[\Delta M(k) + 2\Delta e(k)]\Delta M(k) \end{aligned} \quad (12)$$

Where $\Delta e(k)$ is the error difference that can be expressed based on the first-order Taylor series expansion of error with respect to the IT2ANFIS parameters as follows:

$$\begin{aligned} \Delta e(k) &= \Delta e_1(k) + \Delta e_2(k) = \left(\frac{\partial e(k)}{\partial x(k)}\right)^T \Delta x(k) + \\ &\quad + \left(\frac{\partial e(k)}{\partial \Gamma(k)}\right)^T \Delta \Gamma(k) = M_1(k) + M_2(k) \end{aligned} \quad (13)$$

The error at the time $k+1$ is defined as

$$e(k+1) = \Delta e(k) + e(k) \quad (14)$$

And $\Delta M(k)$ is the difference of $\Delta e(k)$ (the rate of error difference) that can be expressed in the same way as (13), as follows:

$$\begin{aligned} \Delta M(k) &= \Delta M_1(k) + \Delta M_2(k) = \\ &= \left(\frac{\partial M_1(k)}{\partial x(k)}\right)^T \Delta x(k) + \left(\frac{\partial M_2(k)}{\partial \Gamma(k)}\right)^T \Delta \Gamma(k) = \\ &\left(\frac{\partial}{\partial x(k)}\left[\left(\frac{\partial e(k)}{\partial x(k)}\right)^T \Delta x(k)\right]\right)^T \Delta x(k) + \\ &\quad + \left[\left(\frac{\partial}{\partial \Gamma(k)}\left[\left(\frac{\partial e(k)}{\partial \Gamma(k)}\right)^T \Delta \Gamma(k)\right]\right)^T \Delta \Gamma(k)\right] \end{aligned} \quad (15)$$

The error difference at the time $k+1$ is defined as

$$\begin{aligned} \Delta e(k+1) &= \Delta M(k) + \Delta e(k) = \\ &\quad \Delta M_1(k) + \Delta M_2(k) + \Delta e(k) \end{aligned} \quad (16)$$

Where x and Γ are the antecedent and consequent parameters in vector form, respectively. x and Γ are denoted by

$$\begin{aligned} x &= [m_{1,1}^1 \sigma_{1,1} m_{1,1}^2 \dots m_{k,i}^1 \sigma_{k,i} m_{k,i}^2]^T \\ \Gamma &= [r_1 p_1 \dots s_1 r_2 p_2 \dots s_2 \dots r_k p_k \dots s_k]^T \end{aligned} \quad (17)$$

2.3.2. Stabilizing conditions of the IT2ANFIS identification trained with PSO+KF

Theorem. Given that the antecedent and the consequent parameters of the IT2ANFIS are

trained with PSO and KF respectively, Lyapunov stability theorem of this study requires the following conditions:

$$([\alpha(\omega)^2 + \beta\omega\rho_g + \gamma(\rho_g)^2 + \xi^2 e_p(k)] + 2[D^T \omega v(k) + D^T \rho_g d(k) - \xi e_p(k)] \times [\alpha(\omega)^2 + \beta\omega\rho_g + \gamma(\rho_g)^2 + \xi^2 e_p(k)] < 0 \quad (18)$$

Proof. To attain the above condition according to the proposed theorem, $\Delta e_1(k)$ and $\Delta M_1(k)$ in the PSO algorithm, and $\Delta e_2(k)$ and $\Delta M_2(k)$ in the KF algorithm, should be calculated separately. Then by substituting them into (12), $\Delta V(k)$ can be defined. Therefore, sections 2.3.2.1 and 2.3.2.2 give a detailed explanation of acquiring the $\Delta e(k)$ and $\Delta M(k)$.

2.3.2.1 Particle Swarm Optimization (PSO) algorithm

The PSO is an intelligent algorithm that was introduced by Eberhart and Kennedy in 1995 [37]. This algorithm is an iterative technique that is artificially inspired by the social behavior of flocks of birds or schools of fish. In this algorithm, each particle is a possible solution to the optimization problem. Each particle's velocity and position span a multidimensional space at each stage of the optimization process. Each element of the position vector is a variable for the problem (here, each component of the antecedent parameters). The participation of particles in the optimization problem improves the state of the particle population, and the algorithm reaches an optimal solution. The optimization procedure of this algorithm is summarized as follows:

1. Initializing with p_i particles with random position x_i and velocity v_i ,
2. Evaluating the performance of each particle $F(x_i(k))$ according to its present position x_i
Where the objective function is defined as follow:

$$F(x_i(k)) = \frac{1}{2} e^2(k) \quad (19)$$

3. Comparing each particle's performance with its best experience (p_{best_i}), so that if

$$F(x_i(k)) < p_{best_i} \text{ then } \begin{cases} p_{best_i} = F(x_i(k)) \\ x_{best_i} = x_i(k) \end{cases}$$

4. Comparing each particle's performance with the best particle among all particles (g_{best}), so that if $F(x_i(k)) < g_{best}$ then

$$\begin{cases} g_{best} = F(x_i(k)) \\ x_{g_{best}} = x_i(k) \end{cases}$$

5. Updating the velocity vector of each particle by:

$$v_i(k) = \omega v_i(k-1) + \rho_1(x_{p_{best_i}} - x_i(k-1)) + \rho_2(x_{g_{best}} - x_i(k-1)) \quad (20)$$

Where, ρ_1 , ρ_2 and ω are random parameters in which, $\rho_1 = r_1 c_1$ and $\rho_2 = r_2 c_2$. r_1 and r_2 are random numbers in the interval $[0,1]$. c_1 and c_2 are the acceleration constants and ω is the inertia weight such that $c_1 + c_2 \leq 4$.

6. Finally Defining the updated position of each particle by:

$$x_i(k) = x_i(k-1) + v_i(k) \quad (21)$$

Proving the stability for the global best particle can guarantee the convergence of other particles. With this condition, It is evident that the best experience of the global best particle equals the best particle in all particles (global best particle or the $x_{g_{best}}$). Therefore, for the stability analysis of the global best particle according to the Lyapunov stability of the proposed theorem, equations (20) and (21) can be written for the global best particle as:

$$(22)$$

Where $\rho_1 + \rho_2 = \rho_g$, $0 < \rho_g < k$, and $k = c_1 + c_2$.

The parameter k is called the maximum gain.

