ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

NEURAL NETWORK PREDICTIVE CONTROLLER FOR TEMPERATURE PROCESS CONTROL SYSTEM

M. Sai Veerraju*

* Professor, Department of Electrical and Electronics Engineering, S.R.K.R Engineering College, Bhimavaram-534204, Andhra Pradesh, India

Abstract. The importance of the temperature control system in industrial processes is growing more and more apparent. Recently, a significant amount of research has been undertaken on temperature control systems that use a variety of control algorithms. Because of their powerful self-learning and parameter altering capabilities, neural networks (NN) have been frequently used to solve highly nonlinear control difficulties in industrial processes. The temperature sensor is responsible for measuring the temperature of the process. The mathematical model of the system is identified based on the input and output data from the process itself. After that, the NN predictive controller is constructed to maintain the process temperature at the predetermined temperature. The constructed controller is evaluated using a variety of different inputs. The performance of the controller is evaluated using the metrics such as integral square error, integral absolute error, and integral time absolute error. The response time of the system is taken into consideration in addition to this. The following goals must be met for the project to be a success and put into action. Measure the temperature of the process using RTD, Design an NN predictive controller to maintain the temperature of the process based on the setpoint, and evaluate the controller using ISE, IAE, and ITAE.

Keywords—PLC, Fire detection, Siren, Sprinkler, Smoke detector, Pump.

INTRODUCTION

Whenever it comes to chemical plant operations, the Continuous Stirred Tank Reactor (CSTR) is a key player in the advancement of technological advancements. CSTR has a wide operating range and nonlinear properties, making it an excellent choice for many applications. Because of increased competition and the need for more production flexibility, CSTR routinely manufactures things that meet a variety of quality criteria [1]. It's challenging to keep up with the rapid advancement of CSTR technology. When the values of a process's variables are constantly changing over time, this is an example of a situation similar to the one described above. Because of this, a process control system must be designed to maintain the CSTR in a stable condition. Using modeling, Control, and system optimization, in general, may be made much more effective. For control system designers to be successful, they must be able to predict how their systems will act in both dynamic and steady-state settings. A mathematical tool such as modeling is used to design an effective process control system, and this is critical for properly managing and optimizing your chemical process. To be used in system design and process control applications, chemical processes must first be reliably predicted before they can be used. When it comes to reading and analyzing physical control parameters, a PID controller is a common choice these days across a wide range of industries. It is practically impossible to achieve the best PID gain since the vast majority of chemical processes are categorized as non-linear systems. PID gain tuning is a frequent industry practice since it has a substantial impact on how the system performs and is therefore important to understand. However, because most of these approaches are based on linear systems, nonlinear systems are unable to profit from any of them. Over the last few years, the application of artificial intelligence to common problems has grown in popularity. To name a few strategies that have been employed successfully, NNs, fuzzy logic, and evolutionary programming are among those that come to mind. Because they are based on the brain's information processing, neuronal networks have gotten a lot of interest in the chemical processing industry. They are a type of NN. Since its effectiveness in nonlinear estimating and suitability for nonlinear systems, Artificial NNs (ANN) have been included in a variety of controller design approaches, including NN estimation and control. Artificial NNs (ANNs) are used in the field of chemical engineering for a variety of tasks, including modeling and process control. Nonlinear models created from input/output data from this industry serve as the foundation for the majority of control system modeling methodologies used in this industry.

