DESIGN AND IMPLEMENTATION OF A NEURAL CONTROL SYSTEM AND PERFORMANCE CHARACTERIZATION WITH PID CONTROLLER FOR WATER LEVEL CONTROL

Md. Selim Hasan[#*1], A. S. M. Shariar Kabir Khan[#*2] S.M. Shibbir Alam [#*3],
Mehedi Hasan Pavel[≠+4]

[#]Department of Electrical & Electronic Engineering, *Khulna University of Engineering and Technology, Khulna, Bangladesh
[1]hsn_selim@yahoo.com;[2]kalpa_khan@yahoo.com;[3] smsa_kuet@yahoo.com

[+] Ahsanullah University of Science & Technology, Dhaka, Bangladesh
[4]mehedi_hasan_pavel@yahoo.com

ABSTRACT*:-The objective of this thesis is to investigate and find a solution by designing the intelligent controller for controlling water level system, such as neural network. The controller also can be specifically run under the circumstance of system disturbances. To achieve these objectives, a prototype of water level control system has been built and implementations of both PID and neural network control algorithms are performed. In PID control, Ziegler Nichols tuning method is used to control the system. In neural network control, the approach of Model Reference Adaptive Neural Network (ANN) Control based on the back propagation algorithm is applied on training the system. Both control algorithms are developed to embed into a standalone DSP-based micro-controller and their performances are compared.*

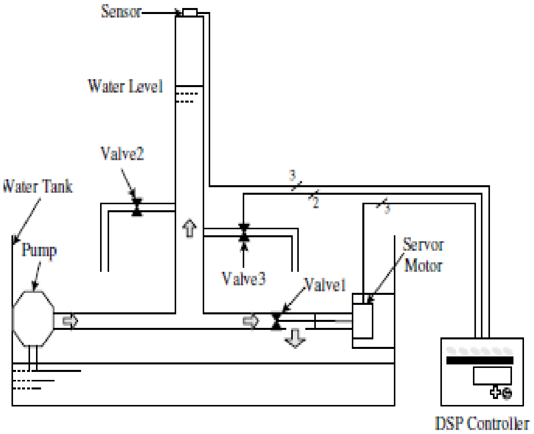


Fig.1: Configuration of water level control system. The system consists of servo motor, valves, pump, infrared sensor and a DSP controller.

KEYWORDS: Neural Network, ANN, Intelligent Controller, PID, Microcontroller.

DOI : 10.5121/ijaia.2011.2206

I. INTRODUCTION:

NOWADAYS, the various parameters in the process of industrial are controlled such as temperature, level, and etc. Some process needs to keep the liquid level in the plant such as water, oil, chemical liquid in its. The level control is a type of control method for common in process system. The level control system must be controlled by the proper controller. The objective of the controller in the level control is to maintain a level set point at a given value and be able to accept new set point values dynamically. The conventional proportional-integral-derivative (PID) is commonly utilized in controlling the level, but the parameter of those controllers must be turned by tuning method either in time response or frequency response to meet their required performances [1,2]. On the other hand, the neural controller is also popularly implemented in many practical industrial automation applications. If the computations required in a task are well understood, and efficient algorithms are known, neural networks are inherently ill-suited for the implementation [12]. There are many papers addressed the PID, fuzzy or neural networks control in the water or liquid level control system. Satean Tunyasrirut chose PID-fuzzy cascade as the model structure for a linear model based predictive control of the liquid level [15]. Riyaz Shariff utilized artificial neural network (ANN) as advanced process control technique for water treatment [3]. Corneliu Lazar Showed a self-learning PID control in the application of level control [5]. Naman et al. presented an adaptive model reference fuzzy controller for controlling the water level in a water tank [10]. Xiao et al. provided a back propagation neural network algorithm used to adjust the parameters of the PID controller and control the liquid level of molten steel smelling non-crystalloid [6].In this paper, we first elaborate the configuration of water level control system. Then, we introduce PID control [5] and model reference adaptive neural network control (MRANNC) [6], [7], [8] strategies based on back propagation algorithm. Finally, some experimental and real-time results using DSP controller for both PID and neural network are presented.

