# Hydraulic Actuator Identification Using Interval Type-2 Fuzzy Neural Networks

Mohsen Vatani, Salman Ahmadi, and Saeid Khosravani

*Abstract*—**In recent years, intelligent based approaches have been introduced as one of the best potential methods for solving many problems in control literature. Neural Networks (NN) and Fuzzy Logic are widely used in nonlinear system modeling and identification. These approaches require a high number of model parameters, which impose more complex computation. Using Interval Type-2 Fuzzy Neural Network (IT2FNN) method, one needs considerably fewer numbers of required parameters. It can also model uncertainty and nonlinearity of the system much more effectively. In this paper, we suggest to use this neuro – fuzzy based network for nonlinear modeling of a hydraulic actuator. Simulation studies of this challenging benchmark confirm the excellent nonlinear modeling properties of the IT2FNN.**

*Index Terms*—**Nonlinear systems, identification, type-2 fuzzy neural network, hydraulic actuator.**

## I. Introduction

System identification is often the first and the critical step in process control, prediction of behavior, fault detection, estimation of immeasurable variables and improving understanding of system behavior. In the control literature, there exist a large number of active researches and various well developed algorithms for linear system identification [1]. However, nonlinear system identification is far from mature both in theory and in practice [2],[3]. When linear models are not appropriate for accurately describing the system dynamic, nonlinear identification should be employed. Because the structure of nonlinear systems is so rich and complicated, it is not expected that a single method could be effectively applied to all nonlinear systems. Indeed, in many cases, knowledge of the physical phenomena involved is often incomplete, some critical variables may not have been measured, and some physical parameters may be unknown. The model must then be determined from observed data.

Several methods have been developed for non-linear system identification. Early works begin by considering the functional series of Volterra. For linear systems, convolution integral models the input-output data set. However, for nonlinear systems, Volterra series serves a generalization of the convolution integral. In [4] wiener series have been used for identification of nonlinear systems. However, by using this method, identification of even a simple system that contains second-order nonlinearity would require the evaluation of, typically, coefficients [3].

Manuscript received April 6, 2012; revised May 20, 2012.

Mohsen Vatani, Salman Ahmadi and SaeidKhosravani were with the Department of Electrical Engineering, Amirkabir University of Technology, 424 Hafez Ave, Tehran, 15875-4413, Iran (e-mail: m.vatani@aut.ac.ir, ahmadi_salman@aut,ac,ir, khosravani_s@aut.ac.ir).

Other methods, including NARMAX and Hammerstein improve mentioned methods. However, it is often difficult to represent the behavior of the system over its full range of operation using such structures.

New methods of system identification are required for such conditions and recently new identification methods have been proposed based on fuzzy logic, neural networks and wavelets [2], [5].

During the past decade, intelligent methods have been found to hold the best potential to solve many engineering problems, which could not be solved before. Fuzzy modeling is an approach which may provide a solution to such problems. Indeed, fuzzy models are regarded as universal approximators of multi-variable functions [6], [7] which can be used to integrate qualitative knowledge. High performance of this approach makes it very attractive in many non-linear system methods. Fuzzy models are based on rules such as "if premise then consequence", where premises evaluate the model inputs, and consequences provide the value of the model output.

On the other hand, fuzzy neural network (FNN) has been developed during the past few years by many researchers to improve the intelligent methodologies with better learning capabilities. In particular, the back propagation (BP) of FNN has been developed to tune the parameters of fuzzy sets and the weighting factors of neural network. Optimization of learning rate for type-1 FNN has been proposed to accelerate the convergence of the BP algorithm. Analysis of stable and optimal learning rates for type-1 FNN was also discussed rigorously as well [6].

In recent years, studies on type-2 FLSs have drawn much attention and a complete theory of type-2 FLSs has been developed [8]. Type-2 FLSs are extensions of type-1 FLSs, where the membership value of a type-2 fuzzy set is a type-1 fuzzy number. Decision making, survey processing, time-series forecasting, time varying channel equalization, control of mobile robots and preprocessing of data are just some of the application areas of Type-2 FLS.

This paper is organized as follows. First, Type-2 Fuzzy Logic system is introduced. Then IT2FNN and its learning method are analyzed in section II. In the next section, nonlinear model of test case benchmark of a hydraulic actuator is described and a comparison between typical identification methods and IT2FNN method is done. Section IV concludes the results and suggests possible future works.

