Design and Implementation of a Neural Network based Controller on the Benchmark of Conical Tank Level System

Sabarna Ghosh Dastidar
Department of Electronics and Communication Engineering
Institute of Engineering and Management
Kolkata, India
gdsabarna353@gmail.com

Atanu Panda
Department of Electronics and Communication Engineering
Institute of Engineering and Management
Kolkata, India
atanu.panda@ iemcal.com

***Abstract*—In the field of process industries, it became a very challenging problem to control the inlet flow of liquid and maintain a desired level of liquid at a constant value in a tank. In this paper, the discussion has been done on the procedure of designing a controller for a conical tank-level system. The modeling for a conical tank system is a little bit more complex than other linear tank systems (e.g. cubical, cylindrical, etc.) due to the change of its cross-sectional area from the bottom to top but it provides complete drainage as a result of it. Here the feed-forward back-propagation neural network is used to implement the architecture of the controller and then the network is trained using the Levenberg-Marquardt algorithm to control the inlet flow of the liquid in order to maintain the desired level of liquid in the tank at a constant value. From the simulation, it is observed that the neural network controller provides good results while performing the servo and regulatory response of the system.**

Keywords—Conical Tank System, Feed Forward Back-propagation Neural Network, Levenberg-Marquardt algorithm

I. Introduction

The control of liquid level in tanks and flow between the tanks is a basic problem in process industries. The process industries require the liquids to be pumped or stored in tanks and then transfer to other tanks. Many times the liquid will be processed by chemical or mixing treatment in tanks, but always the level of the liquid in the tank must be controlled. It becomes more challenging when the process becomes nonlinear. A conical tank level system is considered here for the research work because this is a highly nonlinear process with varying cross-sectional area where its area gets steeper towards the end for the guaranteed drainage of fluids [5] in chemical industries. Conical tanks find wide applications in process industries such as food processing industries, petrochemical industries, and sewage and wastewater treatment industries. So it becomes a very challenging task to control the liquid flow in a conical tank in order to obtain the desired response.

The majority of the control theory deals with the design of linear controllers for linear systems. PID controller proved to be a perfect controller for simple and linear processes. But when it comes to the control of nonlinear and multi-variable processes, the controller parameters have to be continuously adjusted. Non-linear adaptive control strategies [2] are becoming very much useful here by adjusting the control parameters of a feedback control system in real-time based on changes in the system or its environment. The main significance of adaptive control is that it can provide improved performance and robustness in systems that are subject to changes or uncertainties. So, these methods are very effective to compute an approximated input-output model from a large number of given input and output data sets. As the process is adaptive, it updates the controller parameters in real-time to better respond to the variation of the dynamic behavior of a process due to sudden disturbance or plant parameter variation, or model uncertainties.

In this work, the neural network is used as a significant part of the proposed control scheme. This approach leverages the power of machine learning techniques to approximate the nonlinear behavior of the conical tank system [7] and adaptively control the liquid level in real-time. The actual output of the process is compared to the desired response obtained from a reference model, and based on the calculated error, the controller parameters are tuned so that the process output reaches the desired response within a finite time, even in the presence of external disturbances.

II. Mathematical Modelling

The schematic diagram of a conical tank level system is shown in Fig. 1. Here, the input of the system is the inlet liquid flow of the tank and the output is the liquid level of the tank which should be maintained at a constant value. This desired response can be obtained by continuously manipulating the inlet flow of liquid. Thus, the inlet flow rate is referred to as the manipulated variable, while the liquid level is known as the process variable in this system. By manipulating the inlet flow rate of liquid, the control system can maintain the liquid level at the desired set point, ensuring optimal performance of the process.