Based on (22), following relation can be deduced:

$$\begin{aligned} v(k+1) &= x(k+1) - x(k) = \Delta x(k) = \\ &= \omega[x(k) - x(k-1)] + \rho_g[x_{g_{best}} - x(k)] \end{aligned} \quad (23)$$

By defining $d(k)$ as follow

$$d(k) = x_{g_{best}} - x(k) \quad (24)$$

Equation (23) is written as

$$\Delta x(k) = \omega v(k) + \rho_g d(k) \quad (25)$$

Thus, $\Delta e_1(k)$ in (13) can be developed as:

$$\begin{aligned}\Delta e_1(k) &= -\left(\frac{\partial Y(k)}{\partial x(k)}\right)^T \Delta x(k) = \\ &= -\left(\frac{\partial Y(k)}{\partial x(k)}\right)^T [\omega v(k) + \rho_g d(k)]\end{aligned}\quad (26)$$

Similarly, the expansion of the first part of (15) is:

$$\begin{aligned}\Delta M_1(k) &= \left(\frac{\partial M_1(k)}{\partial x(k)}\right)^T \Delta x(k) = \\ &= \left(\frac{\partial}{\partial x(k)} \left(\left(\frac{\partial e(k)}{\partial x(k)} \right)^T \Delta x(k) \right) \right)^T \Delta x(k)\end{aligned}\quad (27)$$

By the further deepening of (27), we have:

$$\begin{aligned}\Delta M_1(k) &= \left[\left(\frac{\partial^2 e(k)}{\partial x^2(k)} \right)^T \Delta x(k) + \right. \\ &\quad \left. + \left(\frac{\partial}{\partial x(k)} \left((\Delta x(k))^T \left(\frac{\partial e(k)}{\partial x(k)} \right) \right) \right)^T \Delta x(k) = \right. \\ &\quad \left. - (\Delta x(k))^T H \Delta x(k) - \left(\frac{\partial Y(k)}{\partial x(k)} \right)^T \frac{\partial (\Delta x(k))}{\partial x(k)} \Delta x(k) \right]\end{aligned}\quad (28)$$

Where $H = \left(\frac{\partial^2 \hat{Y}(k)}{\partial x^2(k)} \right)^T$. In which H is the pseudo

diagonal Hessian matrix that contains second derivatives of $\hat{Y}(k)$ w.r.t the antecedent parameters. We also define the derivative operator

as $D = \frac{\partial \hat{Y}(k)}{\partial x(k)}$. The first part of (28) is developed

as:

$$\begin{aligned}(\Delta x(k))^T H \Delta x(k) &= \\ [\omega v(k) + \rho_g d(k)]^T H [\omega v(k) + \rho_g d(k)] &= \\ = \omega^2 v^T(k) H v(k) + 2 \omega \rho_g v^T(k) H d(k) + \\ &\quad + \rho_g^2 d^T(k) H d(k)\end{aligned}\quad (29)$$

For expanding the second part of (28), the following relation should be determined:

$$\begin{aligned}\frac{\partial (\Delta x(k))}{\partial x(k)} &= \frac{\partial}{\partial x(k)} [\omega v(k) + \rho_g d(k)] = \\ = \omega \frac{\partial v(k)}{\partial x(k)} + \rho_g \frac{\partial d(k)}{\partial x(k)}\end{aligned}\quad (30)$$

The term $\frac{\partial v(k)}{\partial x(k)}$ in particle dynamics is

defined as:

$$\frac{\partial v(k)}{\partial x(k)} = \frac{a(k)}{v(k)}\quad (31)$$

$a(k)$ is the particle acceleration vector.

Denoting that this study has a discrete-time nature, and the dimension of particles is R^n , equation (31) can be rewritten as follows:

$$\begin{aligned}\frac{a(k)}{v(k)} = \frac{\Delta v}{\Delta x} &= \frac{\partial}{\partial [R_{x_1}(k) R_{x_2}(k) \dots R_{x_n}(k)]} \begin{pmatrix} R_{v_1}(k) \\ \vdots \\ R_{v_n}(k) \end{pmatrix} = \\ \begin{pmatrix} R_{v_1}(k) - R_{v_1}(k-1) & \dots & R_{v_n}(k) - R_{v_n}(k-1) \\ R_{x_1}(k) - R_{x_1}(k-1) & \dots & R_{x_1}(k) - R_{x_1}(k-1) \\ \vdots & \ddots & \vdots \\ R_{v_1}(k) - R_{v_1}(k-1) & \dots & R_{v_n}(k) - R_{v_n}(k-1) \\ R_{x_n}(k) - R_{x_n}(k-1) & \dots & R_{x_n}(k) - R_{x_n}(k-1) \end{pmatrix} = A_v\end{aligned}\quad (32)$$

Indices "1" to "n" are the numbers attributed to each variable of the particle's position and velocity. Moreover, $\frac{\partial d(k)}{\partial x(k)} = -I$, in which I is the

identity matrix. Therefore,

$$\frac{\partial (\Delta x(k))}{\partial x(k)} = \omega \frac{\partial v(k)}{\partial x(k)} + \rho_g \frac{\partial d(k)}{\partial x(k)} = \omega A_v - \rho_g I\quad (33)$$

consequently,

$$\begin{aligned}\left(\frac{\partial Y(k)}{\partial x(k)} \right)^T \frac{\partial (\Delta x(k))}{\partial x(k)} \Delta x(k) &= \\ = D^T [\omega A_v - \rho_g I] [\omega v(k) + \rho_g d(k)] &= \omega^2 D^T A_v v(k) + \\ &\quad + \omega \rho_g [D^T A_v d(k) - D^T I v(k)] - \rho_g^2 D^T I d(k)\end{aligned}\quad (34)$$

Finally, the $\Delta M_1(k)$ is derived as:

$$\begin{aligned}\Delta M_1 &= -\omega^2 v^T(k) H v(k) - 2\omega \rho_g v^T(k) H d(k) - \\ &\quad - \rho_g^2 d^T(k) H d(k) - \omega^2 D^T A_v v(k) - \\ \omega \rho_g [D^T A_v d(k) - D^T I v(k)] &+ \rho_g^2 D^T I d(k) = \\ = \omega^2 [-v^T(k) H v(k) - D^T A_v v(k)] &+ \\ \omega \rho_g [-2v^T(k) H d(k) - D^T A_v d(k) &+ D^T I v(k)] + \\ + \rho_g^2 [-d^T(k) H d(k) + D^T I d(k)]\end{aligned}\quad (35)$$