## II. Interval Type-2 Fuzzy Neural Network

### A. Type-2 Fuzzy Logic Systems (T2FLS)

The concept of type-2 fuzzy set was introduced by Zadeh as an extension of the ordinary fuzzy set (type-1 fuzzy set) [9]. Type-2 fuzzy sets are fuzzy sets whose membership

functions are themselves type-1 fuzzy sets. Due to their high level of uncertainty, Type-2 FLSs can be used when the plant is too uncertain to determine exact MF. For example, when the sampling data are corrupted by noise, T2FS is a good alternative to model the system. Moreover, when the nonlinearity of system increases, the T2FLS can handle the large computational complexity of the model.

The membership function of the type-1 fuzzy variables can be any subset in [0, 1]. In Type-2 fuzzy Logic Set, for each primary membership, there is a secondary membership in [0, 1], that defines the possibilities for the primary membership [10]. Type-2 FLSs are computationally intensive because the process of type reduction is very complex. To simplify the computation, the possibility of the secondary membership function (MFs) can be set to 1. If so, the type-2 FLSs become interval type-2 FLSs. A Gaussian primary MF with uncertain mean and fixed standard deviation (see Fig.1) can be used as an IT2FS. Equation.1 will be used to show the MF of an IT2FS:

$$\mu_A(x) = exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right] \equiv N(m,\sigma;x) \qquad (1)$$
$$m \in [m_1, m_2]$$

As in a type-1 fuzzy system, for a Type-2 Fuzzy Logic System, we can define fuzzifier, a rule base, a fuzzy inference engine, and an output processor. Once the rules have been established, a Fuzzy Logic System can be viewed as crisp inputs to crisp outputs mapping. This mapping can be shown as y=f(x). The structure of rules in the T2FLS and its inference engine is similar to T1FLS. We should define unions and intersections of T2FLS as well as compositions of type-2 relations. The output of this system is a type-1 fuzzy function, thus it has more information than the output of a type-1 fuzzy system. Several defuzzification methods have been introduced to convert the output of T2FS to a crisp number.

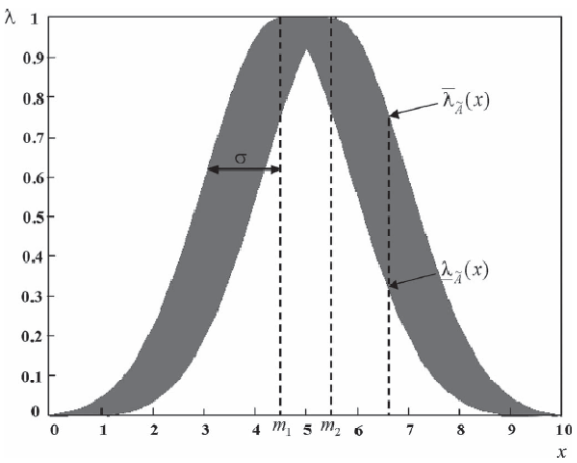


Fig. 1. Type-2 Fuzzy membership function

### B. Interval Type-2 Fuzzy Neural Network (IT2FNN)

Soft computation is a combination of methods, which are subjected to imprecision and uncertainty. They provide a basis for design and use of intelligent systems. The principal methods in this concept are fuzzy logic, neural network, evolutionary algorithm and machine learning. In general, experiences show that better results can be obtained by using these methodologies altogether [11]. In this perspective, neurofuzzy systems can be seen as a combination of neural networks and fuzzy logic. It uses neural networks, the techniques related to learning and approximation, and fuzzy logic, the techniques related to uncertainty. Therefore, a lot of research was conducted using FNNs to represent complex plants and construct advanced controllers. However, all of these analyses focused on the type-1 FNN.

T2FNN can be similarly defined as of type-1 FNN. T2FNN consists of the interval type-2 fuzzy linguistic language as the antecedent part and Neural Network layers as the consequent part. As mentioned before, the Type-2 Fuzzy Logic is computationally intensive. Therefore, in this paper, the interval T2FNN is used to simplify the computational process. This leads to a real time response of the net. The IT2FNN is a multilayer network for the realization of the type-2 fuzzy inference system, and it can be constructed from a set of type-2 fuzzy rules.