II. SYSTEM CONFIGURATION

Figure 1 shows the system configuration. The system consists of the servo motor, valves, a pump, an infra-red sensor and a DSP controller. Water is constantly pumped into a vertical tube by a pump. Water can outflow from the horizontal tube. The general control objective is to reach and stabilize a certain water level in the vertical tube. As shown in the figure, a sensor measures the water level and sends height signal to the DSP micro-controller. The controller controls the servo motor, which adjusts the valve 1 to maintain a desired water level in the vertical tube. Valve 2 constantly opens and valve 3 randomly opens to provide disturbances to the control system.

III. PID CONTROLLER

The most industrial process can be controlled with PID control (Proportional-Integral-Derivative) provided in equation:

$$u(t) = K\left(e(t) + \frac{1}{Ti}\int_0^t e(t)\,dt + Td\ \frac{e(t)}{dt}\right)$$

Where u is the control variable and e is the control error (e=ysp-y). The parameters of its can be determined by ultimate sensitivity method that using Kc=4.5 and received data from Fig. 2. All data is used for calculated the Parameters of PID controller as shown in Table I.

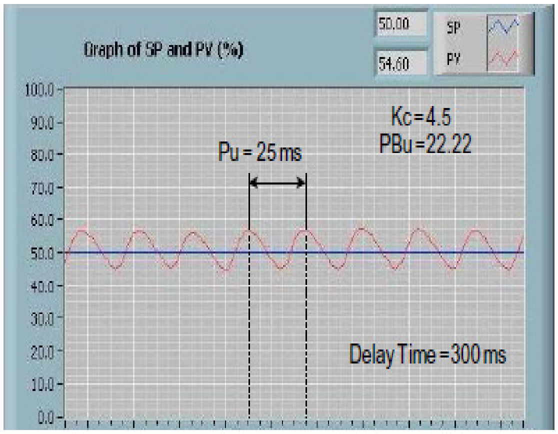
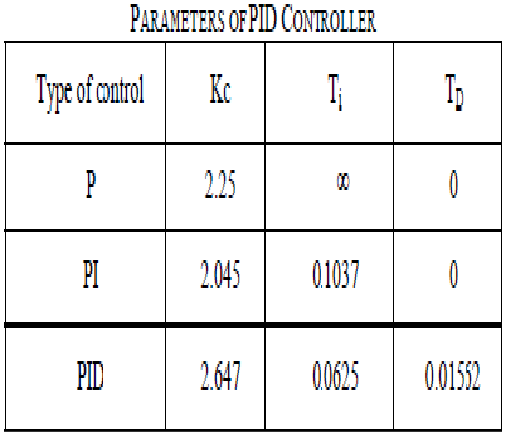


Fig.2. Data result from ultimate sensitivity method



PARAMETERS OF PID CONTROLLER

Type of control	Kc	T_i	T_D
P	2.25	∞	0
PI	2.045	0.1037	0
PID	2.647	0.0625	0.01552

TABLE I

IV. NEAURAL NETWORK CONTROL

In this paper, the Model Reference Adaptive Neural Network Control approach based on back propagation algorithm is applied to implement the water level control system. 3 show the control structure. TDL in Figure 3 denotes time delays of the input or output signals.

A. Neural Network for Identifier

The neural network for identifier is designed as a three layer neural network. It has an input layer, a hidden layer, and an output layer. N_i^i ,N_j^i and N_k^i are their output values respectively. The neuron numbers in the hidden layers can be chosen depending on the practical training result.

The neural network identifier models are trained the forward dynamics of the plant. Six inputs and one output are selected as the identifier model for the system. These six inputs are the control signals: u(t - i), i = 0...2, and the previous output signals: y(t - j), j = 1...3.These current and previous two control signals, and also incorporate in the historical trend from the last three output of the system.

