

Hardware Implementation of the Neural Network Predictive Controller for Coupled Tank System

Ammar A. Aldair *

Electrical Engineering Department, University of Basrah, Basrah, Iraq

*Corresponding author: mmr.ali2@googlemail.com

Received April 24, 2014; Revised June 17, 2013; Accepted January 25, 2014

Abstract In this paper, a neural network based predictive controller is designed for controlling the liquid level of the coupled tank system. The controlled process is a nonlinear system; therefore, a nonlinear prediction method can be a better match in a predictive control strategy. The neural network predictive controller that is discussed in this paper uses a neural network model of a nonlinear plant to predict future plant performance. The simulation results are compared with PID control. The results show that the effectiveness of using the neural predictive controller for the coupled tank system. The Simulink Toolbox in MATLAB has been used to simulate the controlled system with the proposed controller. The VHDL has been used to describe the implementation of neural controller. Xilinx ISE Project Navigator Version 10.1 is used to obtain the compilation and timing test results as well as the synthesized design. The hardware implementation of the neural network predictive controller using FPGA board is proposed. To make sure that the FPGA board works like the simulated neural predictive controller, MATLAB programme is used to compare between the set of the data that are obtained from the ModelSim program and the set of the data that are obtained from the MATLAB Simulink model. Simulation results show that the FPGA board can be used as neural predictive controller for controlling the liquid level of the coupled tank system.

Keywords: neural network predictive controller, coupled tank system, FPGA

Cite This Article: Ammar A. Aldair, "Hardware Implementation of the Neural Network Predictive Controller for Coupled Tank System." *American Journal of Electrical and Electronic Engineering*, vol. 2, no. 1 (2014): 40-47. doi: 10.12691/ajeec-2-2-2.

1. Introduction

The coupled-tank liquid level control system is regarded as the relevant plant to emulate the process control in petrol and chemical industries. The processing plants in these industries largely involve in controlling the liquid level and the flow rate from one tank to another in the presence of nonlinearity and disturbance.

Many authors investigated the problem of controlling liquid flow of single or multiple tanks. The neurofuzzy controller based on the radial basis function neural network is tuned automatically using genetic algorithm to design a control system for the coupled tank [1]. In reference [2], the direct model reference adaptive control is designed for coupled tank system. While, an iterative learning control is used to tune an adaptive neurofuzzy inference system inverse controller for tank system [3]. Khan and Spurgeon presented a second order sliding mode control algorithm for a class of MIMO nonlinear system. This algorithm has been applied for robust control of liquid level in interconnected twin-tanks [4]. Intelligent control including fuzzy control [5,6], neural network control [7], and genetic algorithms [8] have also been applied to the coupled tanks system.

Model predictive control (MPC) is used successfully in some practical application since it was developed in 1980s

[9]. The most popular MPC algorithms that found a wide acceptance in industry are: Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Predictive Functional Control (PFC), Extended Prediction Self Adaptive Control (EPSAC), Extended Horizon Adaptive Control (EHAC) and Generalized Predictive Control (GPC). GPC has been originally developed with linear plant predictor model, which leads to a formulation that can be solved analytically. But in many industrial processes which usually have nonlinear dynamical behaviours and such complicated systems may be not easily to control by the linear GPC method. In recent years, incorporation of neural networks as intelligent control techniques and adaptive control system has been claimed to be a new method for the control of systems with significant nonlinearities. Those neural network based adaptive control system is called neural network predictive control (NNPC). It is widely used as a controller for many nonlinear systems [10,11,12,13,14]. In this work, a neural network predictive controller is designed for controlling the liquid level of the coupled tank system.

The special-purpose VLSI chips is used to implement the NNPC in real time applications with enhanced processing speed, but due to different neural models and limited use, it has been prohibitively expensive and time consuming to develop such chips. In this paper, the network predictive control is mapped into hardware using field programmable gate array FPGA board to design a

control system for coupled tank level. A Very High speed integrated circuit Hardware Description Language (VHDL) codes have been used to describe the operation of the NNPC. The Xilinx Integrated Software Environment (Xilinx ISE 10.1) has been used as environment to type the VHDL codes. The Xilinx Spartan XC3S500E FPGA board has been suggested to use in this work. The Spartan 3E starter Kit board highlights the unique features of the Spartan_3E FPGA family and provides a convenient development board for embedded processing applications. The Hirose 100-pin FX2 Edge connector (J3) can be used as input and output port to receive the digital input error data from the A/D converter (error signal is the difference between the output of the controlled system and desired output) and send the digital control signal to the D/A converter. The analogue control signal is used to force the output of the controlled system to follow the desired output.

For the simulation results, ModelSim XE III 6.4b simulation program will be used with the FPGA-based design. In order to compare the responses of FPGA design with the Simulink design, an M-file will be used to plot the data collected from the ModelSim program and the other data collected from the Simulink design to make sure the FPGA boards work like the neural network predictive controller for coupled tank system. Simulation results show that the FPGA board can be used as neural predictive controller for controlling the liquid level of the coupled tank system.

