academicJournals

Vol. 8(21), pp. 875-884, 4 June, 2013
DOI 10.5897/SRE11.1538
ISSN 1992-2248 © 2013 Academic Journals
http://www.academicjournals.org/SRE

Scientific Research and Essays

Full Length Research Paper

Simulation of the adaptive neuro-fuzzy inference system (ANFIS) inverse controller using Matlab S-function

Tarek Benmiloud

University of Science and Technology of Oran, 1505 Elmnaouar Oran-Algeria.

Accepted 23 April, 2013

In this paper, for the purpose of simulating the mathematical model of adaptive neuro-fuzzy inference system (ANFIS), we use Matlab/Simulink environment with its powerful S-functions. The simulated model of ANFIS network can be then used to make the simulation of the identification and the control of linear or nonlinear systems. Created Simulink block of ANFIS give flexible exploitation of parameters of ANFIS network like learning rates and initial local parameters. We use the S-function of ANFIS to make the direct-inverse adaptive control of DC-motor. The obtained results of Direct-inverse control of DC-motor are compared with these of a classical controller which is the simplest type of controller, selected only to check the effectiveness of the proposed intelligent controller in terms of control performances and disturbance rejection.

Key words: Direct inverse control, adaptive neuro-fuzzy inference system (ANFIS) controller, S-function of Matlab.

INTRODUCTION

Jang (1993) proposed the famous adaptive neuro-fuzzy inference system (ANFIS), which is one of the best in function approximation among the several neuro-fuzzy models (Hiroki et al., 2009). It has been successfully applied in various fields (identification, prediction and control). To design ANFIS based controllers Matlab provides several tools, Simulink based ANFIS toolbox (Howard and Mark, 1997). These tools support offline simulation process. It means prior to use the ANFIS model we need to simulate and train the ANFIS by presenting the input and target data set. The main idea of this paper is to provide a custom tool to investigate better usage of ANFIS Networks. Most of the simulation tools suggest their own architecture and training algorithms which are fixed and uses offline training. The proposed Simulink model is implemented using S-function which allows making online training of systems. It can be applied to any linear or non-linear system. Moreover learning rates and step time can be easy modified. The created S-function of ANFIS is used to make the control of DC-motor.

ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) ARCHITECTURE

ANFIS (Figure 1) (Jang, 1993) makes use of a hybrid learning rule to optimize the fuzzy system parameters of a first order Sugeno system. The output of the nodes in each respective layer is represented by O_i, where i is the i^{th} node of layer l. The following is a layer by layer description of a two inputs first order Sugeno system (Jang, 1993; Hongxing et al., 2001). The 1st layer is for fuzzification of the input variables. It generates the

E-mail: tarek_bm2004@yahoo.fr.

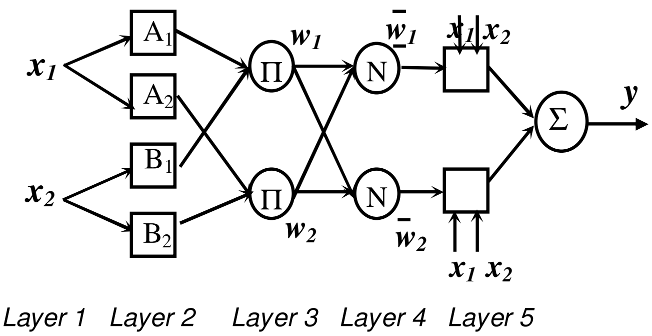

Figure 1. ANFIS architecture

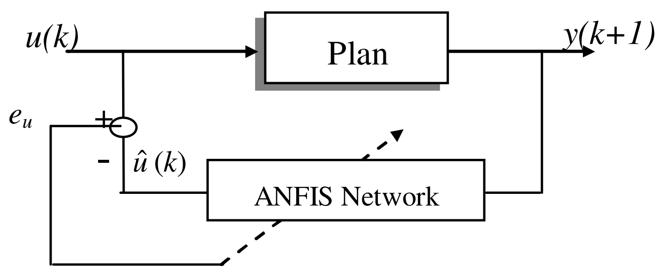

Figure 2. Block diagramme for on-line inverse learning using ANFIS.

membership grades:

$$O_i^1 = g(x) \quad (1)$$

Where g; is the membership function (MFs) of adaptive neuro-fuzzy system. The second layer generates the firing strength:

$$O_i^2 = w_i = \prod_{j=1}^{m} g(x) \quad (2)$$

The third layer normalizes the firing strengths.

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2} \quad (3)$$

Layer 4 calculate rule outputs based on the consequent parameters.

$$O_i^4 = y_i = \overline{w}_i \cdot f_i = \overline{w}_i (p_i \cdot x + q_i \cdot y + r_i) \quad (4)$$

Layer 5 sums all the inputs from layer 4. This is the overall output of the ANFIS system.

$$y = O_i^5 = \sum_i y_i = \sum_i \overline{w}_i \cdot f_i = \sum_i \overline{w}_i (p_i \cdot x + q_i \cdot y + r_i) \quad (5)$$

The learning procedure of ANFIS network has got two steps (Jang, 1993; Carrano et al., 2008): In the 1st step the input patterns are propagated, and the optimal consequent parameters are estimated by an iterative least mean square procedure (Xuan, 2006; Farzad et al., 2005). In the 2nd step the patterns are propaged again, and back-propagation is used to modify the local parameters (the values which compose each membership functions A1, A2, B1, B2). This procedure is then iterated until the error criterion is satisfied (Jang, 1993). The consequent parameters thus identified, are optimal under the condition that the local parameters are fixed. The form of the last layer is:

$$Y = X * W \quad (6)$$

Where X is a vector of predictors, and *W* is the vector of regression parameters to be estimated. To make the correction of local parameters of ANFIS network, ANFIS network uses the sum of the gradient of the error e_y of the output. The signal of error is back-propagated and local parameters are updated. We have the formula of modification of the first local parameter of the first membership function of ANFIS network (Hongsheng and Feng, 2007);

$$a_i(t+1) = a_i(t) - \frac{h}{p} \cdot \frac{\partial e_y}{a_i} \quad (7)$$

Where *h*; is the learning rate for local parameter a_i

The following rule is used to calculate partial derivatives, employed for update of the parameters of membership function.

$$\frac{\partial E}{\partial a_i} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_i} \cdot \frac{\partial y_i}{\partial w_i} \cdot \frac{\partial w_i}{\partial g} \cdot \frac{\partial g}{\partial a_i} \quad (8)$$