A fuzzy controller can be used as an alternative to a regular PID controller in some situations. Because of the realistic portrayal of control expertise, it is simple to grasp the concept. Fuzzy controllers, on the other hand, are used to control industrial process systems, and the rate of error change is similar to that observed in PD controllers. Fuzzy controllers typically have two inputs, which is standard for the industry. When the steady-state error of the control system is removed, it becomes possible to integrate error in the input to the fuzzy controller. Using a fuzzy controller as an input, it is possible to operate a system if the system's errors, error change rates, and integration of errors are all measured and used as inputs. This technique is difficult to apply in practice as a result of the difficulties associated with building a control rules foundation and the difficulties associated with monitoring the incorporation of mistakes. Adding one more control rule to a fuzzy controller will result in a large rise in the number of control rules in the fuzzy controller [2]. When adopting this control technique, the selection and training of NNs have a significant impact on the overall performance of the system. Using this method, the weight of the NN is tweaked regularly to ensure that it reaches the optimal value, even if the network is used for the same job numerous times. On its own, the Lyapunov stability theory cannot govern NNs in nonlinear systems, and hence cannot be used to do so. The radial RBF NN, which can recognize nonlinear systems, is being investigated by researchers as a possible solution to this problem. The persistently stimulating property of the RBF NN can be used to assure the convergence of NN weights to a certain extent [3], as previously stated. The article [4] discusses how to select an oil refining production rectification neural network controller. A rectification process study model is used to gauge the controller's efficiency. Refinement process control parameters are established so that controllers may be identified and evaluated. A numerical analysis shows that the rectification process can be controlled by a NN controller. For the sake of this study, we choose to employ a PID regulator, which is the industry standard. As seen in this example, the use of a NN controller can be beneficial for constructing several target paths. It is possible to customize the NN controller paradigm for computer-controlled correction. In the publication [5] a fuzzy neural network (FNN) controller for brushless drive speed tracking was designed. The extended Kalman filter (EKF) training algorithm is used to train the PI and PD FNN controllers. FNN employs techniques derived from artificial neural networks to determine the parameters of a fuzzy logic system. Each FNN controller contains four layers. Membership functions and weights are adjusted based on EKF training capabilities. The primary purpose of this project is to control parallel PI FNN and PD FNN using the EKF training algorithm. A comparison of parallel PI and PD FNN controllers shows that they outperform traditional PID controls. The hardware design is implemented using MATLAB and a dSPACE DS1104 DSP. The results show that the FNN controller has superior learning capabilities and robust responsiveness in real-time for a variety of operational circumstances.

In the paper [6] the PID controller coefficients of a non-linear exoskeleton control system for the lower limbs may be modified using analog neural networks (NN) with a radial basis function (RBF) and self-tuning. An analog neuron in a network controller, which is capable of learning and adapting over time, can be used to manage nonlinearities with uncertain parameters and external disturbances precisely and accurately. We may use the PID controller's NN to adjust for errors produced by parameter variation and uncertainty as the exoskeleton's lower extremities move. The efficiency of the proposed control algorithm is demonstrated via simulation in the Matlab/Simulink environment. Personalized short-distance travel on a Segway. The Segway, which is presently on the market, can be steered by just shifting one's weight back and forth. Because of the services this machine delivers, it is vital to analyze its performance and devise effective techniques for controlling it. Traditional PID controllers are the most well-known and commonly utilized controllers in industrial settings. Three functions govern the input signals before they reach the plant or process unit. In the study [7], the PID controller for the Segway's cart position and handlebar angle is being replaced with another control strategy based on NN control. MATLAB was used to mimic the findings of this investigation, which revealed that the NN controller used to operate and recognize the Segway performed better than the standard PID controller. Pressure and water level are related in a deaerator at a steam power plant. Controlling pressure and level in a deaerator with a conventional PID controller is quite difficult utilizing the most commonly used control approach. The research [8] proposes a neural network-based control strategy for emulating a single PID controller loop. Furthermore, the PID controller's simple design, as well as its high durability and robustness, makes it an excellent choice for a variety of applications. As a result, the NN can learn for itself and control nonlinear processes. By using the latest tech, the NN and the PID controller may work in tandem. As a result, overshoot and the time required for transient processes to settle in the control system will be considerably decreased. As a result, the NN will be effective for controlling the deaerator's pressure and water levels.

METHODOLOGY

To control the temperature of the CSTR the NN controller is designed. The CSTR and NN controller architecture, as well as working, are detailed in this chapter.

A. CSTR

Continuous stirred tanks (CSTR) are the most straightforward type of continuous reactor when it comes to design. Half-pipe coil reactor with jacket, such as the CSTR seen in figure 1. Material can enter or escape the CSTR at any point in time, and the conditions inside the reactor do not change with time. As open systems, CSTRs allow materials to enter and exit at any moment without causing damage to the system [9]. In the reactor, the addition of reactants and removal of products occurs continuously.

