

**Adaptive MLP Network Reference Model for Controlling Flow of Split Level
Coupled Tank**

by

Samer Sherif Kassem

Dissertation submitted in particular fulfillment of
the requirements for the
Bachelor of Engineers (Hons)
(Electrical & Electronics Engineering)

December 2010

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

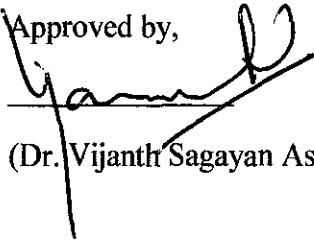
**Adaptive MLP Network Reference Model for Controlling Flow of Split Level
Coupled Tank**

by

Samer Sherif Kassem

A project dissertation submitted to the
Electrical Engineering Program
Universiti Teknologi PETRONAS
in particular fulfillment of the requirements for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,



(Dr. Vijanth Sagayan Asrivadam)

DR. VIJANTH SAGAYAN ASIRVADAM
Senior Lecturer
Electrical & Electronic Engineering Department
Universiti Teknologi PETRONAS
31750 Tronoh, Perak.
Tel: 05-3687881 Fax: 05-3657443

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



SAMER SHERIF KASSEM

ABSTRACT

This draft report explains about the background of tank systems such as the single tank system as well as the split coupled tank and how to model such tanks using Simulink blocks. A brief about neural network architectures were added to this report as well as the ARX model and Neural Network. The report also shows the project work flow such as the scope of study and literature review which have been found during this semester. The results and discussion consist of the designing of coupled tanks, PI and PID controllers as well as ARX model using Simulink/Matlab. The methodology shows the procedures of the project. The conclusion and recommendation of the project are added as well. Last but not least is the appendix which shows the calculations which were made and a CD is attached in the end containing all the Simulink blocks of the project.

ACKNOWLEDGMENT

My utmost gratitude and appreciation is extended to Dr. Vijanth Sagayan, my supervisor, who gave his great attention and dedication to my project and gave me the chance to learn from him. Thank you to his great effort and time.

Thanks to my classmate and friend “Majda”, who also helped me during the project and has always been there for me.

Last but not least, thanks to my family who have supported me during not only my final year but also during the whole five years especially my Dad who was always by my side even though his far away.

TABLE OF CONTENT

ABSTRACT.....	iv
ACKNOWLEDGMENT.....	v
LIST OF FIGURES	viii
LIST OF TABLES	x
CHAPTER 1 - INTRODUCTION.....	1
1.1 Background Of Study	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Scope Of Study.....	2
CHAPTER 2 - LITERATURE REVIEW	3
2.1 Single Tank System.....	3
2.2 Coupled Tank System	4
2.3 Adaptive Inverse Control	5
2.4 Nonlinear Internal Model Control.....	6
2.5 Model Predictive Control	7
2.6 Model Reference Control or (Neural Adaptive Control)	7
2.7 ARX Model Structure	8
2.8 Artificial Neural Network	9
CHAPTER 3 - METHODOLOGY	11
3.1 Procedure Identification	11
3.2 Tools and Equipment.....	12
CHAPTER 4 - RESULTS AND DISCUSSION.....	13
4.1 Design of Coupled Tank System using Simulink	13
4.2 Process Reaction Curve of the Plant	14
4.3 Designing of Controller.....	15
4.3.1 Ciancone Correlations with PID Controller.....	15
4.3.2 Cohen Coon Tuning Correlations.....	17
PI Controller.....	19
PID Controller.....	20
4.3.3 Ziegler Nicholas Closed Loop Method.....	21
PI Controller.....	22
PID Controller.....	23

4.4	Fine Tuning	24
4.5	Changing the Output to the Level H_1	26
4.6	Designing of ARX model	28
4.7	Training of Neural Network	33
CHAPTER 5 - CONCLUSION AND RECOMMENDATION		37
5.1	Conclusion	37
5.2	Recommendation	38
REFERENCES		39
APPENDICES		40
APPENDIX A – CALCULATIONS OF THE PROCESS REACTION CURVE PARAMETERS OF THE OUTPUT FLOW (Q_{OUT})		41
APPENDIX B – CALCULATIONS OF CIANCONE CONTROLLER		42
APPENDIX C – CALCULATIONS COHEN COON TUNING CORRELEATIONS		44
APPENDIX D – CALCULATIONS ZIEGLAR NICHOLS CLOSED LOOP TUNING METHOD CALCULATIONS		46
APPENDIX E – M-FILES		47
APPENDIX F – CD OF THE PROJECT		49

