IJCCCE Vol.15, No.2, 2015

# Control System for Sluice Gates Flow in Irrigation Canals †

**Assist Prof Dr. Arif A. Al-Qassar[1] Shafaq Raad Abduljaleel[2]**

[1] Control and Systems Department , University of Technology, Baghdad, Iraq
[2] General comm. For Dams and Reservoir, Ministry of Water Resources, Iraq

e-mail: arifqassar@yahoo.com,shafaqalnajim@yahoo.com

Received: 16/6/2014

Accepted: 19/5/2015

***Abstract –*** Water has become the most important problem in relations between the countries of the Middle East in the recent years. It occupies an important place on the agenda of several international organizations. Water control and reduction loss of water discharge is a major challenge facing the design of new irrigation projects. A downstream control algorithm for demand operation of irrigation system is proposed in this paper through maintaining downstream end discharge of the canal at the target point by manipulating the upstream sluice gate in real time. The control of the water level and discharge for canal irrigation system has non-linear, time-varying and uncertainty characteristics. This paper compares three control algorithms; conventional PID, fuzzy neural network PID, and PID neural network control based on fuzzy neural network model. The simulation results show that the first control has larger over-shoot, longer adjusting time and poorer anti-interference ability. The second control overcomes above-mentioned short-comings, small overshoot, faster response speed, very small steady state error. Third control produces better effects than previous controllers in both steady performance and dynamic performance, including shorter steady-state time, non-overshot, no oscillator, and higher dynamic tracking rate.

.

**Keywords** – Sluice gate flow, Irrigation canals, PID, FNN.

† *This paper has been presented in ECCCM-2 Conference* and accredited for publication according to IJCCCE rules.

## 1. Introduction

The main function of irrigation canals is to deliver water in an accurate and flexible way, the delivery is said to be accurate if the actual supply matches the intended supply. It is said to be flexible if the delivery meets the changing water requirements of the users. This main function can be translated into a water level control problem consisting of two parts. First, the water levels in the system located just up-stream of the off takes and control structures need to be controlled within a sufficiently small range. Second, the control structure located at the upstream end of the canal is adjusting to control the preferably water levels. These requirements guarantee that the delivery matches the demands, one of the most useful control strategies that satisfy this requirement is the downstream end of pool water level control in real-time. The control algorithm should be mathematically simple in order to require small computing effort [1]. In the control system of water gate, gate opening measurement and control are major components of essential water automatic monitoring and control systems. To achieve accurate control of the gate opening we need to take into account various factors. Previously, gate flow control is the most traditional PID control which has nonlinear, time-varying, and hysteresis characteristics of complex systems, are often difficult to be satisfied with the results. This paper proposed three control algorithms and comparing their performance;

First; it is the conventional PID. Second; is the fuzzy neural network PID for lockage flow parameter setting. Neural network has the self-learning ability and massively parallel processing ability, which is good at the cognitive processing. Fuzzy control system makes full use of scientific knowledge in the field, with fewer rules to express knowledge, and it is quite good at handling skills, connected with PID control greatly improves control quality [2]. Third; it is the PID neural network control based on a fuzzy neural network model which combines the fuzzy neural network model with the PID neural network. PID neural network weight is adjusted online by using fuzzy neural network model and gradient descent method [3].

## 1. Design of the PID Control Algorithm

PID is the most used controller type in Industry. Its use is so diversified that the control engineer must tune the PID values according to specific needs [4]. The PID controller can be used to operate a structure in such a way that a specific hydraulic parameter (e.g. water level or discharge). The feedback control system is illustrated in Fig.1.

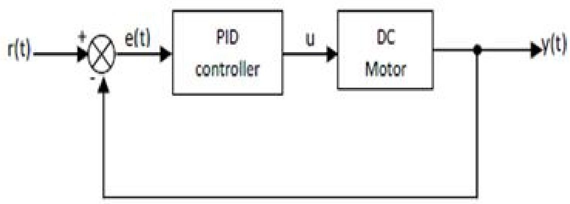



Fig.1 Common feedback control system

The PID controller is described in equation as:

$$u\,(t) = K_P\, e(t) + K_i \int_0^t e(t)dt + K_d \frac{de}{dt} \;....((1)$$

Where $u_t$ is the controller output, $e_t$ is the error, and t is the sampling instance. The factors $k_p$, $k_i$ and $k_d$ are the proportional, integral and derivatives gains (or parameters) respectively that are to be tuned, and estimated from the characteristics of the canal system [1, 5].