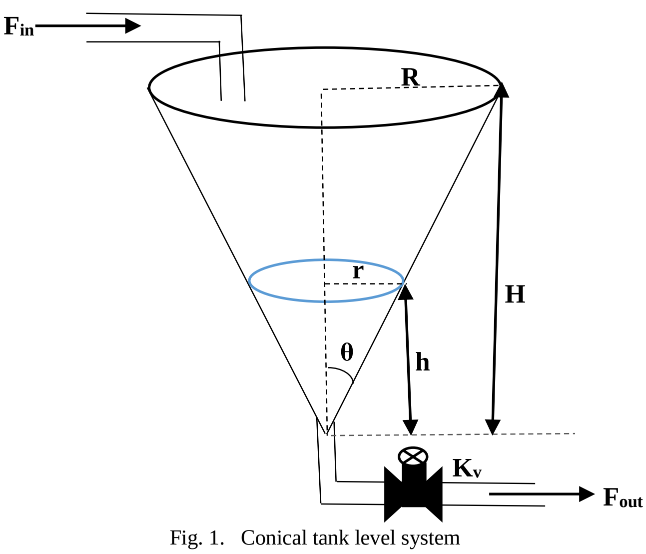

Fig. 1. Conical tank level system

F_{in} – the rate of inlet flow in the conical tank

F_{out} – the rate of outlet flow in the conical tank

K_v – outlet valve coefficient

R – maximum radius of the conical tank

H – maximum height of the conical tank

r – the radius of the water level in the tank

h – the height of the water level in the conical tank

Now, from the above figure, we can say that,

$$\tan\theta = \frac{r}{h} = \frac{R}{H} \tag{1}$$

Now, if we assume V as the volume of the liquid level in the tank, then we can say that the rate of accumulation of the liquid in the tank can be expressed as,

$$\frac{dV}{dt} = F_{in} - F_{out} \tag{2}$$

Now, we know that the volume of the liquid in a conical tank can be expressed as,

$$V = \frac{1}{3}\pi r^2 h \tag{3}$$

Now, from equation (1), we can say,

$$r = \frac{R}{H}h$$

So, putting the value of r in equation (3), we get,

$$V = \frac{1}{3}\pi\left(\frac{R}{H}h\right)^2 h$$

Or, $$V = \frac{1}{3}\pi\frac{R^2}{H^2}h^3$$

Differentiating on the both sides, we get,

$$\frac{dV}{dt} = \frac{1}{3}\pi\frac{R^2}{H^2} \cdot 3h^2\frac{dh}{dt}$$

Or, $$\frac{dV}{dt} = \frac{\pi R^2}{H^2} \cdot h^2\frac{dh}{dt}$$

Now, putting the value of $\frac{dV}{dt}$ in equation (2), we get,

$$F_{in} - F_{out} = \frac{\pi R^2}{H^2} \cdot h^2\frac{dh}{dt} \tag{4}$$

Now, the rate of outlet flow [1] can be expressed as,

$$F_{out} = K_v\sqrt{h}$$

So, from equation (4), we get,

$$F_{in} - K_v\sqrt{h} = \frac{\pi R^2}{H^2} \cdot h^2\frac{dh}{dt}$$

Or,

$$\frac{dh}{dt} = \frac{1}{h^2} \cdot \frac{H^2}{\pi R^2} \cdot (F_{in} - K_v\sqrt{h}) \tag{5}$$

Now, based on the above equation, the mathematical model of the conical tank level system is built in Matlab Simulink and the internal specifications of the system are mentioned in Table I,

TABLE I. INTERNAL SPECIFICATIONS OF CONICAL TANK LEVEL SYSTEM

Name of the Variables	Values
Range of inlet flow (F_{in})	0-220 LPH
Outlet valve coefficient (K_v)	22
Height of the tank (H)	73 cm
Radius of the tank (R)	19.25 cm
Range of liquid level (h)	10-60 cm

III. CONTROL METHODOLOGIES

1. *Proposed Neural Network-based control scheme*

1.1. Designing of the controller architecture

The neural network is an adaptive system that processes information using a connectionist approach. It consists of artificial neurons that are interconnected and its structure changes based on external or internal information that flows through the network during the learning phase. The design of a neural network is fully incorporated into the learning strategy of the trained identifier. The weights of the neural network identifier are constantly verified against the actual plant output to ensure proper prediction. The feed-forward back-propagation neural network is used as the network architecture here. It consists of one input layer, one hidden layer, and one output layer. It processes information in a unidirectional manner, from the input layer through hidden layers to the output layer, without any cycles or loops [2]. For the training of the network, the dataset obtained from the simulation of the reference model is imported into the network. From that dataset, the values of the generated errors and the previous plant outputs are imported as the input dataset and the values of the present control flow are imported as the target dataset to train the network. Hence, two neurons are taken in the input layer and one neuron is taken in the output layer of the architecture. Using six neurons in the hidden layer, a better training result is observed. In the hidden layer, the hyperbolic tangent sigmoid transfer function (tansig) is used as the activation function and the linear transfer function (purelin) is used in the output layer. The implementation of the internal architecture of the network is shown in Fig. 3. The main concept of this architecture is to use a back-propagation algorithm, which involves adjusting the weights of the neurons in each layer based on the difference between the actual output and the desired output. The aim is to minimize the error between the actual and desired outputs, which is accomplished through multiple iterations of training.