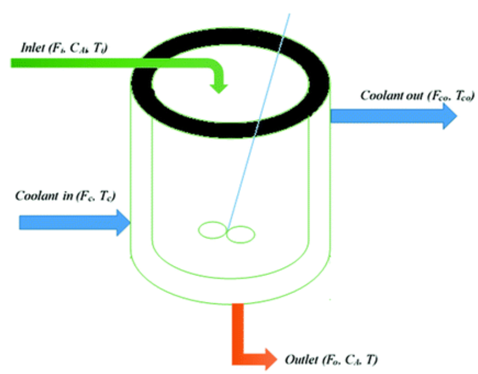

FIGURE 1. *CSTR* [10]

Because of the well-mixed nature of the components, consistent thermal and density metrics may be found throughout CSTRs. Additionally, the reactor's exhaust stream contains identical circumstances. The image depicts a CSTR that does not have a stirring system installed. When a reaction is taking an excessive amount of time, systems involving numerous CSTRs are used. A large number of CSTRs can be used to agitate a mixture of immiscible or viscous liquids that must be stirred regularly. Reactants can be continuously introduced into the reactor using ports located at the top of the reactor. The stirring mechanism of the unit ensures that the contents of the tank are fully mixed. Products are being disposed of regularly. Tanks and stirring systems are employed in CSTRs, which have a constant volume because of their design. There are feed and exit pipelines that are used to bring in reactants and remove them from the system. It is possible to glimpse the interior of a CSTR in the photograph below, which shows a part of one of the sides that has been removed. Agitators, which are another term for stirring blades, are used to combine the reactants in a reaction. The agitators depicted in the photographs below could all be located within a CSTR, according to the authors. A CSTR can also be used as a loop reactor if a hot, pressurized fluid is introduced into the system to encourage stirring. The absence of an agitator simplifies maintenance while simultaneously increasing heat and mass transfer rates. A cell culture reactor is depicted in the illustration below. It is only a small number of cells in the fibrous-bed basket at the beginning of the procedure. The reactor is supplied with a steady supply of nutrient-rich media, and the materials created are harvested from the reactor. As the cells expand, the byproducts of the reaction are continually removed from the reactor. Through the use of a pitched-blade impeller, the reactants in the reactor below are continuously mixed.

B. NEURAL NETWORK CONTROLLER

The considerable literature on the predictive controller, which has piqued the interest of both academics and industry, provides an overview of the controller [11]. The predictive controller is distinguished by the computation of future control actions based on model output values that have been forecasted in advance. Presented in this section are predictive control concepts based on NPC, which make use of the standard optimization functions and control rules, and which are applied to conventional predictive controllers and the working of the NN controller is depicted in figure 2. The optimization functions listed below are examples: The optimization function, which is frequently denoted by the index J, represents the function that the control action is attempting to minimize through the control action. As an illustration, let us consider the most straightforward example of an optimization function: the difference between the output of a plant and its desired value.

$$J = y_{ref}(k) - y(k) = e(k)$$

The output of the plant is denoted by y(k), and the reaction we desire is denoted by $y_{ref}(k)$. The error in estimating is represented by the symbol $e(k)$. The integer k represents the amount of time spent sampling. In optimization, the square error is one of the most commonly used functions, and it can be stated as follows:

$$J = [y_{ref}(k) - y(k)]^2 = [e(k)]^2$$

An optimization index may be represented by a more complex function. Predictive controllers using models that are capable of forecasting N steps ahead can achieve satisfactory results with straightforward use of the square error technique. In this circumstance, an optimization function can be created by using a vector of N-predicted mistakes as input. Its goal is to optimize the complete trajectory of future control operations across an N-step time horizon, which is some steps.

$$J = \sum_{j=1}^{N} \quad [y_{ref}(k+j) - \hat{y}(k+j)]^2 = \sum_{j=1}^{N} \quad [e(k+j)]^2$$

More complex optimization functions can take into account the amount of work required for control. The optimization index J can be represented in the following ways for GPC (Generalized Predictive Control) applications:

$$J = \sum_{j=N_1}^{N_y} \quad [y_{ref}(k+j) - \hat{y}(k+j)]^2 + \sum_{j=1}^{N_u} \quad \propto (j). [\Delta u(k+j)]^2$$