By defining α, β , and γ as follow:

$$\begin{aligned}\alpha &= -v^T(k) H v(k) - D^T A_v v(k) \\ \beta &= -2v^T(k) H d(k) - D^T A_v d(k) + D^T I v(k) \\ \gamma &= -d^T(k) H d(k) + D^T I d(k)\end{aligned}\quad (36)$$

Equation (35) is written as below:

$$\Delta M_1 = \alpha \omega^2 + \beta \omega \rho_g + \gamma \rho_g^2\quad (37)$$

2.3.2.2 Kalman Filter (KF) algorithm

The equations related to the KF are accessible in the literature [6, 36]. Suppose $\hat{Y}(k)$ is the output of IT2ANFIS model, and Γ is the consequent

parameters of this model as defined in (17), then IT2ANFIS model is represented in state-space form as:

$$\begin{aligned}\Gamma(k+1) &= \Gamma(k) + \omega(k+1) \\ Y(k+1) &= G(X(k))^T \Gamma(k) + v(k+1)\end{aligned}\quad (38)$$

Where $\omega(k)$ and $v(k)$ are the process and the measurement white noises, respectively, which are artificially added to the state-space equation. The $G(X(k))$ is a transfer function between the output of the IT2ANFIS and the parameters to be estimated, which is built up by inputs, governing Fuzzy rules, and the antecedent parameters. The task of the KF is to estimate Γ in the model. Thus, the updated rules of the KF is written as below:

$$\begin{aligned}L(k+1) &= P(k)G(X(k+1))(G^T(X(k+1)) \times \\ &\quad \times P(k)G(X(k+1)) + R(k+1))^{-1} \\ P(k+1) &= P(k) - L(k+1)G^T(X(k+1))P(k) + \\ &\quad + Q(k+1) \\ \Gamma(k+1) &= \Gamma(k) + L(k+1)e_p(k)\end{aligned}\quad (39)$$

Where $R(k)$ and $Q(k)$ are the covariance matrices of $\omega(k)$, and $v(k)$, respectively. Here, $e_p(k)$ is the prediction error, $P(k)$ is the covariance matrix of the state variables $\Gamma(k)$, and $L(k)$ is the Kalman gain. With some straight mathematical operations on (38) and (39), the second part of (13) can be written as follows:

$$\begin{aligned}\Delta e_2 = M_2(k) &= \left(\frac{\partial e(k)}{\partial \Gamma(k)} \right)^T \Delta \Gamma(k) = \\ &\quad -G(X(k+1))^T L(k+1)e_p(k)\end{aligned}\quad (40)$$

By similar mathematical operations, the second part of (15) is expanded as:

$$\begin{aligned}\Delta M_2(k) &= \left[\frac{\partial M_2(k)}{\partial \Gamma(k)} \right] \Delta \Gamma(k) = \\ &\quad \left[\frac{\partial}{\partial \Gamma(k)} \left[\left(\frac{\partial e(k)}{\partial \Gamma(k)} \right)^T \Delta \Gamma(k) \right] \right]^T \Delta \Gamma(k) = \\ &= \left[\frac{\partial}{\partial \Gamma(k)} \left[-G(X(k+1))^T L(k+1)e_p(k) \right] \right]^T \times L(k+1)e_p(k) = \\ &= -G(X(k+1))^T L(k+1) \left[\frac{\partial}{\partial \Gamma(k)} [e_p(k)] \right]^T \times \\ &\quad \times L(k+1)e_p(k) = \\ &= -G(X(k+1))^T L(k+1) \left(-G(X(k+1))^T L(k+1) \right) e_p(k) = \\ &= [G(X(k+1))^T L(k+1)]^2 e_p(k)\end{aligned}\quad (41)$$

As stated in (36), and defining ξ as

$$\xi = [G(X(k+1))^T L(k+1)] \quad (42)$$

The change of Lyapunov function, $\Delta V(k)$, can be written as bellow:

$$\begin{aligned}\Delta V(k) &= \frac{1}{2}([\alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k)] + \\ &\quad 2[D^T \omega v(k) + D^T \rho_g d - \xi e_p(k)]) \times \\ &\quad \times [\alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k)]\end{aligned}\quad (43)$$

For stability of the identification process according to the proposed theorem, equation (43) must be negative definite. For that reason, we have the following conditions:

1. If the negative definiteness of (43) is as follows:

$$\begin{cases} ([\alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k)] + \\ \quad + 2[D^T \omega v(k) + D^T \rho_g d - \xi e_p(k)]) < 0 \\ \& [\alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k)] > 0 \end{cases}\quad (44)$$

Considering the sign of the second part of (44) and rearranging its first part, we can obtain the following inequality which results in the following inequality:

$$0 < \alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k) < -2D^T \omega v(k) - 2D^T \rho_g d + 2\xi e_p(k) \quad (45)$$

Although there are several solutions to satisfy the negative definiteness of (43), one of the most available solutions for (45) is to consider the following inequalities:

$$\begin{cases} \alpha\omega^2 + \frac{1}{2}\beta\omega\rho_g + 2D^T \omega v(k) < 0 \\ \gamma\rho_g^2 + \frac{1}{2}\beta\omega\rho_g + 2D^T \rho_g d < 0 \\ (\xi^2 - 2\xi)e_p(k) < 0 \end{cases}\quad (46)$$

The stabilizing boundaries of the proposed theorem based on PSO+KF are:

$$\begin{cases} 0 < \omega < \frac{-\frac{\beta\rho_g}{2} - 2D^T v(k)}{\alpha} & \text{or} & \begin{cases} 0 > \omega > \frac{-\frac{\beta\rho_g}{2} - 2D^T v(k)}{\alpha} \\ 0 > \rho > \frac{-\frac{\beta\omega}{2} - 2D^T d(k)}{\gamma} \end{cases} \\ 0 < \rho < \frac{-\frac{\beta\omega}{2} - 2D^T d(k)}{\gamma} & & \end{cases}\quad (47)$$

& $\begin{cases} \text{if } e_p(k) > 0 & \text{then } 0 < \xi < 2 \\ \text{if } e_p(k) < 0 & \text{then } \xi > 2 \end{cases}$