DIRECTE-INVERSE ADAPTIVE CONTROL ANFIS NETWORK

Identification and control using adaptive neuro-fuzzy inference system (ANFIS)

The training mode that will be used is the inverse model (Toha and Tokhi, 2009; Gonzalez-Gomez et al., 2011; Kasuan et al., 2011) of the DC-motor. In this case, The input of the ANFIS network will be the output *y(t)* of the DC-motor (its speed) and the output of the ANFIS network will be the estimation of the signal of the control of the motor $e_u(t)$, as shown in Figure 2. We know that the adaptive control of systems is constituted by two loops; one loop of control having a regulator with adjustable parameters, and a second loop which acts on the

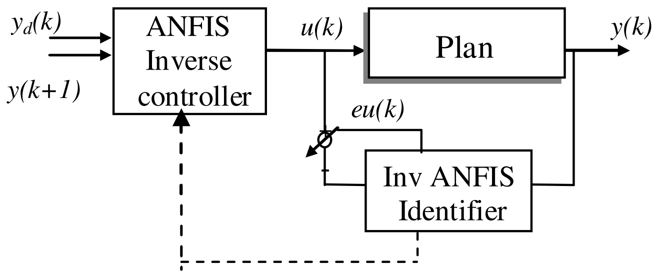

Figure 3. Block diagram of the direct-inverse adaptive control

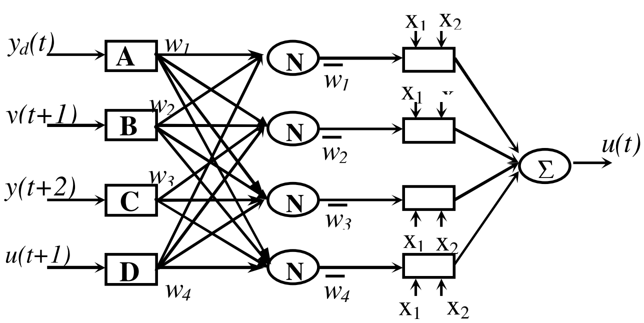

Figure 4. Topology of the ANFIS Network used in the DC Motor control.

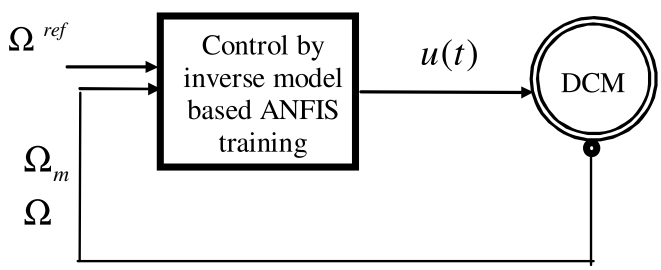

Figure 5. Speed control of DCM with online identification of the inverse model based ANFIS Network.

parameters of the regulator, to maintain parametric variations. The structure of this direct inverse control is shown by Figure 3.

Choice of the model of the adaptive neuro-fuzzy inference system (ANFIS) controller

To represent non-linear processes, several structures of models of type non-linear black boxes were developed as: FIR, NARX, NOE, NARMAX. These models can be used in order with the neuro-fuzzy network (Azeem et al., 2000; Denai et al., 2004; Toha and Tokhi, 2009; Kasuan et al., 2011). To identify the inverse model of the MAS, we are going to use the model NARX (non-linear, autoregressive with exogenous input, which is the most used for its simplicity and its not-recursive structure. The predictor is given by the following formula:

$$\hat{y}(t) = \phi(\hat{y}(t-1), \ldots, \hat{y}(t-ny), u(t-1), \ldots, u(t-nu)) \quad (9)$$

$\hat{y}$: output of ANFIS network,
u : input of the ANFIS network

The regression consists of outputs and the past inputs. The function $\varphi(.)$ is the nonlinear function which we want to approximate. The choice of the number of inputs of ANFIS is not deducted of a main rule, but by successive tries (Figure 4).

The used ANFIS network for the Direct-inverse adaptive control contains 4 neurones in the input layer with triangular functions as membership function. The output of the ANFIS network is the signal of control *u(t)* of the DC-motor. The first input of the ANFIS network is the reference $y_d(t)$, the other inputs are; the output of the DC-motor with a delay *y(t+1*), the output with two delay *y(t+2*), and the signal of control *u(t+1*).

Adaptive neuro-fuzzy inference system (ANFIS) control of the DC-motor

The control based on the adaptive ANFIS network uses the inverse model of the DC-motor (identified by an ANFIS), to control the speed of this motor. It is direct inverse control; the identified inverse model is directly used as controller. It can be a real-time identification or separate time identification. When the identification is real time made the control is adaptive. To control DC-motor MAS, we will use an adaptive control with identification of the inverse model of the system (Figure 5). The stability of the control of the DC-motor using the intelligent controller is ensured since the model DC-motor has a stable model, so, if the inverse model of the induction motor is a good estimation of this model, the total system (DC-motor and the inverse controller) must be a stable system.

SIMULATION OF ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) NETWORK USING S-FUNCTION

For the simulation purpose, MATLAB/SIMULINK version 6.5 is used. For the simulation ode15 s (stiff/NDF) solver is used. The simulation results were taken for 80 s. For each step time, ANFIS provided the value which make the learning of inverse model of the DC-motor to obtain the error e_u *(t*), from which the signal of control *u(t)* is calculated by ANFIS controller. ANFIS is trained via back

Table 1. Parameters of DC-motor.

Parameter	Symbol	Value
Resistance	*Ra*	0.65 Ω
Inductance	*La*	0.0117 H
Inertia mass	*J*	0.652 kg/m^2
Friction	*F*	0.0587 N