LIST OF FIGURES

Figure 1 - Single Tank System.....	3
Figure 2 - Split Coupled Tank System.....	4
Figure 3 - Adaptive Inverse Control System.....	6
Figure 4 - Nonlinear Internal Model Control.....	6
Figure 5 - Model Predictive Control.....	7
Figure 6 - Model Reference Adaptive Control.....	8
Figure 7 – Black Box System.....	8
Figure 8 - Graphical Presentation of an MLP.....	9
Figure 9 – Flow chart of project.....	11
Figure 10 - Interacting Couple Tank.....	14
Figure 11 - Process Reaction Curve.....	14
Figure 12 - Ciancone PID Block.....	15
Figure 13 - Ciancone PID Block.....	16
Figure 14 - Input & Output flow of Ciancone method.....	17
Figure 15 - Level of the tanks using Ciancone controller.....	17
Figure 16 - Simulink Block for Cohen Coon Method.....	18
Figure 17 - Level of the tanks of PI controller.....	19
Figure 18 - Flow of the PI Controller.....	19
Figure 19 - Flow of the PI controller.....	19
Figure 20 - Levels for PID Controller.....	20
Figure 21 - Input & Output flow for the PID Controller.....	20
Figure 22 - Bode Plot of the Coupled Tank System.....	21
Figure 23 - Ziegler Nichols Closed Loop Method.....	22
Figure 24 - Levels for the PI controller.....	23
Figure 25 - Flow for PI controller.....	23
Figure 26 - Comparison between Old & Modified PI Controller using ZN Closed Loop.....	24
Figure 27 - Comparison between the levels of the Old & Modified PI Controller.....	25
Figure 28 - Comparison between the flows of the Old & Modified PI Controller.....	26
Figure 29 - The plant with the new output (H_1).....	26

Figure 30 - Random Input Variables block for New Output	27
Figure 31 - Output of H_1	28
Figure 32 - The values of alphas & betas for different ARXs	28
Figure 33 - ARX model	29
Figure 34 - a graph between ARX model (Blue), PI controller (Red) & plant without controller (Green) in the level H_1	29
Figure 35 - ARX Model as an Input to the PI controller	30
Figure 36 - The Whole System with plant, PI Controller & ARX Model	31
Figure 37 – a graph of level H_1 between PI controller with ARX model as input (Blue), PI controller with input flow (Red) & the Plant (Green).....	32
Figure 38 – a graph of Output flow (Q_{out}) between PI controller with ARX model as input (Blue), PI controller with input flow (Red) & the Plant (Green)	32
Figure 39 - Neural Network fitting tool window	33
Figure 40 - Selecting the neural network size.....	34
Figure 41 - Selecting the input & target data	34
Figure 42 - Training the network	35
Figure 43 – The Training Panel	35
Figure 44 - Performance curve.....	36
Figure 45 - Regression values	36

LIST OF TABLES

Table 1 - Cohen Coon Calculations	18
Table 2 - Ziegler Nicholas Closed Loop Calculations	22
Table 3 - Comparison of performance between the three methods	24
Table 4 - Guidelines for Selecting Controller Modes	27

CHAPTER 1

INTRODUCTION

1.1 Background Of Study

The control of liquid level in tanks and flow between tanks is a basic problem in the process industries. The process industries require liquids to be pumped, stored in tanks, and then pumped to another tank. Many times the fluids will be processed by chemical or mixing treatment in the tanks, but always the level of fluid in the tanks must be controlled, and the flow between tanks must be regulated. Often the tanks are so coupled together that the levels interact and this must also be controlled. Vital industries where liquid level and flow control are essential include:

- Petro-chemical industries.
- Paper making industries.
- Water treatment industries.

Our lives are governed by level and flow control systems. For example, medical physiology involves many fluid bio-control systems. Bio-systems in our body are there to control the rate that blood flows around our body. Other bio-systems control the pressure and levels of moisture and chemicals in our body.