2. If the negative definiteness of (43) is as follows:

$$\left\{ \begin{array}{l} ([\alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k)] + \\ \quad + 2[D^T \omega v(k) + D^T \rho_g d - \xi e_p(k)]) > 0 \\ \& \quad [\alpha\omega^2 + \beta\omega\rho_g + \gamma\rho_g^2 + \xi^2 e_p(k)] < 0 \end{array} \right. \quad (48)$$

By a similar procedure, the stabilizing boundaries are:

$$\left\{ \begin{array}{l} \omega > 0, \quad \omega > \frac{-\beta\rho_g - 2D^T v(k)}{\alpha} \\ \rho_g > 0, \quad \rho_g > \frac{-\beta\omega - 2D^T d(k)}{\gamma} \end{array} \right. \text{ or } \left\{ \begin{array}{l} \omega < 0, \quad \omega < \frac{-\beta\rho_g - 2D^T v(k)}{\alpha} \\ \rho_g < 0, \quad \rho_g < \frac{\beta\omega - 2D^T d(k)}{\gamma} \end{array} \right. \quad (49)$$

$$\& \quad \begin{cases} \text{if } e_p(k) > 0 & \text{then } \xi > 2 \\ \text{if } e_p(k) < 0 & \text{then } 0 < \xi < 2 \end{cases}$$

The desirable regions attained from (47) and (49) on the ω - k axis cover the first and the third quadrants of the trigonometric coordinate. Since these boundaries are not predefined and are obtained during simulation, for presentable boundaries and better performance of the proposed theorem we derive the intersection of boundaries in the theorem with the boundaries provided in [37]. According to [37], PSO was represented as the following state-space form:

$$\zeta(k) = \begin{bmatrix} 1 & \omega \\ 0 & \omega \end{bmatrix} \zeta(k-1) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k-1) \quad (50)$$

$$y(k) = [1 \quad 0] \zeta(k-1)$$

Where the state variables and the input of the system were determined by:

$$\zeta(k) = \begin{bmatrix} x(k) - x_{gbest} \\ v(k) \end{bmatrix} \quad (51)$$

$$u(k) = -\rho_g (y(k) - x_{gbest})$$

Therefore, for the asymptotic stability of the origin, the following condition was obtained:

$$K < \left(\frac{2(1-2|\omega|+\omega^2)}{1+\omega} \right), \quad |\omega| < 1, \quad \omega \neq 0 \quad (52)$$

Finally, the reliable regions obtained from the intersection of the proposed theorem and [38] are shown in Fig. 3. This region is broader than what was defined in [22]. One of the reasons for this occurrence is the existence of matrix H that adds more flexibility in selecting the ω and the ρ_g . Moreover, minimizing the error rate besides the error increases the capability of the algorithms in the optimization problem.

2.4. Computational burden

In the IT2ANFIS model of this study, n inputs, m Fuzzy rules per inputs, and h training data are

considered. In the proposed theorem, if p is considered as the number of particles, then the computational burden is $4mnhp+3mhp+9mnp+3hp+const$. Since h has the highest dimension among others, when all parameters have the same stopping condition, the computational burden of the proposed theorem will be of order $O(h^4)$.

3. Simulation Results

The adaptive stabilizing boundaries of this study are utilized in the identification of the Mackey-Glass Chaotic time series and a stochastic non-linear system. The parameters of IT2ANFIS are tuned adaptively in such a way as to predict the future values of these systems. The results of this study are compared with the results of the LM algorithm in [23, 39], EKF and PSO+KF in [40], GD+FFRLS in [28], and GD+KF in [29]. Root-mean-square errors (RMSEs) derived in both the training and the checking data are the expected values of 10 separate runs. MATLAB R2015b is the simulation software of the benchmark. The hardware used in this study is a computer with a 2.4 GHz processor and 4 GB installed memory.

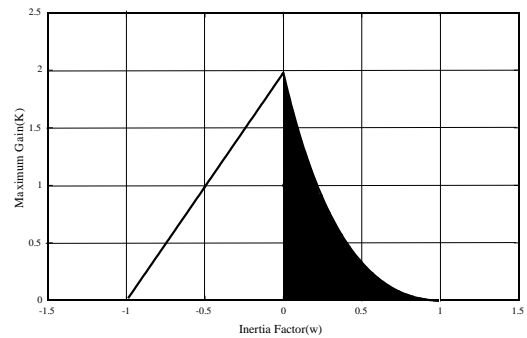


Fig. 3. Reliable regions of inertia factor and maximum gain obtained from the intersection of the proposed theorem and [38]

3.1. Implementation of the proposed theorems in predicting the future values of Mackey-Glass chaotic time series

Example 1. The equation concerning the dynamics of the Mackey-Glass chaotic time series is given below [41]:

$$\frac{dx(t)}{dt} = \frac{0.2 * x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1 * x(t) \quad (53)$$

Where $x(0)=1.2$ and $\tau=17$. The trained IT2ANFIS model of this study should be able to predict the values of $x(t+1)$ that are defined as the output of the system, using the input values of $x(t)$, $x(t-1)$, $x(t-2)$, and $x(t-3)$. For the training

phase, 750 sets of data are considered, and 335 sets of data are considered for the prediction phase (checking data). In this system, the number of tuning parameters in the antecedent and the consequent are 48 and 20, respectively. For comparison, a noise with the level of signal-to-noise ratio (SNR) of 10 dB is added to the data during the identification process. The following relation is used for describing the SNR:

$$SNR = 10 \log_{10} \left(\frac{\text{var}(\text{signal})}{\text{var}(\text{Noise})} \right) \quad (54)$$

In the simulation of the benchmarks based on the proposed theorem, seven particles are considered for tuning the consequent parameters by the PSO. Fig. 4 shows the training RMSE of the IT2ANFIS model in example 1 when the inertia factor and the maximum gain in PSO and ξ in KF are defined within the boundaries of the proposed theorem.