When dealing with a control problem of this nature, it is critical to identify locations where the first-order differential is equal to zero. For non-linear systems, predictive control is the ideal approach to controlling the process flow. A neural model for the nonlinear plant, which is capable of both forward and backward predictions, is used to generate multi-step forecasts and backward predictions. When using predictive control, each sample point represents an opportunity to reduce the cost function. Using predictive control, J.

$$J\big(t, U(k)\big) = \sum_{t=N_1}^{N_2} \quad [r(k+i) - \hat{y}(k+i)]^2 + \sum_{i=1}^{N_u} \quad \rho[\Delta u(k+i-1)]^2$$

Concerning the N_u future controls,

$$U(k) = [u(k) \ldots \ldots \ldots . . u(k+N_u-1)]^T$$

and subject to constraints:

$$N_u \le i \le (N_2 - n_k)$$

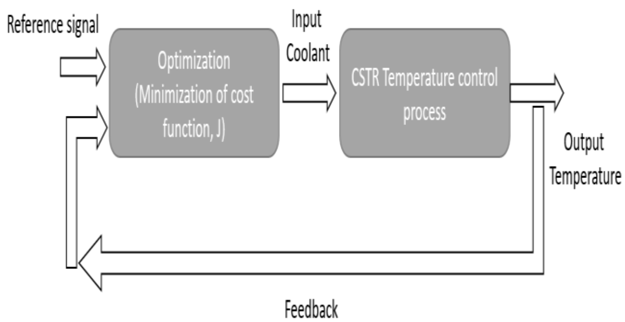

FIGURE 2. *NN CONTROLLER*

With the help of the predictive control approach and the widely used NARX model, it is possible to construct an optimal control sequence for a nonlinear plant (NNMPC) [12]. The prediction horizons N_1 and N_2 are the lowest and maximum prediction horizons, respectively; N_u is the control horizon, and the control penalty factor is the control increment; and Nu is the control horizon, respectively. The needed reference plant output is denoted by $r(k+i)$, whereas the expected output of the NN model is denoted by $y(k+i)$, which denotes the predicted model output. Strategy for shortening the time horizon: It solves the previously specified optimization problem for a finite future at present and implements the first optimal control input as current control input by employing the predictive control approach. It is necessary to utilize the following formula to minimize the cost function J for a given set of control parameters (N_1, N_2, N_u), which is written as $[u(k) \ldots \ldots \ldots .. u(k+N_u-1)]$. These control factors are responsible for determining the predictive control performance. The prediction horizon, i.e., the number of times the plant response can be recursively anticipated in the future, is specified by N_1 and N_2, which are commonly set to one and two, respectively, depending on the application.

The iterative minimization of the optimization problem of the criterion, J, is achieved in NNMPC. Iterative search strategies, which are similar to NN training procedures, are employed to locate the smallest possible number of results.

$$\theta^{i+1} = \theta^i + \mu^i . d^i$$

The formula specifies the number of iterations (a number I), the direction of the search (d^i)), and the step size (I) for each step. Based on the search direction and step size that algorithms employ, algorithms can be categorized in a variety of ways. It is the Levenberg–Marquardt (LM) approach that is being used in this study, which is a Newton-based method. The following is the search direction of the LM algorithm:

$$H[U^i(t)] = \frac{\partial^2 J(t, \widetilde{U}(t)}{\partial U(t)^2}|_{U(t)=U^t(t)}$$

$$= \frac{\partial}{\partial U(t)}\left[\frac{\partial \hat{Y}(t)}{\partial U(t)} E(t)\right] + 2\rho \frac{\partial \widetilde{U}^T(t)}{\partial U(t)} \frac{\partial \widetilde{U}(t)}{\partial U(t)}|_{U(t)=U^t(t)}$$

Making Predictions about Time Series Using NNs Our NN model is intended to predict the output of a plant throughout a given period. To estimate the plant output series yN a series of control signals u, as well as historical data, must be provided. With the present control signal u_t and the current plant output, the network is trained to predict the plant output y_{t+1}. It will be implemented through the use of a NN. Making Predictions about Time Series Using NNs Our NN model is intended to predict the output of a plant throughout a given period. Assuming that u and previous data are available, it is desirable to estimate the plant output series yN based on the available data and assumptions. It is trained to predict the plant output y_{t+1} from the current control signal u_t and the current plant output y_t using the current control signal ut and current plant output y_t It will be implemented through the use of a NN.