2. Dynamic Model of the Two Tank System

The coupled tank system is shown in Figure 1. It consists of two identical hold-up tanks coupled by an orifice. The input is supplied by a variable speed pump which supplies liquid to the first tank. The orifice allows the liquid to flow into the second tank and hence out to a reservoir. The control task is to adjust the inlet flow rate $q(t)$ by pumping the liquid into the first tank so as to force the level of the liquid in the second tank, h_2 , close to a desired set point level, H . It is assumed that the back pressure created by the water-head of the second tank does not affect the flow rate of the pump significantly. The controller has been designed based on the neural predictive control.

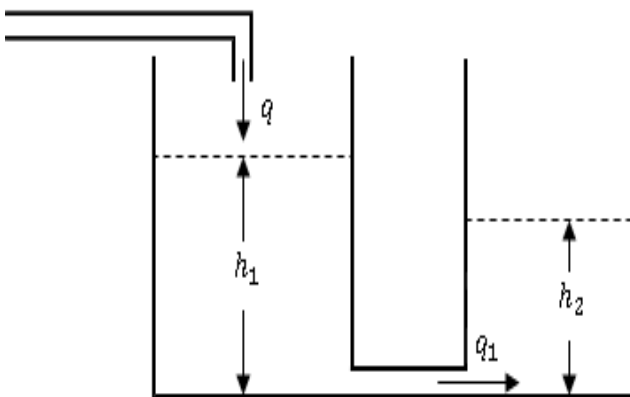


Figure 1. Two Tanks System

2.1. System Modelling

The twin-connected tanks system is a nonlinear dynamical system and the governing dynamical equations can be written as

$$\dot{h}_1 = -\frac{a_{12}}{A} \sqrt{2g|(h_1 - h_2)|} \operatorname{sgn}(h_1 - h_2) + \frac{1}{A} q \quad (1)$$

$$\dot{h}_2 = \frac{a_{12}}{A} \sqrt{2g|(h_1 - h_2)|} \operatorname{sgn}(h_1 - h_2) - \frac{a_2}{A} \sqrt{2g|h_2|} \quad (2)$$

where h_1 is the level in the first tank;

h_2 is the level in the second tank;

q_1 is the flow rate from the first tank to the second tank;

q_2 is the flow rate from the second tank to a reservoir;

q is the inlet flow rate;

g is the gravitational constant;

A is the cross-section area of first tank and second tank;

a_2 is the cross-section area of the outlet orifice;

a_{12} is the cross-area of the coupled orifice.

2.2. System Constraints

The pump can only pump the water into the first tank, therefore, the inlet flow rate q is always positive. The constraint on the inlet flow rate is given by

$$q \geq 0 \quad (3)$$

In the steady state, for constant water level set points, the respective derivatives must be zero separately i.e.,

$$\dot{h}_1 = \dot{h}_2 = 0 \quad (4)$$

At the steady state, Equation. 1 and Equation 2 can be rewritten as

$$0 = -\frac{a_{12}}{A} \sqrt{2g|(h_1 - h_2)|} \operatorname{sgn}(h_1 - h_2) + \frac{1}{A} Q \quad (5)$$

$$0 = \frac{a_{12}}{A} \sqrt{2g|(h_1 - h_2)|} \operatorname{sgn}(h_1 - h_2) - \frac{a_1}{A} \sqrt{2g|h_2|} \quad (6)$$

where Q is the steady state inflow rate. From Equation 5 and Equation 6, to satisfy the constrain in Equation 3, the following constrain should be satisfy as

$$h_1 \geq h_2 \quad (7)$$

3. Neural Network Predictive Control

To design the neural network predictive control, two steps should be carried out:

- Determine the neural network plant model for the given nonlinear system (system identification).
- The neural network plant model is used by the controller to predict future performance.

The following subsection described the steps design of the NNPC.

3.1. Nonlinear System Modeling Using Neural Networks

System identification is the process of developing a mathematical model of a dynamic system based on the input and output data from the actual process. An important stage in control system design is the

development of a mathematical model of the system to be controlled. Previous studies in system identification have demonstrated that neural networks are successful in modelling many non-linear systems [13]. Before neural networks are investigated for identification, linear techniques such as auto regressive with exogenous input (ARX) and auto regressive moving average with exogenous input (ARMAX) will be applied to nonlinear system.

The model of any nonlinear system in the form of NARMA is given as

$$y(t+1) = f\left(y(t-1), y(t-2), \dots, y(t-n_y), u(t-d-1), \dots, u(t-d-n_u)\right) \quad (8)$$

where $f(\cdot)$ represents an unknown nonlinear system, d represented the time delay of the system, n_y and n_u are the orders of the system. For an unknown model plant, many forms of NN model can be applied. In this work, the multilayer perceptron neural network is used to model the nonlinear system as shown in Figure 2. The neural network design to identify the plant is expressed as

$$\begin{aligned} \hat{y}(t+1) &= \hat{f}\left(\hat{y}(t-1), \hat{y}(t-2), \dots, \hat{y}(t-n_y), u(t-d-1), \dots, u(t-d-n_u)\right) \\ &= F\left(\sum_{j=0}^q v_j g\left(\sum_{l=1}^m w_{lj} u_l + b_{1j}\right) + b_2\right) \end{aligned} \quad (9)$$