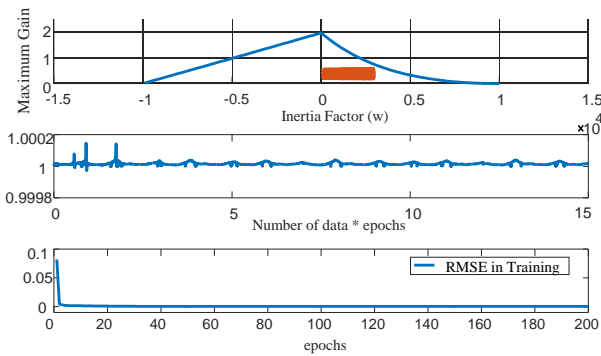


Fig. 4. Training RMSE of the IT2ANFIS, stabilizing values of inertia factor and the maximum gain in PSO, and ξ in KF when the APAs are defined within the boundaries of the proposed theorem (Ex. 1)

The orange-colored area at the uppermost part of Fig. 4 is the region that is selected for the APAs of the PSO (Inertia factor and maximum gain). The middle part of Fig. 4 shows the variations of the ξ in the KF algorithm during the tuning of the consequent parameters. As shown, the variations of the ξ are bounded in the boundaries defined in the theorem. The lower part of Fig. 4 shows the convergence of the training RMSE based on the boundaries shown. Fig. 5 shows the predicted output of the IT2ANFIS model (blue dashed line) versus real checking data (solid red line) that confirms the proper performance of this theorem

in terms of prediction RMSE. The convergence of the antecedent parameters in example 1 is shown in Fig. 6. Similarly, Fig. 7 shows the convergence of the consequent parameters. The fixation of these parameters at a constant value after some iterations indicates the stability of the training process. As shown in Fig. 8, when the APAs are chosen outside of the boundaries defined in the proposed theorem, instability will be inevitable. To compare the performance of this study with other related researches, white noise with SNR of 10 dB is added to the data sets. The result of this case is shown in Fig. 9. As shown, the solid blue line is the predicted noisy output of IT2ANFIS during checking data, and the red line represents the real checking data without noise. Table (1) shows the comparison of the RMSEs resulting from example 1 based on the proposed study with those in EKF and PSO+KF [40], LM [23, 39], GD+FFRLS [28], and GD+KF [29].

As stated, the results of this study have better RMSE compared with others. But has a higher simulation time (35 seconds) than [28], [29], and [40], because in PSO each particle is considered as a candidate solution that increases the computational burden. However, it has a less simulation time than [39] and the PSO+KF algorithm used in [40]. The results of table (1) are given in two situations. Initially, the results are obtained when the data sets are not subjected to noise. Although, whenever the KF is utilized as an optimizer, a small value of artificial white noise (higher than 50 dB) should be added to the data sets. In this case, the RMSEs of the training and checking data are the smallest value. Then, the results are given when the data sets are subjected to the white noise of about 10dB. Two states are considered:

At first, the RMSEs are given Based on the difference between predicted noisy data and real noisy data. Then they are given Based on the difference between predicted noisy data and real non-noisy data. The mean value, the best value, and the standard deviation of the RMSEs are computed for 10 separate runs. Given that the checking data are more important, it makes sense to compare these data. The Mean checking RMSE of the proposed theorem is about 0.0642 which is the smallest value among the other methods.

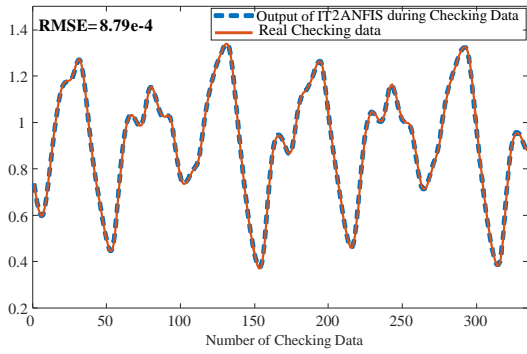


Fig. 5. Predicted data during checking data using the identified model versus the real checking data (Ex. 1)

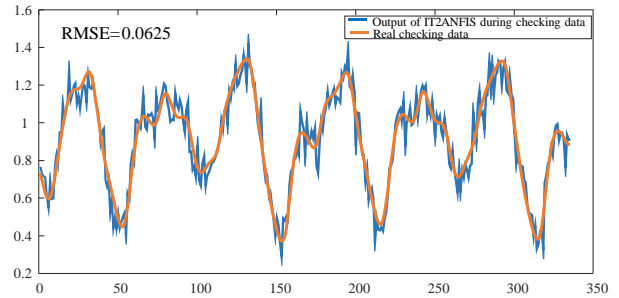


Fig. 9. The predicted noisy output of IT2ANFIS during checking data (the blue solid line) and the real checking data without noise (the red line) (Ex. 1)

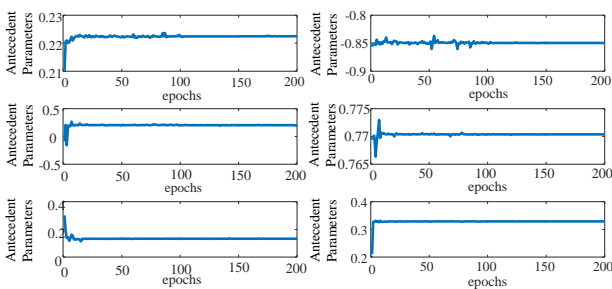


Fig. 6. Convergence of the antecedent parameters of IT2ANFIS (Ex. 1)

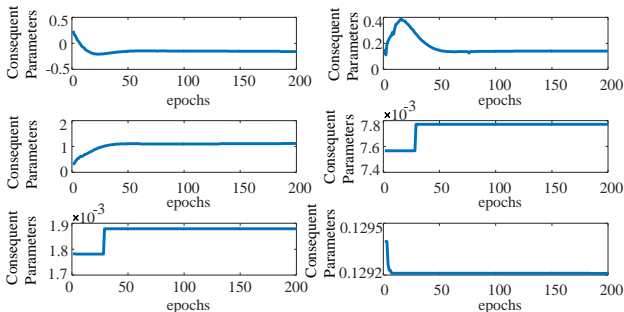


Fig. 7. Convergence of the consequent parameters of IT2ANFIS (Ex. 1)

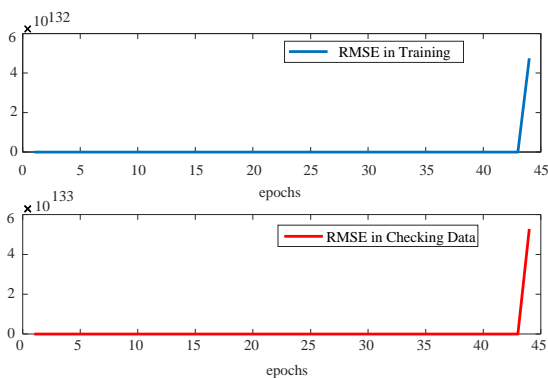


Fig. 8. Training instability when the APAs are chosen outside of the boundaries defined in the proposed theorem (Ex. 1)

3.2. Application to the Prediction of the future values in a stochastic non-linear system.

Example 2. The system in this example is a stochastic nonlinear system represented by $u = 1 + x^{0.5} + y^{-1} + z^{-1.5}$ (55)

The variables x , y , and z in the training process are random numbers in the range [1,6].