Tank level control systems are everywhere. All of our process industries, the human body and fluid handling systems depend upon tank level control systems. It is essential for control systems engineers to understand how tank control systems work and how the level control problem is solved [1].

The PI and PID controllers have been used heavily in the process industries, mostly concerned about improving its performance and efficiency without using other approaches. In this paper, couple tanks are used to select the best controller (either PI or PID) based on its performance and stability, and then the best

controller is compared with an intelligent controller using “ARX” model based on their performance in controlling the couple tanks.

1.2 Problem Statement

To control by the levels of both tanks of the coupled tanks and the output flow from the second with random input flow from the first tank.

1.3 Objectives

- Design a coupled tank using Simulink.
- Implement a controller to the coupled tank
- Design an ARX model for the coupled tank.
- Implement Neural Network training.
- Compare the performances of the ARX model and neural network model with the controller.

1.4 Scope Of Study

The scope of study is designing a single tank model first from linearized flow equations using Simulink then implementing a PI controller to the model.

The couple tank is then designed on Simulink as well and three different PI controller calculations are implemented. The best controller in performance is then selected.

An “ARX” model will be designed for the split coupled tank and a comparison will be made between the performances of the two models on the split coupled tank.

CHAPTER 2

LITERATURE REVIEW

It's important to understand the mathematics of how the coupled tanks system behaves. System modeling is a very important part in control system analysis. Before modeling of the split coupled tanks, a model is made for the single tank. This will facilitate the modeling and simulation on the split coupled tanks as well as the controlling of flow. Fig. 1 shows the single tank level system.

2.1 Single Tank System

The system model is determined by relating the flow Q_i into the tank to the flow Q_o leaving through the valve at the tank bottom. Using a balance of flows equation on the tank, it is possible to write:

$$Q_i - Q_o = A \frac{dH}{dt} \quad (\text{Equation 1}) \quad \text{and} \quad Q_o = k\sqrt{h} \quad (\text{Equation 2})$$

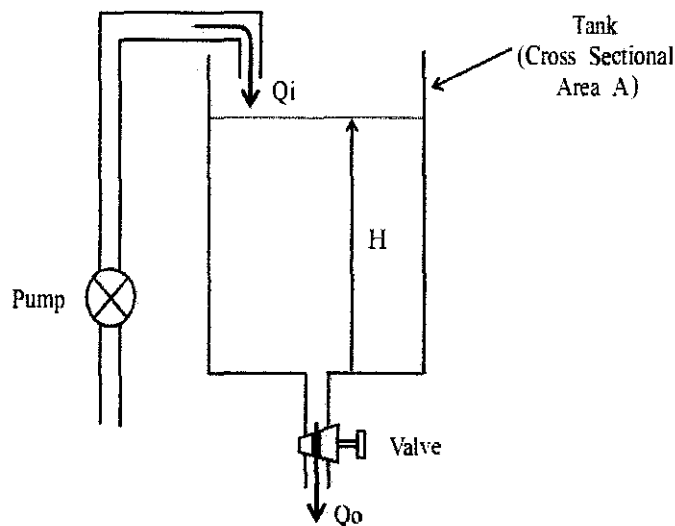


Figure 1 - Single Tank System

Where, A is the cross sectional area of the tank, H is the height of the fluid in the tank and k is the valve resistance of the tank.

2.2 Coupled Tank System

Similar to the single tank system is the split coupled tank, the equations of flows in the coupled tank can be determined where the system states here are the levels h_1 and h_2 .

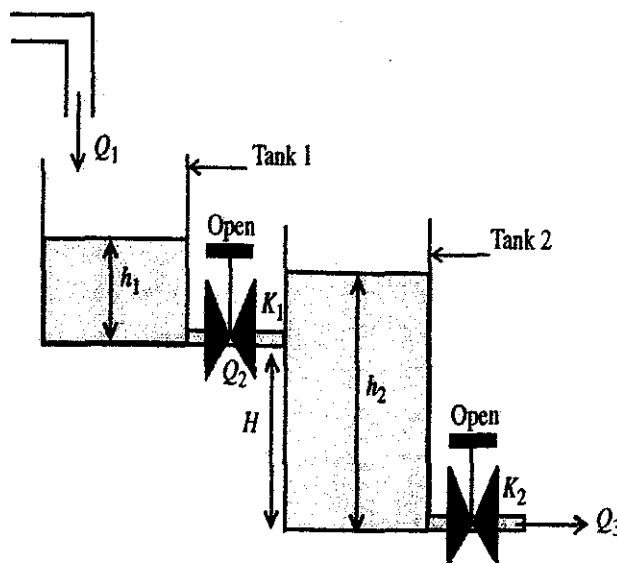


Figure 2 - Split Coupled Tank System

The flow equations of the split coupled tank are:

$$\frac{dh_1}{dt} = \frac{Q_1 - Q_2}{A_1} \text{ (Equation 3) \quad \& \quad } \frac{dh_2}{dt} = \frac{Q_2 - Q_3}{A_2} \text{ (Equation 4)}$$

Where

$$Q_2 = k_1 \sqrt{h_1}, \quad h_2 < H \text{ (Equation 5)}$$

$$Q_2 = k_1 \sqrt{(h_1 - (h_2 - H))}, \quad h_2 > H \text{ (Equation 6)}$$

Where

Q_1	Input flow into tank 1
Q_2	Output flow out of tank 1
Q_3	Output flow out of tank 2
h_1	Liquid level of tank 1
h	Liquid level of tank 2
H	Height of the split tank level
A_1	Cross section area of tank 1
A_2	Cross section area of tank 2
K_1	Valve resistance of tank 1
K_2	Valve resistance of tank 1

After applying and selecting the best PID controller to the plant which was explained in details in the previous progress report, multi-layer perceptron (MLP) will be applied as well. A paper has been reviewed, prepared by Martin T. Hagan, showing examples of neural network architectures for system identification and control. These architectures are Adaptive Inverse Control, Nonlinear Internal Model Control, Model Reference Adaptive Control, and Model Predictive Control. They are briefly explained below [2].

2.3 Adaptive Inverse Control

The adaptive algorithm receives the error between the plant output and the reference model output. The controller parameters are updated to minimize that tracking error. The basic model reference adaptive control approach can be affected by sensor noise and plant disturbances.

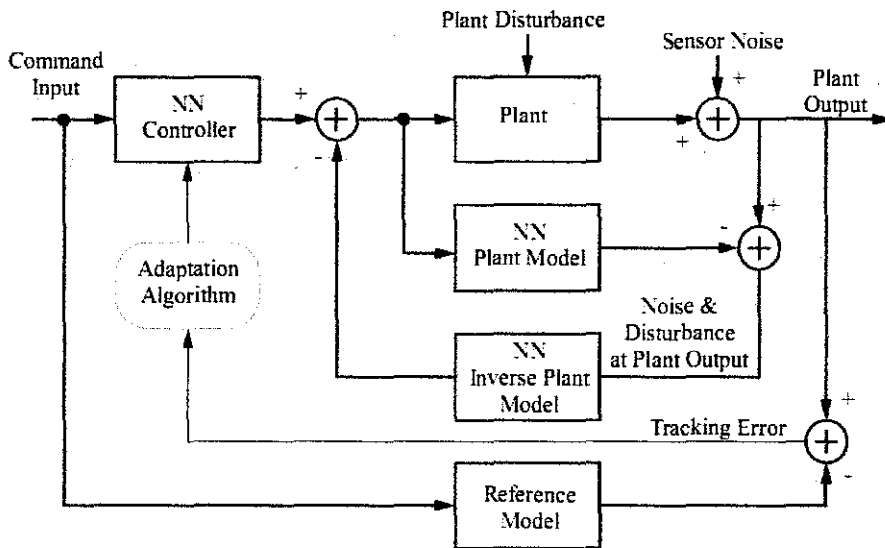


Figure 3 - Adaptive Inverse Control System

An alternative which allows cancellation of the noise and disturbances includes a neural network plant model in parallel with the plant. That model will be trained to receive the same inputs as the plant and to produce the same output.

2.4 Nonlinear Internal Model Control

Here, the neural network controller is generally trained to represent the inverse of the plant, if the inverse exists. The error between the output of the neural network plant model and the measurement of plant output is used as the feedback input to the robustness filter, which then feeds into the neural network controller.