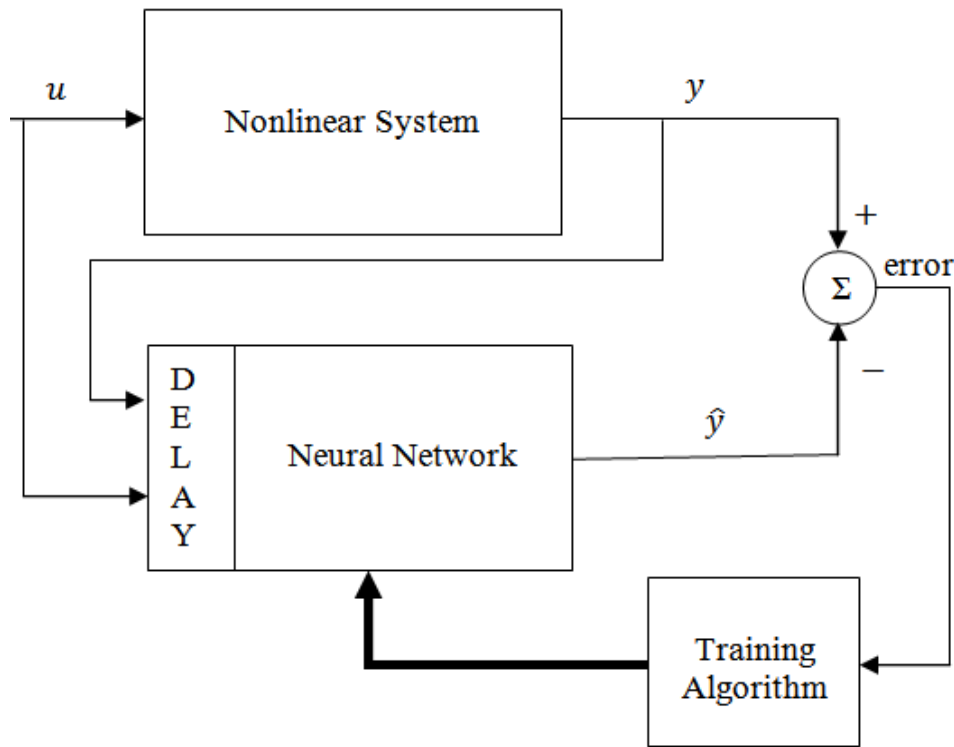


Figure 2. Nonlinear system identification

Where $\hat{y}(t)$ is the output of the neural network system, u_l represents the l^{th} input of the neural network; v_j , w , b_1 and b_2 are the weights and biases of the NN (the adjustable parameters of the NN); $F(\cdot)$ and $g(\cdot)$ are nonlinear functions ; q and m are the number of the nodes in the hidden and input layer, respectively.

The adjustable parameters of the NN are determined from a set of examples through the process called training. The examples, or the training data as they are usually called, are a set of inputs, $u(t)$, and corresponding desired outputs, $y(t)$.

Specify the training set by

$$Z^N = \{[u(t), y(t)] | t = 1, \dots, N\} \quad (10)$$

where N is the number of the samples in training set.

The objective of training is then to determine a mapping from the set of training data to the set of possible weights. Therefore, the neural network will produce

predictions $\hat{y}(t)$, which in some sense are “close” to the true outputs $y(t)$

Because the training of neural network is often operated off-line, any training algorithms in batch version can be developed to minimize the following criterion

$$V_N = \frac{1}{2N} \sum_{i=1}^N [y(t) - \hat{y}(t|\theta)]^T [y(t) - \hat{y}(t|\theta)] \quad (11)$$

where θ is the trainable parameters vector.

The weights are then found as $\theta = \arg \min V_N$. By some kind of iterative minimization scheme:

$$\theta^{(i+1)} = \theta^{(i)} + \mu^{(i)} S^{(i)} \quad (12)$$

Where $\theta^{(i)}$ specified the current iterate; $S^{(i)}$ is the search direction; and $\mu^{(i)}$ the step size.

A large number of training algorithms exist, each of which is characterized by the way in which search

direction and step size are selected. The MATLAB Neural toolbox provides a lot of training algorithms such as: Backpropagation algorithm, Variable Learning Rate algorithm, Resilient Backpropagation algorithm, Conjugate Gradient algorithm, Quasi-Newton Algorithm, Levenberg-Marquardt algorithm, and Reduced Memory Levenberg-Marquardt algorithm.