Similarly, for checking data, these variables are randomly selected in the range [1.5, 5.5]. For this, 216 and 125 sets of data are considered for the training and checking data, respectively. Fig. 10 shows the training and the checking RMSE of the IT2ANFIS when the APAs in the PSO and KF are defined within the boundaries of the proposed theorem. Fig. 11 shows the predicted output of the IT2ANFIS (blue dashed line) versus real checking data (solid red line). The figure demonstrates the proper performance of this theorem in terms of prediction RMSE for this example. Figure 12 shows the convergence of some of the antecedent and consequent parameters that suggests the correctness of the proposed theorem. As depicted in Fig. 13, training instability occurs when the APAs are chosen outside of the defined boundaries. Table (2) shows the comparison of the RMSEs resulting from this study with those in GD+FFRLS [28] and [29]. Although the results of [28] and [29] show lower simulation time, they have higher training and checking RMSE compared with this study.

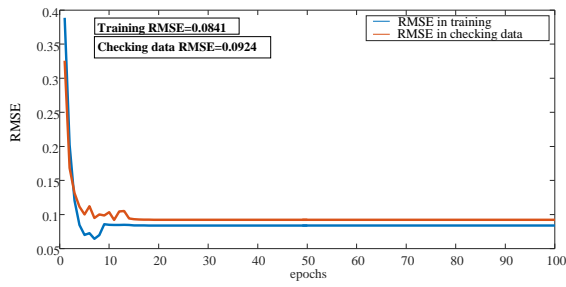


Fig. 10. RMSE of training and checking data (Ex. 2)

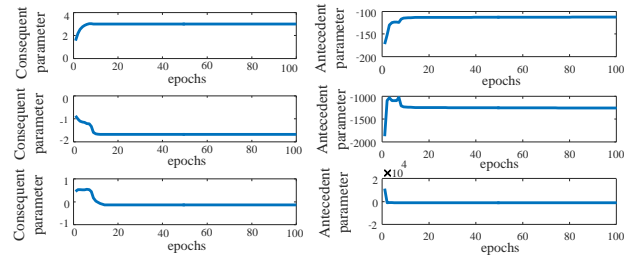


Fig. 12. Convergence of some of the antecedent and consequent parameters of IT2ANFIS (Ex. 2)

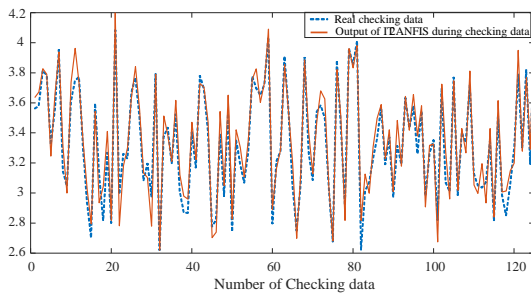


Fig. 11. Predicted data during checking data using the identified model versus the real checking data (Ex. 2)

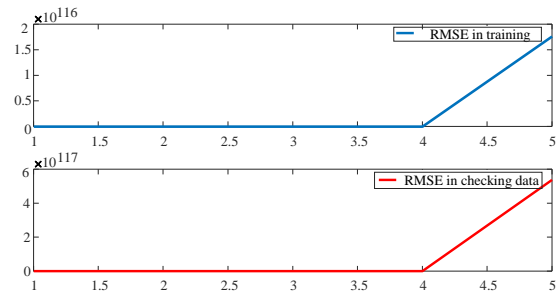


Fig. 13. Training instability when the APAs are chosen outside of the boundaries defined in the proposed theorem (Ex. 2)

Table (1): Comparison of the RMSEs resulting from the proposed study with other references (Ex. 1)

Example 1			Inputs: $x[t], x[t-1], x[t-2], x[t-3]$		
			Output: $x[t+1]$		
			No. of training/Checking data 750/335		
Signal to Noise Ratio (SNR)			Without Noise	10 dB	
				Based on the difference between predicted noisy data and real noisy data	Based on the difference between predicted noisy data and real non-noisy data
LM [23, 39]	Training	Mean	0.008		0.0914
	Checking	Mean	0.04		0.0913
	Simulation Time			70 sec	
EKF [40]	Training	Mean	-----		0.0770
	Checking	Mean	-----		0.0943
	Simulation Time			32.2 sec	
PSO+KF [40]	Training	Mean	-----		0.0765
	Checking	Mean	-----		0.0933
	Simulation Time			59.53 sec	
GD+FFRLS [28]	Training	Mean	9.61e-4	0.0256	0.0594
		Best	8.6e-4	0.0225	0.0588
		Standard Dev.	6e-5	0.0027	0.002
	Checking	Mean	9.77e-4	0.0261	0.0695
		Best	8.7e-4	0.023	0.0664
		Standard Dev.	5.1e-4	0.0026	0.0025
Simulation Time			21 sec		

GD+KF [29]	Training	Mean	9.5e-4	10.2e-4	0.0536
		Best	9.1e-4	9.6e-4	0.0524
		Standard Dev.	2.35e-4	4.28e-5	0.0015
	Checking	Mean	9.7e-4	11.3e-4	0.067
		Best	9.4e-4	10.8e-4	0.0645
		Standard Dev.	5.53e-5	4.15e-5	0.0018
Simulation Time			18 sec		
Proposed theorem based on PSO+KF	Training	Mean	8.3e-4	9.8e-4	0.0510
		Best	7.9e-4	9.2e-4	0.0485
		Standard Dev.	3.5e-4	1.93e-5	0.0016
	Checking	Mean	9.2e-4	11e-4	0.0642
		Best	8.79e-4	10.6e-4	0.0625
		Standard Dev.	3.2e-5	4.9e-5	0.0015
Simulation Time			35 sec		

Abbreviations: LM: Levenberg-Marquardt; GD: gradient descent; FFRLS: forgetting-factor recursive least square; KF: Kalman filter; EKF: Extended Kalman filter; PSO: particle swarm optimization.

