Neural Network Predictive Controller for Pressure Control

ZAZILAH MAY[1], MUHAMMAD HANIF AMARAN[2]
Department of Electrical and Electronics Engineering
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, Tronoh, Perak
MALAYSIA
zazilah@petronas.com.my[1], anep88@gmail.com[2]

Abstract: - Proportional-integral-derivative (PID) controllers provide the simplest, robust and effective solutions to most of the control engineering applications. This project focused on neural network model development to fine-tune PID Controller based on a first-order with dead time plant model. The algorithm was based on Ziegler Nichols Process Reaction Curve. A Neural Network (NN) Model was built to predict the tuning parameters for the PID controllers of the plant model. The development of Neural Network Predictive Controller (NNPC) yield better performance compared to conventional PID-type controller. Future work suggests allocating disturbances and time delay in the design process for a real plant system.

Key-Words: - PID Controller, Neural Network Predictive Controller, Ziegler Nichols, Parameters Tuning

1 Introduction

"PID control" is the method of feedback control that uses the PID controller to adjust some process variables at the set point automatically. Fig. 1 illustrates the basic structure of conventional feedback control systems using a block diagram representation. In this figure, the process is the object to be controlled. The purpose of control is to make the process variable *y* follow the set-point value *r*. To achieve this purpose, the manipulated variable *u* is changed at the command of the controller.

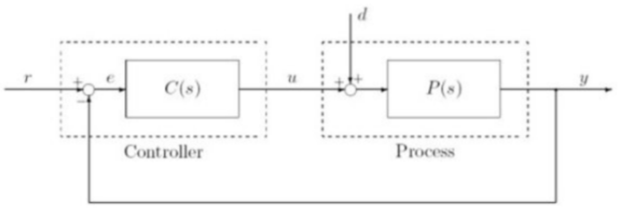

Fig. 1: Conventional feedback control system

The Proportional-integral-derivative (PID) controllers provide the simplest, robust and effective solutions to most of the control engineering applications. It is reported that 95% of the controller system in the process application utilized PID type [1]. In this controller, there are three adjustable parameters, namely the proportional term (K_p), integral term (K_i) and derivative term (K_d). These parameters are necessary to be adjusted to appropriate values to maximize the system performance. The transfer function of PID controller is

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

where K_p is the proportional term, K_i is the integral terms and the K_d is the derivative term.

Shortages of tuning rules and lack of understanding on the part of the users on tuning procedures limit the capability of most tuning methods. Several applicable techniques of tuning PID Controller such as computational methods, intelligent systems, genetic algorithm, fuzzy systems and neural network were discussed, followed by the technique proposed in this project. To increase the performance of PID controllers, Giri *et al*. had developed a computational intelligence (CI) method based on Genetic Algorithm (GA) [2]. The PID tuning was implemented on a closed-loop real time industrial process. This method is simple, involved low computational cost and gave good performance.

Zulfatman *et al*. developed a self-tuning fuzzy PID controller to improve the performance of the electro-hydraulic actuator [3]. Appropriately selecting fuzzy rules in tuning the parameters K_p, K_i and K_d of the PID controller improved the performance of the hydraulic system significantly compared to conventional PID controller.

Neural Network (NN) has become tremendously popular in the control application due to its ability in adaptive learning and approximating function. The type of NN most commonly used is the feed forward multilayer NN, where no information is fed back during operation. There is however a feedback information available during training. The Back Propagation (BP) algorithm is perhaps the most popular and widely used learning algorithm in the feed forward multilayer NN.

Rezazadeh *et al.* designed an NN predictive controller to control the voltage of at the presence of fluctuations of the Proton Exchange Membrane Fuel Cell (PEMFC) control system [4]. The identification approach was used, based on the single layer feed forward neural network with Levenberg-Marquardt training algorithm. Simulation results indicated that the performance of NN predictive controller was better than PID with higher accuracy and speed of convergence. Results showed that this controller can reduce the effect of noise as an adaptive filter.

Anna *et al.* discussed the NN predictive controller that uses an NN model of a nonlinear plant to predict future plant performance [5]. In the paper, simulation of the NN based predictive control of the continuous stirred tank reactor was presented. The simulation results showed that the NN predictive controller produced better performance compared to fuzzy and PID controller.