A set of corresponding input-output training patterns is selected from the open-loop continuous signal response of the system. The control input signal is directly added to servo motor of the system and system output signal is reflected with the sensor, which is actual water level. In training a neural network to learn a forward dynamic model of a plant, the back propagation error signal between the output and the hidden layer is expressed as

k $$\delta_k = T_k - N_k^{i} \qquad (1)$$

Where T_k is the target pattern and N_k^{i} is the actual output of the identifier, and between the hidden and input layers, it is expressed as

$$\delta_j = f\,(net_j)\sum^{-} \delta_k \cdot w_{kj} \qquad (2)$$

Here, $f'(net_j)$ is the derivative of the activation function $f(net_j)$

Where $$f(net) = \frac{1}{1 + \exp\,(-net_j)} \qquad (3)$$

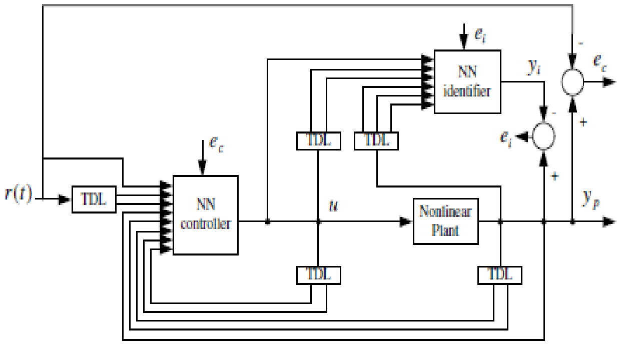


Fig.3.Neural Network in water level control

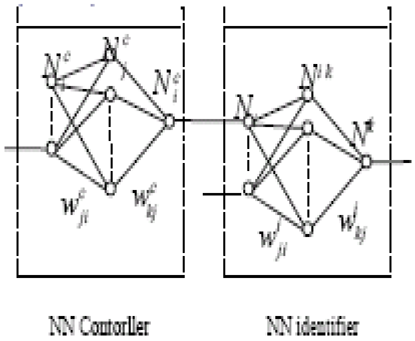


Fig. 4. The Connection between the Controller and the Identifier

The weights between the input and hidden layers are updated as:

$$_{i} \quad \Delta w_{ji}(t+1) = \eta \cdot \delta_{j} \cdot N^{i} + \alpha \cdot \Delta w_{ji}(t) \tag{4}$$

and the weights between the hidden and output layer are updated as:

$$^{i} \quad \Delta w_{kj}(t+1) = \eta \cdot \delta_{k} \cdot N_{j} + \alpha \cdot \Delta w_{kj}(t) \tag{5}$$

Where

N^i_i and N^i_j are the outputs of the input and hidden layers respectively, η is the learning rate and α is the momentum coefficient. The constants η, α are all chosen empirically.

B. *Neural Network for Controller*

The neural network controller is created directly based on the neural network identifier. Its design is fully incorporated the learning strategy into the trained identifier. The weights of the neural network identifier are constantly verified against the actual plant output. This ensures that the weights allow the neural network identifier to properly predict actual plant output. The neural network identifier is used as means to back propagate the performance error to get the equivalent error at the output of the neural network controller. The accuracy of the plant model is critical in ensuring that the controller is accurate as well. The error between the plant output and the identifier output is also checked for the accuracy level of the identifier. This error is used to back propagate adjust the weights of the identifier to provide the most accurate representation of the plant. The neural network for controller is also designed as a three-layer neural network. It has an input layer, a hidden layer and an output layer.
The neuron numbers in the hidden layers can be chosen also depending on the practical training result.