3.2. Predictive Controller

The objective of the predictive control strategy using neural predictors is twofold: (i) to estimate the future output of the plant and (ii) to minimize a cost function based on the error between the predicted output of the processes and the reference trajectory. The cost function, which may be different from case to case, is minimized in order to obtain the optimum control input that is applied to the non-linear plant. In most of the predictive control algorithms a quadratic form is utilized for the cost function:

$$J_p = \sum_{i=N_1}^{N_2} \left[y(k+1) - r(k+i) \right]^2 + \rho \sum_{i=1}^{N_u} \left(\dot{u}(k+i-1) - \dot{u}(k+i-2) \right)^2 \quad (13)$$

with following requirements:

$$\Delta u(k+i-1) = 0, 1 \leq N_u < i \leq N_2 \quad (14)$$

where N_u is the control horizon; N_1 is the minimum prediction horizon; N_2 is the prediction horizon; i is the order of the predictor; r is the reference trajectory; ρ is weight factor; and Δ is the differentiation operator.

The command u may be subject to amplitude constraints

$$u_{\min} \leq u(k+i) \leq u_{\max} \quad (15)$$

The cost function is often used with the weight factor $\rho = 0$. A very important parameter in the predictive control strategy is the control horizon N_u , which specifies the instant time, since when the output of the controller should be kept at a constant value.

The output sequence of the optimal controller is obtained over the prediction horizon by minimizing the cost function J_p with respect to the vector U . This can be achieved by setting

$$\frac{\partial J_p}{\partial U} = 0 \quad U = [u(k-d), \dots, u(k-d+N_u-1)]^T \quad (16)$$

The analytical approach to the optimization problem needs for the differentiation of the cost function and, finally, leads to a nonlinear algebraic equation; unfortunately this equation cannot be solved by any analytic procedure. This is why a computational method is preferred for the minimization of the cost function, also complying with the typical requirements of the real-time implementations (guaranteed convergence, at least to a sub-optimal solution, within a given time interval). Figure 3 depicts the structure of the NNPC. The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of u' that minimize J_p , and then the optimal u is inputted to the plant (nonlinear system).

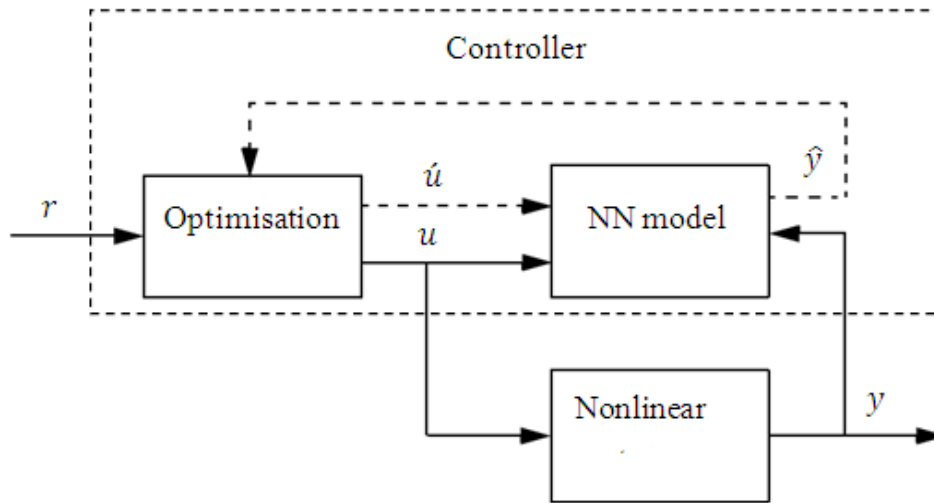


Figure 3. Structure of neural network predictive control

4. Hardware Implementation of the Neural Predictive Control

During the last years, consumer digital devices have been built using either application specific hardware modules (ASICs) or general purpose software programmed microprocessors, or a combination of them. Hardware implementations offer high speed and efficiency but they are tailored for a specific set of computations. If an alternative implementation is needed, a new and expensive design process has to be performed. On the contrary, software implementations can be modified freely

during the life-cycle of a device, through patches and updates. However, they are much more inefficient in terms of speed and area. Reconfigurable computing is intended to fill the gap between hardware and software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware. Reconfigurable devices, including Field-Programmable Gate Arrays (FPGAs), contain an array of computational elements whose functionality is determined through multiple programmable configuration bits. These elements, usually called logic blocks, are connected using a set of routing resources that are also programmable.

Neural networks can be implemented using analogy or digital systems. The digital implementation is more popular as it has the advantage of higher accuracy, better repeatability, lower noise sensitivity, better testability, higher flexibility, and compatibility with other types of pre-processors. Two implementation methods can be used to implement the NNPC: (i) FPGA implementation; and (ii) DSP implementation. FPGA is a suitable hardware for neural network implementation as it preserves the parallel architecture of the neurons in a layer and offers flexibility in reconfiguration.

The VHDL coding developed is tested using Xilinx 10.1 software tool to design the NNPC. This software is synthesised the VHDL codes and generated a configuration file that used to program the FPGA board. The ModelSim 6.4b software tool is used to simulate the output response of the FPGA boards. The same NNPC is

designed and simulated using MATLAB software tool. This Software-Based Controller (SBC) will be used to make a comparison with the proposed design. This comparison is important because it tells us to which extent our FPGA-Based Controller (FBC) is close to similar controller designed as a computer program.

5. Simulation and Results

In this work, the Simulink Toolbox is used to simulate the NNPC and coupled tank system. Figure 4 depicts the MATLAB Simulink model of coupled tank system. While, Figure 5 shows the connection between NNPC and coupled tank system.