## 2. Fuzzy Neural Network PID

Fuzzy neural networks (FNN) are

neural networks that realize a set of fuzzy rules and a fuzzy inference machine in a connectionist manner. The FNN (which has been proposed) is a connection of feed-forward architecture with five layers of neurons and four layers of connections. The first layer of neurons receives the input information. The second layer calculates the fuzzy membership degrees to which the input values belong to predefined fuzzy membership functions, e.g. small, medium, or large. The third layer of neurons represents associations between the input and the output variables, fuzzy rules. The fourth layer calculates the degrees to which output membership functions are matched by the input data, and the fifth layer does defuzzification and calculates values for the output variables [6]. Neural networks and fuzzy logic systems are both numerical model-free estimators and dynamical systems. They share the common ability to deal with difficulties arising from uncertainty, imprecision, and noise in this natural environment. Both systems and their techniques have been successfully applied to various applicable areas to improve their machine intelligence. A promising approach to get both the benefits of neural networks and fuzzy logic systems and to solve their respective problems is to combine them into an integrated system so that it can bring the low-level learning and computation power of neural networks into the fuzzy logic systems, and also, provide the high-level human-like thinking and reasoning of fuzzy logic systems into the neural networks [7]. In these control algorithms BP back propagation (BP) training algorithms have been developed for FNN. BP neural network is a typical multilayer networks, it is divided into the input, hidden, and output layers. The basic processing units of BP network (except for the input layer unit) are non-linear input-output relations. Generally, (0, 1) s-function is used and input and output values of processing unit can change continuously, that is:

$$f(x) = \frac{1}{1+e^{-x}} \quad ..... (2)$$

BP algorithm consists of two parts: information forward transmission and error back propagation. In the forward propagation, the input information is calculated layer-by-layer from the input layer, through the hidden layer, to the output layer. The state of neurons on each layer only influences the state of neurons on the next layer. If the expected output has not been achieved in the output layer, the error value on output layer is calculated, and then turning back propagation. Through the network, the error signal is returned along the original path to modify the weight value of neurons on each layer until the desired target is reached [2].

In this article a fuzzy PID controller based on BP neural network is used to control the sluice flow. Fuzzy BP neural network structure is shown in the Fig 2. The network consists of input layer, fuzzation layer, fuzzy inference layer and output layer. The network output is $k_p$, $k_i$, $k_d$ .

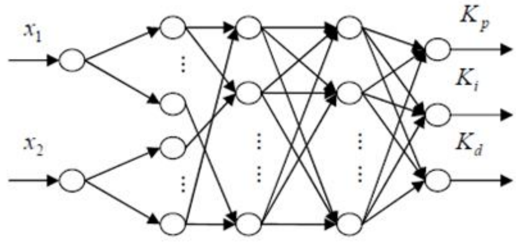



Fig.2 The structure of fuzzy neural network

The first layer is the input layer which contains two nodes $x_1$ and $x_2$ Where,

$x_1 = e = T_I - T_{i-1}, x_2 = e_c = [e(t+1) - e(t)]/T$ ,

both are input nodes. They transmit the signal to the next layer directly, that is, the connection weight is 1. The formula is:

$net_i^{(1)} = x_i, i = 1,2$

$o_i^{(1)} = x_i$

Where, $net_i^{(1)}$ is the net input of ith neuron in the jth layer.

$o_i^{(1)}$ is the output of the ith neuron in the jth layer. $x_1$ is the flow rate error. $x_2$ is the change rate of flow error. T is sampling period.

The second layer expresses language variable value of input variable. The first fuzzy set of input variable includes 8 language variables: positive big (PB), positive middle (PM), positive small (PS), positive zero (PZ), negative zero (NZ), negative small (NS), negative middle (NM) and negative big (NB). The second fuzzy set of input variable includes 7 language variables: positive big (PB), positive middle (PM), positive small (PS), zero (ZO), negative small (NS), negative middle (NM) and negative big (NB). That is, the fuzzy segmentation numbers of two input variables are 8 and 7 separately. The nodes on this layer are used to represent the membership function of each input language variable, they are as follows:

$$net_i^{(2)} = (o_j^{(1)}, i=1,2,\ldots\ldots,15 \quad \begin{cases} j = 1 (i \le 8) \\ j = 2 (8 \le i \le 15) \end{cases}$$

$$o_1^{(2)} = \exp\left[\frac{-net_i^{(2)} - m_i^{(2)}}{\sigma_i^{(2)}}\right]$$

Where, $m_i$ is the center of membership function in the form of Gauss for input $x_i$ language value. $\sigma_i$ is the width of membership function in the form of Gauss for input $x_i$ language value and can be adjusted. The connection weight for the second layer is 1.