To illustrate the derivation of the error signals for the neural network controller, Figure 4 provides the connections between the controller and identifier networks. The adaptation of the weights of the neural network controller between the hidden and output layers can be derived as follows:

$$\Delta w^c_{kj} = \eta . \frac{\delta E^i}{\delta W^c_{ki}} \tag{6}$$

Where

$$E^{i} = \frac{1}{2}.\,(y_{p} - r)^{2} \qquad (7)$$

Where y_{p} and **r** are the actual and desired plant output. The superscript c denotes the variables belonging to the neural network controller and superscript i denote the variables belonging to the neural network identifier. Using chain rule, equation (6) can be expanded as follows:

$$\Delta w^{c}_{kj} = \eta.\,\frac{\delta E^{i}}{\delta net^{c}_{ki}}.\,\frac{\delta net^{c}_{kj}}{\delta W^{c}_{ki}}$$

So,

$$\Delta w^{c}_{kj} = \eta.\ \delta^{c}_{k}\,.N^{c}_{j} \qquad (8)$$

Where

$$\delta^{c}_{k} = \frac{\delta E^{i}}{\delta net^{c}_{ki}}$$

And

$$N^{c}_{j} = \frac{\delta net^{c}_{kj}}{\delta w^{c}_{ki}}$$

δ^{c}_{k} is the error signal between the hidden and output layers of the neural network controller .Linear functions are used at the input and output neurons of the neural networks between the controller and identifier. Therefore, δ^{c}_{k} can be represented as:

$$\delta^{c}_{k} = \frac{\delta E^{i}}{\delta N^{c}_{k}} = \frac{\delta E^{i}}{\delta\, net^{i}_{i}} = \frac{\delta E^{i}}{\delta N^{i}_{i}} \qquad (9)$$

Where net^{i}_{i} and N^{i}_{i} are the input and output of the identifier input layer neurons. Further using chain rule

$$\delta^{c}_{k} = \frac{\delta E^{i}}{\delta net^{i}_{j}}.\,\frac{\delta net^{i}_{j}}{\delta N^{i}_{i}}$$

$$\frac{\delta E^{i}}{\delta net^{i}_{j}} = \delta^{i}_{j} \qquad (10)$$

And

$$\frac{\partial net^{i}_{j}}{\partial N^{i}_{i}} = w^{i}_{ji}\ .$$

So,

$$\partial^{c}_{k=\partial^{i}_{j}.w^{i}_{ji}} \qquad (11)$$

The error signal between the input and hidden layer of the neural network controller can be derived as follows:

$$\partial^{c}_{j} = \frac{\partial E^{i}}{\partial net^{c}_{j}} = \frac{\partial E^{i}}{\partial net^{c}_{j}}.\frac{\partial N^{c}_{j}}{\partial net^{c}_{j}} \qquad (12)$$

By using chain rule

$$\frac{\delta E^{i}}{\delta net^{i}_{j}} = \Sigma\frac{\delta E^{i}}{\delta net^{c}_{k}}\,.\,\frac{\delta net^{c}_{k}}{\delta N^{c}_{j}}$$

$$\frac{\partial N^{c}_{j}}{\partial net^{c}_{j}} = f(net^{c}_{j}$$

Where

$$f(net^{c}_{j}) = N^{c}_{j}\,.(1-N^{c}_{j})$$

So,

$$\partial_j^c = N_j^c \,.\, (1- N_j^c).\,(\textstyle\sum \partial_k^c)\,.\, w_{kj}^c \qquad (13)$$

The weights of the neural network identifier can be further improved online if necessary. This can be reached by back- propagation of the following error equation through the neural network identifier at every sample. y_p and y_i are the outputs of the actual plant and neural network identifier respectively

$$e=0.5*(y_p - y_i)^2 \qquad (14)$$

V. EXPERIMENTAL RESULT:

The neural network control algorithm were first trained and tested with C language with a normal PC. Then, the algorithm were ported to DSP-based controller and tested.

*A. **Training and Testing for Neural Network Control***

1) Parameters Setup: In order to have quicker calculation without sacrificing performance, some parameters were setup as shown in Table III. In Table III, n_i,n_j and n_k denote the number of nodes in the input layer, hidden layer and output layer, respectively, η and α denote learning rate and momentum term respectively.

