IDENTIFICATION AND CONTROL OF ROBOT MANIPULATOR USING FUZZY WAVELET NETWORK

Dr.Yhya R. M. Kuraz *, Prof.Dr.Waleed Ameen Mahmoud Al-Jawher**

*Computer Engineering Dep.Faculty of Engineering. Mosul Univ. Iraq
yhya.kuraz@gmail.com

** *Dean of Engineering College,University of Al-Isra,Jordan-Amman*
profwaleed54@yahoo.com

ABSTRACT

Wavelet transform is a useful tool for function estimation and signal processing; nevertheless wavelets are limited to small dimensions. The combinations of wavelet transform (WT) and neural networks (NN) will lead to overcome these limitations of large dimension efficiently. Wavelet networks (WN)s have been developed for their abilities of self-learning and self organizing. WNs have been successfully demonstrated to have potential in many applications. Although, controller design can be considered as signal approximation, and wavelet transform is a powerful tool in this field, yet its applications in the process control and identification areas have not been investigated deeply . In this paper a new structure of Fuzzy Wavelet Network (FWN) is proposed to identify Multi-input Multi-output complex nonlinear systems. It was found that the FWN performance depends on the selection of mother wavelet basis function and the associated number of wavelons.

The FWN is used to replace the linearization feedback of a robot arm that has four inputs and four outputs. Thus the FWN was employed as an identifier and it gave good results and fast convergence for the non parametric function under consideration in comparison with conventional Neural Network as well as, it was shown that one set of data is sufficient during off-line learning.

In Two Flexible Joints Robot Manipulation under consideration, the static learning of FWN structure can achieve robustness behavior in dynamic control. The stiffness of the FWN to avoid change of robot parameters was up to ±50% of its nominal values.

Keywords: *identification, wavelet network, fuzzy logic, fuzzy wavelet network and robot manipulation*

1. INTRODUCTION

Robot is a nonlinear system by nature, so that conventional linearization procedures cannot implement a perfect accurate controller design. A feedback linearization technique of a nonlinear system performs a methodological procedure for robot control system design. The basic idea is to construct a nonlinear control law as a so-called inner loop control which theoretically exactly linearizes the nonlinear system by suitable state space change of coordinated [1]. Then the design is completed by an outer loop control that satisfies the required performance such as transient characteristics tracking a reference signal, disturbance rejection… etc. Such theoretical technique has sever drawback when such controllers are implemented in practice; specifically they have no acceptable degree of robustance. Small changes in system and/or controller may lead to unsatisfactory behavior or unstable system.
In this chapter the control laws derived by linearization techniques as state in [1] will be replaced by the proposed FWN obtained in section 2.

2. FUZZY WAVELET NETWORK (FWN)

The main block diagram of this proposed Fuzzy Wavelet Network FWN is given in Figure 1, this structure introduces a multi input multi output MIMO function approximator model. It consists of four layers namely: the input layer, Fuzzification layer, rule layer and defuzzification layer [2].

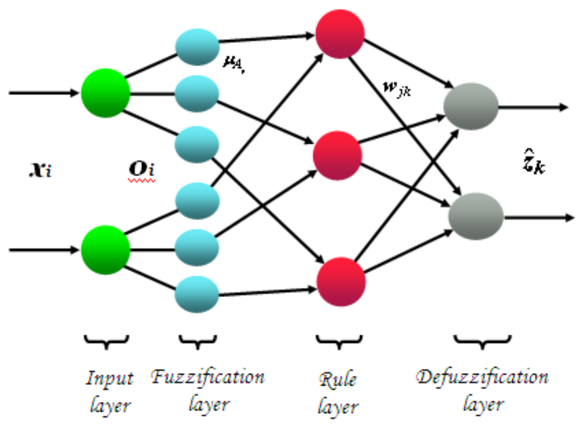

Figure 1: Structure of Fuzzy Wavelet Networks

the computation of the FWN is illustrated in the following steps:
step one: the fuzzification setup: in this step it is necessary to build the fuzzification membership through the following procedure.
a) the input to the structure, input layer, is xi and its outputs o_i will be given to the fuzzification layer

$o_i = x_i, \quad i = 1,2,....,N$ …………………… (1)

where n is the number of inputs
b) for each fuzzy term A_{ji} in jth rule, associated with ith input o_i, the membership μA_{ji} is

$\mu A_{ji}(t) = x_i(t).\psi_{a_{ji},b_{ji}}(t)$ ……………….. (2)

the structure of WN [3,4] is extended to be used here as a membership function by translating and dilating the mother wavelet basis function to build up fuzzification layer. It is shown that a set of 10 to 40 wavelet basis functions memberships is required for its success.

step two: selection of implicator rule: this requires setting of the implicators rules that are necessary for convergence of the network, the number of implicators used here is equal to the number of memberships, which use mini-operation implicator [5].

step three: the defuzzification setup: the fourth layer in FWN is the output layer, which realizes the k-th approximated output by:

$$\hat{z}_k = \sum_{j=1}^{J} w_{jk} y_j \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

where J is the number of wavelet filters that are used as membership functions for each input and J rules are used with minimum implicator rule to produce crisp output value.

step four: the learning algorithm of the proposed FWN: here the gradient descent algorithms for tuning the parameters, dilations a_{ji} , translations b_{ji} and weights w_{jk} of the FWN is used.