The third layer has 8×7 nodes and each node represents a fuzzy rule. The connection between this layer and the second layer is used to match the conditions of fuzzy rules. Its output represents incentive intensity of each rule. That is

$$net_i^3 = (o_j^{(2)}.o_k^{(2)}) (i = 1,2,\ldots.56)$$
$$(j = 1,2,\ldots 8)\ (k = 9,10,\ldots,15)$$

$$o_i^{(3)} = net_i^{(3)}, (i = 1,2,\ldots.56)$$

The fourth layer contains 21 nodes, in which seven nodes are the fuzzy set of output variable $k_p$ seven nodes for output variable $k_i$ and seven for $k_d$. Each node of this layer performs fuzzy "or" operation to compose the consequent rules with the same output, its output actually uses the Mamdani fuzzy inference rules. The function of this layer is expressed as

$$net_i^{(4)} = \frac{\sum_{j=1}^{56} w_{ij} \cdot o_j^{(3)}}{\sum_{j=1}^{56} w_{ij}} \quad i = 1,2\ldots.,21$$

$$o_i^{(4)} = net_i^{(4)} \quad i = 1,2\ldots.,21$$

Where, $w_{ij}$ is the connection intensity between the ith output language value and the jth rules and its value can be changed. The fifth layer plays a role of unfuzzy. The output of the fourth layer is the membership degree of output variable fuzzy sets. Therefore, an improved gravity method is used here for unfuzzy, that is

$$k_p = \frac{\sum_{j=1}^{7} k_j \cdot o_j^{(4)}}{\sum_{j=1}^{7} k_j} \quad k_i = \frac{\sum_{j=8}^{15} k_j \cdot o_j^{(4)}}{\sum_{j=8}^{15} k_j} \quad k_d = \frac{\sum_{j=15}^{21} k_j \cdot o_j^{(4)}}{\sum_{j=15}^{21} k_j}$$

Where, $k_j$ is the weight of the fifth layer here $k_j$ equals 0.2 [2].

## 3. PID Neural Network Based on Fuzzy Neural Network

The block diagram of the fuzzy neural model based PID neural network control system is shown as Fig.3. And PIDNN is the PID neural network controller, FNNM being the fuzzy neural model with the controller output and the system output to be its inputs. In order to establish the control system, firstly, FNNM should be established according to the collected field data; Secondly, PID neural network controller should be introduced on the basis of model, and the controller parameters are tuned online in light of the model output and the error between system input and output.

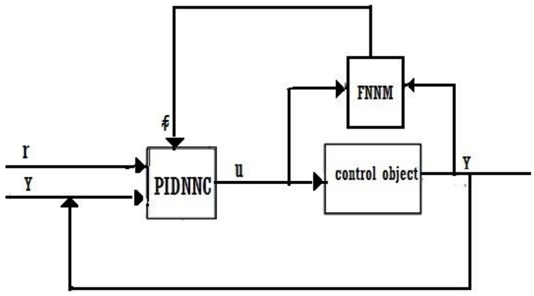
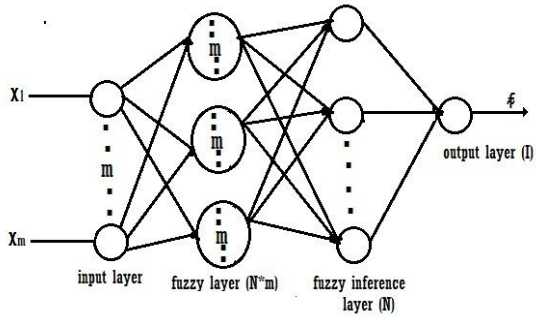