2) Identifier Training

Identifier training resulted to a level of error of 0.0005 after 100k epochs. At the cut-off of 100k epochs, the identifier was still converging on an error, but extremely slowly. Then, this level of an error was used to move onto controller training. Figure 5 shows the trend of decreasing error in the identifier training.

3) Controller Training and Testing: In order to effectively create a controller for the system, the existing system was analyzed in an off-line setup. The goal in the project is to develop a controller through simulation and use more computer resources in a highly effective manner. It proved far more effective to design the training and testing in an off-line format.

1. Controller Training

In Figure 8, convergence towards the goal of small error is clearly seen in the controller training algorithm. However, the error does not reach the desired cut-off. Despite this, the weights trained as a result of this training provide a very good response when the testing data is used to evaluate the controller weights.

2. Controller Testing

As shown in Figure 10, the weights for both the identifier and controller had been trained to a level that provided an acceptable level of prediction, and showed that the controller was working as expected.

NN	n_i	n_j	n_k	η	α
Identifier	6	10	1	0.00001	0.82
Controller	10	20	1	0.00005	0.66

TABLE III

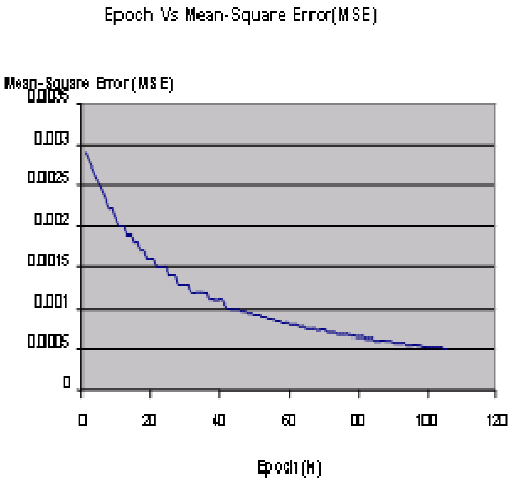


Fig:5: Identifier Training Error

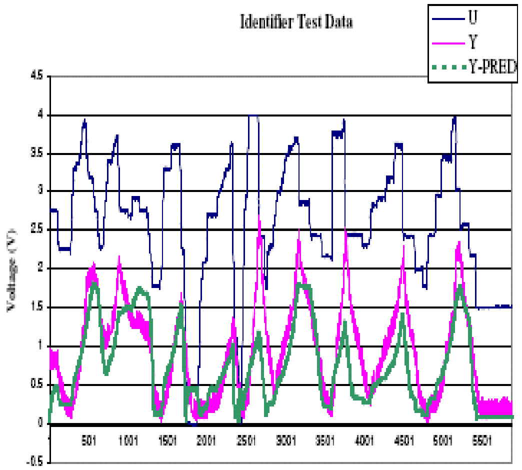


Fig .6.Identifier Testing Results

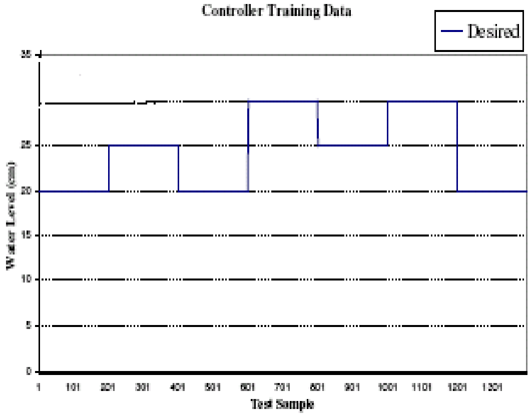


Fig.7. Controller Training Data