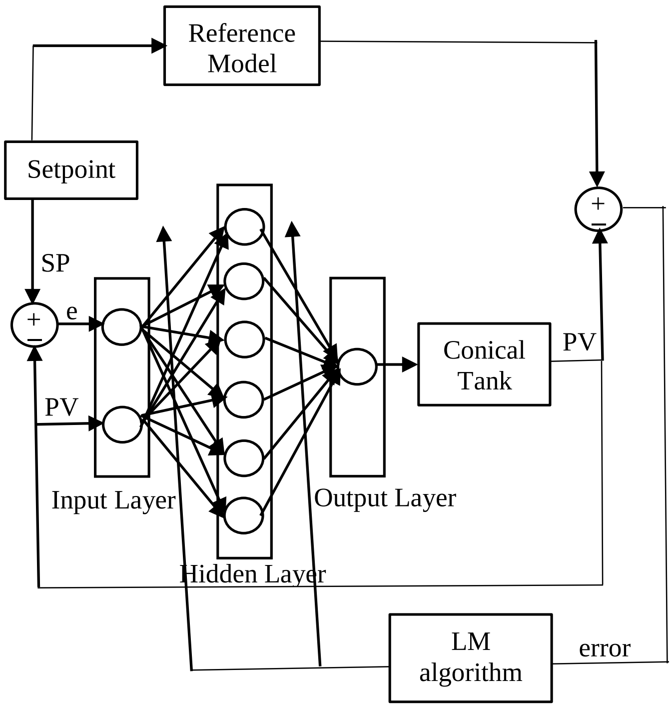

Fig. 2. Architecture of Neural Network based model

1.2. Levenberg-Marquardt training algorithm

Levenberg-Marquardt (LM) back-propagation algorithm [3] is used here to train the network. This algorithm is fast, has a very stable convergence, and provides a numerical solution for the problem of minimizing a non-linear function. This algorithm is quite a simple but extremely robust method for function approximation. It is a combination of the steepest descent algorithm and the Gauss-Newton algorithm that can converge quickly and stably for small and medium-sized problems. The LM algorithm is a numerical optimization technique that is widely used for solving non-linear least squares problems. It is a modification of the Gauss-Newton method that adds a damping parameter to improve the stability and convergence of the algorithm. This damping parameter enables the LM algorithm to handle ill-conditioned problems that the Gauss-Newton algorithm may fail to converge. The expression of this algorithm can be given as:

$$(JJ^T + \lambda I)\ \delta = J^T E \tag{6}$$

Where J represents Jacobian matrix, λ is Levenberg damping factor, δ is the update vector of weight and E represents the error vector. The damping factor λ is adjusted for every iteration. The Jacobian matrix can be given as:

$$J = \begin{bmatrix} \frac{\partial F(x_1,w)}{\partial w_1} & \cdots & \frac{\partial F(x_1,w)}{\partial w_N} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \frac{\partial F(x_N,w)}{\partial w_1} & \cdots & \frac{\partial F(x_N,w)}{\partial w_N} \end{bmatrix} \tag{7}$$

The Jacobian matrix J is a matrix of partial derivatives that describes the sensitivity of the output of the network to changes in its weights. Equation (7) shows the general form of the Jacobian matrix. Here, $F(x_i, w)$ represents the network function, and w is the weight vector of the network. The Levenberg damping factor λ is adjusted for every iteration to ensure that the algorithm converges to the optimal solution. If λ is too small, the algorithm may converge slowly or even diverge. If λ is too large, the algorithm may overshoot the optimal solution.

1.3. Working principle of neural network

Neural networks are composed of interconnected neurons, as depicted in Fig. 3, where each connection between neurons represents the flow of data. Each connection has a weight associated with it, which determines the signal between the two neurons. If the network output is accurate, the weight values need not be changed. However, if the output does not match the expected output, an error is calculated as the difference between the expected and actual outputs. This error is used to adapt the network and improve its performance in the next iteration.

Each neuron in the hidden and output layers can be modeled using a perceptron [6], which processes inputs to generate an output. The perceptron algorithm multiplies each input by its weight, adds the results, and applies an activation function to calculate the output. The activation function can be defined as the sign of the sum obtained, where a positive sum produces an output of 1 and a negative sum produces an output of -1. To handle cases where both inputs are zero, a bias input with a value of 1 is added.