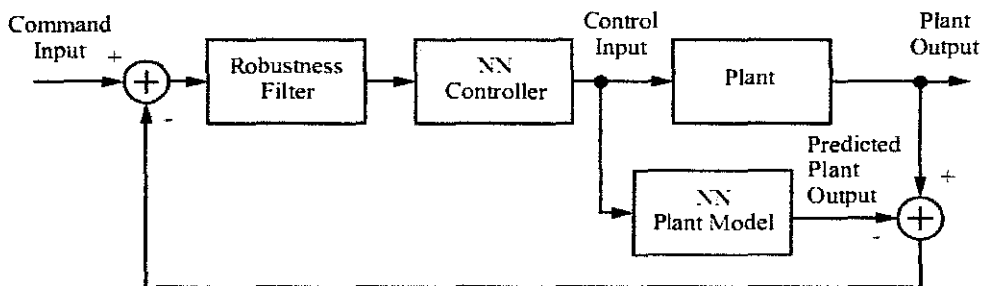


Figure 4 - Nonlinear Internal Model Control

The NN plant model and the NN controller (if it is an inverse plant model) can be trained off-line, using data collected from plant operations. The robustness filter is a first order filter whose time constant is selected to ensure closed loop stability.

2.5 Model Predictive Control

This architecture requires a neural network plant model, a neural network controller, a performance function to evaluate system responses, and an optimization procedure to select the best control input. The optimization procedure can be computationally expensive.

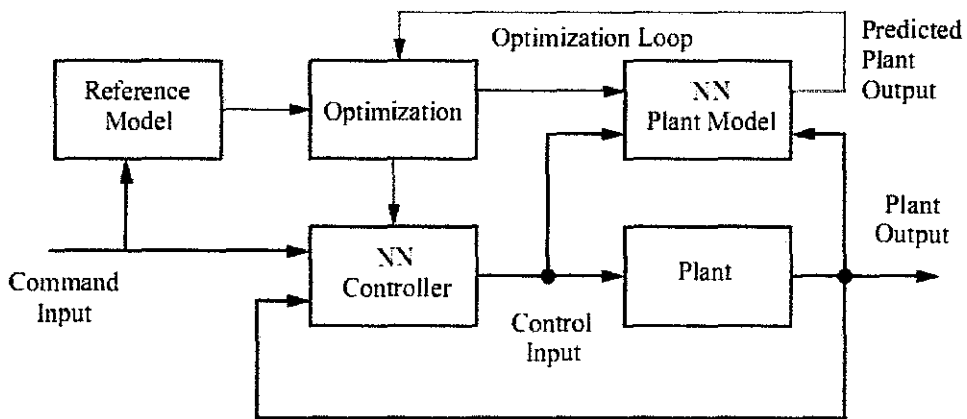


Figure 5 - Model Predictive Control

It requires a multi-step ahead calculation, in which the neural network model is used to predict the plant response. The neural network controller learns to produce the input selected by the optimization process. When training is complete, the optimization step can be completely replaced by the neural network controller.

2.6 Model Reference Control or (Neural Adaptive Control)

The model network can be trained off-line using historical plant measurements. The controller is adaptively trained to force the plant output to track a reference model output. The model network is used to predict the effect of controller changes on plant output, which allows the updating of controller parameters [3] [4].

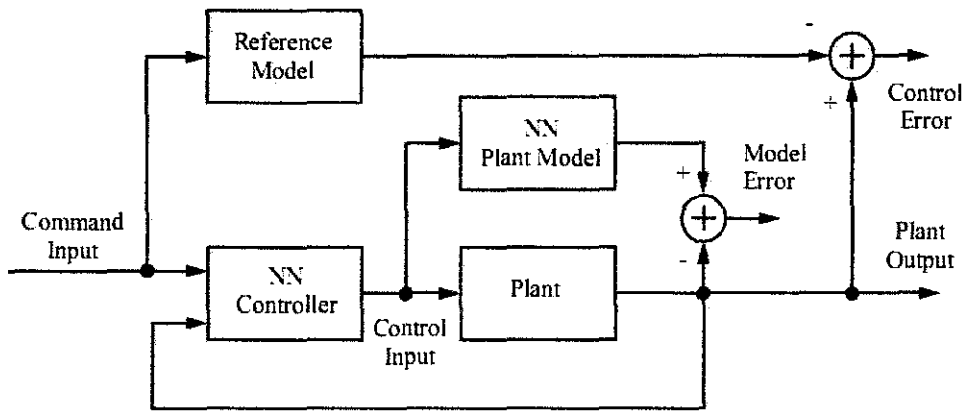


Figure 6 - Model Reference Adaptive Control

2.7 ARX Model Structure

“ARX” stands for “AutoRegressive, eXternal input”. It is considered as a black box system which can be viewed in terms of its input, output and transfer characteristics without any knowledge of its internal workings.