$$\hat{y}_{t+1} = f(u_t, y_t)$$

For this forecast to be accurate, the information included in y_t must be sufficient, as previously indicated. It is assumed that y_t is a multivariate function. As errors accumulate, using this technique can result in a rapidly increasing divergence as the number of errors increases. This necessitates the need for the model to be exceedingly exact. In the process of getting closer to accurate representations of real-world plants, the importance of errors decreases in importance. This method is beneficial in lowering the accumulation of errors since it decreases the number of steps that must be completed within a certain period. A NN that has been trained to predict one step forward will be used to model the plant. When this model is acquired, the process is referred to as System Identification.

The feed-forward network with a sigmoidal activation function was selected after a series of trials with several multilayer perceptron topologies. The buried layer contained seven neurons that were responsible for the lowest level of error. As a result, it was determined to be the optimal architecture for an ANN. Each of the four neurons in the input layer, seven neurons in the hidden layer, and one neuron in the output layer of the ANN used in this example is connected to the other neurons. Backpropagation is used to train the CSTR model, which is then used to test it. Before training, the algorithm's weights are established using random integers generated by the computer. As a precaution, the weights for each training set are changed to ensure that no mistakes are made. The instruction is terminated as soon as the overall quantity of error is deemed acceptable by the instructors. The validation of a model is the final step in the construction of a model. During the validation process, the model's performance is evaluated by comparing it to both trained and test data. It was decided to employ the Levenberg-Marquardt technique to train the network on both the input and the target simultaneously.

B. SYSTEM PROCEDURE

The dynamic properties of the CSTR should be well-represented in the data used to train the network so that the network can be effectively trained. It was decided to utilize the typical CSTR model to sample the input and output data at 0.02 sampling instants, and the sampled data was then used to train the network until it was completely trained. The following diagram depicts the breakdown of the input sequence training and output sequence training processes. The feed-forward network with a sigmoidal activation function was selected after a series of trials with several multilayer perceptron topologies. The buried layer contained seven neurons that were responsible for the lowest level of error. As a result, it was determined to be the optimal architecture for an ANN. Each of the four neurons in the input layer, seven neurons in the hidden layer, and one neuron in the output layer of the ANN used in this example is connected to the other neurons. Backpropagation is used to train the CSTR model, which is then used to test it. Before training, the algorithm's weights are established using random integers generated by the computer. As a precaution, the weights for each training set are changed to ensure that no mistakes are made. The instruction is terminated as soon as the overall quantity of error is deemed acceptable by the instructors. The validation of a model is the final step in the construction of a model. During the validation process, the model's performance is evaluated by comparing it to both trained and test data. It was decided to employ the Levenberg-Marquardt technique to train the network on both the input and the target simultaneously. How the suggested system will function in practice is depicted in figure 3, respectively.

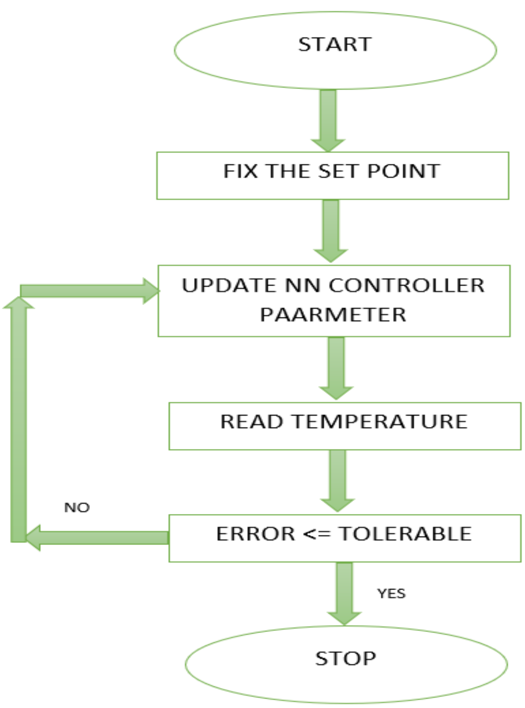

FIGURE 3. *FLOWCHART OF THE PROPOSED SYSTEM*

RESULT AND DISCUSSION

The proposed system consists of three main phases namely mathematical model of the system, designing the NN controller, and evaluating the designed system. For identifying the process mathematical model, the first principal method is used. The NN controller is designed by giving appropriate parameters like a horizon, time, etc., next the model is validated using the error function such as IAE, ISE, ITAE, ITSE. The RTD is used to measure the temperature of the process. This sensor is used as a feedback element.