first the following cost function (E) for this case was used [6]:

$$E = \frac{1}{2K} \sum_{t=1}^{T} \sum_{k=1}^{K} (\hat{z}_k(t) - z_k(t))^2 \quad \ldots\ldots \quad (4)$$

where $z_k(t)$ is the kth desired output, $\hat{z}_k(t)$ is the k_{th} approximated output of FWN, K is the number of output signals and T is the length of the output signal. the training algorithm, that is extended from WN training algorithm, the parameters of FWN, dilations a_{ji} , translations b_{ji} and weights w_{jk}, are updated such as to minimize the cost function (E) defined in equation (4). thus the FWN weights can be update using the following equations:

$$\left.\begin{aligned} w_{jk}(k+1) &= w_{jk}(k) - \eta_1 \frac{\partial E}{\partial w_{jk}}, \\ a_{ji}(k+1) &= a_{ji}(k) - \eta_2 \frac{\partial E}{\partial a_{ji}}, \\ b_{ji}(k+1) &= b_{ji}(k) - \eta_3 \frac{\partial E}{\partial b_{ji}}, \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots \quad (5)$$

where:

$$\left.\begin{aligned} \frac{\partial E}{\partial w_{jk}} &= -\sum_{t=1}^{T} e(t)\psi(\tau)o_i(t), \\ \frac{\partial E}{\partial b_{ji}} &= -\sum_{t=1}^{T} e(t)o_i w_{jk} \frac{\partial \psi(\tau)}{\partial b_{ji}}, \\ \frac{\partial E}{\partial a_{ji}} &= -\sum_{t=1}^{T} e(t)o_i w_{jk} \tau \frac{\partial \psi(\tau)}{\partial b_{ji}} = \tau \frac{\partial E}{\partial b_{ji}}, \end{aligned}\right\} \quad \ldots \quad (6)$$

Where: $\tau = \frac{t - b_{ji}}{a_{ji}}$, i is the input indicator, j is the rule indicator, k is the output indicator, μ_A is the membership function which is here considered as the mother wavelet basis function and o_i is the output of the input layer. An adaptive learning rate is used here to updating FWN parameters. Also in learning algorithm, the modification (updating) is done only in the gene that activates (or is firing) specific rule.

an attempt was made in order to simplify this procedure in a form which is clearly denoted in software engineering as the mechanization of proposed FWN or the cycle time and it is shown in Figure 2.

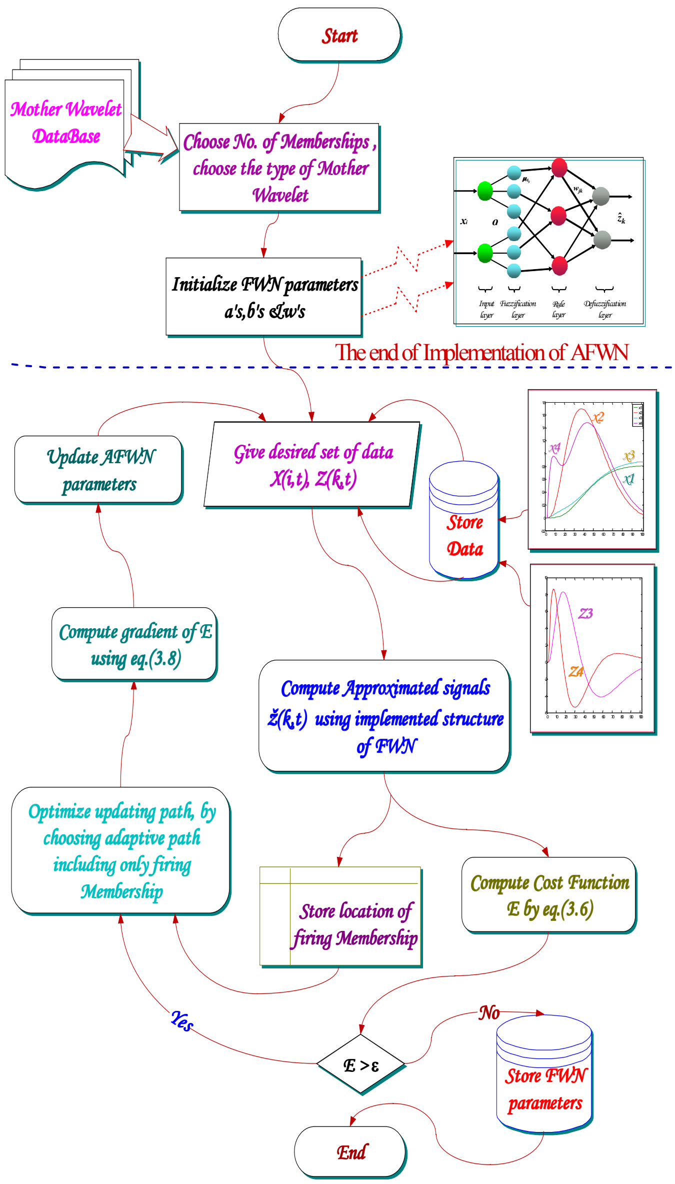

Figure 2: Mechanization of the proposed FWN

3. TWO FLEXIBLE JOINTS ROBOT MANIPULATOR SIMULATION

The simulation of two flexible joints robot manipulator considers the following parameters [7].

