A Fast On-Line Learning Algorithm for Multidimensional Fuzzy Neural Networks

MOHAMED S. IBRAHIM
Knowledge Based Systems and Robotic Department
Mubarak City for Scientific research and Technology applications
EGYPT

*Abstract:-*This paper presents a fast on-line algorithm for constructing multidimensional fuzzy neural networks based on the possibility theory (MFNN-P).The multidimensional fuzzy neurons have a center vector and radius vector and the dimensions of these vectors are the same as the dimensions of the input vectors. The possibility measure is incorporated at the neurons in the output layer. A new on-line algorithm is developed. Simulation results show that the proposed network can deal with nonlinearities and uncertainties of the system.

Key –Words: - fuzzy, neural network, possibility theory, multidimensional neurons.

1 Introduction

In recent years, self organizing algorithm has been developed in hybrid systems [1, 2] to create adaptive models for nonlinear and time varying systems. Problems that arise in these systems are large dimensions, time varying characteristics, large amount of data and noisy measurements. A self organizing dynamic fuzzy neural network stated in [3]. With the number of input-output space partitions is given, a range of techniques is suggested to achieve the best set of fuzzy rules. Adaptive network based fuzzy inference system (ANFIS) is an example of this work [4].One of the learning algorithms is the iterative learning one [5].

This paper presents a new fast learning algorithm for the construction of multidimensional fuzzy neural networks for dealing with nonlinearities and uncertainties of the system dynamics. In the next sections, the architecture of the network is described. The learning algorithm is developed. The proposed network is tested using nonlinear and uncertain dynamics finally; results are summarized in the last section.

2 Multidimensional Fuzzy Neural Networks Based on Possibility Theory

The structure of MFNN-P has three layers of compact multidimensional fuzzy neurons; each one is a T-norm of nonlinear bipolar continuous function. The neurons in the input and hidden layers are of nonlinear types where the neurons at the output layer are of linear type. Each neuron has its center vector and width vector and the dimensions of these vectors are the same as the dimensions of the input vectors.

For the jth neuron in the hidden layer, the response is φ^j , j = 1, 2,…, h where h is the number of hidden neurons.

Each neuron in the output layer has two inputs .One input is the output of the winner neuron in the hidden layer and the other is the resultant possibility measure m^j defined in (1). The output of each neuron in the output layer is then given by:

$$y^k = \sum_{j=1}^{h} m^j \varphi^j \qquad (1)$$

Definition 1 : Consider two fuzzy sets with fuzzy membership functions $\mu_{C_i}(x_i)$ *and* $\mu_{A_i^j}(x_i)$ where $\mu_{C_i}(x_i)$ *and* $\mu_{A_i^j}(x_i)$ denotes the ith element of the input data and of the fuzzy rule R^j respectively , the possibility measure for the two fuzzy sets C_i *and* A_i^j is given by:

$$m^j = poss(\mu_{C_i} \mid \mu_{A_i^j})$$

$$= \mu_{C_i}(x_i) o \mu_{A_i^j}(x_i) \qquad (2)$$

Where the operator o denotes the composition rule of inference. The possibility measure has an important property, where it is a good fuzzy similarity measure. Now, let us assume some form of fuzzy similarity rules,

If $|y_k^* - y_k|$ is small then m^j is large

or

If $\left|y_k^* - y_k\right|$ is medium then m^j is medium

or

If $\left|y_k^* - y_k\right|$ is large then m^j is small

(3)

Where y_k^* is the kth desired output and y_k is the actual output.

The similarity measure m^j from definition (1) is

$$m^j = w^{jk} \circ \mu^{jk} \qquad (4)$$

Where, μ^{jk} is the fuzzy membership function of the kth element in jth rule and w^{jk} is the fuzzy membership function of m^j.

For multi-input multi-output case, the then part of jth rule has the form:-

$$y_1 \ \ is \ \ \mu^{j1} \text{ and } y_2 \ \ is \ \ \mu^{j2} \text{ and } y_k \ \ is \ \ \mu^{jk} \qquad (5)$$

The similarity measure becomes, $(\underset{k=1}{\overset{K}{T}} m^j)$ and if the number of input variables i=1,2,3,…n equals the number of output variables k =1,2,3,…,K then the similarity measure of the then part of jth rule becomes $(\underset{i=1}{\overset{n}{T}} m^j)$ this value is fed to the corresponding hidden neurons.