Fig.3. PIDNN control system structure based on fuzzy neural network model



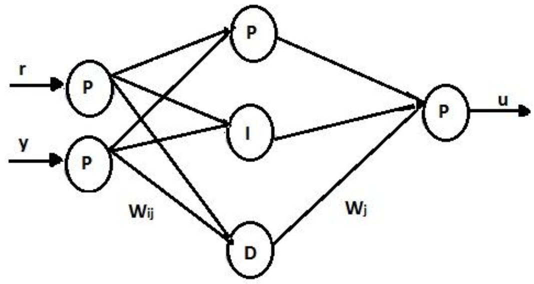




Fig. 4 Block diagram showing fuzzy neural network model

Fig.5. Structure diagram of PID neural network controller

FNNM is a multilayer feed-forward network model, consisting of TSK fuzzy logical system and RBF neural network. There are four fuzzy neural network layers in the fuzzy neural network model, which are input layer, fuzzy layer, fuzzy inference layer and output layer respectively. Input layer: composed of m neurons, passing m input signals to the next layer; Fuzzy layer: consisting of N arrays with every array having m neurons. The neurons in the l$^{th}$ array joined to each neuron in the output layer, and each neuron represents a Gaussian membership function; Fuzzy inference layer: composed of N neurons, the lth neuron connects to all the neurons of the l$^{th}$ array in the second layer output layer: made up of 1 neurons which is employed to calculate output y^ [3], as shown in Fig.4. The relationships between inputs and outputs for all the layers are as follows:

Layer 1: input $I_i^{(1)} = x_i$; output $o_i^{(1)} = I_i^{(1)}$ : 1=1 ,2 ,…..,m .

Layer 2: input $I_i^{(2)} = o_i^{(1)}$ ; output

$o_{1i}^{(2)} = exp\,(-\frac{(I_i^{(2)} - c_{1i})^2}{\sigma_1^2})$, i =1, 2,…. m, l=1,2, …..,N

Layer 3: input $I_i^{(3)} = o_i^{(2)}$ ; output

$o_{1i}^{(3)} = exp\,(-\sum_{i=1}^{m} \frac{(I_i^{(2)} - c_{1i})^2}{\sigma_1^2})$ , l=1,2, …..,N

Layer 4: input $I_i^{(4)} = o_i^{(3)}$ ; output

$o_1^{(4)} = \sum_{i=1}^{n} h_1\, o_i^{(3)}$ , l=1,2, …..,N

In summary, the total output of fuzzy-neuron model is:

$y\hat{} = \sum_{i=1}^{n} h_1 \;\; exp\,(-\sum_{i=1}^{m} \frac{(I_i^{(2)} - c_{1i})^2}{\sigma_1^2})$ ,

i =1,2,…., m, l=1,2, …..,N …..(3)

Learning and amending are necessary for the weights wi and wij among each layer of PID neural network controller. In this research based on fuzzy model, the weights are tuned online through gradient information given by the fuzzy model. The objective function is defined as follows:

$E=\frac{1}{2}(r(k) - y\hat{}(k))^2$…… (5)

Where r(k) is the system input at time k, and y^ (k) is the model output at time k. The control target is to seek for optimal weights wj and wij (i=1,2 ; j=1,2,3) in order that the objective function E can be minimized. The concrete regulating algorithm is as follows:

$w_{ij}(k) = w_{ij}(k-1) - \eta\Delta w_{ij}$ , η is learning rate

$w_j(k) = w_j(k-1) - \eta\Delta w_j$ , η is learning rate

$\Delta w_j = \frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y\hat{}}\frac{\partial y\hat{}}{\partial U}\frac{\partial U}{\partial w_j} = -(r(k) - y\hat{}(k))\frac{\partial y\hat{}}{\partial U}\frac{\partial U}{\partial w_j}$ ….. (6)

$$\Delta w_{ij}=\frac{\partial E}{\partial w_{ij}}=\frac{\partial E}{\partial y^{\wedge}}\frac{\partial y^{\wedge}}{\partial U}\frac{\partial U}{\partial w_{ij}}= -(r(k)-y^{\wedge}(k))\frac{\partial y^{\wedge}}{\partial U}\frac{\partial U}{\partial w_{ij}} \quad \ldots\ldots (7)$$

Where i= 1,2; j=1,2,3, it can be shown from Fig.3 that one of the inputs is u(k), the following equation can be obtained through equation (3):

$$\frac{\partial y^{\wedge}}{\partial U}=\sum_{i=1}^{N} -2\, h_1\, t_1 \times \left(\frac{u-c_{1m}}{\sigma_1^2}\right) \quad \ldots\ldots (8)$$