Ching *et al.* proposed a robust PID controller tuning method for parametric uncertainty systems using fuzzy neural networks (FNNs) [6]. This robust controller is based on robust gain margin (GM) and phase margin (PM) specifications that satisfy user requirements. The FNN system is used to identify the relation between the PID controller parameters and robust GM and PM. The trained FNN system was used to determine the parameters of the PID controllers in order to satisfy robust GM/PM specifications that guarantee robustness and performance. Simulation results are shown to illustrate the effectiveness of the robust controller scheme.

Akesson *et al*. applied an NN controller to the optimal model predictive control of constrained nonlinear systems [7]. The NN controller was designed by minimizing a model predictive control (MPC) type cost function off-line for a set of training data. Results proved that the NN model predictive controller can be trained to achieve near-optimal control performance using both centralized and decentralized controller structures.

Frahani *et al*. dealt with control of a single link flexible joint Robot [8]. An NN based predictive controller using Multi Layer Perceptron (MLP) was designed to govern the dynamics of the proposed Robot. Simulation results showed that this technique performs better in case of mean square error, the percent overshoot and the settling time.

The objective of this project was to focus on NN development to fine-tune PID Controller based on a first-order with dead time plant model. Started with the algorithm based on Ziegler Nichols Process Reaction Curve, an NN Model was then built to predict the tuning parameters for the PID controllers of the plant model. This then lead to the development of NN Predictive Controller to further improve the plant performance.

2 Material and Methods

Fig. 2 illustrates the flow chart of processes conducted throughout the project.

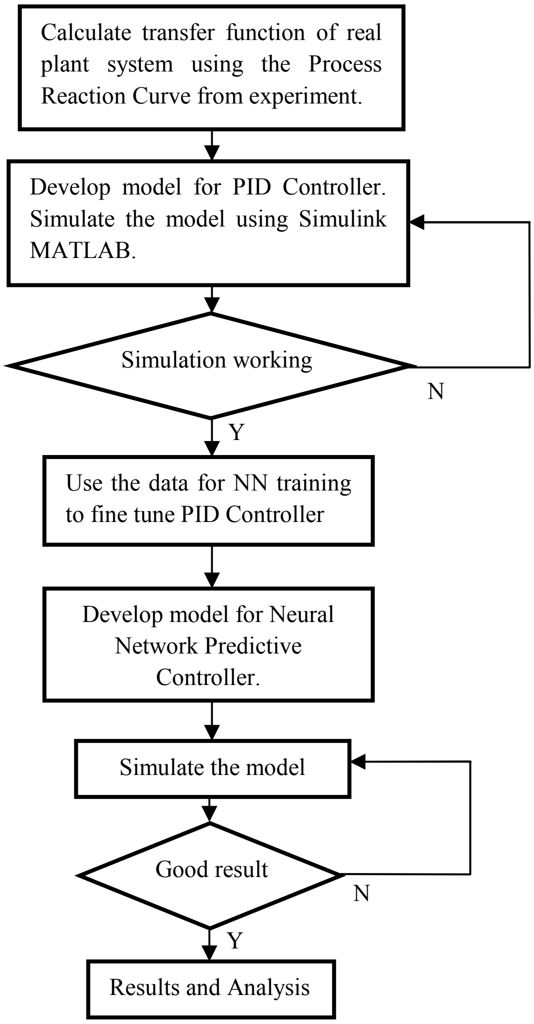

Fig. 2: Project Flow Chart

2.1 Process Reaction Curve

The process reaction curve is the most commonly used method for identifying dynamic models. It is simple and provides adequate models for many applications. The process reaction curve method involves four actions. Firstly, the process is allowed to reach steady state. Secondly, a single step change is introduced in the input variable. Input and output response data is then collected until the process reaches steady state again. The final stage is to perform calculations on the graphical process reaction curve to acquire the transfer function of a particular plant system. For this project, the process reaction curve is is restricted to the first-order with dead time plant model. The form of the model expressed in Equation (2), with X(s) denoting the input and Y(s) denoting the output.

$$\frac{Y(s)}{X(s)} = \frac{K_p\, e^{-\theta s}}{\tau s+1} \qquad (2)$$

2.2 Neural Network
Neural network are usually used for fitting a function, recognizing patterns, and clustering of data. In this project, the neural network was used for the clustering of data which means grouping of data based on similarities. Neural Network is useful when the data is very complex and the design is too impractical to be implemented manually.