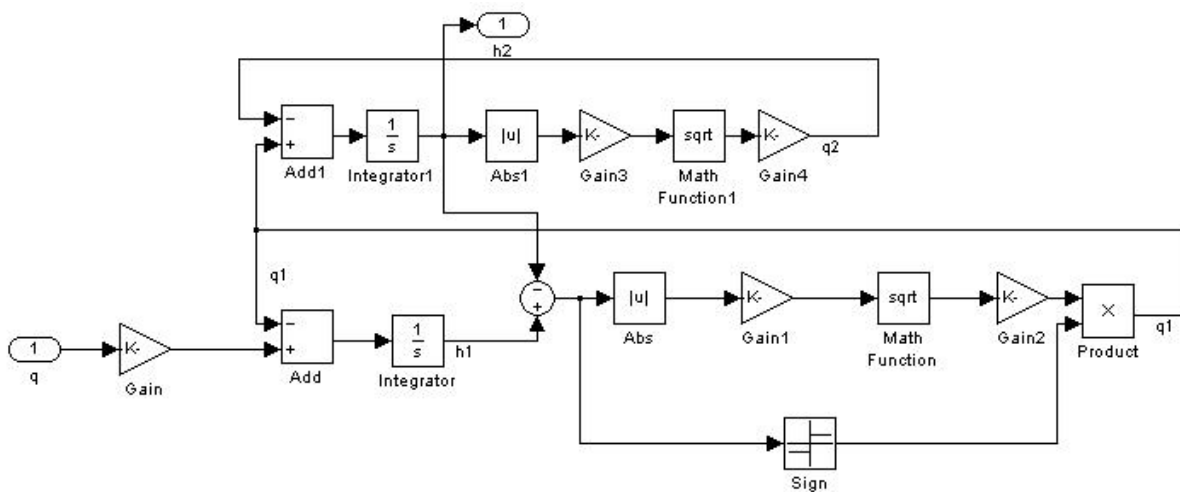


Figure 4. MATLAB Simulink model of coupled tank system

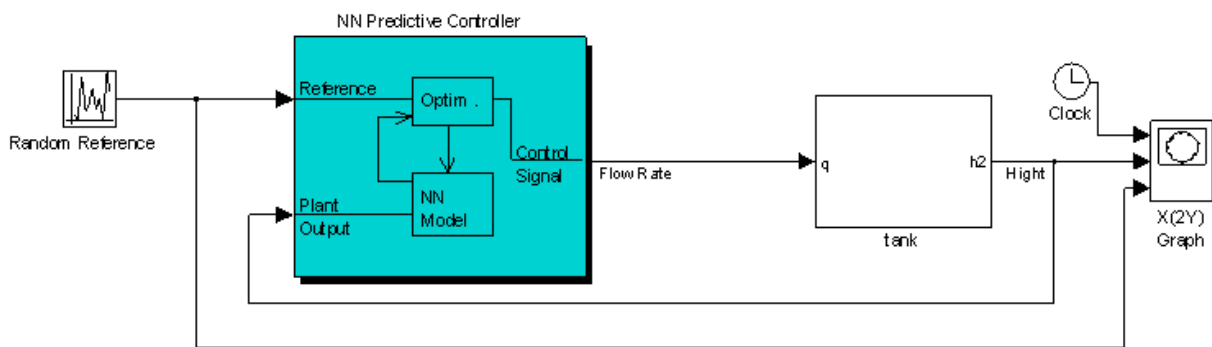


Figure 5. MATLAB Simulink model of NNPC with coupled tank system

As it is mentioned before, to design the neural network predictive control, two steps should be carried out:

- Determine the neural network plant model for the given nonlinear system (system identification).
- The neural network plant model is used by the controller to predict future performance.

To carry out the first step, the neural network is used to design the identification system for the coupled tank system. The training data should be generated by applying a random step inputs to the coupled tank Simulink model. Figure 6 shows the input/output data set that is used to design the neural network identifier.

The Levenberg-Marquardt Algorithm (LMT) is an iterative technique that locates the minimum of a

multivariate function expressed as the sum of squares of nonlinear real-valued functions. It has become a standard technique for nonlinear least-squares problems [15], widely adopted in a broad spectrum of disciplines.

To design the NN identifier, we assume that it has four inputs ($y(t-1), y(t-2), u(t-1)$) and $u(t-2)$ and one output $\hat{y}(t)$. The size on hidden layer is assumed seven.

In this work, the Levenberg-Marquardt algorithm is used as training algorithm to adjust the trainable parameters of the NN. After 166 optimization iteration steps, the objective function V_N has been minimized to very small value as shown in Figure 7.