Where $t_1=\exp(-\sum_{i=1}^{m}\frac{(I_1-c_{1i})^2}{\sigma_1^2}$

u is the mth input of fuzzy-neuron model, the following equations are obtained from equation (4) and the relationships between inputs and outputs for all layers

$$\frac{\partial U}{\partial w_j}=O_j^2(k) \quad (j=1,2,3) \ldots\ldots (9)$$

$$\frac{\partial U}{\partial w_{ij}}=\frac{\partial U}{\partial I^{(3)}}\frac{\partial I^{(3)}}{\partial O_j^{(2)}}\frac{\partial O_j^{(2)}}{\partial I_j^{(2)}}\frac{\partial I_j^{(2)}}{\partial w_{ij}}=w_j\frac{\partial O_j^{(2)}}{\partial I_j^{(2)}}I_i^1$$

( i =1,2 ; j =1,2,3 )

Based on [9] , $\frac{\partial O_j^{(2)}}{\partial I_j^{(2)}}$ can be replaced by

$$\mathrm{Sign}\left(\frac{O_j^{(2)}(k)-O_j^{(2)}(k-1)}{I_j^{(2)}(k)-I_j^{(2)}(k-1)}\right)$$

and this can simplify and standardize the computing method without affecting the convergence direction. To incorporate it into equation (10), the following equation can be obtained

$$\frac{\partial U}{\partial w_{ij}}=w_j\mathrm{Sign}\left(\frac{O_j^{(2)}(k)-O_j^{(2)}(k-1)}{I_j^{(2)}(k)-I_j^{(2)}(k-1)}\right)I_i^1 \quad (i=1,2)\ (j=1,2,3) \quad \ldots\ldots (10)$$

After incorporating equations (8), (9) into equation (10) respectively, the online-tuning mode of each weight can be obtained [3].

## 4. SIMULATION

In the sluice flow control, the system has the character of time delay. Considering delay factor and the liquid level system, the system transfer function (between upstream liquid level with downstream flow rate), can be described as:

$$G(S)=\frac{(64.12\ e^{-0.3\,s})}{s+8.376} \ldots\ldots (11)$$

Simulation block diagram of PID control, and fuzzy neural network PID, are shown in fig. (6), fig. (7) Respectively.

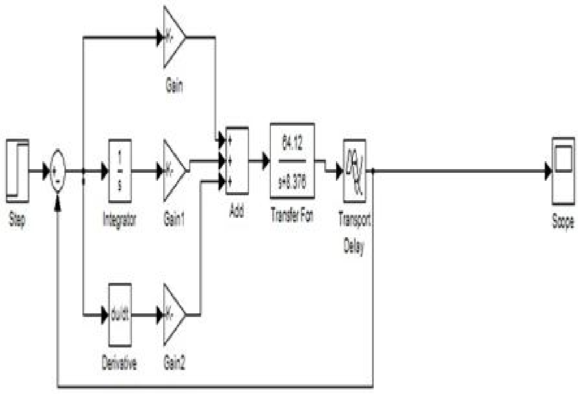



Fig.6 Simulation block diagram of PID control

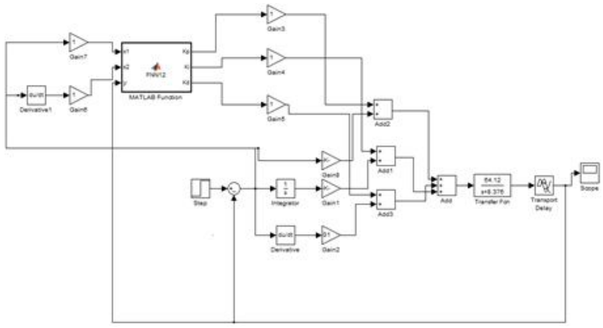


Fig.7 Simulation block diagram of fuzzy neural network PID

From Fig.8, it can be seen that, the conventional PID control has larger overshoot, adjusting time is longer and anti-interference ability is poor. The control algorithms based on fuzzy neural network PID overcomes above-mentioned shortcomings, not only has small overshoot, faster response speed, but also the steady state error is zero. The fuzzy neural network controller for static and dynamic properties are both better, as illustrated in Fig. 9. This algorithm combines the advantage of fuzzy and neural network with PID control algorithms and has good adaptability. Because of the ability to adjust parameters online, the adaptive capacity of algorithm is enhanced further. The PID neural network control system based on fuzzy-neuron model as shown in Fig.10, produces better effects than the previous controllers in both steady and dynamic performance, including shorter steady-state time , non-overshot, non- oscillator, and stronger dynamic tracking capability .