2.3 Neural Network Structure
Neural networks are models of biological neural structures. Fig. 3 illustrates a model neuron where neuron consists of multiple inputs and a single output. Every input is modified by a weight, which multiplies with the input value. The neuron will combine these weighted inputs with reference to a threshold value and activation function. The output is determined by these values.

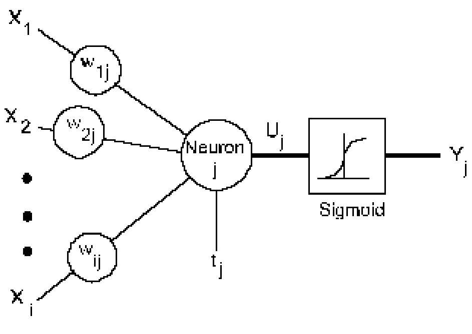

Fig. 3: A Model Neuron

2.4 Neural Network Training
In the process of NN training, the error, which is the difference between the desired response and the actual response, was calculated. The error was then propagated backward through the network. At each neuron in the network, the error was used to adjust the weights and threshold values of the neuron. This was done to ensure that in the next round, the error would be reduced for the same inputs. Fig. 4 illustrates the basic concept of neuron weight adjustment.

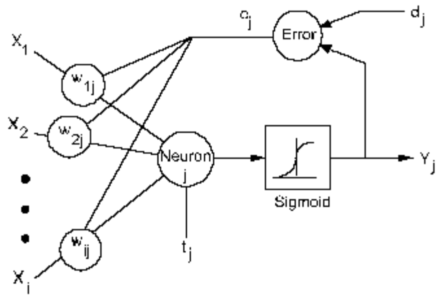

Fig. 4: Neuron Weight Adjustment

Backpropagation is the corrective procedure to reduce the error. It is applied repeatedly for each set of inputs and for its resultant outputs. This procedure continues on as errors in the responses exceed a specified level or until there are no measurable errors. At this point, the neural network has learned the training material.

2.5 Neural Network Predictive Controller
The neural network predictive controller as illustrated in Fig. 5 implements neural network to predict future plant performance. The controller will calculate the input that will optimize plant performance in a particular time. Firstly, the neural network plant model is established. This is done by training the plant using neural network to represent forward dynamics of the plant. The error between the neural network output and the plant output is set as input. Then, the neural network plant model is used by the controller to predict future performance of the plant.

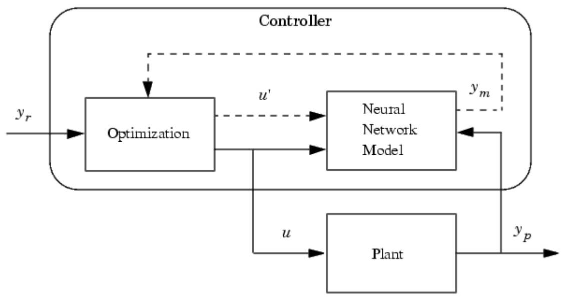

Fig. 5: Neural Network Predictive Controller

3 Results and Discussions
The result from the laboratory experiment for the Simple PID Pressure Control was obtained. The Process Reaction Curves of the experiment for two

different valve openings, 20–40% and 20–50% were illustrated in Fig. 6 and Fig. 7, respectively.

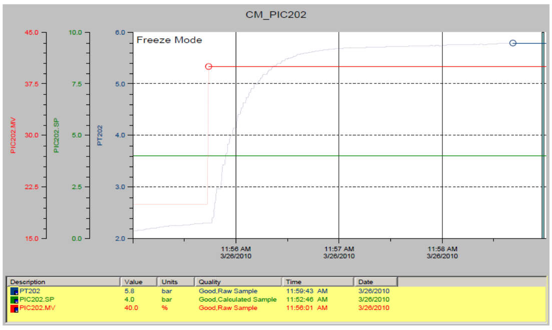

Fig. 6: Process Reaction Curve (Valve Opening 20 – 40%)

From the Process Reaction Curve with Valve Opening from 20 – 40% in Fig. 6, the parameters K_p, τ and θ, were calculated based on the second method of Ziegler Nichols Process Reaction Curve.