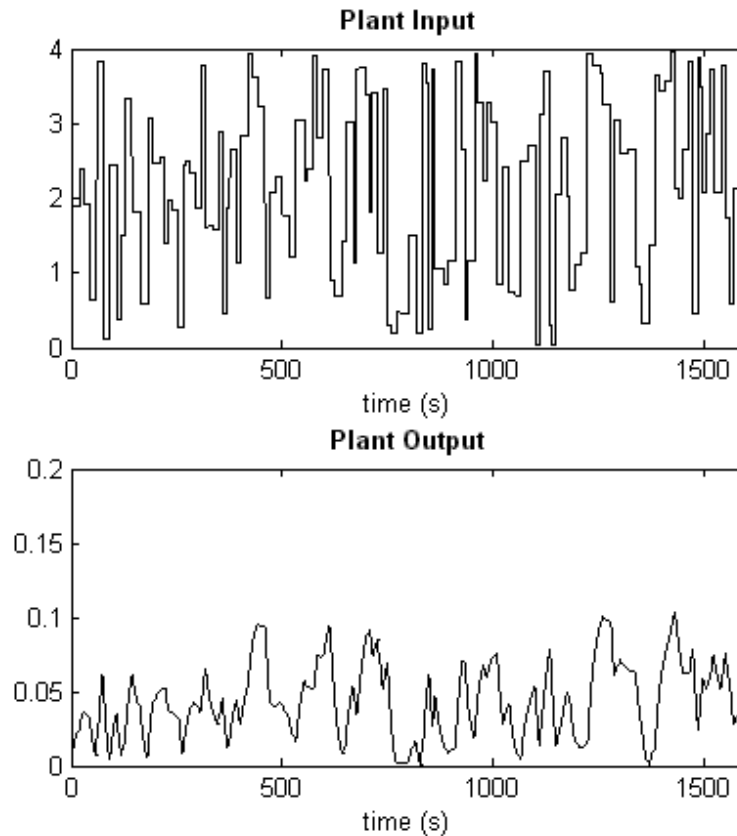


Figure 6. Input/output data training set

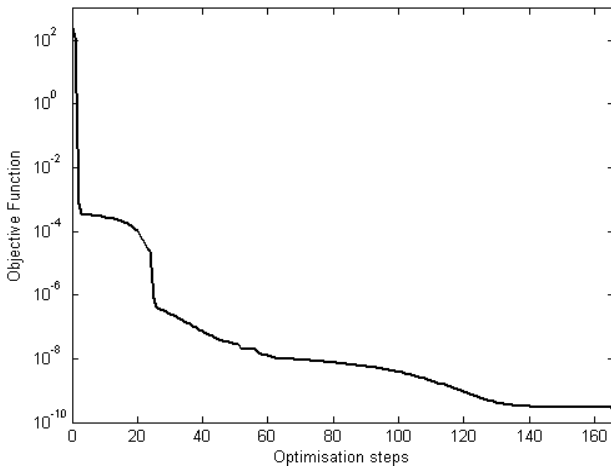


Figure 7. Objective Function during optimization steps

The optimization problem was addressed in accordance with the computational scenario built in Subsection 3.1. With respect to the notations introduced in Subsection 3.1, the following concrete values were chosen for the tuning parameters of the predictive control algorithm: $N1=1$, $N2=10$, $Nu=5$, and $\rho =0.05$. The NNPC is generated the suitable control signal to force the liquid level of the second tank to track the reference input signal. In this work, the random step input is used as the reference input signal. In order to investigate the effectiveness of the neural network based predictive controller, the output response of the controlled system with NNPC is compared with output response of the controlled system with PID controller as shown in Figure 8. The Internal model control theory is used to design the PID controller. Figure 8 shows that the overshoot and settle time of the output response are minimized.

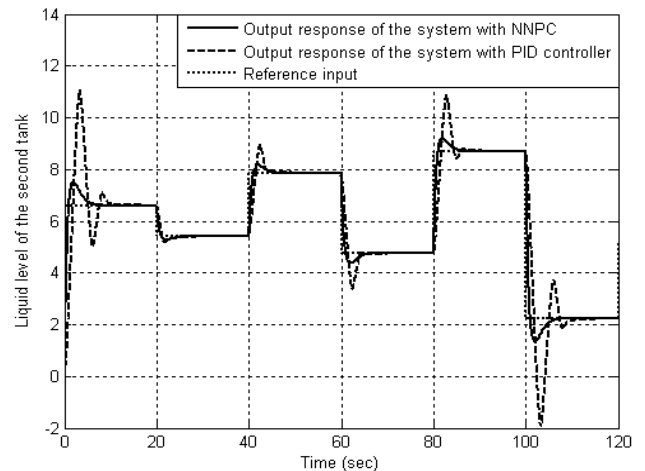


Figure 8. the output response of the coupled tank system with different control system

After the design of neural network predictive controller has been completed, the FPGA board can be used to implement the proposed control system. First, VHDL codes, which describe the operation of NNP controller, are written (the flowchart of VHDL program is shown in Figure 9). This program is downloaded into the FPGA chip (XILINX Spartan XC3S700AN) by using USB cable. Then, the Hirose 100-pin FX2 Edge connector (one part of the FPGA board) can be used to connect the FPGA board with the tank system. The error between the reference input and the system output can be applied as input to the A/D converter. The digital output of the A/D converter has been applied as input data to the FPGA boards. The FPGA board generates digital inputs to the NNP controller (error and error rate). The NNP controller generates a

suitable digital control signal based on the program that is stored in the FPGA chip. The digital control signal is sent to the D/A convertor to generate an analogy control signal which is applied as input to the pump of tank system. Figure 10 shows the control signal obtained from ModelSim program, while, Figure 11 shows the output signal of the NNP controller obtained from MATLAB simulation. By comparing Figure 10 and Figure 11, it can be seen that, the two output signals are identical.

