# Tuning of PID Parameters Using Artificial Neural Network

Nisha[1], Avneesh Mittal[2], O.P.Sharma[3], Nitu Dhyani, Vijay Sharma, Avinashi Kapoor[1] and T.K. Saxena
*National Physical Laboratory, Dr. K.S. Krishnan Road, New Delhi – 110 012*
*[1] Department of Electronics, University of Delhi South Campus, New Delhi 110 021*
*[2]Bhaskaracharya College of Applied Sciences, University of Delhi, Dwarka, ND-110 075*
*[3]Kirori Mal College, University of Delhi, Delhi-110 007*
***e-mail Corresponding Author: tushyaks@gmail.com***

*Abstract:* **Tuning of the PID controller in a varying environment is extremely difficult. For this purpose one has to use the adaptive PID controller. In the present paper a novel method for fast tuning of the PID controller has been presented and implemented on designed and developed hardware around the 89C51 microcontroller. Varying environment in the very old existing MLW-MK70, former East German bath has been created with the help of two microcontrollers. The artificial neural network (ANN) has been used to tune the PID parameters. The software has been written in Visual BASIC5.0 language**.

**Keywords:** PID, Artificial Neural Network (**ANN**), Temperature Controller, Adaptive Control

## I. INTRODUCTION

Adaptive nature of Artificial Neural Network controllers[1-6] have made them a major area of interest among researchers in widespread fields[7-13], mainly because ANN controllers can efficiently learn the unknown or continuously varying environment and act accordingly. Industrial automation applications prefer PID (proportional Integral Derivative) controllers because of its simple structure and robustness etc**.** In this paper an ANN has been employed for tuning of PID parameters. The hardware has been designed and fabricated around Atmel's 89C51 microcontroller to control the temperature in a 30 year old, former East German, water bath MLW MK 70. A specially designed varying environment has been created in the water bath using two microcontrollers. In this system, fresh water is allowed to flow continuously at a rate according to outflow of the hot water. The level of the water is kept constant inside the bath.

## II. DESIGN APPROACH

For training/ learning the environment ANN has been utilized. By varying the environment the neural network was allowed to learn the system and accordingly the PID Parameters.

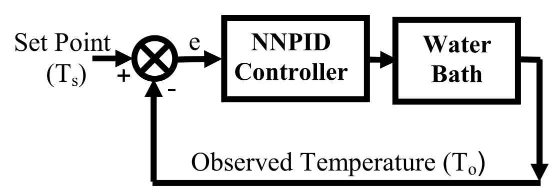
A well known continuous PID controller is described using

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

where u is the controller output, $K_P$ is the proportional gain, $K_I$ is the integral time, $K_D$ is the derivative time, and e is the error between the set point and the process output. For a digital control of $t_s$ sampling periods, we can write

$$u = K_P\left(e_n + \frac{1}{K_i}\sum_{j=1}^{n} e_j t_s + K_D\frac{e_n - e_{n-1}}{t_s}\right) \qquad (2)$$

The figure (1) shows the block diagram of the approach followed in the present work




$$e = T_s - T_o$$

Fig.1: Block Diagram of the approach followed

A Neural Network tuned PID (NNPID) which has two inputs, one output and three layers which are input layer, hidden layer and output layer. The input layer has two neurons and the output layer has one and their neurons are P-neurons. The hidden layer has three neurons and they are P-neuron ($H_1$), I-neuron ($H_2$) and D-neuron ($H_3$) respectively. The NNPID is shown in Fig.2

In NNPID when suitable connective weights are chosen, a NNPID becomes a conventional PID controller.

In the present case weights for the first layer to hidden layer are takes as

$w_{1hj}= +1$, $w_{2hj}= -1$, $w_{1ho} = K_P$, $w_{2ho} = K_I$, $w_{3ho} = K_D$, then,

$$H_{1i}=w_{1h1}I_1+ w_{2h1}I_2=T_s - T_{os} = T_{err}$$
$$H_{2i}=w_{1h2}I_1+ w_{2h2}I_2=T_s - T_{os} = T_{err} \quad (3)$$
$$H_{3i}=w_{1h3}I_1+ w_{2h3}I_2=T_s - T_{os} = T_{err}$$

Where $T_s$ is the set point and temperature $T_{os}$ is observed temperature and $T_{err}$ is the error. $H_{1i}$, $H_{2i}$ and $H_{3i}$ are input part of hidden layer nodes. The output of the hidden layer nodes are:

$$H_{1o}= T_{err}$$
$$H_{2o}=\int T_{err}\, dt \quad (4)$$
$$H_{3o}=d\, T_{err} /dt$$

then,

$$O_{1i}= w_{1ho}H_{1o} + w_{2ho}H_{2o} + w_{3ho}H_{3o} \quad (5)$$
$$=K_P T_{err} + K_i \int T_{err}\, dt + K_d \frac{dT_{err}}{dt}$$

where $H_{1o}$, $H_{2o}$ and $H_{3o}$ are output part of hidden layer nodes. $O_{1i}$ is the input part of output layer.