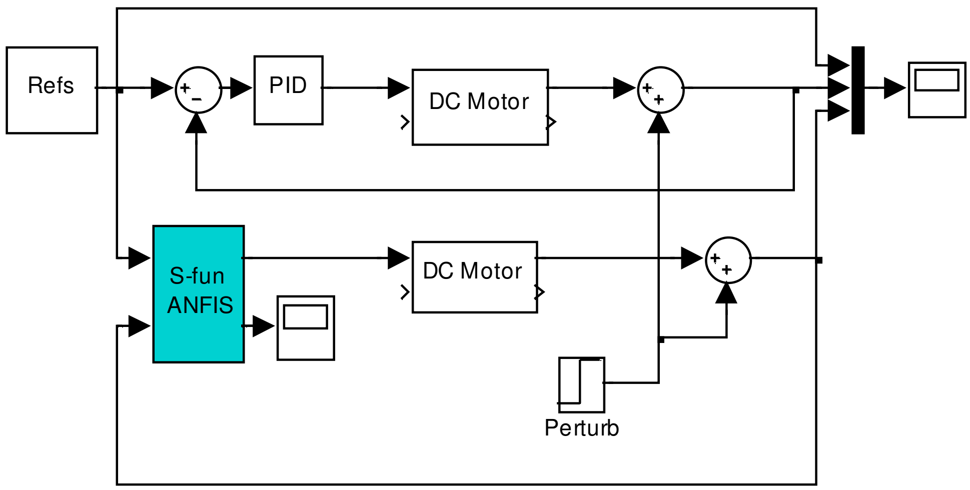

Figure 6. Simulink model of direct-inverse adaptive control based ANFIS Learning.

propagation algorithm (Jang, 1993; Eleftherios, 2012). The direct-inverse control of the DC-motor based on ANFIS learning is designed to make the output speed track the reference $\Omega_{.ref}$ and at the same time achieve the desired dynamic performance. To verify the effectiveness of the ANFIS controller we use the PID controller. The aim of using the PID controller is to validate the results of the ANFIS controller in terms of rapidity and precision as well as the disturbance rejection. The parameters of the DC-motor are given in Table 1. The parameters of the PID controller are arbitrarily chosen to have rapid response of the speed of the DC-motor. We take: K_p = 150, K_i= 7, K_d=13.

Figures 7 and 8 give the response of the speed of the DC-motor with direct-inverse adaptive control based ANFIS training and PID controller. Figure 8 shows the zoomed version of the Figure 7. Figures 9, 10 and 11 show the evolution of consequent and local parameters during the control (and the training of the model) of the DC-motor. Figures 12 and 13 give the response of the speed of the DC-motor with direct-inverse control and PID controller with application of load torque T_L= 5 *Nm* at the instant *t* =47 *s*. Figures 14 and 15 give the response of the load torque of the DC-motor using the two controllers; direct-inverse controller and the PID controller. The convergence rate is mainly affected by the four Learning rates parameters, in the Simulink block (Figure 6) we can change these by simply changing the gain values of the S-function block in Appendix. In present context it is set to 0.02.

The simulation result obtained shows that the speed of DC-motor tracks rapidly its reference (Figures 7 and 12) with static error of tracking almost zero for the direct-inverse control based ANFIS training, what is not the case of the PID controller which, in spite of we have chooses high gains does not allow the cancelation of the static error. Direct-inverse control based ANFIS learning keep very small static error (Figure 13) despite the application of the load torque. We noted small variation of local parameters of ANFIS (Figure 10 and 11) during the control of the DC-motor which is not the case of consequent parameters (Figure 9). Figures 14, 15, 16 and 17 show that direct-inverse control based ANFIS learning give best response of load torque that these obtained by PID controller which present high values of picks during the variation of speed references.

CONCLUSION

A direct-inverse adaptive control based ANFIS learning is proposed as Matlab-Tool for identification and control of dynamic systems. The proposed Simulink block is implemented as S-function of MATLAB software. The application of this block to control of speed of DC-motor gives good results of tracking of speed and load torque.

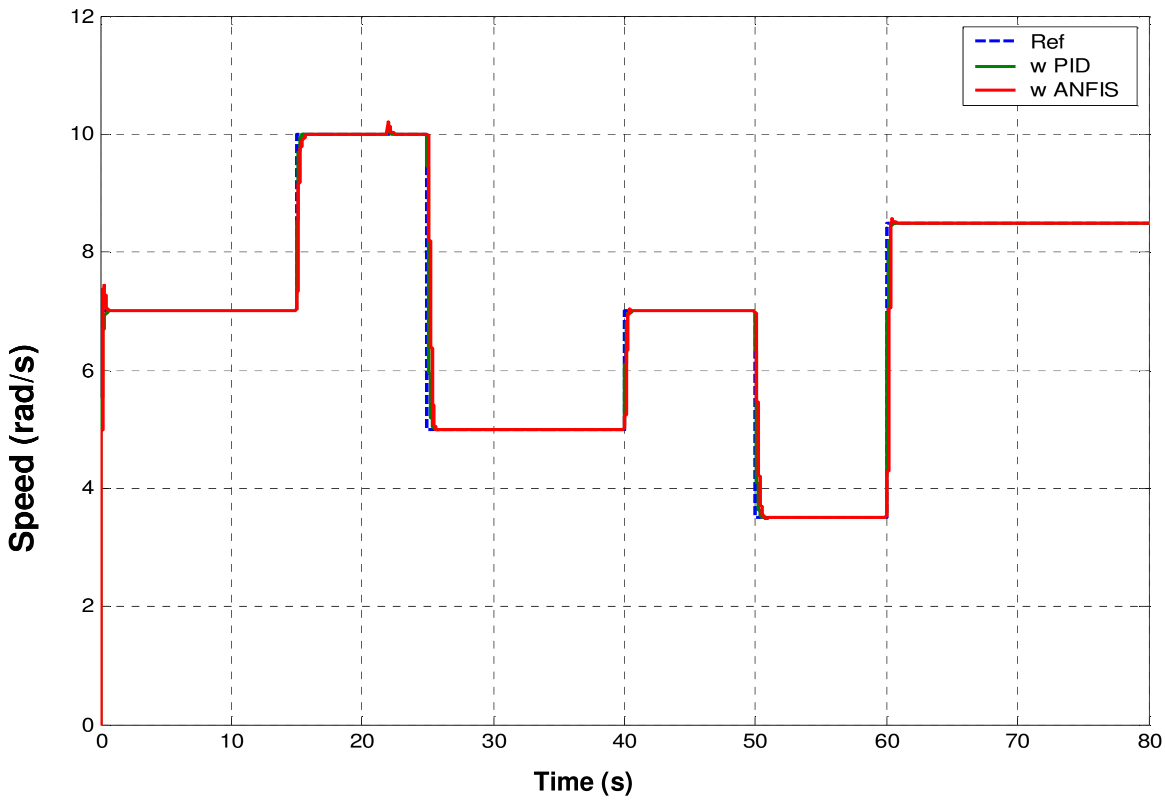

Figure 7. Matlab response of speed with direct-inverse control based ANFIS learning and PID controller.