Substitute into equation (1) we get the normalized algorithm,

$$y^k = \frac{\sum_{j=1}^{h} w^{jk} \ (\underset{i=1}{\overset{n}{T}} m^j) \varphi^j}{\sum_{j=1}^{h} \varphi^j}, \qquad \text{k=1, 2,…, K} \ ; \text{ the}$$

number of output variables

(6)

In this paper, we selected the fuzzy membership function in the form of Gaussian type.

Note that w^{jk} is the centroid of the output neurons and is adapted using the instantaneous gradient with respect to the above functional mapping.

3 The proposed learning algorithm

The learning algorithm of MFNN-P achieves both the structure learning and the parameter learning. The structure learning means compact and economical network size. The parameter learning means that the network achieves the desired goals quickly using the proposed on-line self organizing algorithm. The important features of the algorithm are:

1.Determine the winner neuron using the minimum distance

$$\left\|x_i - c^j\right\| \le d_o$$

$$\left\| d_o \right\| = \sqrt{x^T x} \qquad (7)$$

Then adjust the weights using

$$\Delta w^j = \alpha_1 \left[x_i(t) - w^j(t-1)\right] \qquad (8)$$

Where $x_i(t)$ is the input vector and c^j is the centroid of the jth neuron.

2.Compute the outputs of the neurons in the output layer after incorporating the possibility measure.

3.The global learning rule can be derived as

$$\Delta w^{jk} \ (t) = \frac{\alpha_2 (u_k^* - u_k)(\underset{i=1}{\overset{n}{T}} m^j) \varphi^j}{\sum_{j=1}^{h} \varphi^j} \qquad (9)$$

This rule adjusts the weights between the output and the hidden neurons where u_k^* the desired control is signal and computed at the beginning of each iteration k, and u is the response of each neuron at the output layer .T is the t-norm.

The iterative learning algorithm [5] is given by

$$u_{k+1}^* = u_k^* + P.e + Q.ce \qquad (10)$$

Where P, Q are constant learning gain matrices. The error and change of errors are e=yd-y and ce .The desired and actual responses are yd and y respectively.

4 Simulation Results

Simulation studies are devided into two parts.the first part has studied the performance of the proposed multidimensional fuzzy neural network based on possibility measure(MFNN-P). The second part introduces a comparative study analysis with previous works.

4.1 Performance of The Proposed Approach

The performance of the proposed multidimensional fuzzy neural network based on the possibility theory is tested for controlling two link robotic arm .In the simulation 50% mass uncertainties are considered. The dynamics of such robotic system are strongly nonlinear and also suffer from uncertainties. The equations used in the simulation can be obtained from LaGrange Euler formulation [6].

Table 1 parameters of a two-link robot arm are given.

Length of the first link $(L_1) = 0.4m$
Length of the second link $(L_2) = 0.4m$
Mass of the first link $(m_1) = 0.5Kg$
Mass of second link $(m_2) = 0.5Kg$
Moment of inertia of the first link $(I_1) = 0.1Kgm^2$
Moment of inertia of the second link $(I_2) = 0.1Kgm^2$
Distance from joint 1 to C.M. of the first link $(l_1) = 0.2m$
Distance from joint 2 to C.M. of the second link $(l_2) = 0.2m$

$$u_1 = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 - 2h\dot{\theta}_1\dot{\theta}_2 - h\dot{\theta}_2^{\,2}$$
$$u_2 = H_{22}\ddot{\theta}_2 + H_{12}\ddot{\theta}_1 + h\dot{\theta}_1^{\,2} \tag{11}$$

Where u1 and u2 are the controlled torques at the joints of links 1 and 2 respectively and,

$$H_{11} = I_1 + I_2 + m_1 l_1^2 + m_2(L_1^2 + l_2^2 + 2L_1 l_2 \cos\theta_2)$$
$$H_{12} = I_2 + m_2 l_2^2 + m_2 L_1 l_2 \cos\theta_2$$
$$H_{22} = I_2 + m_2 l_2^2$$
$$h = m_2 L_1 l_2 \sin\theta_2 \tag{12}$$

The inputs to the network are e1, $\dot{e}1$, e2 and $\dot{e}2$ where e1= $\theta_{1d} - \theta_1$, $\dot{e}1$ is the derivative of error. The output of the network has two neurons. From simulations the size of the network is 4-6-2. The robot was initially at (θ_1 = 65 deg., θ_2 = 70 deg) is commanded to move one step to (80 deg., 50 deg). From the Figs.1, 2 and 4, we showed that the proposed MFNN-P can cope with the mass uncertainties to achieve better performance. It also showed that the proposed architecture performed well in bringing the system to the desired configurations in one second. Fig.3 shows the growth of the multidimensional hidden neurons.