This way NNPID as auto tuned PID in a varying environment.. It is well known then most neural networks can't be practically used in system control because the initial connective weights of the neural networks are randomly selected thus the neural controller cannot work as a stable system.

PID controllers have been widely used in industry and there are much more experiences to choose P, I, D parameters in order to suit the stability. So, if a NNPID equals to a PID controller, it has the useful foundation of the use. Then, via training and study, NNPID can get better control performances.

### A. Back-propagation algorithms

In the present control system, the aim of the NNPID algorithms is to tune the PID parameters in such a way that the mean square error (MSE) is minimum which is given by

$$MSE = \int_0^\infty [T_{err}]^2 dt$$

The weights of NNPID are changed by Steepest descent in on-line training process. The details of the weight adjustments used are as given in reference [5]. The training of the neural network has been done by varying the PID parameters and taking the sample online. The NNPID was trained with a total of 50 sets of PID parameters each having 360 data points

## III. EXPERIMENTAL SETUP

Experimental setup is shown in the block diagram of Fig. 3. An Artificial Neural Network tuned PID controller has been designed and fabricated. The hardware for controlling the bath has been designed and fabricated around the Atmel microcontroller 89C51. The temperature of the bath is acquired with the help of Platinum Resistance Thermometer (PRT). A current of 1mA is fed to the PRT with the help of constant current source. The PRT gives the output in voltage form. The voltage is then amplified to give the temperature directly. This voltage is then fed to the 14 bit ADC. The digitized voltage is then sent to the PC by microcontroller. The program in PC does the calculations using the NNPID algorithm. After doing all the calculations it generates the firing angle to control the energy in the water bath and sends the same to the microcontroller. Thereafter microcontroller triggers the triac accordingly. The NNPID program in PC

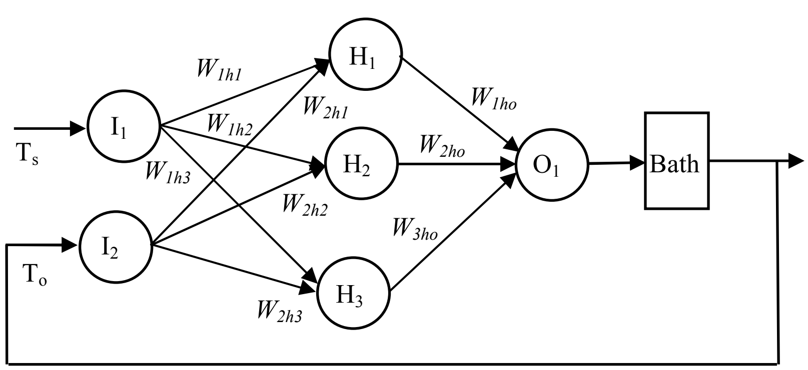
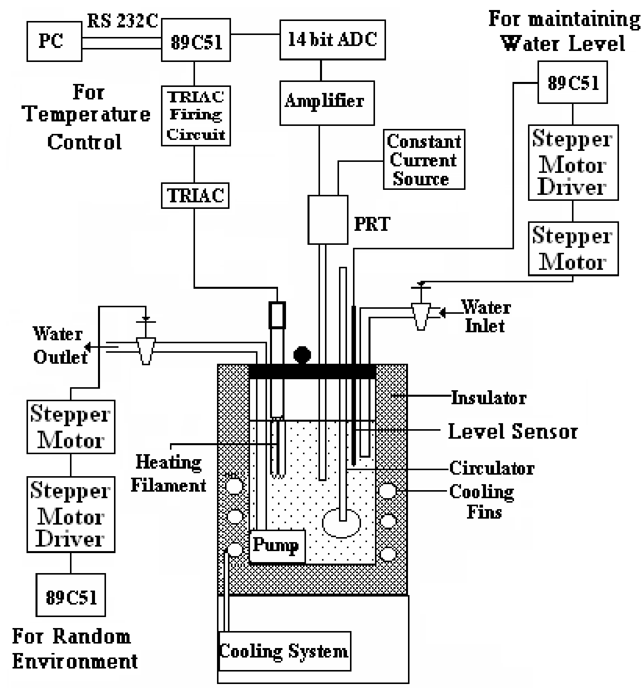