The neural network compares its output with the expected output for each input and learns from mistakes by adjusting the weights of the perceptron. The process involves giving different inputs to the perceptron with known answers, guessing the value, calculating the error, adjusting the weight based on the error, and repeating the process. The error, defined as the difference between the expected and actual outputs, is the critical factor that determines how the perceptron weights should be varied. By continuously adapting and improving based on errors, the neural network can learn to accurately predict outputs.

2. Conventional PI control scheme

Conventional controllers play a vital role in process industries and are exceedingly popular for their simple structure and robust performance. They are used to control the plant inputs in order to track the desired setpoint of the plant. Here, the most conventional controller i.e. the PI controller [9] is used for the reference model implementation. The block diagram is shown in Fig. 2. The primary objective of this controller is to calculate an error value as the difference between a measured process variable and a desired set point. This error value is used as the input to the PI controller and the process input (i.e. the inlet flow of the liquid) is used as the output of the controller and the values are limited to the specified range, mentioned in Table I. Then it attempts to minimize the error by continuously adjusting the control input in order to obtain the desired process output from the tank. The Ziegler and Nichols method is used here to tune the controller parameters (i.e. the proportional gain, K_p, and the integral gain, K_i), and the parameters are found as $K_p = 1.3950$, and $K_i = 0.0042$. So, using these values, the controller is tuned to obtain the desired input-output dataset which is used to train the neural network controller of the actual model.

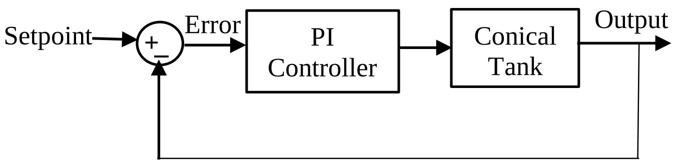

Fig. 3. Block diagram of conventional PI controller

IV. RESULTS & DISCUSSIONS

The architecture of the neural network model is made in Matlab-Simulink software and the simulation is performed using automated training commands available in Matlab library. After the training is completed, a sequence of step inputs is given to the system as the setpoint, and the performance of the conical tank system and the neural network controller is checked for each of the responses discussed below and it is clearly observed that the controller is providing satisfactory results while tracking the setpoint.

1. Servo response

Setpoint is set at 10 cm for the first 1000 seconds, then it is set at 47 cm for the next 2000 seconds and then it is set again at 23 cm for the next 2000 seconds and the output of the neural network model satisfactorily follows the setpoint. The tracking response is shown in Fig. 4(a) and the controller output response is shown in Fig. 4(b). The variation of weights in the hidden layer is shown in Fig. 4(c).