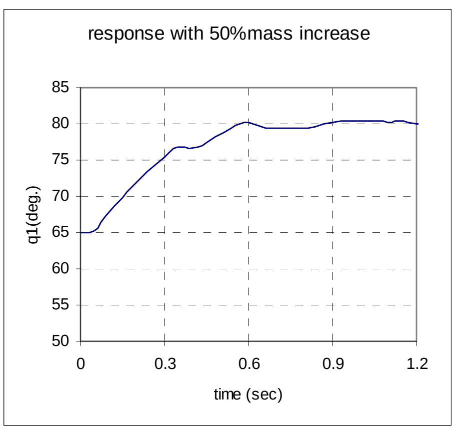

Fig. 1.a Robot angle q1= θ_1

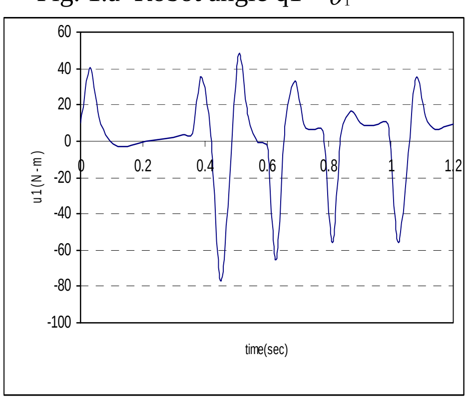

Fig.1.b applied torque

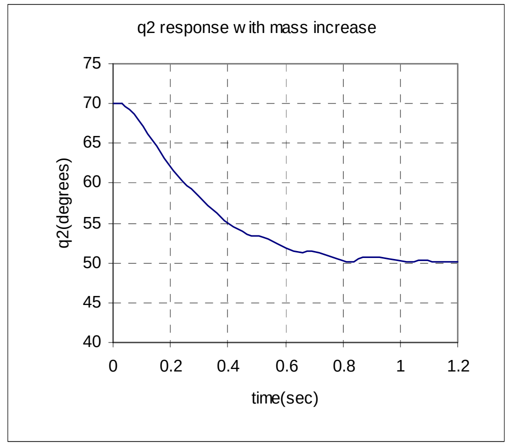

Fig. 2a Robot angle of the second link

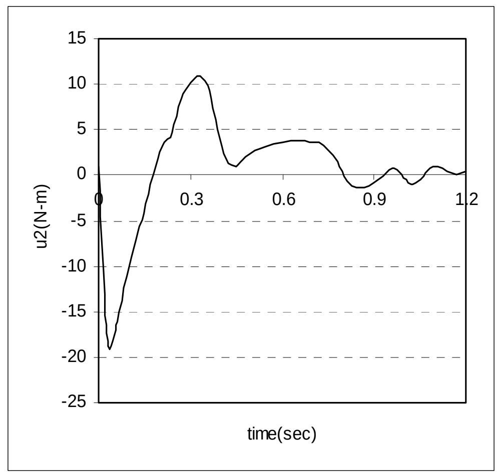

Fig.2b Torque u2 on the joint of the second link

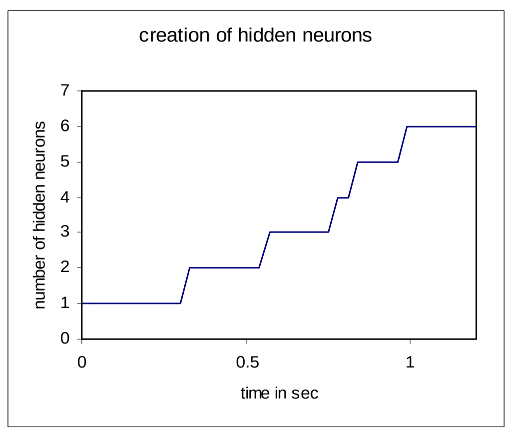

Fig.3 growth of hidden neurons

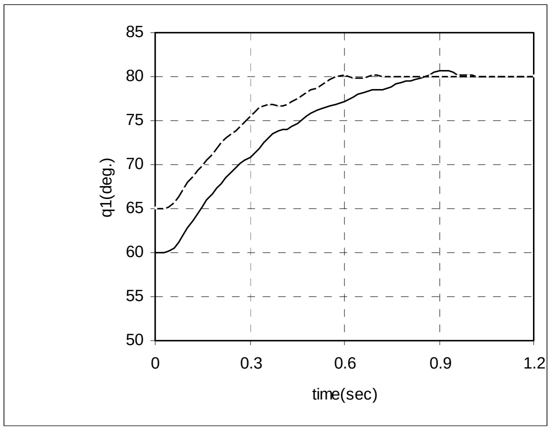

Fig.4 q1= θ_1 robot angle with initial conditions

4.2 Comparative Study with Previous Works

Comparative study analysis is summarisied in Table 2 in appendix A . The structure of the MFNN-P is 4-6-2 when the possibilistic concept is incorporated into the network , the number of iterations for convergence are 50 and the max. steady state error is 0.3 degree.While the structure of MFNN without incorporating the possibilistic concept is 4-9-2 , the number of iterations are 61 and the max. steady state error is 0.5 degree.more details are given in Table2.It shows smaller network size and less number of iterations needed for convergence compared with previous works.