$$K_p = \frac{\text{magnitude of steady state change in output}}{\text{magnitude of change in input}} \quad (3)$$

$$= \frac{5.8 - 2.3}{40 - 20} = 0.175$$

$$\tau = \frac{\text{magnitude of steady state change in output}}{\text{maximum slope of the output vs time plot}} \quad (4)$$

$$= 1.5\,(t_{63\%} - t_{28\%}) = 1.5\,(11.875) = 17.8 \text{ sec}$$

The intercept of maximum slope with initial value, θ

$$= t_{63\%} - \tau = 20.625 - 17.8 = 2.8 \text{ sec} \quad (5)$$

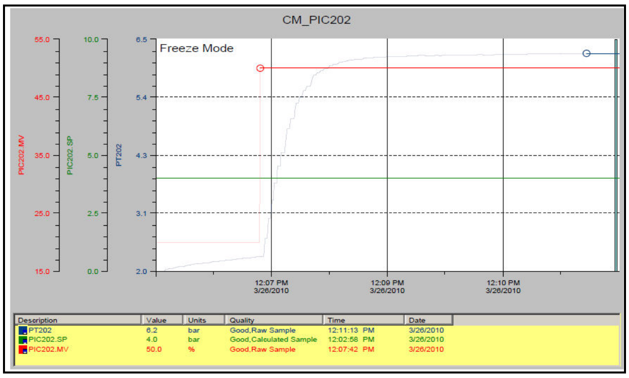

Fig. 7: Process Reaction Curve (Valve Opening 20–50%)

From the Process Reaction Curve with Valve Opening from 20 – 50% in Fig. 7, similar formulas described in Equation (3) to Equation (5) were used to calculate the three parameters

$$K_p = \frac{6.2 - 2.26}{50 - 20} = 0.13 \quad (6)$$

$$\tau = 1.5\,(t_{63\%} - t_{28\%}) = 1.5\,(15) = 22.5 \text{ sec} \quad (7)$$

$$\theta = t_{63\%} - \tau = 27.5 - 22.5 = 5 \text{ sec} \quad (8)$$

From all the values of parameters obtained, the average value was then calculated to obtain more accurate values of the following parameters:

$$K_p = (0.175 + 0.13) / 2 = 0.153 \quad (9)$$

$$\tau = (17.8 + 22.5) / 2 = 20.15 \quad (10)$$

$$\theta = (2.8 + 5) / 2 = 3.9 \quad (11)$$

Substituting the values of K_p, τ and θ obtained in Equation (9), (10) and (11) respectively into Equation (2) yields the transfer function of the plant shown in Equation (12).

$$\frac{Y(s)}{X(s)} = \frac{0.153\,e^{-3.9s}}{20.15s+1} \quad (12)$$

To verify the transfer function, simulations on five different step inputs were conducted. The graphs obtained were about the same for each different step inputs. The transfer function was verified to be functional and can be used as the plant for the neural network training.

The plant model was simulated using Simulink, as shown in Fig. 8, to obtain the three parameter values of the P, PI and PID controllers, namely K_p, T_i and T_d, in Table 1 , PI and PID controllers were initially tuned using the Ziegler-Nichols Open Loop method based on the Process Reaction Curve. The simulation results for P, PI and PID controllers are shown in Fig. 9, Fig. 10 and Fig. 11, respectively.

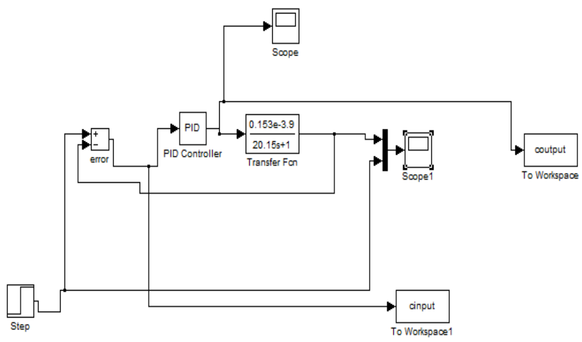

Fig. 8: Plant Model in Simulink

Table 1: Results For Ziegler-Nichols Open Loop Tuning Correlations

Controller	**K_p**	**T_i**	**T_d**
P-only	33.8	-	-
PI	30.4	12.87	-
PID	40.5	7.8	1.95