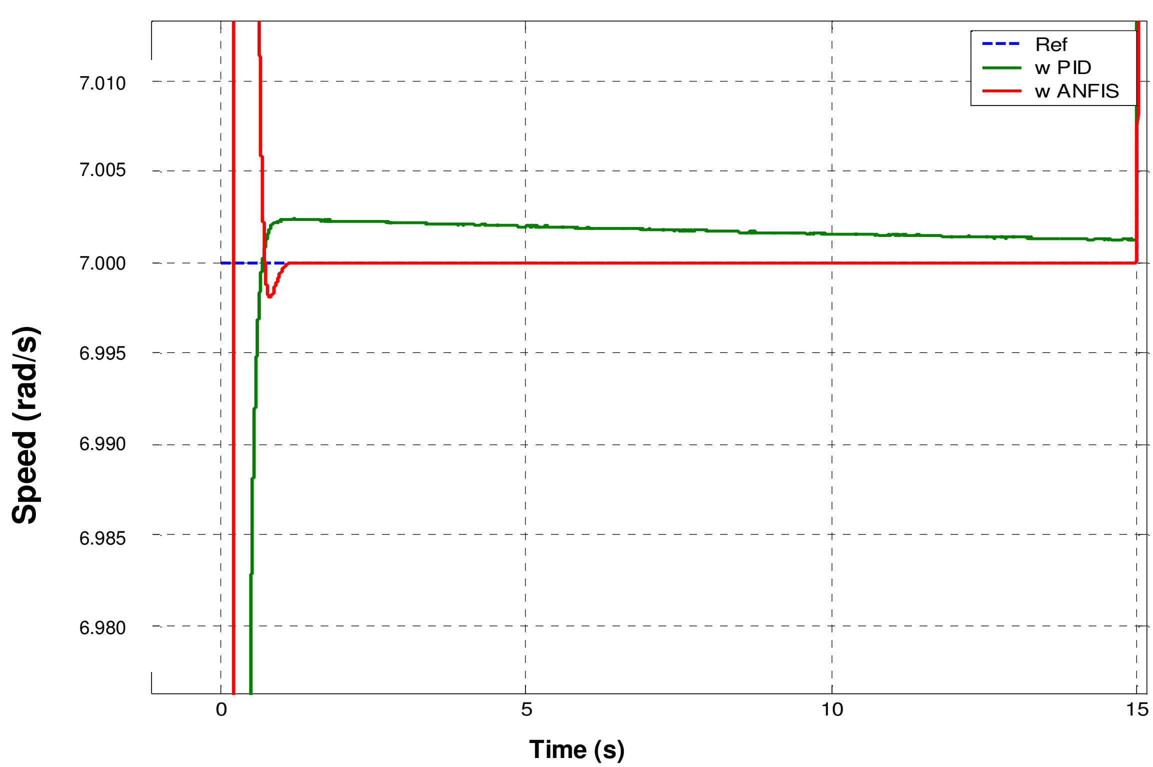

Figure 8. Speed of the DC-motor with direct-inverse control and PID controller (zoomed).

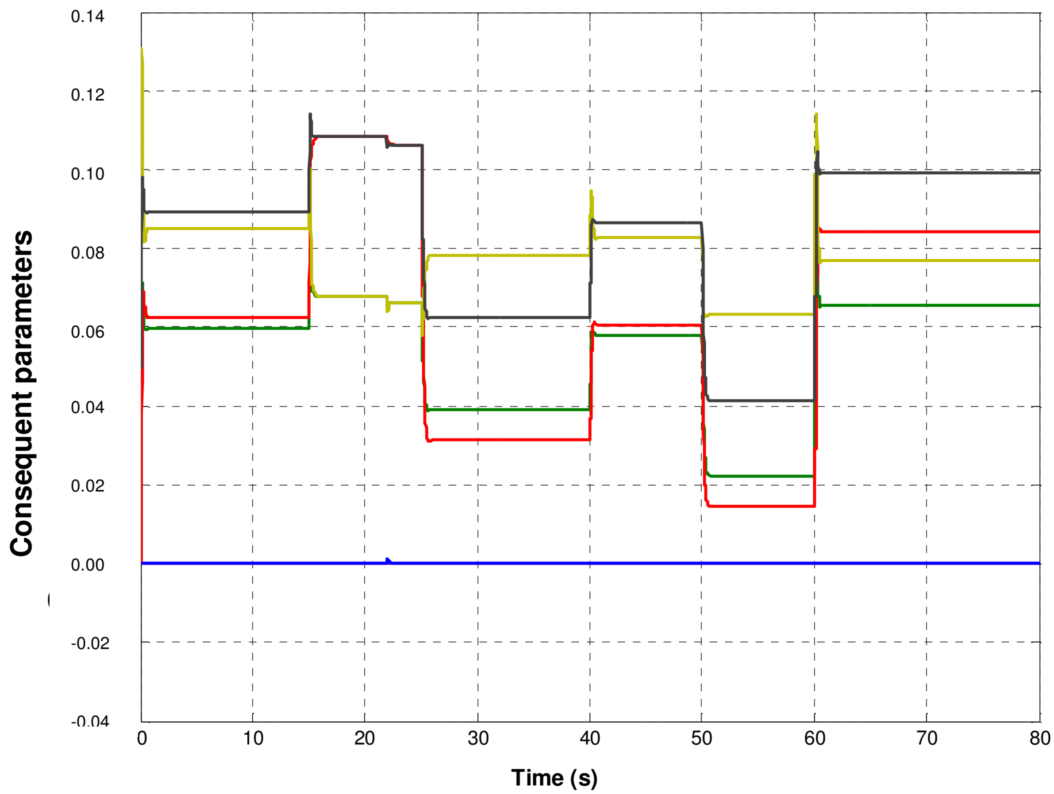

Figure 9. Evolution of consequent parameters of ANFIS during the direct-inverse control of the DC-motor.