The Simulink diagram is drawn as shown in figure 4. the setpoint is given using a random generator. The NN controller block is used in the place o controller. For feedback here, unity type is used. next, the mathematical model of the CSTR is drawn using various numeric and mathematical blocks. To validate the model the errors are used. Here the ISE, IAE, ITAE, ITSE is employed. For visualizing and comparison of output the XY graph is dragged and placed

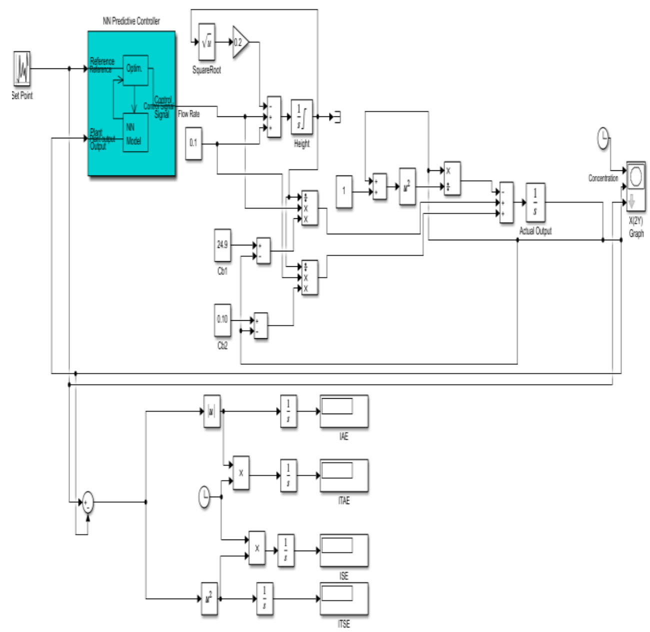

FIGURE 4. *SIMULINK DIAGRAM FOR NN CONTROLLER*

First, the setpoint is given as 22.1 using the random number generator. The output obtained is given in figure 5. The setpoint is shown in the figure by red color and the actual output is given in the figure by blue color. From the figure, it is clear that the controller is designed very well. The undershoot does not happen in the controller. Then the rise time, peak time, settling time, and steady-state error are very less. The error is displayed in figure 6. The error obtained through the process is shown in the below figure. The ISE obtained in this work is 0.01667. The IAE obtained is 0.2378, ITAE obtained is 0.6147, ITSE obtained is 0.0132. All the errors obtained are very few.