A typical IT2FNN consists of 6 layers [8]. In the first layer (input layer), the crisp values of input data normalize to [-1, 1]. Second layer performs the fuzzification task. For each input, (1) is used in this process and the output of each node can be represented as a bounded interval in terms of lower membership function and upper membership function, i.e. $[\underline{\mu}_A, \overline{\mu}_A]$.

$$\overline{\mu}_A(x) = \begin{cases} N(m_1, \bar{\sigma}\ x) & x < m_1 \\ 1 & m_1 < x < m_2 \\ N(m_2, \sigma; x), & x > m_2 \end{cases} \qquad (2)$$

$$\underline{\mu}_A(x) = \begin{cases} N(m_2, \bar{\sigma}\ x) & x < \dfrac{m_1+m_2}{2} \\ & \\ N(m_1, \sigma; x), & x > \dfrac{m_1+m_2}{2} \end{cases} \qquad (3)$$

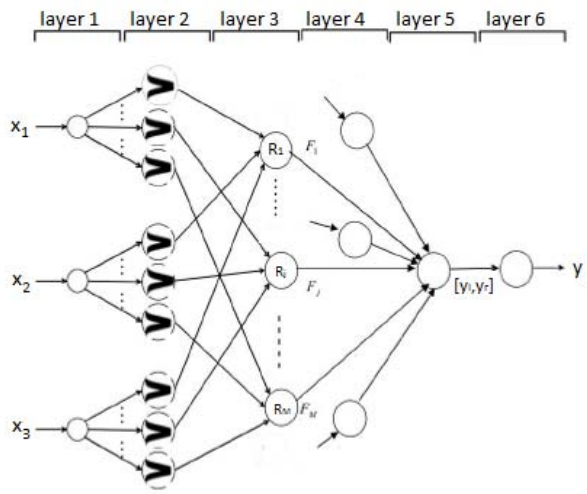


Fig. 2. Structure of interval type-2 fuzzy neural network

We only consider singleton input fuzzification throughout this paper. In firing layer (third layer), each node is a rule

node that performs the fuzzy operation using algebraic production. The output of these nodes is the firing strength.

$$F^i = \left[\underline{f}^i, \overline{f}^i\right] \tag{4}$$

$$\overline{f}^i = \prod_{j=1}^{n} \overline{\mu}_A \quad , \underline{f}^i = \prod_{j=1}^{n} \underline{\mu}_A \tag{5}$$

Each node in layer 4 (consequent layer) is called a consequent node. The output of each consequent node is a type-1 fuzzy set denoted by $[w_l, w_r]$. Next layer, called output processing layer, computes bounds of output parameters.

This layer handles more computational load than other layers. We use the algorithm proposed in [6] to compute the lower and upper bound of output $[y_l, y_r]$. In the last layer, nodes compute the output linguistic variable using a defuzzification operation. By computing the average of upper and lower bound of output, the type-1 fuzzy set of previous layers defuzzify in this layer:

$$y = \frac{y_l + y_r}{2} \tag{6}$$

### C. Back Propagation Equations for the IT2FNN

We use BP method to train the net [12]. For P input-output data, the following error function should be minimized:

$$\boldsymbol{e} = \frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{d}\right]^2 \tag{7}$$

By using chain rule to compute the derivative of error function, we can derive the mean of the Gaussian membership function by following equation, where α represents learning rate of BP method.

$$m_k(p+1) = m_k(p) - \alpha \left.\frac{\partial e}{\partial m}\right|_p$$
$$m_k(p+1) = m_k(p) - \alpha \left.\left[\frac{\partial e}{\partial y}\frac{\partial y}{\partial y_l}\frac{\partial y_l}{\partial m_k} + \frac{\partial e}{\partial y}\frac{\partial y}{\partial y_r}\frac{\partial y_r}{\partial m_k}\right]\right|_p \tag{8}$$

The same method can be used to calculate weighting factors. The weighting factors $w_l$ *or* $w_r$ depends on which branch is active in the process of calculating left-most point $y_l$ or right-most point $y_r$.

$$w_l(p+1) = w_l(p) - \alpha \left.\left[\frac{\partial e}{\partial y}\frac{\partial y}{\partial y_l}\frac{\partial y_l}{\partial w_l} + \frac{\partial e}{\partial y}\frac{\partial y}{\partial y_r}\frac{\partial y_r}{\partial w_l}\right]\right|_p \tag{9}$$

## III. Simulation Results

In order to demonstrate the effectiveness of the method, we use Interval Type-2 Fuzzy Neural Network for nonlinear identification of the hydraulic robot arm as a benchmark test case.