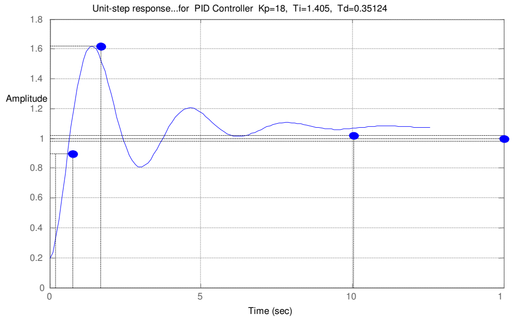


Fig. 9.Unit Step Response of PID

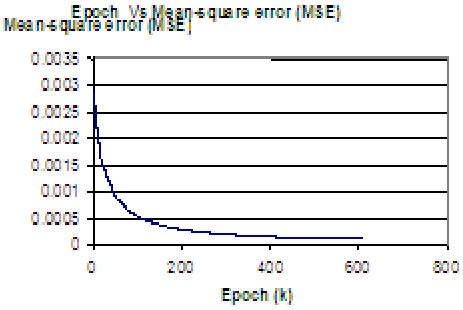


Fig:8: Controller Testing Error

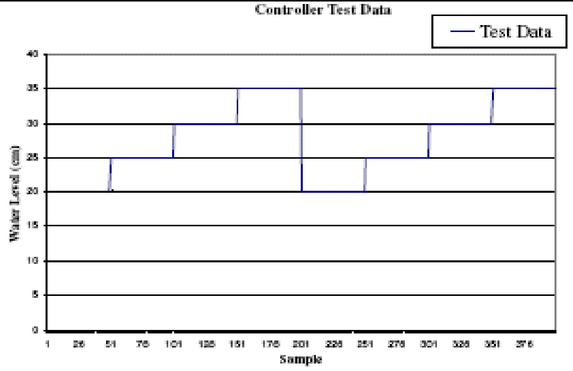


Fig:10:Controller Testing Data

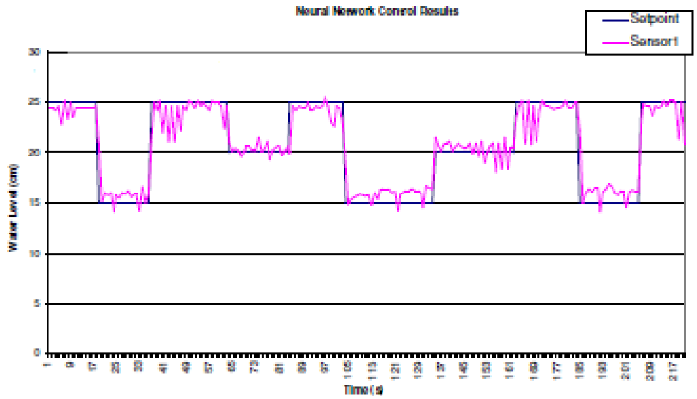


Fig.11. Online Neural Network Control Result

B. Experiments on micro-controller

The neural network control algorithms were ported to a stand-alone VRP MTSC32 micro-controller. The main board of micro-controller is based on Texas Instruments' TMS320C32 DSP with a CAN bus communication port that facilitates the communication with a PC for monitoring online parameters. The control algorithms were developed with C language, complied with *TI* Compiler and run from the micro- controller. To provide an objective comparison between the PID and neural network control algorithms, two tests were carried out in the VRP MTSC32 DSP micro-controller. Figure 09 shows PID control result and Figure 11 shows the neural control result.

	PID	NNC
Plant Math Model	Necessary	Not Necessary
Tracing Performance	Good	Better
Disturbance Rejection	Poor	Good

TABLE IV COMPARISON BETWEEN THE PID CONTROL AND NEUR-AL NETWORK CONTROL BASED ON THE EXPERIMENT

VI. CONCLUTION:

This thesis proposes the development of neural network based linearizers to linearize highly nonlinear input-output characteristics of a liquid level control system. Two possible solutions employing two different variants of Back-propagation Neural Network (BPNN) are investigated. Between these BPNN with adaptive learning rate and momentum with momentum factor based neural linearizer showed the best possible results.