ISBN: 978-1-61804-049-7

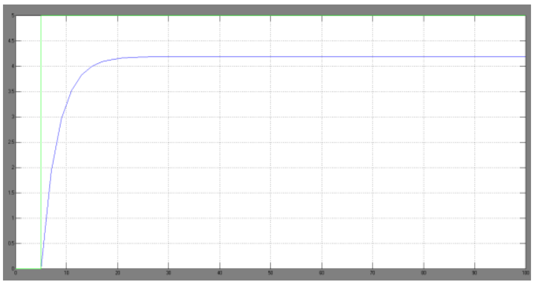

Fig. 9: Simulation result for P Controller

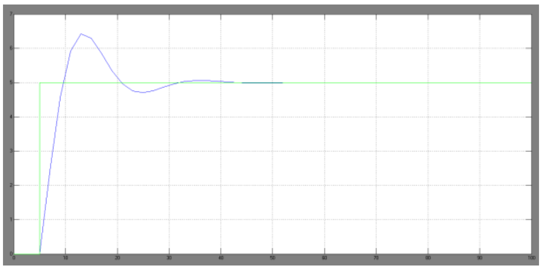

Fig. 10: Simulation result for PI Controller

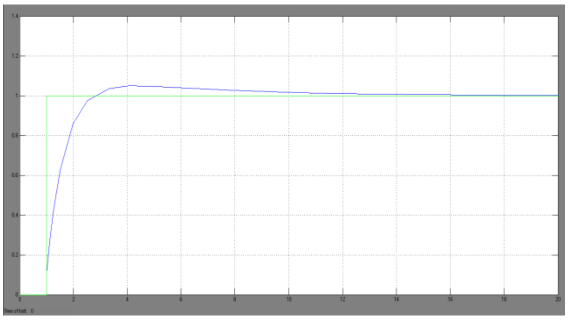

Fig. 11: Simulation result for PID Controller

Based on the simulation results of P, PI and PID controllers, the response for P controller was not acceptable as it did not reach the zero offset. The best response for this plant was given by the PI controller that gave the lowest settling time and rise time of 9.86 seconds and 1.5 seconds respectively as compared to the PID controller which had the settling time of 14.28 seconds and rise time of 1.8 seconds.

Since the PI controller showed the best controller performance, the input values obtained from the response were used for the Neural Network model illustrated in Fig. 12.

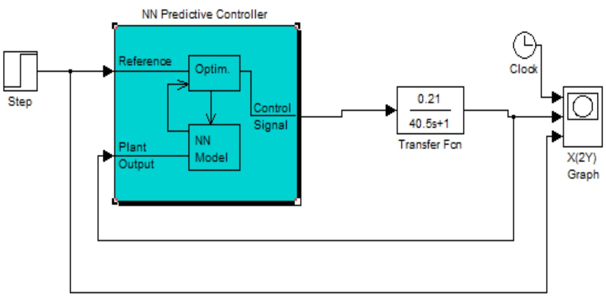

Fig. 12: Plant model using NN Predictive Controller

The input values were the range of error signal while the targets consisted of a set of constant proportional and integral values. The initial values used for training were taken from the initial tuning of the PI Controller. For both K_p and T_i training, the data are randomly divided into three sets which are training, testing and validation. Fig. 13 illustrated the NN performance plot for K_p of PI Controller.

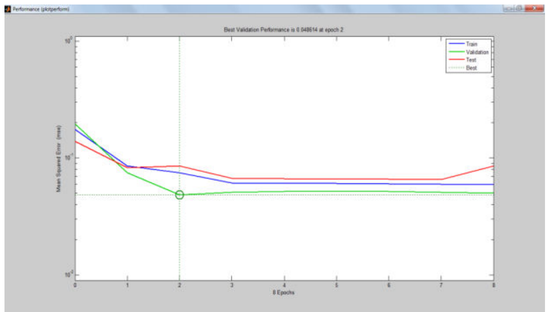

Fig. 13: Neural Network Performance Plot for K_p of PI Controller

The mean squared error (MSE) for testing, training and validation of K_p value were 0.065, 0.061 and 0.04 respectively. The MSE for testing set became 0.1 at the end of the training. This network is acceptable as it shows a small MSE values for the training of Neural Network. However, since the testing set MSE value became slightly higher at the end of training, it showed that the network can be further improved to obtain a better result. Based on the training, the output of the network obtained was plotted as shown in Figure 14. Based on the graph, the output for Kp is around 30.07.