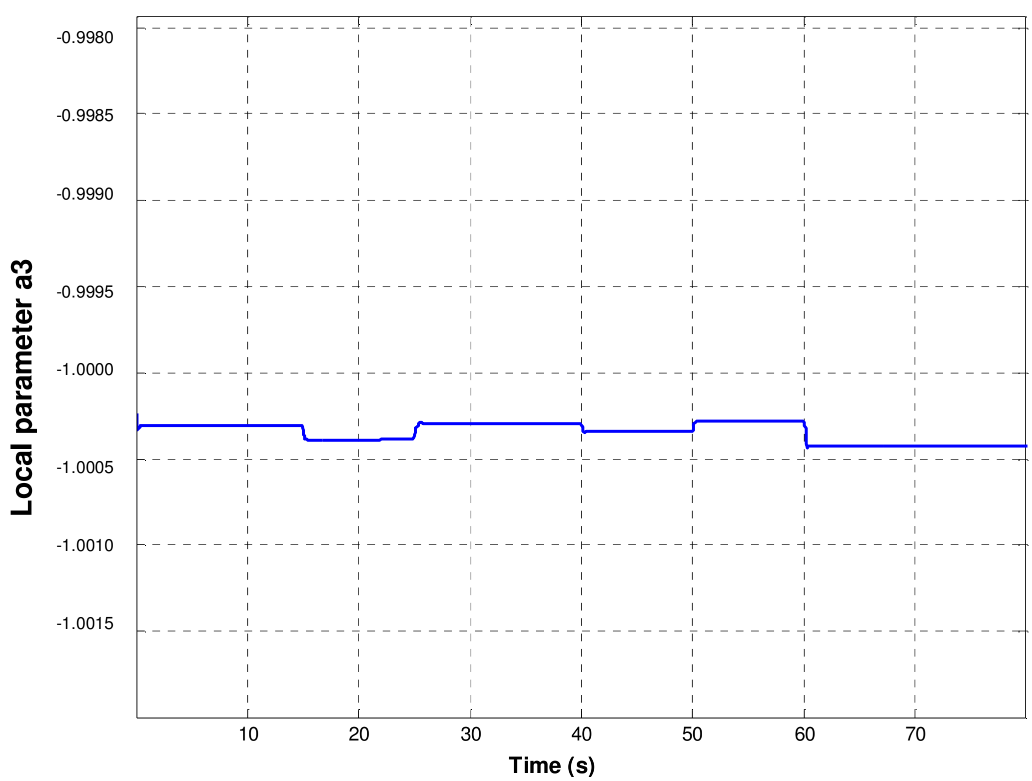

Figure 10. Evolution of the local parameters *a3* of ANFIS during the control of the DC-motor.

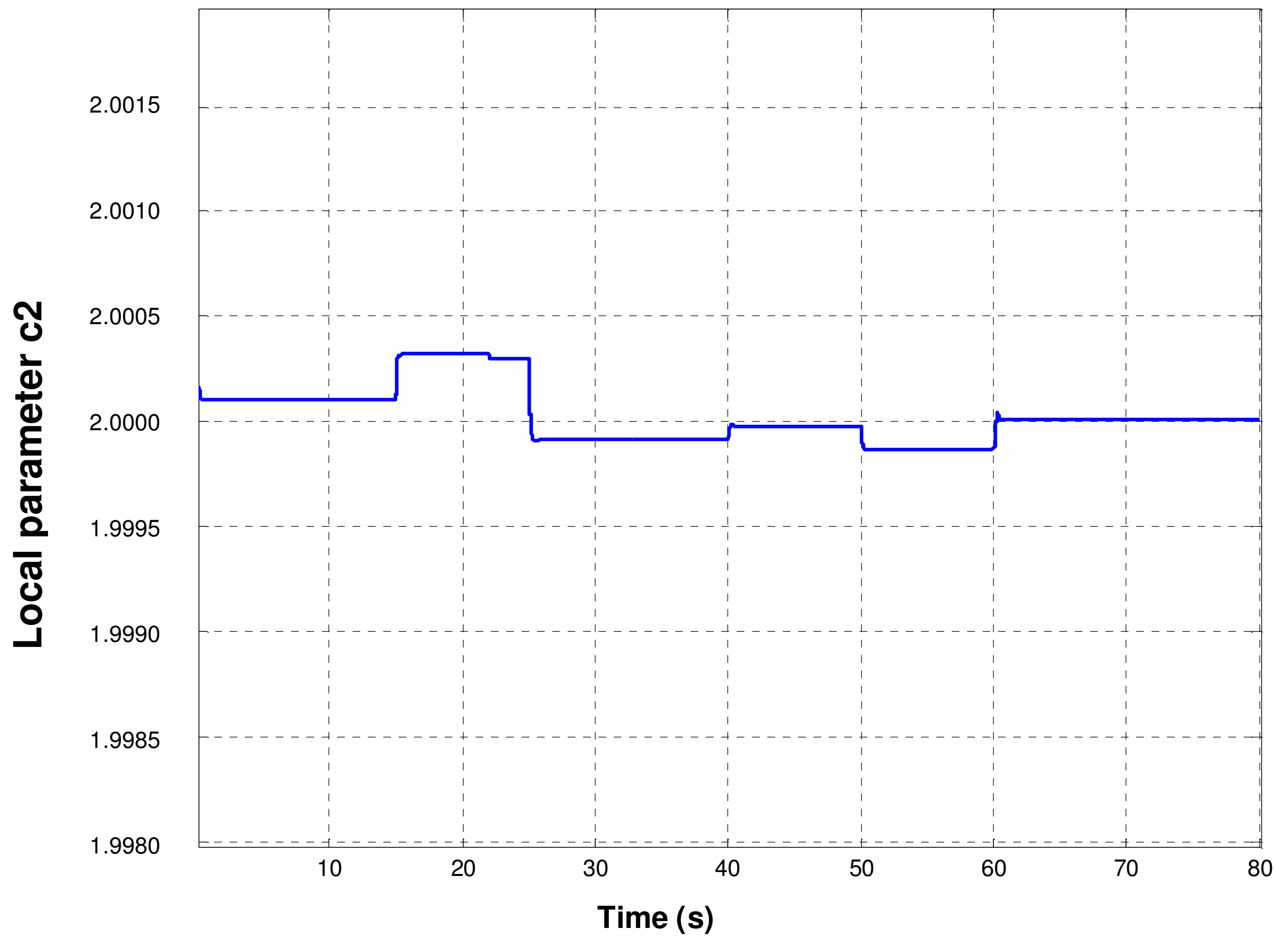

Figure 11. Evolution of the load parameters *c2* of ANFIS during the control of the DC-motor.