I : Moment of inertia of the load	=1	$kg.m.s^2$
M : Rotor mass	= 1	kg

g : Gravity = 9.8 m/s^2

ℓ : Length of the arm = 1 m

K :Stiffness coefficient of the motor= 100 $kg.m/A$

J : moment of inertia of the motor = 1 $kg.m.s^2$

The nonlinear state space of the robot manipulator is given by the following model

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{Mgl}{I}\sin(x_1) - \frac{k}{I} & 0 & \frac{k}{I} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J} & 0 & -\frac{k}{J} & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -9.8\sin(x_1) - 100 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \\ 100 & 0 & -100 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \quad \text{........ (7)}$$

Applying the linearization technique, the state transformation (feedback control law) and the control action (feed forward control law) are respectively given by:

$$y_1 = x_1$$
$$y_2 = x_2$$
$$y_3 = -\frac{Mgl}{I}\sin(x_1) - \frac{k}{I}(x_1 - x_3)$$
$$y_4 = -\frac{Mgl}{I}\cos(x_1)x_2 - \frac{k}{I}(x_2 - x_4)$$

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_2 \\ y_3 &= -9.8\sin(x_1) - 100(x_1 - x_3) \\ y_4 &= -9.8\cos(x_1)x_2 - 100(x_2 - x_4) \end{aligned} \quad \text{........... (8)}$$

$$u = \frac{IJ}{k}v - \frac{MglJ}{k}\sin(x_1)(x_2^2 + \frac{Mgl}{I}\cos(x_1) + \frac{k}{I}) - J(x_1 - x_3)(\frac{k}{I} + \frac{k}{J} + \frac{Mgl}{I}\cos(x_1))$$

$$u = \frac{1}{100}\Big(v - \Big[9.8\sin(x_1)(x_2^2 + 9.8\cos(x_1) + 100) + 100(x_1 - x_3)(100 + 100 + 9.8\cos(x_1))\Big]\Big) \quad \text{... (9)}$$

The outer loop design is completed by design of an optimal state feedback regulator for the linearized model ($1/s^4$), [7]. the normalized gain matrix is:

$K = [1 \quad 0.454 \quad 0.0861 \quad 0.0068 \;]$.

Figure 3 shows both outer and inner loop design based on linearization technique and optimal control theory f or the considered two flexible joints robot manipulator.

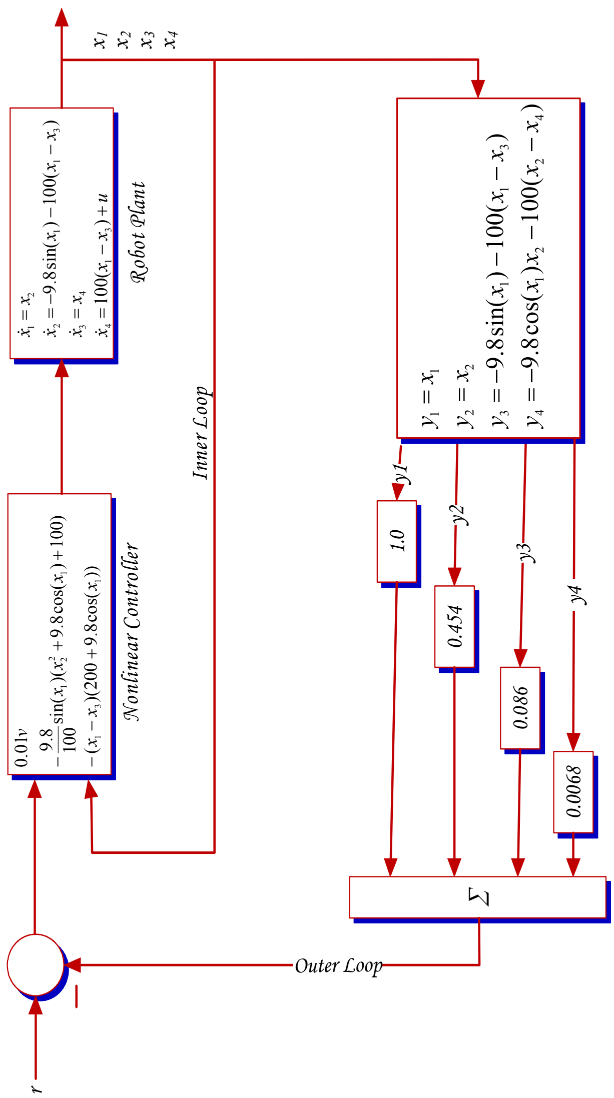

Figure 3: Robot and Control system structure using feedback linearization technique

A Matlab program is performed to obtain the nominal values of the system states, transformed states and the control input, and on the other side to link this program with the proposed software of implementing FWN. Also a Simulink setup is performed as shown in Figure 4 for performance evaluation when the robot manipulator changes its parameter.

The objective now is to replace the feedback control law given in (8) by the proposed FWN of 40-Rasp1 structure and evaluate the controlled system response for nominal conditions. Figure 5 shows the system step responses for both cases; with linearized control law and with proposed FWN. As it is clearly seen no significant difference is there, which indicates successful replacement. However it is important to say that till now no dynamic charges have been considered to test the proposed FWN and the successful replacement is nothing than a static fitting of data to certain nonlinear function. In the next section a dynamic situation will be considered.