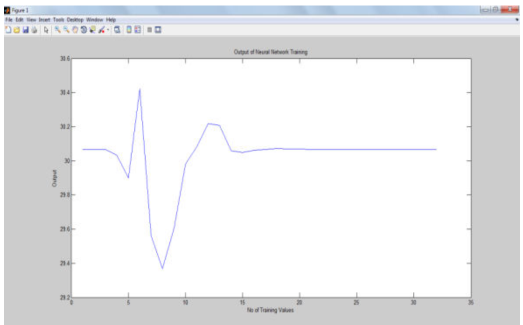

Fig. 14: Output of Neural Network training for K_p of PI Controller

ISBN: 978-1-61804-049-7

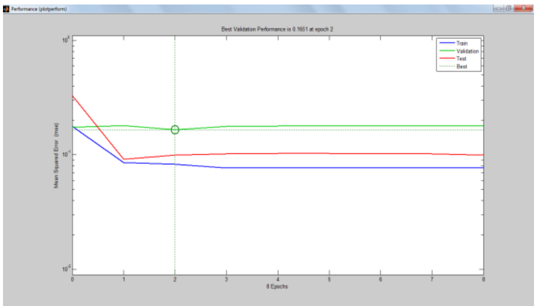

Fig. 15: Neural Network Performance Plot for T_i of PI Controller

Based on the Neural Network Performance Plot for T_i of PI Controller in Fig. 15, the mean squared error (MSE) for testing, training and validation T_i value were calculated to give values of 0.9, 0.8 and 1.07 respectively. This network has slightly higher error compared to the performance plot for Kp but it is acceptable as it still shows a small MSE values. However, since MSE value is slightly higher, it shows that the network can be further improved to obtain a better result. Based on the training, the output of the network obtained was plotted as shown in Fig. 16. Based on the graph, the output for T_i is around 7.39.

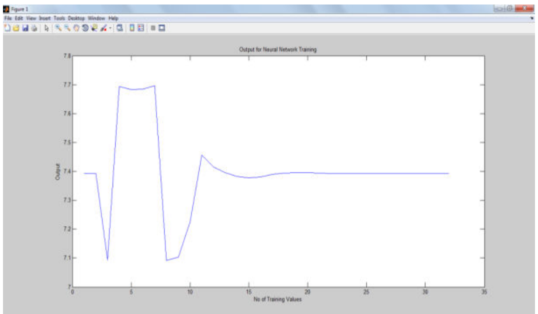

Fig. 16: Output of Neural Network training for T_i

The overall performance of the plant was illustrated in Fig. 17. The decay ratio was lower than the initial plant performance. The settling time also improved than the initial plant performance. Overall, the neural network helped to improve the plant performance.

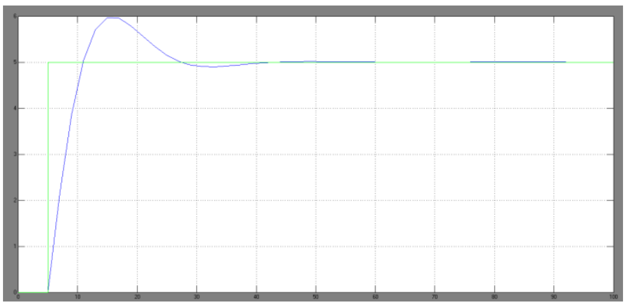

Fig. 17: Plant performance of PI controller after fine-tuning

The plant model shown in Fig. 18 was simulated to generate data that will be used for NN training in NN Predictive Controller.