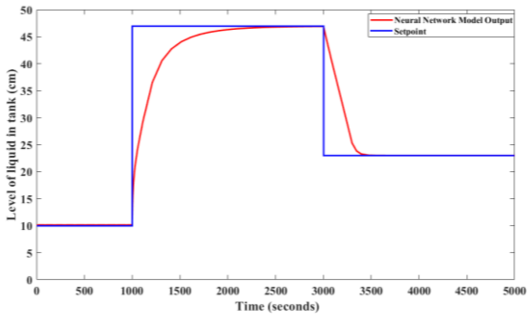

Fig. 4(a). Servo response of process variable

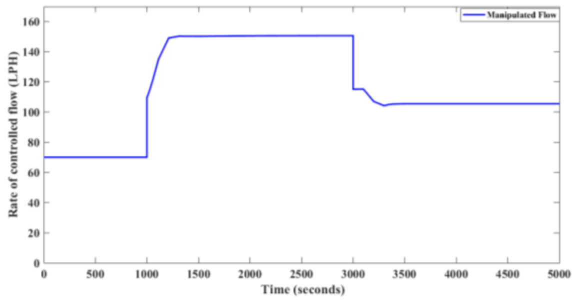

Fig. 4(b). Servo response of controlled flow

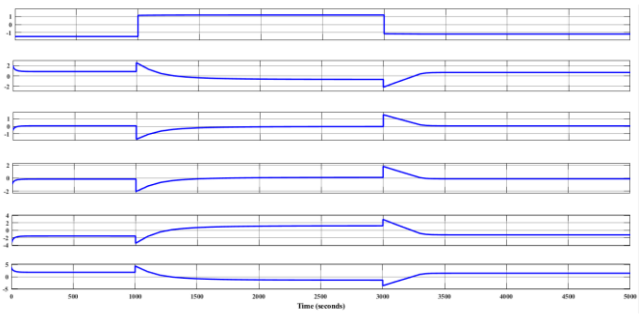

Fig. 4(c). Update of weights in hidden layer in servo response

2. *Regulatory response*

Setpoint is set at 10 cm for the first 100 seconds and then it is set at 27 cm for the total simulation time but during the simulation, at time, t = 2000 sec, a disturbance is added by reducing the flow by 20 LPH for a time duration of 60 seconds and at time, t = 3500 sec, another one disturbance is produced by increasing the flow by 20 LPH again for a time duration of 60 seconds. It is clearly observed that the output of the neural network model satisfactorily handles the disturbance and follows the setpoint. The tracking response is shown in Fig. 5(a) and the controller output response is shown in Fig. 5(b). The variation of weights in the hidden layer is shown in Fig. 5(c).

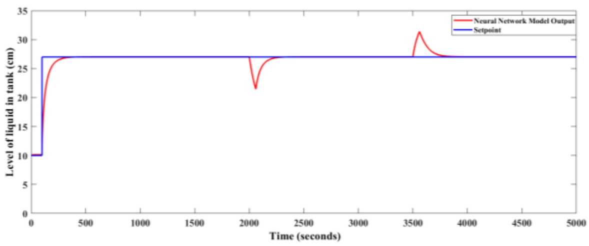

Fig. 5(a). Regulatory response of process variable

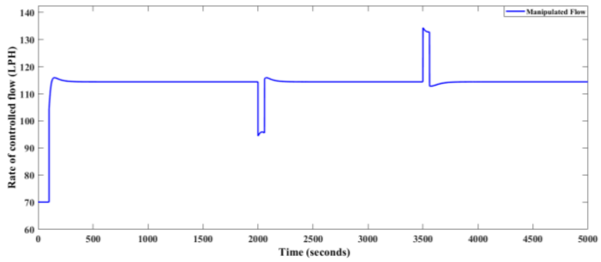

Fig. 5(b). Regulatory response of controlled flow

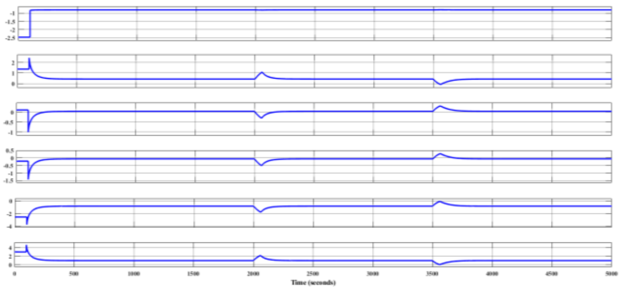

Fig. 5(c). Update of weights in hidden layer in regulatory response

3. *Parametric uncertainty*

Setpoint is set at 27 cm and two pulse disturbances are added at time, t =1500 sec and at time, t = 3500 sec respectively to vary the outlet valve coefficient K_v. The first one is having the amplitude = 4 and pulse width = 100 sec. The second one is having the amplitude = -4 and pulse width = 100 sec. It is clearly observed that the output of the neural network model satisfactorily handles the disturbance and follows the setpoint. The variation of K_v is shown in Fig. 6(a). The tracking response is shown in Fig. 6(b) and the controller output response is shown in Fig. 6(c). The variation of weights in the hidden layer is shown in Fig. 6(d).

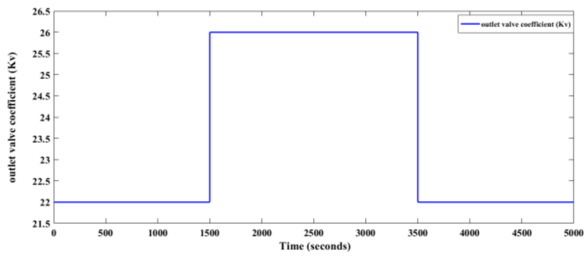

Fig. 6(a) Variation of outlet valve coefficient (Kv)