This data set consists of input and output time series of a hydraulic actuator of a mechanical structure known as crane. This equipment has actually four actuators: one to rotate the whole structure, one to move the arm, one to move the forearm and one to move a telescopic extension of the forearm.

This plant was chosen because of its long arm and long forearm with considerable flexibility on the mechanic structure, which makes the movement of the whole crane oscillative and hard to control.

The position of the arm is controlled by a hydraulic actuator. The oil pressure in the actuator is controlled by the size of the valve opening through which the oil flows into the actuator. The position of the robot arm is then a function of the oil pressure. Thus, we have a very oscillative settling period after a step change of the valve size. These oscillations are caused by mechanical resonances in the robot arm [2].

The generalization capability and accuracy of regression algorithms can be evaluated using the normalized mean-squared estimation (NMS) error:

$$NMSE = \frac{\sum_{k=1}^{M} e^2(k)}{M \cdot \hat{\sigma}_u^2} \tag{10}$$

where $\hat{\sigma}_u^2$ is the variance of the measured values of the valve position $(\{u(k)\}_{k=1}^{M})$ and $M$ is the length of the sequence of residues.

Input time series $\{u(k)\}_{k=1}^{M}$ and output time series (oil pressure {y (t)}) are shown in Fig. 3.

For the sake of comparison, we use the same regressor as that in [13],

$$x(t) = [u(t-1) \quad \cdots \quad u(t-5) \quad y(t-1) \quad \cdots \quad y(t-4)]^T$$
$$y(t) = f\left(u(t-1), \cdots, u(t-5), y(t-1), \cdots, y(t-4)\right) \tag{11}$$

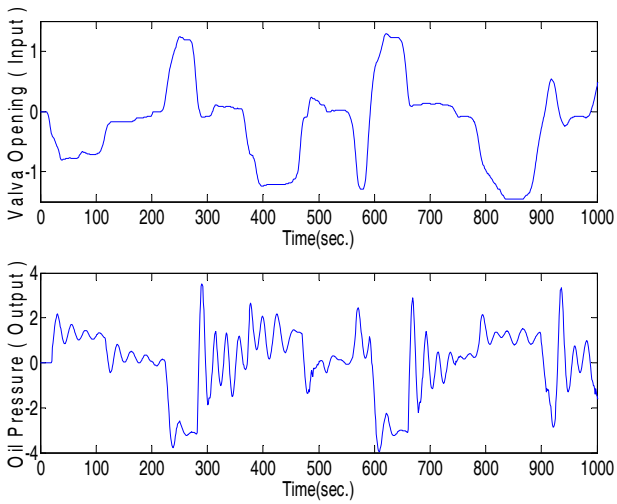


Fig. 3. Measured values of valve opening and oil pressure

We use some typical methods to demonstrate effectiveness of IT2FNN. MLP – Bp1 (or Bp2) is multi-layer perceptron with one (or two) hidden layer using back-propagation (BP) learning algorithm and MLP –LM is multi-layer perceptron with one hidden layer using levenberg–marquardt learning algorithm. The obtained results are shown in Table I. It contains mean, minimum, maximum and variance of the NMSE values, measured along the 100 training runs.

TABLE I: COMPARISON OF IDENTIFICATION METHODS

| ***Models*** | ***NMSE*** | | | |
|---|---|---|---|---|
| | *Mean* | *Min* | *Max* | *Variance* |
| ***IT2FNN*** | 0.0027 | 0.0025 | 0.0028 | $2.37 \times 10^{-8}$ |
| ***MLP– LM*** | 0.0722 | 0.0048 | 0.3079 | 0.0041 |
| ***MLP – Bp1*** | 0.2489 | 0.2 | 0.2961 | $2.6 \times 10^{-5}$ |
| ***MLP – Bp2*** | 0.2425 | 0.0676 | 1.8611 | 0.6642 |

We consider 15 neurons in the hidden layer of MLP which provide the smallest generalization error. For the MLP-Bp2, the number of neurons in the second hidden layer is set to 5 and in the first one is set to 10. MLP-LM models have 15 hidden neurons. The MLP-based models were trained with constant learning rate equal to 0.1. The IT2FNN was trained with constant learning rate equal to $\alpha = 2$. The identification results are illustrated in Figs. 4. In IT2FNN, 19 parameters are estimated to obtain theprecise model, however, as we mentioned before, other methods require estimating a large number of parameters for identification. It can be seen that system output is consistent with the IT2FNN output (see Fig 4). The error signal is portrayed in Fig. 5. Simulation results show that performance of the IT2FNN is far better than the other methods.