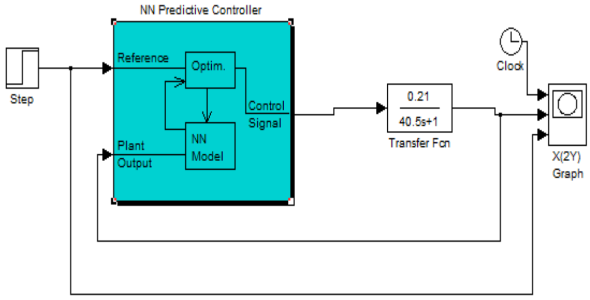

Fig. 18: Plant model using NN Predictive Controller

The inputs for NN training in Neural Network Predictive Controller were the data plant output obtained by simulating the plant model illustrated in Fig. 11. They are randomly divided to training, validation and testing data. The plant output is the output of the Simulink model while the NN output is the NN plant model output, a one step ahead prediction of the plant output. The error is the difference between the plant output and the NN plant model output. The error obtained is less than 0.058 which is closed to zero. The training is done for different step inputs of data and the results obtained are similar.

The differences between training, validation and testing data are that training is done to adjust the weights on the neural network, validation is used to minimize over fitting so that the neural network does not need to use all the data to train the network by confirming that there is accuracy in the training data while validation data stays the same. Testing is done to test the final result and verify the predictive power of the network.

4 Conclusion

The tuning rule based on the Ziegler Nichols Process Reaction Curve is able to fine tune the PID controller to produce a satisfactory step response. This project focused on neural network model development to fine-tune PID Controller based on a first-order with dead time plant model. A Neural Network (NN) Model was then built to predict the tuning parameters for the PID controllers of a first-order with dead time plant model. This then lead to the development of Neural Network Predictive Controller (NNPC). NNPC yield better performance compared to conventional PID-type controller. Future work suggests to accomodate

ISBN: 978-1-61804-049-7

disturbances and time delay in the design process for the real plant system.

References:

[1] Silva, G.J.; Datta, A.; Bhattacharyya, S.P., New Results on the Synthesis of PID Controllers. *IEEE Trans. Automat. Contr.* Vol.47, No.2, 2002, pp. 241-252.

[2] Giri Rajkumar, S. M.; Atal. A. K.; Anantharaman, N., Application of Genetic Algorithm for Tuning of a PID Controller for a Real Time Industrial Process. *Sensors & Transducers Journal*, Vol. 121, 2010, pp. 56-67.

[3] Zulfatman and Rahmat, M. F., Application of Self-Tuning Fuzzy PID Controller on Industrial Hydraulic Actuator Using System Identification Approach, *International Journal on Smart Sensing and Intelligent Systems*, Vol.2, 2009, pp.246-261

[4] Rezazadeh, A. , M. Sedighizadeh,, M. Karimi, Proton Exchange Membrane Fuel Cell control using a Predictive control based on Neural Network, *International Journal of Computer and Electrical Engineering*, Vol. 2, No. 1, 2010, pp. 1793-8163.

[5] Anna V. and B. Monika, Neural Network Predictive Control of a Chemical Reactor, *Journal of Computational Information Systems*, Vol. 6, No. 11, 2010, pp. 3501-3508.

[6] Ching, H.L, H.L.Yi, C.T. Ching, A Novel Robust PID Controller Design by Fuzzy Neural Network, *Asian Journal of Control*, Vol. 4, No. 4, 2008. pp. 433–438.

[7] Akesson, B.M., H.T. Toivonen, A Neural Network Model Predictive Controller, *Journal of Process Control,* Vol. 16, 2006, pp. 937–946.

[8] Frahani, S.S.S.A, M.A. Nekouei, M. Nikzad, Predictive Control of a Single Link Flexible Joint Robot Based on Neural Network and Feedback Linearization, *Australian Journal of Basic and Applied Sciences*, Vol. 3, No.3, 2009, pp. 2322-2333.

[9] Durmus, B. and N. Yumusak, Nonlinear Generalized Predictive Controller Based On Artificial Neural Network for Robot Control, *Journal of Applied Science*, Vol. 8, 2008, pp. 3783-3794.

[10] Hamilton, L.C. *Modern Data Analysis - A First Course of Applied Statistics*, 2nd edition., Brooks/Cole Publishing Company: Pacific Grove, CA, USA, 1990.

[11] Marlin, T.E., *Process Control : Designing Processes and Control Systems For Dynamic Performance*, 2nd edition, McGraw Hill Publishing Company: Singapore, 1999.

[12] May, Z, *Design of a Simply Structured Compensator.* MSc. Dissertation, Dept. of Advanced Control, University of Manchester Institute of Science and Technology (UMIST), UK, 2000.