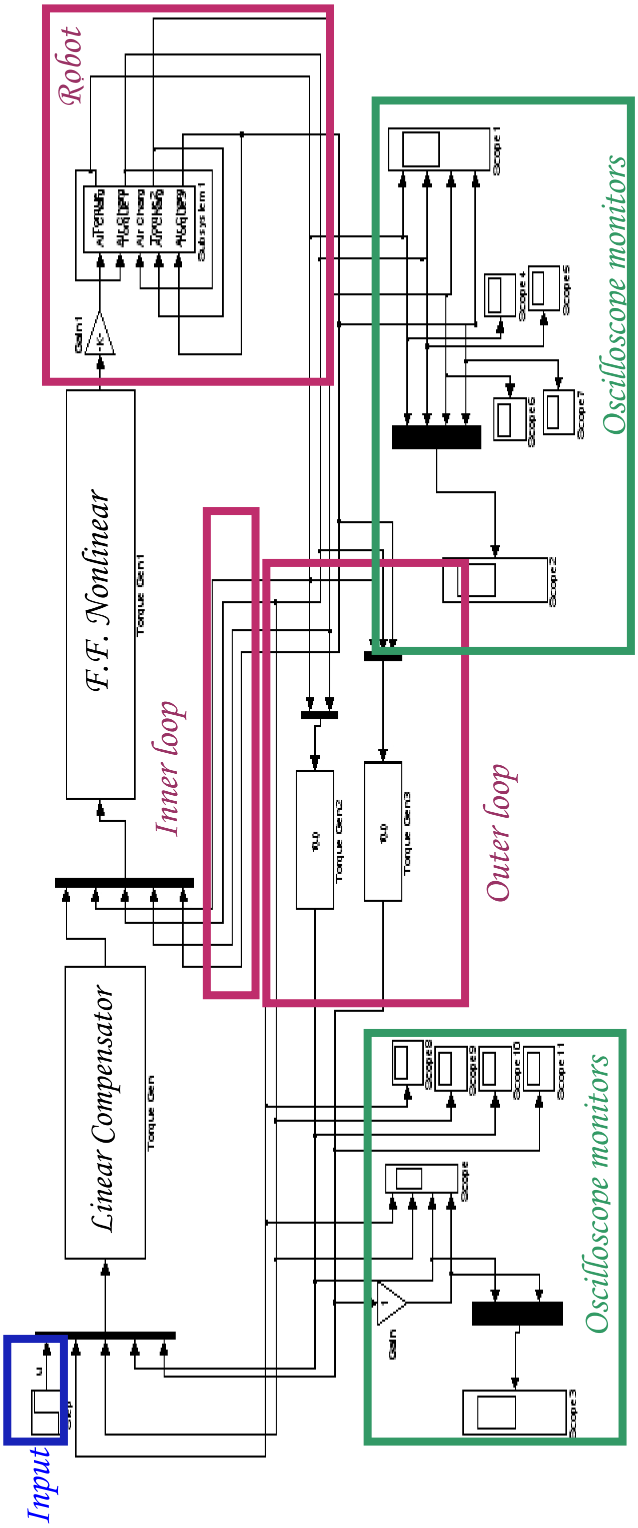

Figure 4: Simplified Simulink model design of Feedback Linearized controller for single link robot manipulator system

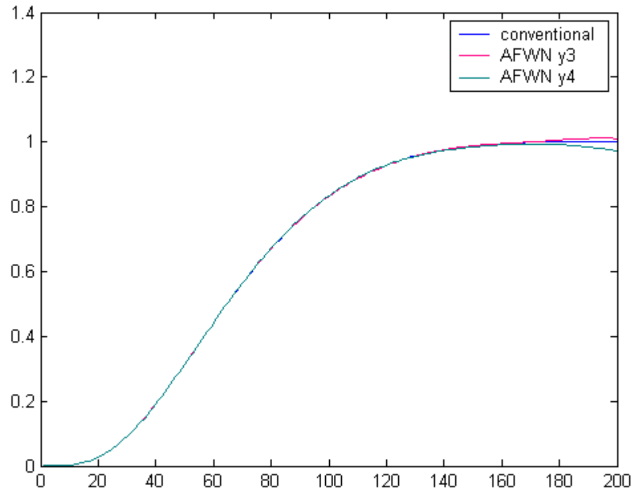

Figure 5: System Step responses using conventional control law and FWN in Feedback controller

4. ROBUSTNESS PERFORMANCE OF FWN CONTROLLER

Due to many physical environments the value of robot parameters may be deviated from their nominal values, and hence the robustness degree of any used controller must be under investigation. In [7] it is shown thoroughly how the performance of the robot manipulator controlled by feedback linearization technique is degraded from its nominal response as the parameters change. In what follows the proposed FWN will be used as a feedback controller where the nonparametric dynamic functions x_1, x_2, x_3 and x_4 represent the elements of the input vector to FWN. Different hypothetical cases of changing the moment of inertia of the motor *J* and the moment of inertia of the load *I* are within specific ranges.

The strategy of simulation will be of two phases. First to replace the nonlinear function y_3 by the proposed FWN keeping y_4 unchanged and in the second phase to replace y_4 keeping y_3 unchanged. Such independency gives more clear view of implementing the FWN as a controller. For both phases the 40-Rasp1 structure whose weights, dilations and translation are updated up to 1000 iteration to minimize the error to less than 10^{-4}. The final values of the weights, dilation and translation are given in Appendix B. Figures 6and 7 shows the performance of FWN for both y_3 and y_4 respectively.