VII. FUTURE WORKS:

The research described in this thesis was concerned with the design and implementation of a neural control system for water level control. The proposed neural control system has been successfully designed with many exciting results. The works remaining for future study are discussed as:

The performance of the neural control system can further increase by more development of the software. If the bit number of the A/D converter increases, then the proposed system can be used for higher range applications.

REFERENCES:

[1] Niimura, T. and Yokoyama, R., “Water level control of small-scale hydro- generating units by fuzzy logic”, Proceedings of IEEE Int’l Conference on Systems, Man and Cybernetics, pp. 2483, 1995.

[2] Roubos,J.A., Babuska,R., Bruijn,R.M. and Verbruggen,H.B., “Predictive Control by Local Linearizatin of a Takagi-Sugeno Fuzzy Model”, IEEE Transactions, 1998.

[3] Riyaz Shariff, Audrey Cudrak and Qing Zhang “Advanced process control techniques for water treatment using artificial neural networks” J. Environ. Eng. Sci. **3**(S1): S61–S67 (2004).

[4] Naman,A.T., Abdulmuin,M.Z. and Arof,H., “Development and Appli- cation of a Gradient Descent Method in Adaptive Model Reference Fuzzy Control”, 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium, pp. 358, 2000

[5] Xiao,Y., Hu,H., Jiang,H., Zhou,J.and Yang,Q., “A Adaptive Control Based Neural Network for Liquid Level of Molten Steel Smelting Noncrystlloid Flimsy Alloy Line”, Proc. of 4t h World Congress on Intelligent Control and Automation, China, 2002.

[6] S. Haykin, Neural Networks: “A comprehensive foundation”, 2002.

[7] Narendra, K.S. and Parthasarathy, K., “Identification and Control of Dynamical Systems Using Neural Networks”, Neural Networks, IEEE Transactions on,Vol.1, No.1,Mar,1990.

[8] Takagi,T. and Sugeno,M., "Fuzzy Identification of System and its Appli- caiton to Modeling and Control", IEEE Transactions on Systems, Man and Cybernetics, Vol.15, No.1, 1985.

[9] Murakami,K. and Sugeno,M., "An Experimental Study on Fuzzy Parking Control Using a Model Car", Industrial Application of Fuzzy Control, North Holland, 1985.

[10] Yamada, T.and Yabuta, T., "Dynamic system identification using neural networks", Systems, Man and Cybernetics, IEEE Transactions on, Vol.23, Issue 1,Jan/Feb, 1993.

[11] Mistry, S.I.and Nair, S.S., "Identification and Control Experiments Using Neural Designs", IEEE Control Systems Magazine, Vol.14, Issue 3, Jun,1994.

[12] N.K Bose, P. Liang, "Neural network fundamentals with graphs, algorithms and applications", 2001.

[13] International Journal of knowledge based and Intelligent Engineering Systems, Volume 9, Number 3, 2005.

[14] G. Castellano and A.M. Fanelli, Fuzzy Interference and rule extraction using a neural network, Neural network world journal 3, 2000.

[15] Ghwanmeh,S.H., Jones,K.O. and Williams,D., "On-line Performance Evaluation of a Self-Learning Fuzzy Logic Controller Applied to Non- Linear Processes", *IEEE Transactions*, 1996

[16] Pu Han, Xin-Jing Mao, Song-Ming Jiao and Hai-Rong Sun "Adaptive Neural Network Control for Drum Water Level Based on Fuzzy Self-Tuning" IEEE Transactions on Control Systems Technology, March 2006.

[17] Parlos, A.G.; Sanjay Parthasarathy; Atiya, A.F.; "Neuro-predictive process control using online controller adaptation "IEEE Control Systems Society, August 2002.

[18] S. Alvisi, G. Mascellani and A. Bardossy; "Water level forecasting through fuzzy logic and artificial neural network approaches " Hydrology and Earth System Sciences, 10, 1-17, 2006.