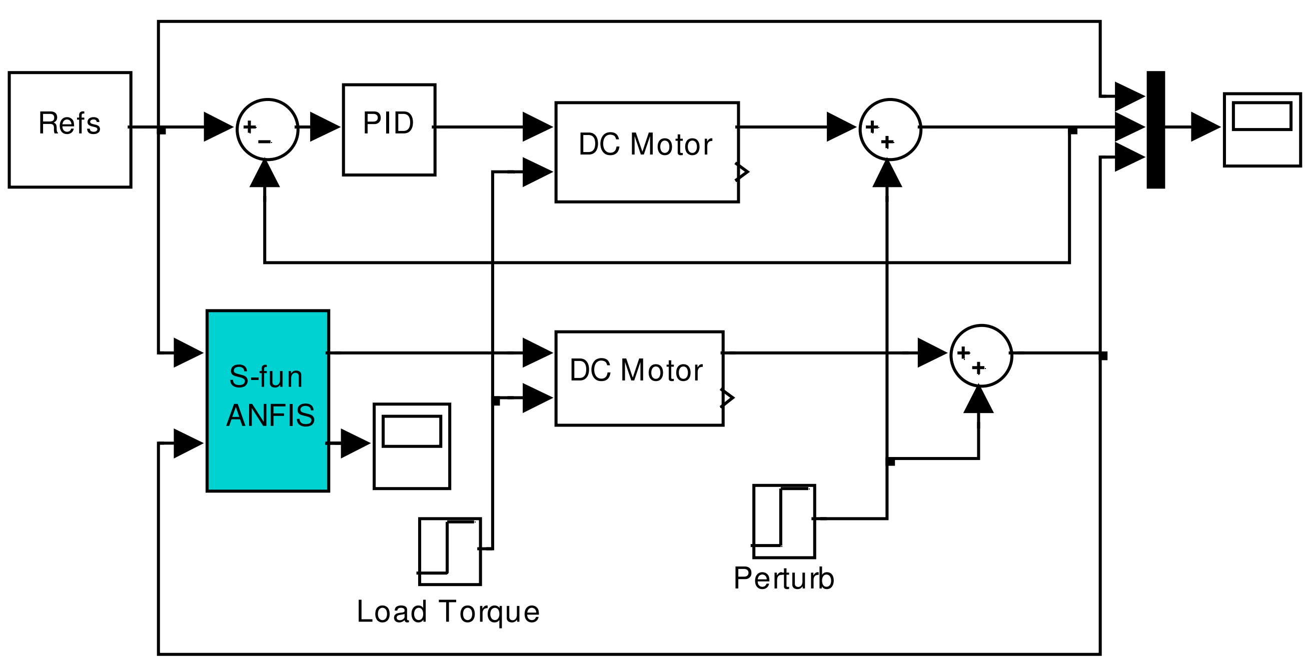

Figure 12. Simulink model of direct-inverse adaptive control based ANFIS Learning –application of load toruqe

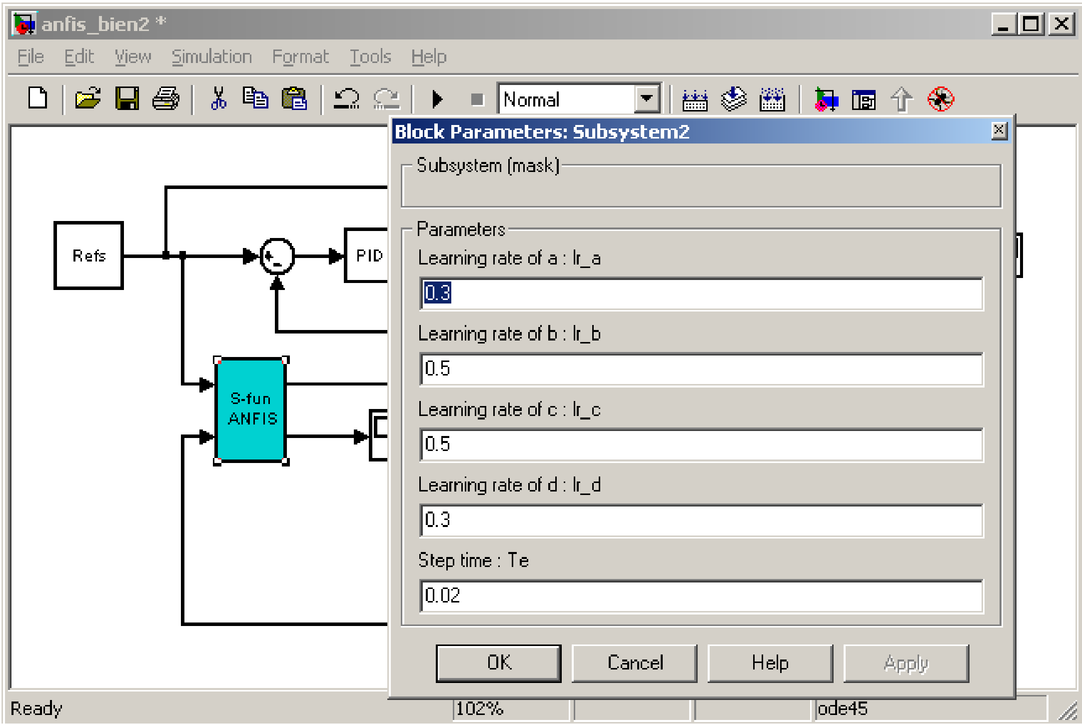

Figure 13. Adjustment of parameters of the S-function Matlab block of ANFIS.