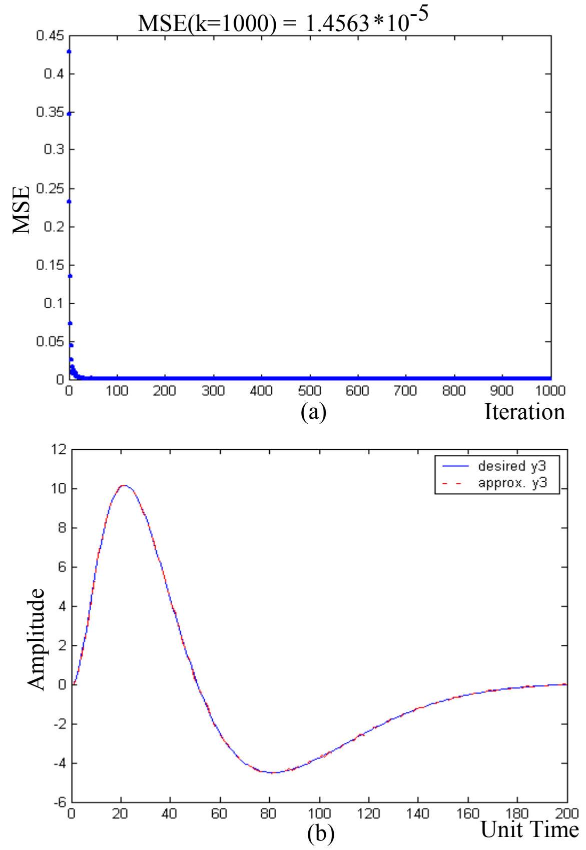

Figure 6 : Adaptive Fuzzy Wavenet Simulation and Performance for 40-Rasp1 a) MSE vs. of iteration b) desired & approximated y3.

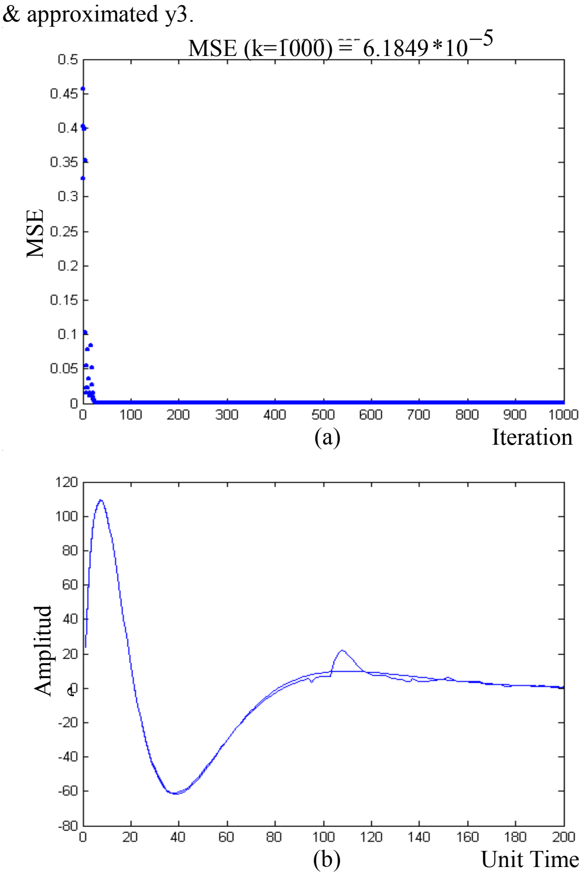

Figure 7: Adaptive Fuzzy Wavenet Simulation and Performance for 40-Rasp1 a) MSE vs. of iteration.
b) desired & approximated y4.

4.1 SIMULATION RESULTS FOR WN y3 CONTROLLER

In this phase, the motor moment of inertia J is varied up and down of its nominal value by 50%. Figure 8 shows the robot manipulation normalized responses for unit step input when J is increased up to 50% of its nominal value. The maximum increment of the overshoot is less than 2% as J is increased by 50%. Similarly Figure 9 illustrates a maximum steady-state error of about 0.04. Therefore one can say that the performance of the proposed FWN controller in such dynamic environments exhibits robust behavior. However as the load moment of inertia I is varied such robustness can not be achieved and the system response can only be accepted for ±5% as shown in Figures 10&11. These butterfly figures indicate the necessity to train the network to more than one input data set as it is done so far, or to train the network on-line.