Table (2): Comparison of the RMSEs resulted from the proposed study with [28] and [29] (Ex. 2)

Example 2		Inputs:	x, y, z
		Output:	u
		No. of training/Checking data 216/125	
GD+FFRLS [28]	Training	Mean	0.1586
		Best	0.0758
		Standard Dev.	0.0312
	Checking	Mean	0.1678
		Best	0.1053
		Standard Dev.	0.0263
Simulation Time		4.5 sec	
GD+KF [29]	Training	Mean	0.1025
		Best	0.0835
		Standard Dev.	0.0121
	Checking	Mean	0.118
		Best	0.0956
		Standard Dev.	0.0124
Simulation Time		1 sec	
Proposed theorem based on PSO+KF	Training	Mean	0.1012
		Best	0.0841
		Standard Dev.	0.0126
	Checking	Mean	0.11
		Best	0.0924
		Standard Dev.	0.0128
Simulation Time		8 sec	

3.3. Discussions

In training the antecedent part of the IT2ANFIS by PSO, matrix H , known as the Hessian matrix, provides more flexibility in inertia factor and maximum gain. This flexibility has led to the choice of a wide range of APAs. The existence of this matrix, which contains second-order derivatives, can increase the computational load in the identification process to some extent. But this matrix is the inseparable part of the proposed

theorem. Matrices R and Q are weight factors between the state (prediction) and the measurement. The ratio of the two is representative of Kalman gain. Large values of Q mean higher uncertainty in the states. In other words, less confidence in the results of the states (prediction), and the filter must be optimized further by measurements. Similarly, large values of R mean more uncertainty in the measurements, and the filter should be updated less with the

measurements. By a large R , the value of the measurements decreases, and the RMSE increases during training and checking data. The performance of the algorithm improves when the APAs of the PSO are selected randomly and purposefully. In this study, in addition to providing adaptive stabilizing boundaries, their randomness is also preserved. One of the points that should be mentioned in the stability of model parameters is that if one of the membership function's mean in the antecedent part of the model goes toward infinity, it results in the zeroing of the firing strength. Similarly, if one of the membership function's standard deviation goes toward infinity, the related firing strength becomes one and instability will not occur.

4. Conclusions

Utilizing the type 2 Fuzzy system in the IT2ANFIS model, has improved the ability of the system in the presence of noise and uncertainties. The main Contribution of this study compared with other related studies was finding a novel Lyapunov function to introduce newer and broader stabilizing boundaries to enhance the performance of the algorithms in an identification process. Therefore, the introduction of the novel Lyapunov function in this study not only has led to providing new adaptive stabilizing boundaries for the APAs but also increased the performance of the identification process (training and checking data). The adaptive stabilizing boundaries attained from this study have resulted in the stability of the training process. While choosing them outside of these boundaries will not guarantee the stability of the training process. Adaptive selection of APAs in the proposed theorem has increased the chance of escaping from the local minimum traps. Implementation of the proposed theorem in predicting the future values of Mackey-Glass chaotic time series and a stochastic nonlinear system shows that the performance of the theorem is comparable with those presented in [28] and [29], and better than [23], [39], and [40]. However, the proposed theorem has slightly better performance than [28] and [29]. In terms of simulation time, the proposed theorem has a higher simulation time compared to [28], [29], and [40], but less than others. This matter is due to the high computational burden of PSO compared to the other algorithms of this study. As shown in

tables 1 and 2, Replacing FFRLS with KF can increase the noise confronting capability of the algorithm. Moreover, replacing GD with PSO reinforced this issue. Another issue is that the derivative-based algorithms may not be implementable to the models with membership functions that do not have explicit derivatives at some points. This problem has been solved by substituting PSO with the derivative-based algorithms in this study.

Acknowledgment

The authors gratefully acknowledge the ISEE journal editorial board for their work on the original version of this document.

References

- [1] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning", *Information sciences*, Vol.8, No. 3, pp. 199-249, 1975.
- [2] J. Mendel, H. Hagsras, W. W. Tan, W. W. Melek, & H. Ying, "Introduction to type-2 fuzzy logic control: theory and applications". John Wiley & Sons, 2014.
- [3] M. D. Zeiler, "ADADELTA: an adaptive learning rate method". ArXiv Preprint arXiv: 1212.5701, 2012.
- [4] S. Ruder, "An overview of gradient descent optimization algorithms", ArXiv preprint arXiv: 1609.04747, 2016.
- [5] J. S. R. Jang, E. Mizutani, "Levenberg-Marquardt method for ANFIS learning", In *Proceedings of North American Fuzzy Information Processing*, pp. 87-91, June 1996.
- [6] A. A. Ibrahim, H. B. Zhou, S. X. Tan, C. L. Zhang, J. A. Duan, "Regulated Kalman filter based training of an interval type-2 fuzzy system and its evaluation", *Engineering Applications of Artificial Intelligence*, Vol. 95, pp. 103867, October 2020.
- [7] I. Eyoh, R. John, G. De Maere, E. Kayacan, "Hybrid learning for interval type-2 intuitionistic fuzzy logic systems as applied to identification and prediction problems", *IEEE Transactions on Fuzzy Systems*, Vol. 26, No. 5, pp. 2672-2685, February 2018.
- [8] N. Vafamand, M. M. Arefi, A. Khayatian, "Nonlinear system identification based on Takagi-Sugeno fuzzy modeling and unscented Kalman filter", *ISA transactions*, Vol. 74, pp. 134-143, March 2018.
- [9] Q. Liang, J. M. Mendel, "Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters", *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 5, pp. 551-563, October 2000.
- [10] L. Bottou, "Stochastic gradient learning in neural networks", *Proceedings of Neuro-Nimes*, Vol. 91, No. 8, 1991.
- [11] C. Darken, J. Chang, J. Moody, "Learning rate schedules for faster stochastic gradient search", In *Neural networks for signal processing*, Vol. 2, 1992.
- [12] J. S. Smith, B. Wu, B. M. Wilamowski, "Neural network training with Levenberg-Marquardt and adaptable weight compression", *IEEE transactions on neural networks and learning systems*, Vol. 30, No. 2, pp.