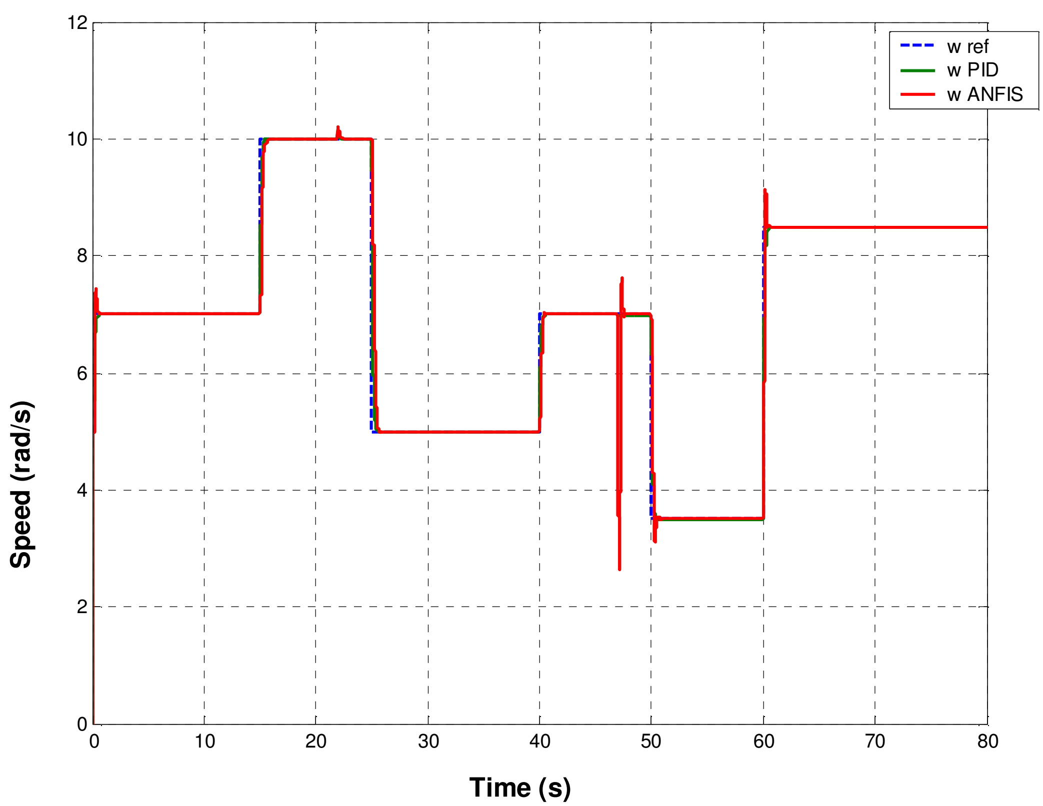

Figure 14. Speed of the DC-motor with direct-inverse control based ANFIS learning and PID controller -application of load torque at t=47 s.

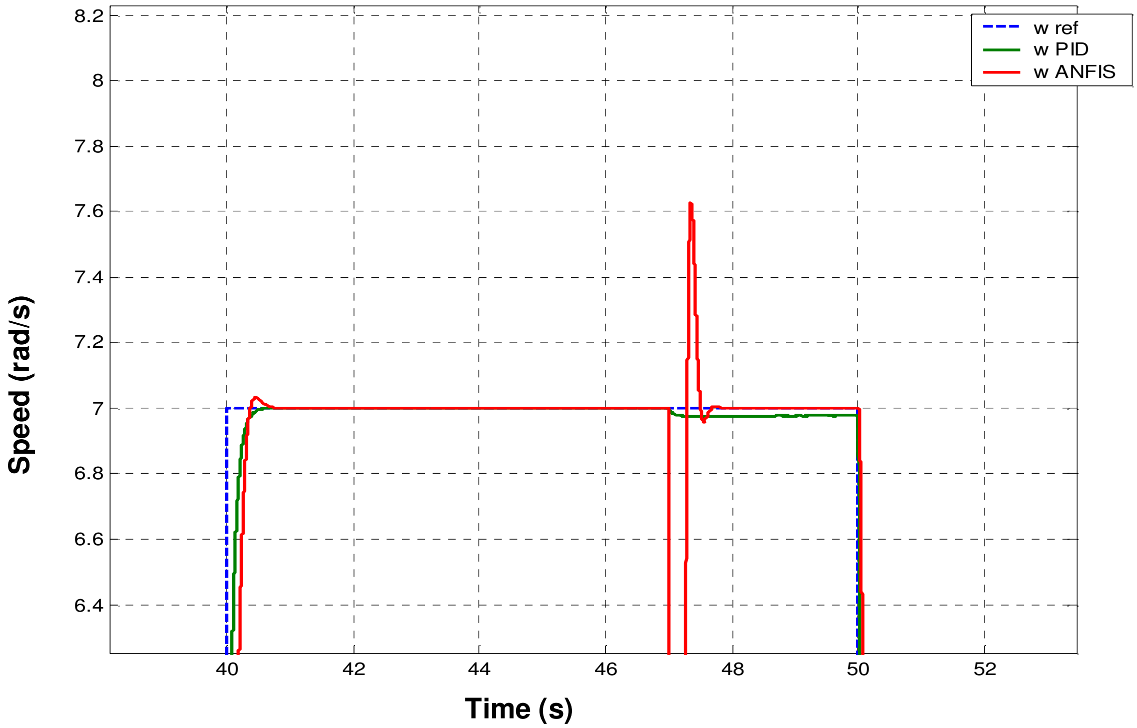

Figure 15. Speed of the DC-motor with direct-inverse control and PID controller –application of load torque at t=47 s. (zoomed).

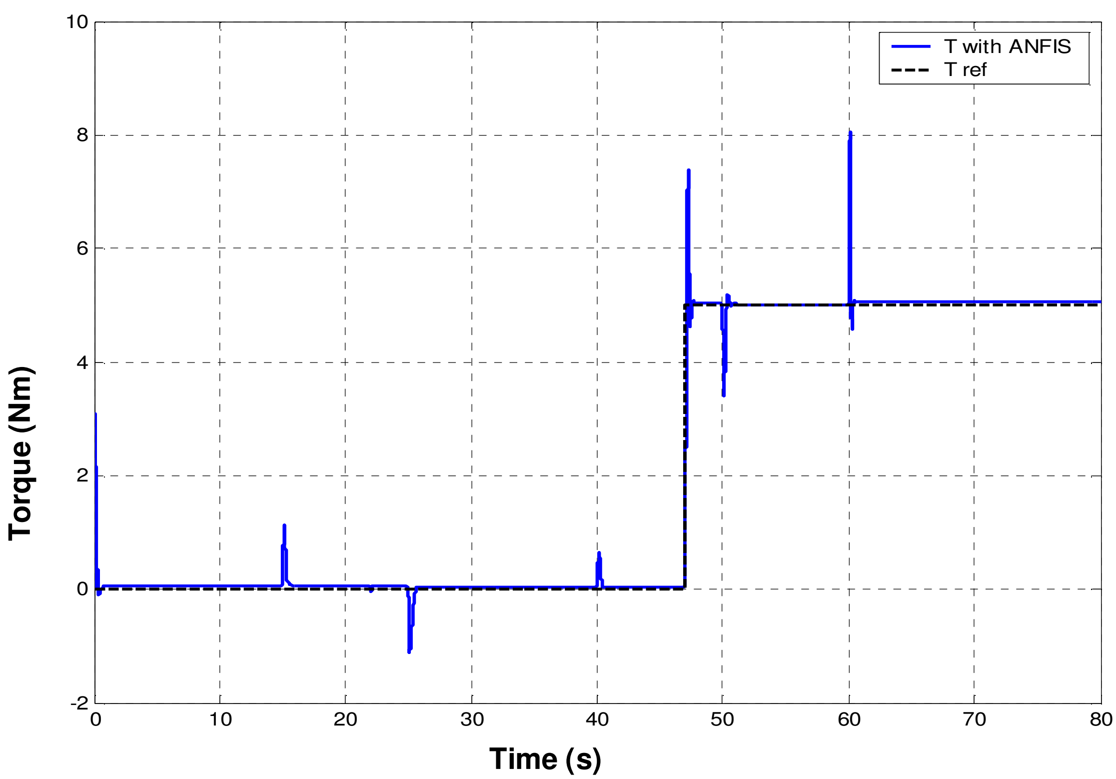

Figure 16. Torque response of the DC-motor with direct-inverse control based ANFIS learning –application of load torque at t=47 s.