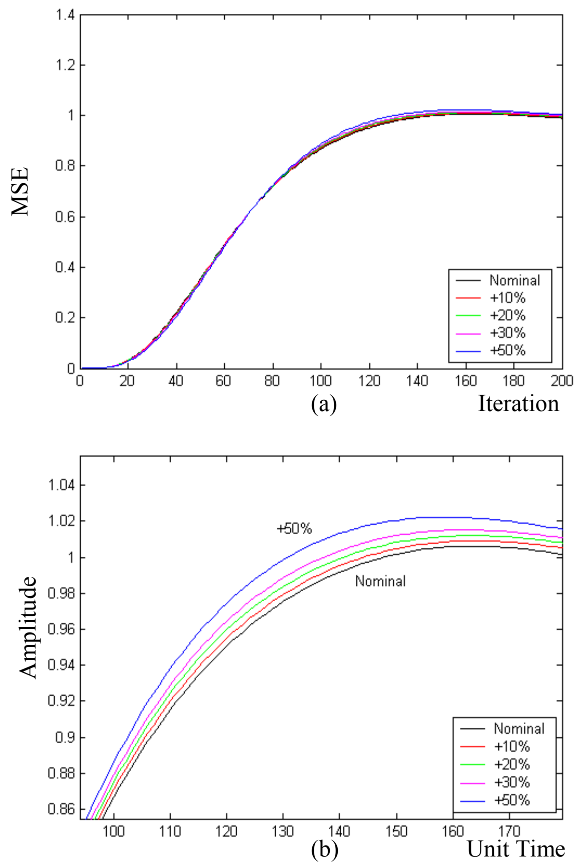

Figure 8: Unit step response as J increases up to 50%
(a) the full range of values from 0 to 200 unit time.
(b) After enlarging the value from 90 to 180 unit time

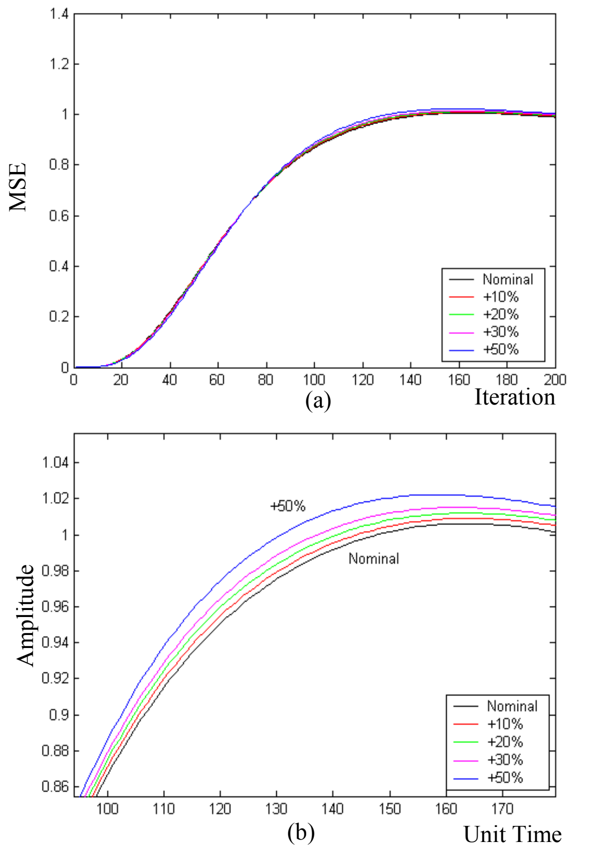

Figure 9: Unit step response as J decreases down to -50% (a) the full range of values from 0 to 200 unit time. (b) After enlarging the value from 100 to 200 unit time

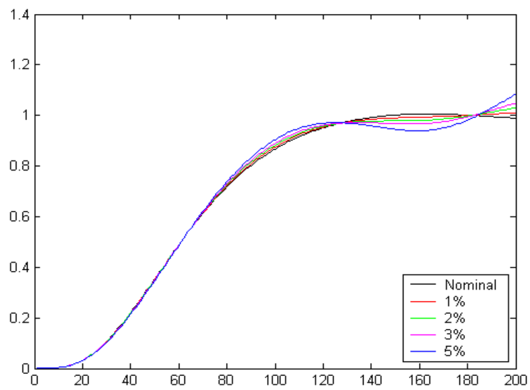

Figure 10: Unit step response as I increases up to 5%

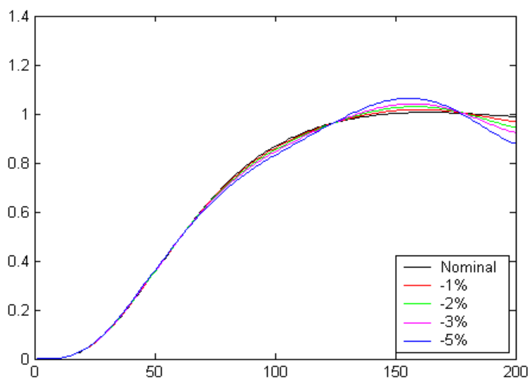

Figure 11: Unit step response as I decreases down to -5%

4.2 SIMULATION RESULT FOR FWN y_4 CONTROLLER

Now the proposed FWN stands as a controller instead of the control law described by the nonlinear function y_4 and the two system parameters J and I are allowed to vary: Figures 12&13 show as before the system response for J variations within $\pm 50\%$ of the nominal value. The maximum increment of the overshoot is not more than $\pm 1.5\%$, while the maximum steady-state error is 0.017. also Figures 14&15 show the case when the I parameter changes within $\pm 50\%$. Unlike the case with FWN y_3 controller the responses with $\pm 50\%$ changes of I show robust result.