6 Conclusions

Architecture of the multidimensional fuzzy neural networks is proposed .Also the learning algorithm for that network is developed. The network consists of three layers. The multidimensional hidden neurons are created automatically to deal with nonlinearities and uncertainties of the system dynamics. The possibility measure is incorporated at the neurons in the output layer. The proposed learning algorithm can be implemented on-line. The multidimensional fuzzy neural network has economic and compact network size. It has the ability to deal with nonlinearities and uncertainties of the system dynamics. Comparative study analysis shows that the MFNN-P has smaller network size (4-6-2)and less number of iterations needed for convergence(50 iterations) compared with previous works .This is due to incorporating the possibilistic concept in the output layer of the network.

References:

[1] Cho K. Wang B. ,” Radial basis function based adaptive fuzzy systems and their applications to identification and prediction,” fuzzy sets and systems ,83,pp 325-339,1996.

[2] Wu S. ,Er.M.,” Dynamic fuzzy neural networks-a novel approach to function approximation,”IEEE trans. On sys. Man,cyber. Part B ,30,pp.356-364,2000.

[3] Er M., Wu S. ,” A fast learning algorithm for parsimonious fuzzy neural system ,” fuzzy sets and systems ,126,pp.337-351,2002.

[4] Jang T. S.,” ANFIS: adaptive network based fuzzy inference system,” IEEE Trans. On sys., man, cyber., 23,665-684,1993.

[5] Armoto S.,” learning control theory for robotic motion,” Int.J. adaptive control and sig. processing,4,pp. 543-564,1990.

[6] Zomaya A.,” modeling and simulation of robot manipulators: a parallel processing approach. Singapore: world scientific publishing .1992

[7] Sahin Yildirim,” Adaptive robust neural controller for robots,” Robotics and autonomous systems ,46,pp. 175-184,2004.

[8] Fuchun Sun, Zengqi sun,Lei Li,Hanxiong Li,”Neuro-fuzzy adaptive control based on dynamic inversion for robotic manipulators,” Fuzzy sets and systems,pp.1-17,2002

[9] Keigo watanabe,Lanka udawatta,Kiyotaka Izumi,Kazuo Kiguchi,”Fuzzy rule selection of switching controllers for underactuated manipulators,” joint 1st International conf.on soft computing and intelligent systems(international session of 18th Soft fuzzy systems symposium)and 3rd Int. symposium on advanced intelligent systems,Tsukuba,Japan,Oct. 2002.

[10] Kumpati S. Narendra,Jeyendran Balakrishnan ,and M. Kemal Ciliz,” Adaptation and learning using multiple models, switching and tuning,”IEEE control systems,pp.37-51 ,June 1995.

Appendix A - Table 2
Case study :2-link robotic arm
Results of comparative analysis

Model	Learning algorithm	Structure	Iterations (or time in Sec) to converge	Steady state error (Ess)
Feedforward NN [6]	BP+ Reinforcement	5-20-2	100	a_
M-FNN+Possibility measure (This approach in the paper)	M-SONN+Iterative Learning	4-6-2	50 (1 sec.)	$0.3^{\circ} \cong 0.005\, rad$
M-FNN(the approach without possibility)	M-SONN+Iterative Learning	4-9-2	61 (1.83sec)	0.5°
NN+Robust control[7]	BP	4-6-2 + PID	50 000	a_
T-S model + NF system [8]	LS+ adaptive learning Algorithm + Lyapunov	8 rules for N Identifiers + 56 h. neurons	1.5sec	$0.4^{\circ} \cong 0.0069\, rad$
Fuzzy Logic [9]	GA	25 fuzzy rules	24 sec.	a_
FF –NN +switching models[10]	BP+models switching	6- 8 NN models+4	2 sec	$0.12^{\circ} \cong 0.002\, rad$

NN:neural network, MFNN:multidimesional fuzzy neural,BP:backpropagation,M-SONN:modified self organizing NN,T-S: Takagi-Sugeno,FF:feedforward,a_:value not reported,LS:least square