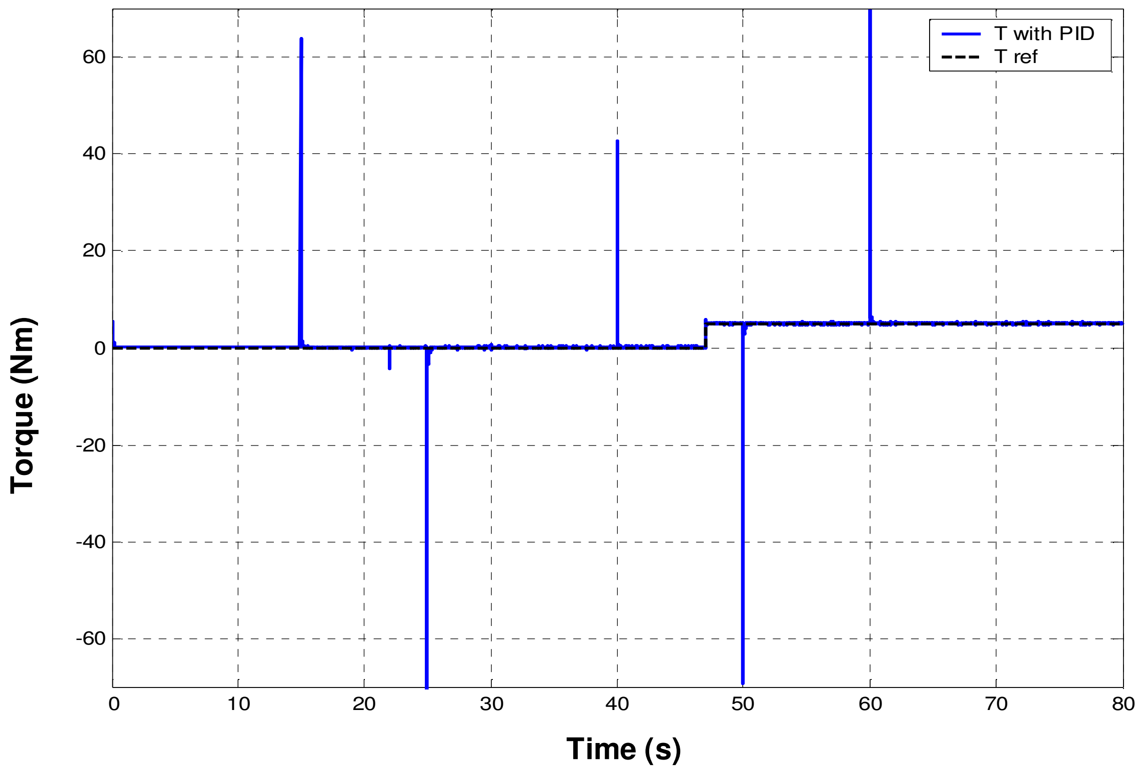

Figure 17. Torque response of the DC-motor with PID controller -application of load torque at t=47 s.

The simulated results show improvement in the static error speed with maintenance of performances in the presence of load torque. The future work is to build an S-function of ANFIS to control real systems.

REFERENCES

Azeem MF, Hanmandlu M, Ahmad N (2000). Generalization of adaptive neuro-fuzzy inference systems. IEEE Trans. Neur. Netw. 11:1332-1346.

Carrano EG, Takahashi RHC, Caminhas WM, Neto OM (2008). A Genetic Algorithm for Multiobjective Training of ANFIS Fuzzy Networks. Congr. Evol. Comput. pp. 3259-3265.

Eleftherios G (2012). Study of Discrete Choice Models and Adaptive Neuro-Fuzzy Inference System in the Prediction of Economic Crisis Periods in USA. Econ. Anal. Pol. 42(1):79-96.

Farzad R, Babak NA, Caro L (2005). An Evolutionary Fuzzy Modeling Approach for ANFIS architecture. IEEE Congress on Evolutionary Computation, Edinburgh -Scotland, pp. 2182-2189.

Gonzalez-Gomez JC, Ruz-Hernandez JA, Garcia-Hernandez R, Sanchez EN (2011). Real-Time Neuro-Fuzzy Inverse Control Applied to a DC Motor. 8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE).

Hiroki T, Koichi T, Hisasi T, Catherine V, Zheng T (2009). Recurrent Type ANFIS Local Search Technique for Time Series Prediction. Proceedings, IEEE Asia Pacific Conference on Circuits and Systems, pp. 380-383, Nov 30-Dec 3, Macao, China.

Hongsheng S, Feng Z (2007). A Novel Learning Method for ANFIS Using EM Algorithm and Emotional Learning. International Conference on Computational Intelligence and Security.

Hongxing L, Philid Ghen CL, Han-Pang H (2001). Fuzzy Neural Intelligent System: Mathematical foundation and the applications in engineering. CRC Press LLC.

Howard D, Mark B (1997). Neural Network toolbox for use with Matlab. Users Guide version 4; Mathworks.

Jang JSR (1993). ANFIS/ Adaptive Neuro-fuzzy Inference System. IEEE Transactions on Systems Man, and Cybernetics, Vol. 23, N3 May/ June1993.

Kasuan N, Ismail N, Taib MN, Rahiman MHF (2011). Recurrent adaptive neuro-fuzzy inference system for steam temperature estimation in distillation of essential oil extraction process. 7th IEEE International Colloquium on Signal Processing and its Applications (CSPA).

Toha SF, Tokhi MO (2009). Dynamic nonlinear inverse-model based control of a twin rotor system using adaptive neuro-fuzzy inference system. Third UKSim European Symposium on Computer Modeling and Simulation.

Xuan FZ (2006). Artificial Intelligence and Integrated Intelligent Information Systems Emerging Technologies And applications. Idea Group Inc (IGI).