Figure 7 – Black Box System

Its equation is as follows:

$$Y_t = \alpha_1 Y_{t-1} + \beta_1 U_{t-1} + \varepsilon \quad (\text{Equation 7})$$

Where

U	System input
Y	System output
ε	Error measurement
α	variable values
β	variable values

A system with one input and one output is called “ARX 1” and described in the above equation. A system with two inputs and two outputs is called “ARX 2” with the following equation.

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \beta_1 U_{t-1} + \beta_2 U_{t-2} + \varepsilon \quad (\text{Equation 8})$$

2.8 Artificial Neural Network

An artificial neural network (ANN) is a computer program that can recognize patterns in a given collection of data and produce a model for that data. It is the same like the brain in where:

- It acquires knowledge by the network through learning. (trial and error)
- It uses of synaptic weight to store the knowledge.

One of the applications of “ANN” is the function approximation where it consists of process modeling, process control and data modeling [3].

The most common neural network model is the multi-layer perceptron (MLP). This type of network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown.

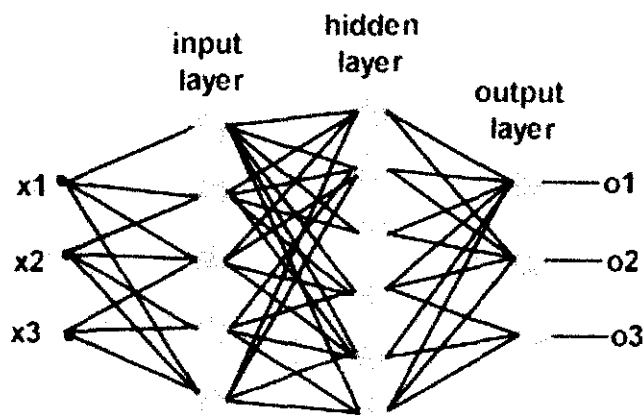


Figure 8 - Graphical Presentation of an MLP

The MLP learn using an algorithm called “back-propagation” where the output of the neural network is compared to the desired output and an error is computed. This error is then fed back to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is called “training”.

The MLP is used in process modeling and control by creating a neural network model for a physical plant then using that model to determine the best control settings for the plant [3].

The features of using neural network are:

- a. Model unknown non-linear systems.
- b. Simplified the modeling process.
- c. Enables implementations of generic tools for control system design.