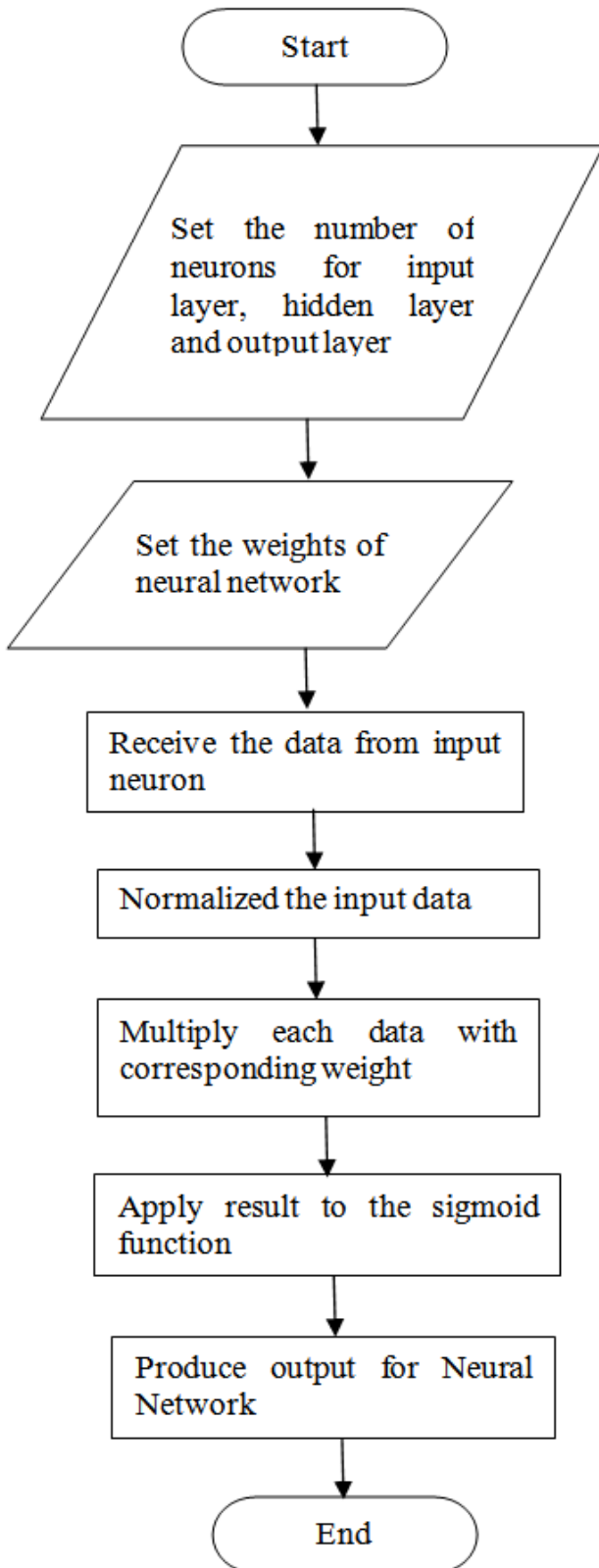


Figure 9. the flowchart of describe the Neural network using VHDL

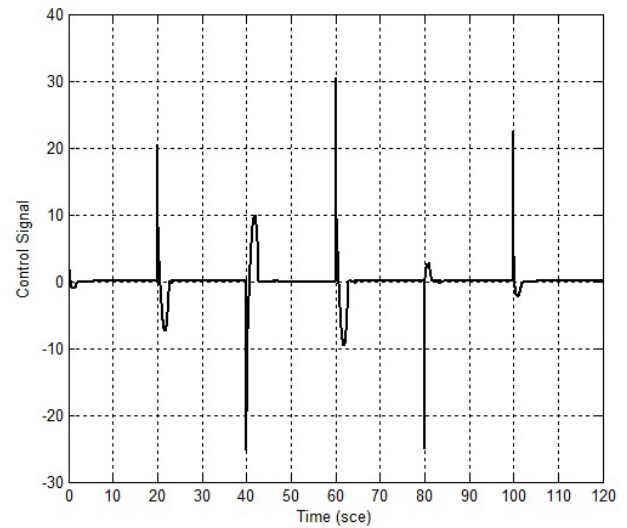


Figure 10. Control signal obtained from FPGA Board

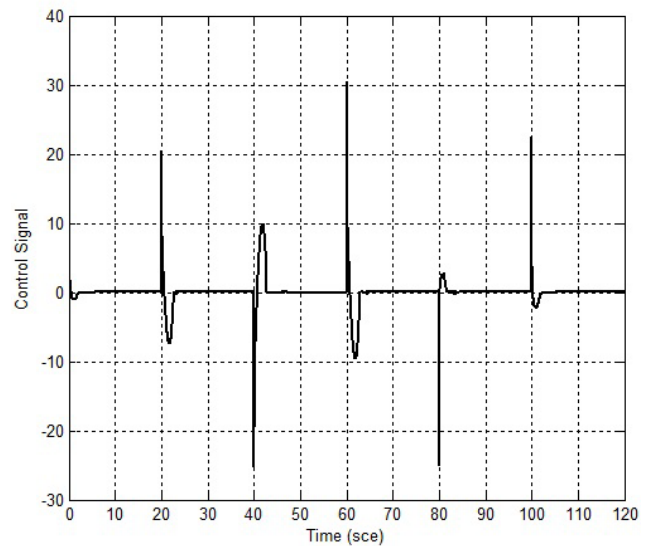


Figure 11. Control signal obtained from MALAB Simulation

6. Conclusion

The neural network based predictive control is proposed in this paper for controlling the liquid level of the nonlinear coupled tank system. Using the neural predictive controller, the liquid level of the second tank is tracked the desired set points by applying the liquid flow rate as a control signal. The Levenberg-Marquardt training algorithm is used to design the neural network identifier which is used to design the predictive control. To investigate the effectiveness of the neural network based predictive controller, the output response of the coupled tank with the NNPC is compared with the response of the controlled system with PID controller. Simulation results show that when the proposed controller is applied, all transient response specification such as: maximum overshoot, settling time delay time, rise time and peak time have been minimized. It means the NNPC can be used effectively to implement the control system for the coupled tank system practically. The obtained adjustable parameters of the NN can be used to implement the controller using an FPGA board. The results show that the control signals obtained from FPGA boards are identical

to the control signals which are obtained from design simulation. Therefore, it can be arrived that the FPGA boards can be used to control the coupled tank system.

References

- [1] Seng, T.L., M. Khalid, and R. Yusof, *Tuning of a Neurofuzzy Controller by Genetic Algorithms with an Application to A Coupled Tank Liquid Level Control System*. International Journal of Engineering Applications on Artificial Intelligence, 1998. 11: p. 517-529.
- [2] Mahyuddin, M.N., M.R. Arshad, and Z. Mohamed. *Simulation of Direct Model Reference Adaptive Control on a Coupled-Tank System Using Nonlinear Plant Model in International Conference on Control, Instrumentation and Mechatronics Engineering*. 2007. Johor, Malaysia.