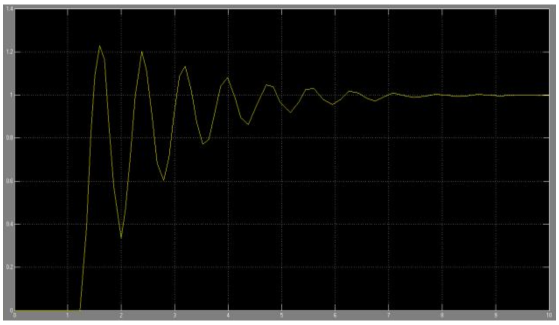


Fig.8 Step response curves of PID control

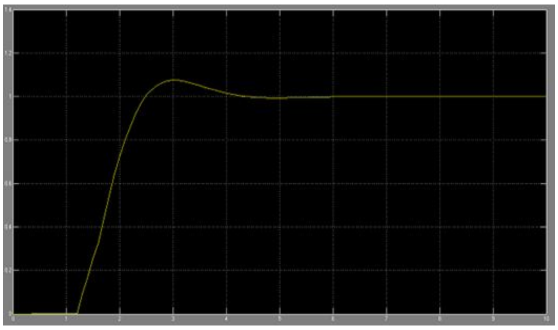


Fig.9 Step response curve of fuzzy neural PID

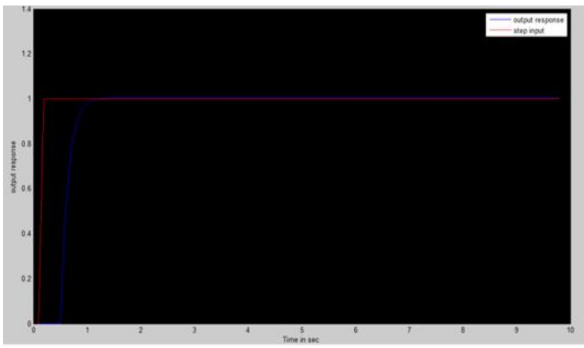



Fig.10 step response curve of PID neural network controller based on fuzzy neural network model

The parameter of control algorithms are summarized in table 1

Table. 1 Parameters of Control Algrithms

| Control algorithm / Parameters | Conventional PID | FNN PID | PID neural network based on FNN |
|---|---|---|---|
| Rise time(sec) | 2.257 | 2.484 | 1.2 |
| Peak time(sec) | 1.592 | 3.0 | 1.4 |
| Settling time(sec) | 9.462 | 6.0 | 1.9 |
| Error steady state | 0.002 | 0.0 | 0.0 |
| Overshoot | 0.227 | 0.076 | 0.0 |

## 5. CONCLUSION

From the simulation result We conclude that the best control algorithm is the third (PID neural network controller based on fuzzy neural network model) because it is characterized by fast response, produces good effects in both steady and dynamic performance, including shorter steady-state time, non-overshot, non- oscillator, and stronger dynamic tracking capability

## References

[1] G. M. El-Kassar , V. Kutija and P. E. OíConnell . design of real-time control algorithm for on-demand operation of irrigation channels . ciheam (2007),p.341-352.

[2] Guohuan Lou , Kangwei Li . Research on Control System for Sluice Gate Flow Based on Fuzzy Neural Network PID , Applied Mechanics and Materials Vols. 198-199 ,201), p1779-1782 .

[3] Changlu Zheng , Jian Fan and MinRui Fei . PID Neural Network Control Research Based on Fuzzy Neural Network Model, Shanghai University, China. .

[4] Dr. Mohammed Yousif Hassan and Mr. Qussay F. Ad'Doory, Design Of Adaptive Fuzzy-Neural PID-Like Controller For Nonlinear MIMO Systems, IJCCCE,VOL.7,NO.1,2007,P: 87-96

[5] S. J. Bassi, M. K. Mishra and E. E. Omizegba, Automatic Tuning Of Proportional–Integral–Derivative (PID) Controller Using Particle Swarm Optimization (PSO) Algorithm, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.4, October 2011.

[6] Michael Watts and Nikola Kasabov, Genetic Algorithms For The Design Of Fuzzy Neural Networks, Department of Information science, University of Otago P.O.Box 56, Dunedin, New Zealand.

[7] C. T. Lin and C. S. G. Lee, Real-time Supervised Structur Parameter Learning For Fuzzy Neural Network, Fuzzy Systems, 1992., IEEE International Conference on, 8-12 Mar 1992, P128