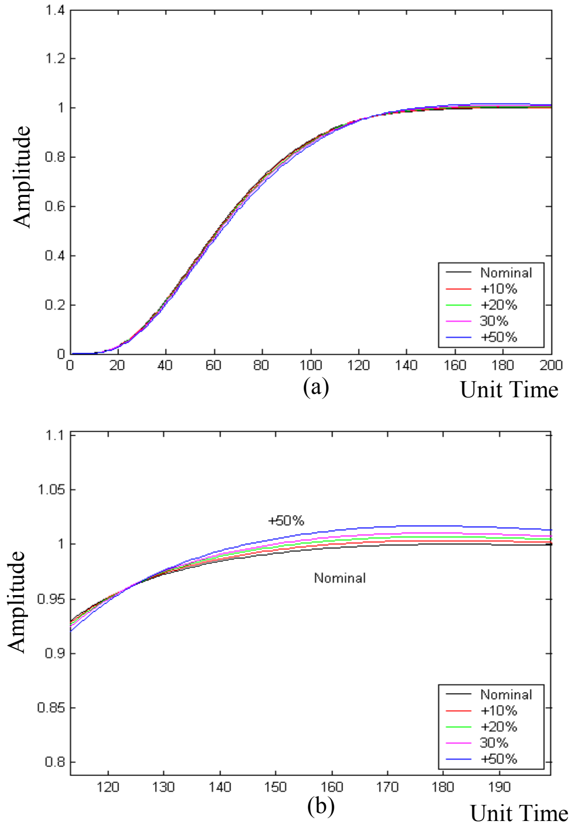

Figure 12 : Unit step response as J increases up to 50% (a) the full range of values from 0 to 200 unit time.
(b) After enlarging the value from 110 to 200 unit time.

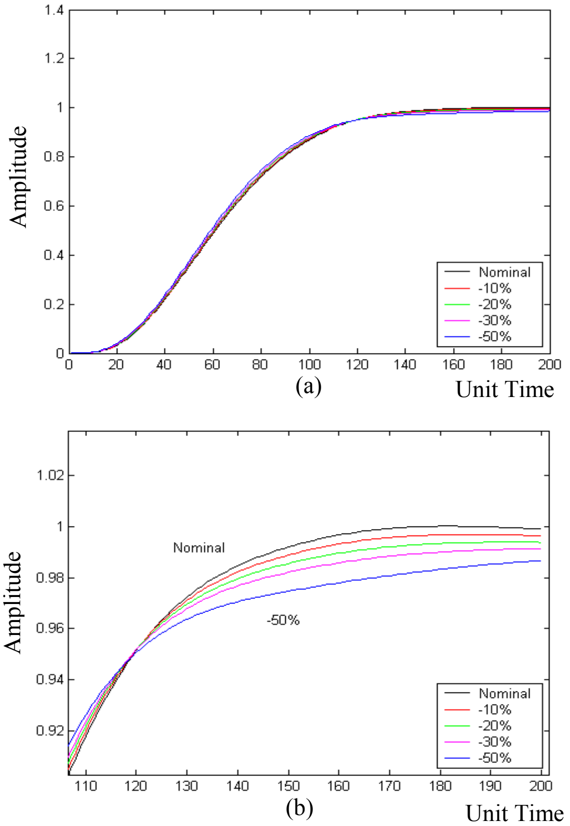

Figure 13: Unit step response as J decreases down to -50%
(a) the full range of values from 0 to 200 unit time.
(b) After enlarging the value from 105 to 200 unit time.

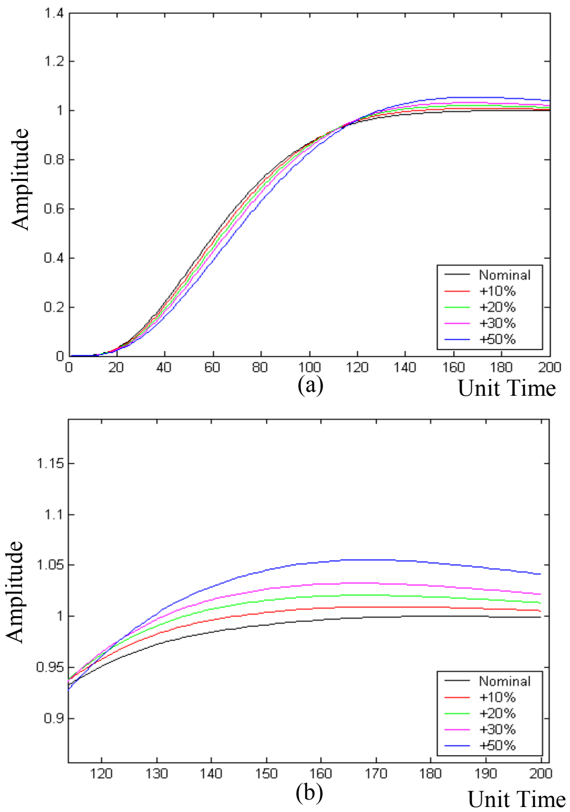

Figure 14 : Unit step response as I increases up to 50%
(a) the full range of values from 0 to 200 unit time.
(b) After enlarging the value from 115 to 200 unit time