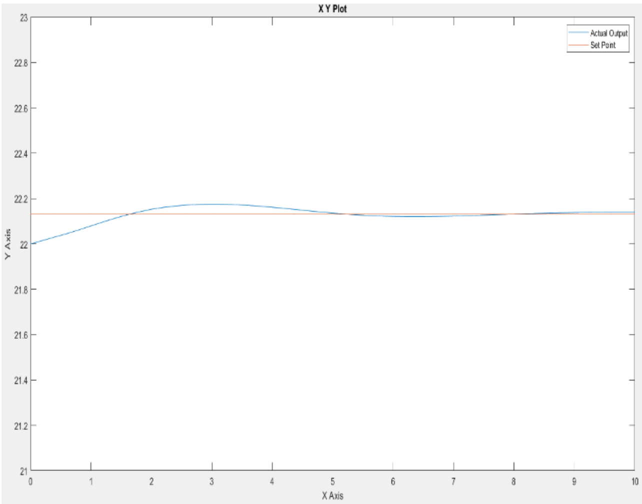

FIGURE 5. *MATLAB SIMULATION WHEN THE SETPOINT IS GIVEN AS 21.5*

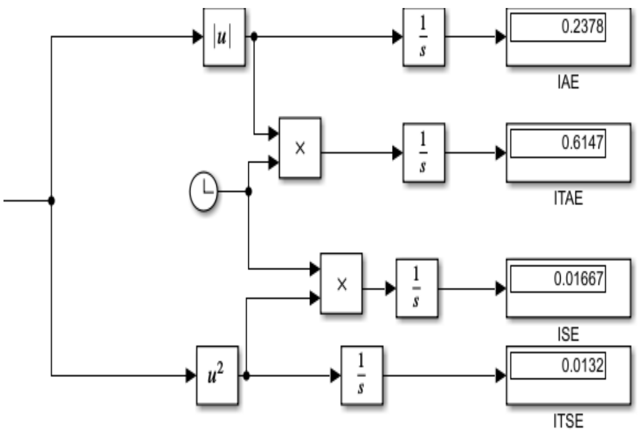

FIGURE 6. *ERROR OBTAINED WHEN THE SETPOINT IS GIVEN AS 21.5*

CONCLUSION

The NN controller is designed very well. The process taken for this research is CSTR. The mathematical model is designed using numeric and mathematical blocks. The setpoint is given based on the requirement. Then the model is run and validated using the error metrics like ISE, IAE, ITAE, and ITSE. Then the time response also validates using rise time, settling time, steady-state, peak time, etc. The proposed methodology has very limited cons and constraints. The system set point is fixed and monitored through MATLAB. To control and monitor the process through wireless is not possible. In the future, IoT is planned to implement. The usage of implementing IoT in the process helps the operator to fix and change the set point through wireless. The process is also monitored through IoT. For implementing the IoT in MATLAB, the ThingSpeak is used.

REFERENCES

1. Ozkan L, Kothare MV, Georgakis C, "Control of a solution copolymerization reactor using multi-model predictive control". *Chemical Engineering Science, vol.* 58, pp. 1207–21, 2003.
2. Saptarshi Das, Indranil Pan, Shantanu Das, "A novel fractional-order fuzzy PID controller and its optimal time-domain tuning based on integral performance indices". *Engineering Applications of Artificial Intelligence*, 2010.
3. J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks". *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.
4. Bukhtoyarov, Vladimir Viktorovich et al. "Neural network controller identification for refining process." *Journal of Physics: Conference Series*, vol. 1399, 2019.
5. L. D. Patil and S. U. Shinde, "PI and PD Fuzzy Neural Network Controller Basedon Extended Kalman Filter for Brushless Drives," *2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 1-7, 2018.
6. M. P. Belov, D. D. Truong and P. van Tuan, "Self-Tuning PID Controller Using a Neural Network for Nonlinear Exoskeleton System," *2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT)*, pp. 6-9, 2021
7. A. A. Ahmed and A. F. Saleh Alshandoli, "On replacing a PID controller with Neural Network controller for Segway," *2020 International Conference on Electrical Engineering (ICEE)*, pp. 1-4, 2020.
8. E. A. Muravyova and A. O. Yurasov, "A Neural Network-Based Control System Using PID Controller to Control the Deaerator," *2020 International Russian Automation Conference (RusAutoCon)*, pp. 73-77, 2020.
9. Deng, Xiaogang & Tian, Xuemin, "Multivariate Statistical Process Monitoring Using Multi-Scale Kernel Principal Component Analysis", *IFAC Proceedings*, vol. 6., pp. 108-113, 2006
10. Parvesh Saini, Rajesh Kumar, Priyanka Sharma, Nalini Rajput, "Design and Comparative Analysis of Controllers for Continuous Stirred Tank Reactor (CSTR)", *In: Singh R., Choudhury S. (eds) Proceeding of International Conference on Intelligent Communication, Control and Devices. Advances in Intelligent Systems and Computing, Springer*, vol 479, 2017
11. Clarke, D.W., C.Mohtadi and P.S.Tuffs, "Generalised Predictive Control-part 1", *The basic algorithm. Automatica*, vol. 23, No. 2, pp. 137-148, 1987
12. H. Ted Su and Tariq Samad, "Neuro-Control Design: Optimization Aspects", *Neural Systems for Control*, pp. 259-288, 1997