**Fig.2: Neural Network tuning of PID Controller**

continuously monitors the temperature and accordingly controls the same in the bath. In case it sees any change in it then itself learns the environment and modifies the parameters of the temperature controller in such a way that it will control the system efficiently. The NNPID program in PC, which has been written in Visual BASIC-5.0 language, has been made very user friendly. It guides the user at every step. The program stores the data in the user defined file as well as plots the online data in the form of graph on the screen. The program for microcontroller has been written in 8051 assembly language. A specially designed varying environment is created by continuously in flowing fresh water in such a way that the level of the water inside the bath remains constant even if the hot water is removed at random outflow rates. Uniform heat distribution is maintained using the circulator and the isolated system is used to minimize external disturbance. The cooling is achieved at a constant rate using the refrigeration system of the bath.

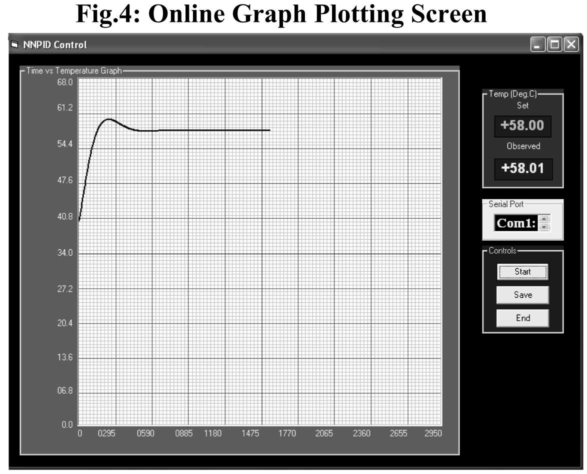
**Fig.3: Block Diagram of the Setup**




## IV. RESULTS AND DISCUSSION

An NNPID temperature control system *has been designed* to be tested at various set point temperatures in the MLW-MK-70 bath under varying environment. The PID parameters of the controller have been automatically tuned to give the satisfactory results by Artificial Neural Network (ANN). A typical temperature Vs time online graph is obtained at a set point temperature is shown in the Fig.4.This online graph shows that, the system is controlling the temperature satisfactorily. This indicates that the NNPID controller can be used in place of the conventional PID controller after proper tuning.

**Fig.4: Online Graph Plotting Screen**




## V. REFERENCES

[1] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural networks for control*, The MIT Press, 1990.

[2] K. S. Narendra and S. Mukhopadhyay, *Adaptive control of nonlinear multivariable systems using neural networks*, Neural Networks, vol. 7, 1994.

[3]. Åström, K. J. and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, MA, 1989.

[4]. Isermann, R., K. Lachmann and D. Matko, *Adaptive Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[5] S.Rajasekaran, G.A. Vijaylakshmi Pai, *Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications*. Printice Hall of India, 2007.

6] G Emila, Marra and Natale, Use of neural networks for quick and accyrate auto-tunning of PID controller, Robotics and Comp Integ Manuf, Vol 23, Apr 2007, pp 170-179.

[7] I.J. Gyo¨ ngy, D.W. Clarke, *On the automatic tuning and adaptation of PID Controllers*, Control Engineering Practice 14 (2006) 149–163

[8] G. Chen, *Conventional and fuzzy PID controllers: An overview*, Int J. Intell. Control Syst., Vol. 1, pp. 235 -246, 1996.

[9] Jantzen J, *Tuning of Fuzzy PID Controllers*, online, Technical University of Denmark, Department of automation, 98-H-871(fpid), Sep 1998.

[10] J. S. R. Jang, *ANFIS: Adaptive-Network-Based Fuzzy Inference System*, IEEE Trans. On Systems, Man, and Cybernetics, Vol. 23, pp. 665 – 685, 1993.

[11] C. Lee, and C. Teng, *Tuning of PID controllers for unstable processes based on gain and phase margin specifications: a fuzzyneural approach*, Fuzzy Sets and Systems, Vol. 128, pp. 95 – 106, 2002.

[12] Seema Chopra, R. Mitra, Vijay Kumar, *A NeuroFuzzy Learning and its Application to Control System*, Int J of Computational Intelligence, Vol 3, 2..0, pp. 72- 78

[13] Gyongy, Clarke, *On the automatic tuning and adaptation of PID controllers*, Control Eng Practice (2006) 149-163