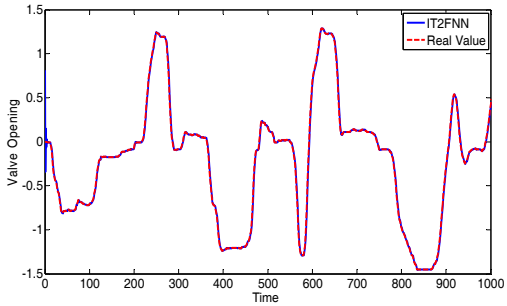


Fig. 4. Comparison between real values and output of IT2FNN

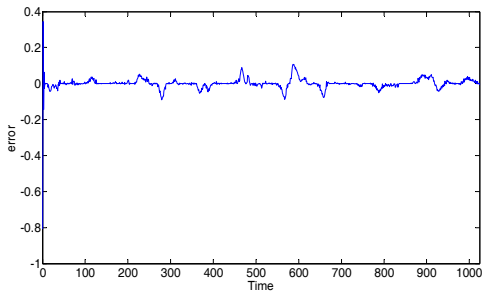


Fig. 5. Error between real and the Type-2 FLS output

## IV. CONCLUSION

In this paper, an extremely effective method for nonlinear system identification is used for hydraulic actuator modeling. The ability of IT2FNN to approximate nonlinear and uncertain systems or functions accurately makes it an appropriate choice for identifying such system. Due to the high nonlinearity of the considered system, the MLP based methods are not adequate to identify a precise model. We showed that IT2FNN leads to better results with fewer estimated parameters.

## REFERENCE

[1] L. Ljung, System Identification: Theory for the User, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[2] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol.31, pp. 1691–1724, 1995.

[3] S. A. Billings, "Identification of nonlinear systems—A survey," in *Proc.Inst. Elect. Eng.*, vol. 127, p. 272, 1980

[4] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley. & Sons, 1980

[5] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjoberg, and Q. Zhang, "Nonlinear block-box models in systemidentification: Mathematical foundations," *Automatica*, vol. 31, pp.1725–1750, 1995.

[6] C.-H. Wang, C. S. Cheng, and T. T. Lee, "Dynamical Optimal Training for Interval Type-2Fuzzy Neural Network (T2FNN)," *IEEE Trans. Syst., Man, Cybern*, vol. 34, no 3, pp. 1462-1477, June 2004

[7] L. Wang and R. Langari, "Complex systems modeling via fuzzy logic," *IEEE* Trans. *Syst, Man, Cybern.*, vol. 26, pp. 100–106, Feb. 1996.

[8] C. F. Juang and Y. W. Tsao "A Self-Evolving Interval Type-2 Fuzzy Neural Network With Online Structure and Parameter Learning," *IEEE Trans on Fuzzy Systems*, vol. 16, no. 6, pp. 1411-1424, December 2008

[9] J. Mendel, " Type-2 Fuzzy Sets and Systems: An Overview," *IEEE* computational *Intelligence Magazine*, vol. 2, no. 1, pp. 20-29, February 2007.

[10] F. J. Lin and P. H. Chou, "Adaptive Control of Two-Axis Motion Control System Using Interval Type-2 Fuzzy Neural Network," *IEEE Trans. On Industrial Electronics*, vol. 56, no. 1, pp. 178-193 January 2009

[11] P. Liu and H. Li," Fuzzy Neural Network theory and applications," *Series in* Machine Perception and Artificial Intelligence, vol. 59, World Scientific Publishing, 2007

[12] H. Hagras, "Comments on Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN) ," *IEEE Trans on Systems, Man and Cybernetics- Part B: Cybernetics*, vol. 36, no. 5, October 2006

[13] L. Gustavo M. Souza, and G. A. Barreto, "On building local models for inverse system identification with vector quantization algorithms," Neurocomputing, vol. 73, pp.1993–2005, 2010.