CHAPTER 3 METHODOLOGY

3.1 Procedure Identification

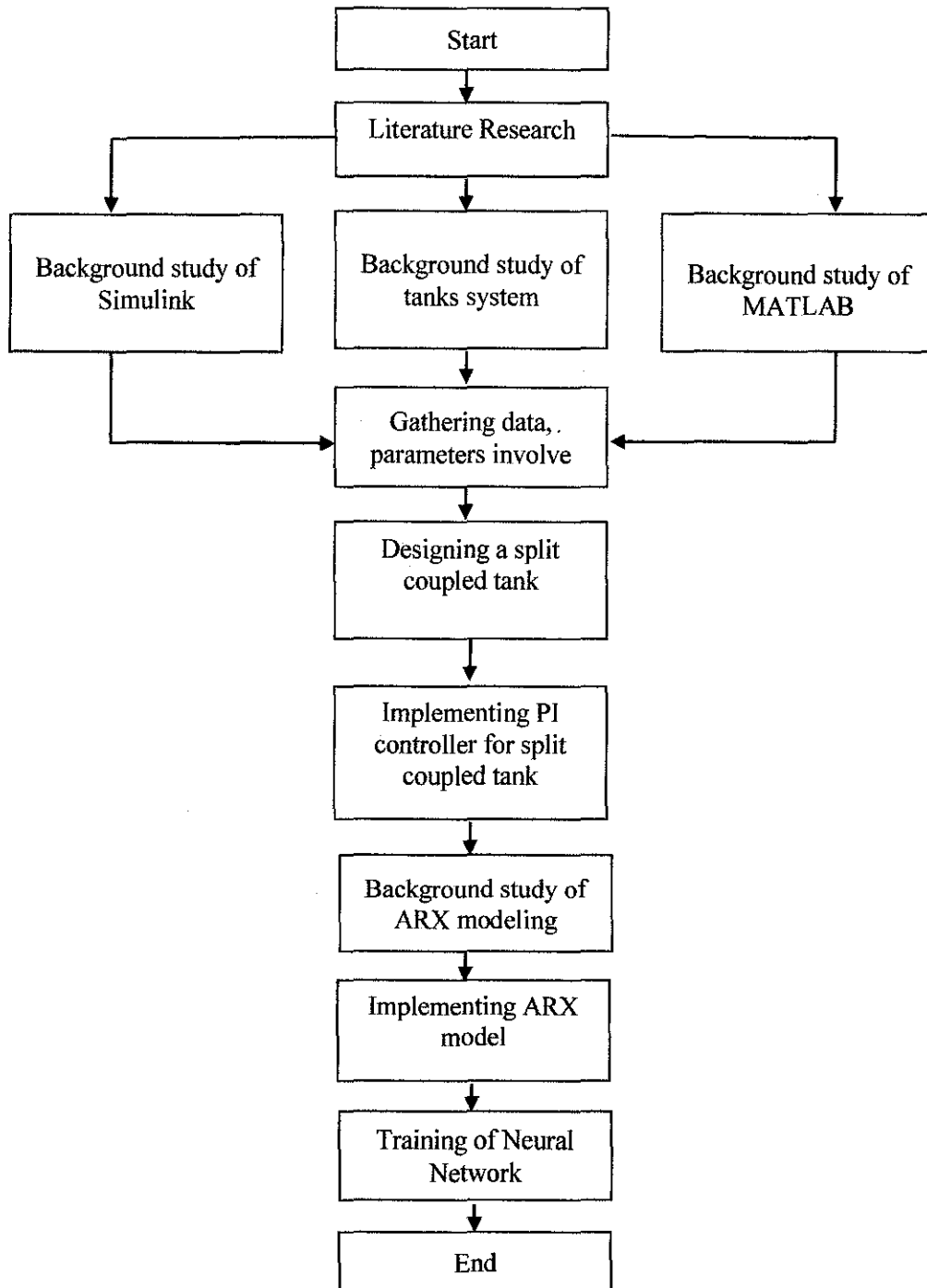


Figure 9 – Flow chart of project

3.2 Tools and Equipment

MATLAB software was used to implement the required designs. The command window, Simulink feature as well as M-file facilitated the construction of the plant, controllers and the “ARX” model. It is recommended for beginners to view the basics of Simulink before viewing the results. This is available from the MATLAB help guide with video tutorials.

CHAPTER 4

RESULTS AND DISCUSSION

The results and findings are composed of:

- 1) Design of coupled tank system using Simulink.
- 2) “Process Reaction Curve” of the coupled tank.
- 3) Designing of controller using:
 - a. Ciancone correlations for PID controller.
 - b. Cohen Coon tuning correlations.
 - c. Ziegler Nichols closed loop tuning correlations.
- 4) Choosing the best controller.
- 5) Fine tuning the selected controller.
- 6) Changing output to the level H_1
- 7) Designing of ARX model.
- 8) Training of Neural Network

4.1 Design of Coupled Tank System using Simulink

The split coupled tank is constructed from the flow equation which was mentioned earlier. The cross sectional areas (A_1, A_2) are as assumed to be 6 cm^2 while the valve resistances (k_1, k_2) equals $1 \text{ cm}^{5/2}/\text{s}$.