- 580-587, July 2018.
- [13] G. T. Reddy, M. P. K. Reddy, K. Lakshmana, D. S. Rajput, R. Kaluri, G. Srivastava, "Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis", *Evolutionary Intelligence*, Vol. 13, No. 2, pp. 185-196, June 2020.
- [14] M. Almarashi, R. John, A. Hopgood, S. Ahmadi, "learning of interval and general type-2 fuzzy logic systems using simulated annealing: Theory and practice", *Information Sciences*, Vol. 360, pp. 21-42, September 2016.
- [15] H. Marzi, A. Haj Darwish, H. Helfawi, "Training ANFIS using the enhanced Bees Algorithm and least-squares estimation", *Intelligent Automation & Soft Computing*, Vol. 23, No. 2, pp. 227-234, 2017.
- [16] K. Thangavel, A. K. Mohideen, "Mammogram classification using ANFIS with ant colony optimization based learning", In *Annual Convention of the Computer Society of India*, pp. 141-152, December 2016.
- [17] S. F. E. H. Noorbin, A. Alfi, "Adaptive parameter control of search group algorithm using fuzzy logic applied to networked control systems", *Soft Computing*, Vol. 22, No. 23, pp. 7939-7960, December 2018.
- [18] S. M. A. Pahnehkolaei, A. Alfi, A. Sadollah, J. H. Kim, "Gradient-based water cycle algorithm with evaporation rate applied to chaos suppression", *Applied Soft Computing*, Vol. 53, pp. 420-440, April 2017.
- [19] E. Kayacan, E. Kayacan, M. A. 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, August 2014.
- [20] A. Ebrahimi, A. Hajipour, R. Roshanfekr, "Stator winding short circuit fault detection in three-phase Induction Motors using combination type-2 Fuzzy logic and Support Vector Machine classifier optimized by Fractional-order Chaotic Particle Swarm optimization algorithm", *Computational Intelligence in Electrical Engineering*, Vol. 12, No.1, pp. 37-48, 2021 (In Persian).
- [21] D. Simon, "Training fuzzy systems with the extended Kalman filter", *Fuzzy sets and systems*, Vol. 132, No. 2, pp. 189-199, December 2002.
- [22] M. A. Shoorehdeli, M. Teshnehlab, A. K. Sedigh, "Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter", *Fuzzy Sets and Systems*, Vol. 160, No. 7, pp. 922-948, April 2009.
- [23] M. A. Khanesar, E. Kayacan, "Levenberg-marquardt training method for type-2 fuzzy neural networks and its stability analysis", In *2015 IEEE International Conference on Fuzzy Systems*, pp. 1-7, 2015.
- [24] R. H. Abiyev, O. Kaynak, E. Kayacan, "A type-2 fuzzy wavelet neural network for system identification and control", *Journal of the Franklin Institute*, Vol. 350, No. 7, pp. 1658-1685, September 2013.
- [25] Y. Y. Lin, J. Y. Chang, C. T. Lin, "A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network and its applications", *IEEE Transactions on Industrial Electronics*, Vol. 61, No. 1, pp. 447-459, February 2013.
- [26] J. Tavoosi, A. A. Suratgar, M. B. Menhaj, "Stable ANFIS2 for nonlinear system identification", *Neurocomputing*, pp. 235-246, March 2016.
- [27] S. Feng, C. P. Chen, "Nonlinear system identification using a simplified Fuzzy Broad Learning System: Stability analysis and a comparative study", *Neurocomputing*, Vol. 337, pp. 274-286, April 2019.
- [28] M. M. Zabihi Shesh Poli, M. Aliyari Shoorehdeli, A. Moarefianpour, "Stability analysis in identification of interval type- 2 adaptive neuro- fuzzy inference system: Contribution to a novel Lyapunov function", *Expert Systems*, Vol. 36, No. 6, e12457, July 2019.
- [29] M. M. Zabihi Shesh Poli, M. Aliyari Shoorehdeli, A. Moarefianpour, " Lyapunov stability analysis in the training of type 2 neuro-fuzzy inference system with a hybrid algorithm based on gradient descent and Kalman filter", *Modeling in engineering*, 2021 (In Persian), doi:10.22075/JME.2021.22189.2017.
- [30] M. M. Jerry, "Uncertain rule-based fuzzy systems: Introduction and New Directions", Springer, USA, 2019.
- [31] J. M. Mendel, R. I. John, F. Liu, "Interval type-2 fuzzy logic systems made simple", *IEEE transactions on fuzzy systems*, Vol. 14, No. 6, pp. 808-821, November 2006.
- [32] N. N. Karnik, J. M. Mendel, "Centroid of a type-2 fuzzy set", *Information Sciences*, Vol. 132, No. 1-4, pp. 195-220, February 2001.
- [33] J. M. Mendel, R. B. John, "Type-2 fuzzy sets made simple", *IEEE Transactions on fuzzy systems*, Vol. 10, No. 2, pp. 117-127, 2002.
- [34] D. Wu, J. M. Mendel, "Enhanced karnik--Mendel algorithms", *IEEE transactions on fuzzy systems*, Vol. 17, No. 4, pp. 923-934, April 2008.
- [35] M. B. Begian, W. W. Melek, J. M. Mendel, "Stability analysis of type-2 fuzzy systems", In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pp. 947-953, September 2008.
- [36] C. I. Li, G. D. Chen, T. Y. Sung, H. F. Tsai, "Novel Adaptive Kalman Filter with Fuzzy Neural Network for Trajectory Estimation System", *International Journal of Fuzzy Systems*, Vol. 21, No. 6, pp. 1649-1660, September 2019.
- [37] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43, October 1995.
- [38] V. Kadiramanathan, K. Selvarajah, P. J. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer", *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp. 245-255, May 2006.
- [39] M. A. Khanesar, E. Kayacan, M. Teshnehlab, O. Kaynak, "Levenberg marquardt algorithm for the training of type-2 fuzzy neuro systems with a novel type-2 fuzzy membership function", In *2011 IEEE symposium on advances in type-2 fuzzy logic systems*, pp. 88-93, July 2011.
- [40] M. A. Khanesar, E. Kayacan, M. Teshnehlab, O. Kaynak, "Extended Kalman filter based learning algorithm for type-2 fuzzy logic systems and its experimental evaluation", *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 11, pp. 4443-4455, May 2011.
- [41] M. C. Mackey, L. Glass, "Oscillation and chaos in physical control system". *Science*. Vol. 197, No. 4300, pp. 287-289, August 1977.