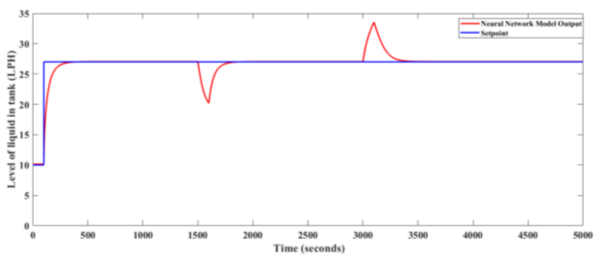

Fig. 6(b). Regulatory response of process variable

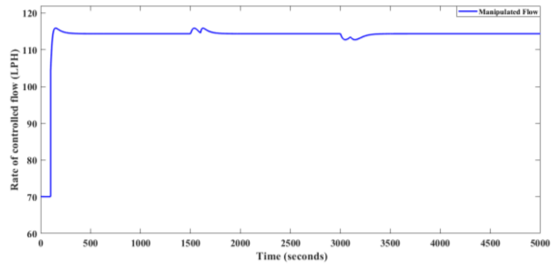

Fig. 6(c). Regulatory response of controlled flow

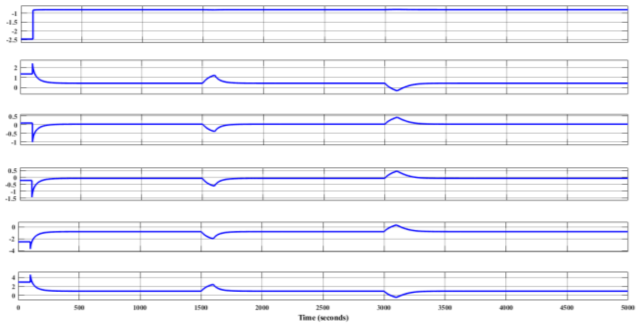

Fig. 6(d). Update of weights in hidden layer in regulatory response

CONCLUSION

This paper presents a case study on the application of an adaptive neural network control scheme to regulate the liquid level in a nonlinear tank-level system. The simulation results demonstrate that the controller achieves the desired performance for both the servo and regulatory response of the conical tank level system. These findings are valuable for the field of process control industries, as they demonstrate the potential for effective control of liquid levels in nonlinear tank systems.

ACKNOWLEDGMENT

We extend our sincerest gratitude to Prof. Atanu Panda of the Department of Electronics and Communication Engineering, IEM Kolkata, for providing us with unwavering support, invaluable insights, and for inspiring within us a strong passion for the subject matter that kept us driven throughout the entirety of this project. We would also like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

REFERENCES

[1] Sanjay Bhadra, Atanu Panda, and Parijat Bhowmick, “A nonlinear model-based control scheme for benchmark industrial processes,” IEEE International Conference on Opto-Electronics and Applied Optics (Optronix), pp. 1-5, 2019.

[2] Sanjay Bhadra, Atanu Panda, Parijat Bhowmick, Shinjinee Goswami, Sayan Podder, Debayan Dutta, and Zeet Chowdhury, “Implementation of Neural Network Based Control Scheme on the Benchmark Conical Tank Level System,” IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1-5, 2019.

[3] R J Rajesh, Preethi R, Parth Mehata, and Jaganatha Pandian B, “Artificial Neural Network Based Inverse Model Control Of A Nonlinear Process,” IEEE International Conference on Computer, Communication and Control (IC4_5270), pp. 1-6, 2015.

[4] N.S. Bhuvaneswari, G. Umab, and T.R. Rangaswamy, “Adaptive and optimal control of a non-linear process using intelligent controllers,” ELSEVIER, Applied Soft Computing 9, pp. 182-190, 2009.

[5] Sudharsana Vijayan, Dr.G Glan Devadhas, Vineed T govind, and Shinu M M, “Soft computation for Conical Tank Level Control,” IEEE International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), pp. 1776-1783, 2017.

[6] K.J. Nidhil Wilfred, S.Sreeraj, B.Vijay, and V.Bagyaveereswaran, “System Identification Using Artificial Neural Network,” IEEE International Conference on Circuits, Power and Computing Technologies (ICCPCT), pp. 1-4, 2015.

[7] C. Febina, and D. Angeline Vijula, “Model based Controller Design using Real Time Neural Network Model and PSO for Conical Tank System,” CEAI, Vol.22, No.3, pp. 13-24, 2020.

[8] Sukanya R. Warier, and Sivanandam Venkatesh, “Design of Controllers based on MPC for a Conical Tank System”, IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM), pp. 309-313, March 30, 31, 2012.

[9] T.Pushpaveni, S.Srinivasulu Raju, N.Archana, and M.Chandana, “Modeling and Controlling of Conical tank system using adaptive controllers and performance comparison with conventional PID,” International Journal of Scientific & Engineering Research, Volume 4, Issue 5, pp. 629-635, May-2013.

[10] Santanu Mallick, and Ujjwal Mondal, “Performance Analysis of a Conical Tank System using Lyapunov based MRAC Technique,” Proceedings of IEEE Region 10 Symposium (TENSYMP), pp. 168-172, 2019.