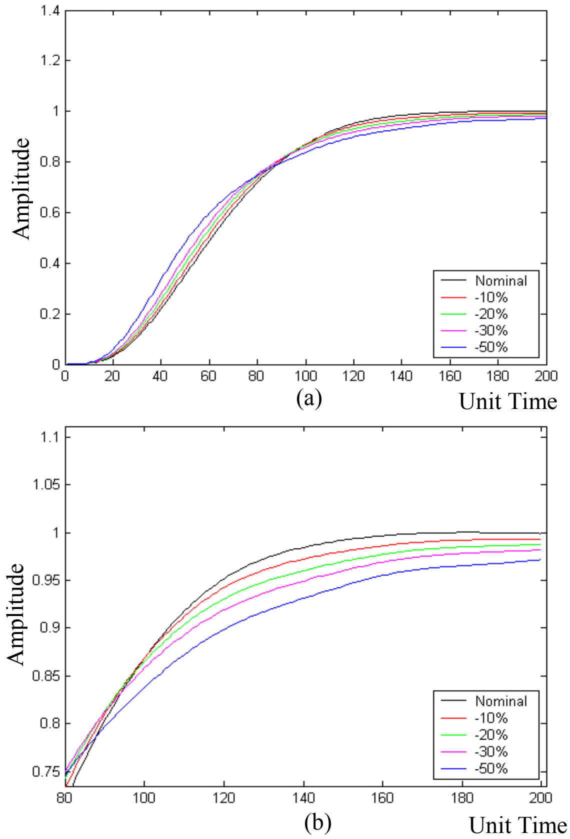

Figure 15 : Unit step response as I decreases down to -50%
(a) the full range of values from 0 to 200 unit time.
(b) After enlarging the value from 80 to 200 unit time.

By comparing NN results in [7], with those acquired by using FWN, it is found that NN solution is safer from the following limitations:

1) The structure of NN is given by large amount of trial and error. Hence, it is very sensitive to the change in the parameters of the system. Thus changing the data set of operation in a new desired structure must be redesigned that needs a hard searching process using the trial and error procedure.
2) The NN structure must be learned all the desired ranges of parameters changes. Hence it can't satisfy the unknown changes.
3) Capability of the system is limited to moment of inertia of the load and moment of inertia motor on which its range of change will be between -25% to 25% of its nominal values.
From the same figures the FWN overcomes these limitations. It can be clearly defined that FWN affords wider range of parameters change up to 50% from its nominal values. Moreover the facility is offered from using FWN as rapid reaching of the optimal structure with respect to those using NN, which requires many attempts of trial and error. Besides, FWN needs only one set of nominal data for its training procedure and it is capable of standing unknown changes within 50% of the nominal values.

5. CONCLUSION

The WN was developed to a new structure called Fuzzy Wavelet Network (FWN). This structure was employed and tested as an identifier of a complex and nonlinear Multi-Input Multi-Output function. Firstly, it can be concluded that this structure is an alternative to neural network to approximate nonlinear system. Then, this structure was used to achieve identification. Therefore, the proposed structure was useful for control system design of unknown (or uncertain) nonlinear systems. It can be concluded from the simulation results, that approximation errors can be successfully attenuated using the proposed FWN design method within a desired Wavelet basis function, thus identification of MIMO is achieved. Significant contribution is dedicated to algorithms that were proposed as identification structures of unknown nonlinear system. This is a very difficult problem especially when the black-box systems are highly nonlinear and under parameter changes environments. Hence, a structure named Fuzzy Wavelet Network (FWN) was proposed and simulated. It was used within a fuzzification defuzzification structure to achieve the adaptation off learning for rapid access to the goal of identification. As well as, the Adaptive Fuzzy Wavelet Network structure was implemented to achieve feedback control of a complex four inputs two outputs system. It was shown that this structure can be used to improve the performance of the trained network with fast convergence, minimum variability between runs, and high complexity to learn and track of unknown nonlinear systems. It is well known that the worst scenario to all of the control schemes, in terms of performance, occurs when of the plant system are changed. The conventional scheme with NN shows that it requires a longer time adapting to changes and performs poorly to system parameter changes. It can be concluded from the comparison of the performance of this new structure with the NN that for the same plant it gives a robust implementation. The new structure withstands parameter change of up to 50% over the conventional NN.

REFERENCES

[1] : Spong M.W. and Vidyasagar, " Robot Dynamic and Control", New York; John Wielly, 1989.

[2] : Kuraz Y. R.," Identification and Control of Robot Manipulator Using Adaptive Fuzzy Wavelet Network", college of engineering/electrical department, PHD thesis ,Baghdad, IRAQ, 2005.

[3] : Lekutai G.," Adaptive Self-Tuning Neuro Wavelet Network Controllers", Virginia Polytechnic Institute PHD thesis ,Blacksburg, Virginia, 1997.

[4] : Bellei E., Guidotti D., Petacchi R., Reyneri L., Rizzi I., "Applications of Neuro-Fuzzy Classification, Evaluation and Forecasting Techniques in Agriculture", ESANN'2001 proceedings – European Symposium on Artificial Neural Networks Bruges (Belgium), 25-27, D-Facto public., ISBN 2-930307-01-3, pp. 403-408, 2001.

[5] : Frey C.W., Kuntze H.B., "A Neuro-Fuzzy Supervisory Control System for Industrial Batch Processes", Fraunhoferstrabe 1,D-76131 Karlsruhe, Germany, Phone:+49- 721-6091-332/fax:+49-721-6091-413/Email: fry@iitb.fhr.de.

[6] : Karim1A. and Adeli H., " Comparison of Fuzzy-Wavelet Radial Basis Function Neural Network Freeway Incident Detection Model with California Algorithm", Journal of Transportation Engineering PP.21-30, January, 2002.

[7] : yaseen C.A.," Controller Design for Robot Manipulator using Feedback Linearization /Neural Network Technique", Military College of Engineering, Master thesis in 2000.