- [3] Hassan, K.H., *Self Learning of ANFIS Inverse Control using Iterative Learning Technique*. International Journal of Computer Applications, 2011. 21(8): p. 24-29.
- [4] Khan, M.K. and S.K. Spurgeon, *Robust MIMO Water Level Control in Interconnected Twin-tanks Using Second Order Sliding Mode Control*. Control Engineering Practice 2006. 14: p. 375-386.
- [5] Rojas, I., et al. *Analysis and Electronic Implementation of a Fuzzy System for the Control of a Liquid Tank*. in *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems* 1997. Barcelona.
- [6] Saini, J.S. and Y.P. Singh, *Use of Causal Knowledge in a Real-time Fuzzy Logic Controller*. IEEE Trans. Ind, Application, 1999. 35: p. 554-560.
- [7] Evans, J.T., et al., *Implementation and Performance Evaluation of an On-line Neural Network Control Scheme*. International Conference on Control, 1994.
- [8] Ng, K.C., et al., *Genetic Algorithms Applied to Fuzzy Sliding Mode Controller Design*. First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995.
- [9] Clarke, D., C. Mohtadi, and P. Tuffs, *Generalized Predictive Control-Part I: the Basic Algorithm*. Automatica, 1987. 23(2): p. 137-148.
- [10] Sedighzadeh, M., M. Rezaei, and V. Najmi, *A Predictive control based on Neural Network for Proton Exchange Membrane Fuel Cell*. World Academy of Science, Engineering and Technology, 2011. 74: p. 461-465.
- [11] Lawrynczuk, M., *A Family of Model Predictive Control Algorithms with Artificial Neural Networks*. International Journal of Applied Mathematics and Computer Science, 2007. 17(2): p. 217-232.
- [12] Chen, X. and Y. Li, *Neural Network Predictive Control for Mobile Robot Using PSO with Controllable Random Exploration Velocity*. International Journal of Intelligent Control and Systems, 2007. 12(3): p. 217-229.
- [13] Li, X., Z. Chen, and Z. Yuan, *Simple Recurrent Neural Network-Based Adaptive Predictive Control for Nonlinear Systems*. Asian Journal of Control, 2002. 4(2): p. 231-239.
- [14] Chidrawar, S. and B. Pater, *Generalized Predictive Control and Neural Generalized Predictive Control* Leonardo Journal of Sciences, 2008. 13: p. 133-152.
- [15] Wilamowski, B.M., et al., *An Algorithm for Fast Convergence in Training Neural Networks*, in *International Joint Conference on Neural Networks*. 2001: Washington, DC p. 1778-1782.