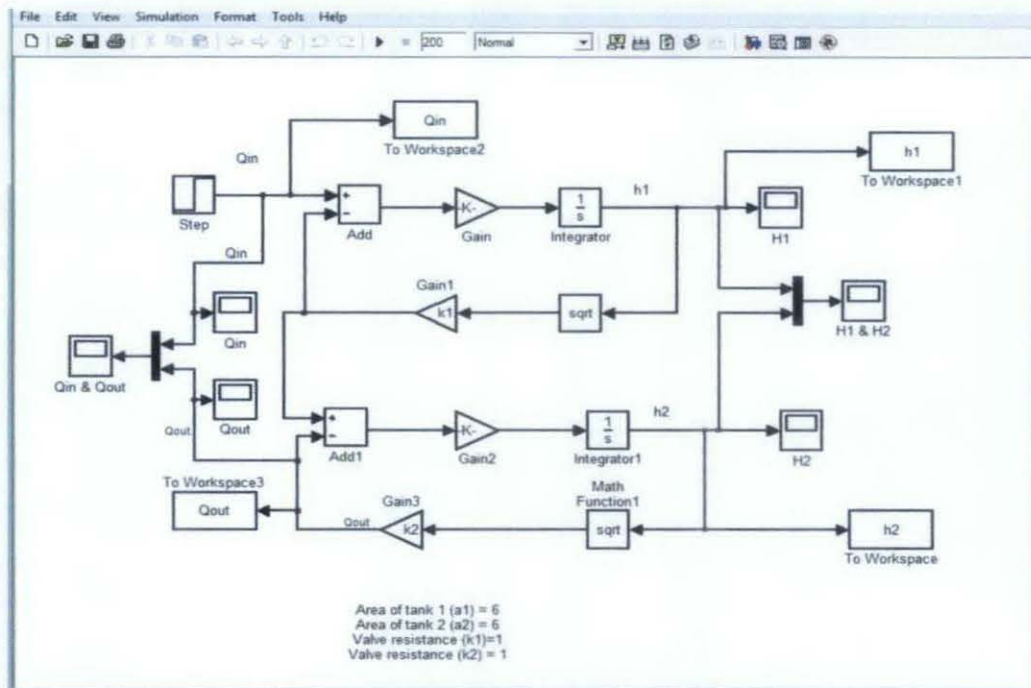


Figure 10 - Interacting Couple Tank

Figure 10 shows the coupled tank with a step input (Q_{in}) and the output is considered as the output flow (Q_{out}).

4.2 Process Reaction Curve of the Plant

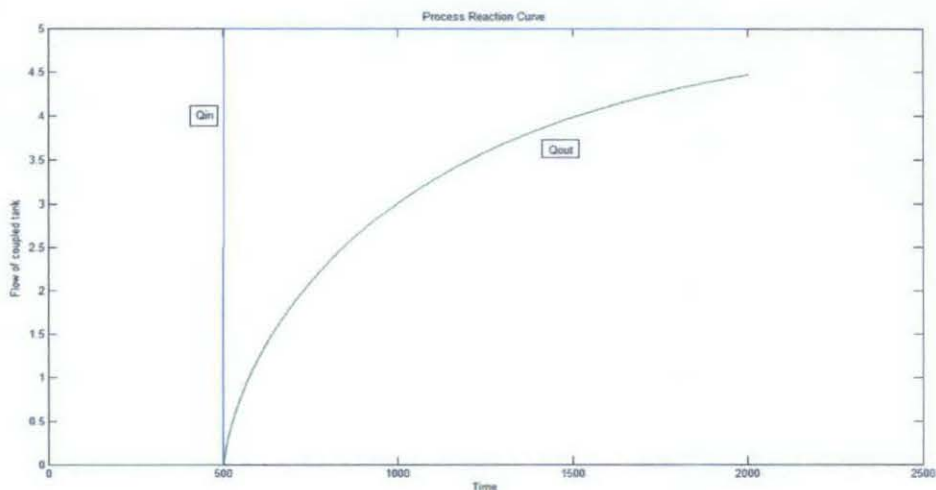


Figure 11 - Process Reaction Curve

Based on the relationship between the input and the output of the coupled tank shown in figure 5, some parameters have been calculated using it such as process

gain (K_c), dead time (θ) and time constant (τ). These parameters helped in determining the suitable controllers for the tank. This is also called “Process Reaction Curve”.

Three different approaches have been used to determine the suitable controllers which are Ciancone correlations with PID controller, Cohen con tuning correlations and Ziegler Nichols closed loop tuning correlations.

4.3 Designing of Controller

This section shows how the controller parameters have been estimated by three different methods which are:

4.3.1 Ciancone Correlations with PID Controller

The proportional gain, the integral and derivative time are calculated from the parameters of the process reaction curve. The controller block is connected in series with the plant block as shown in the figure below.

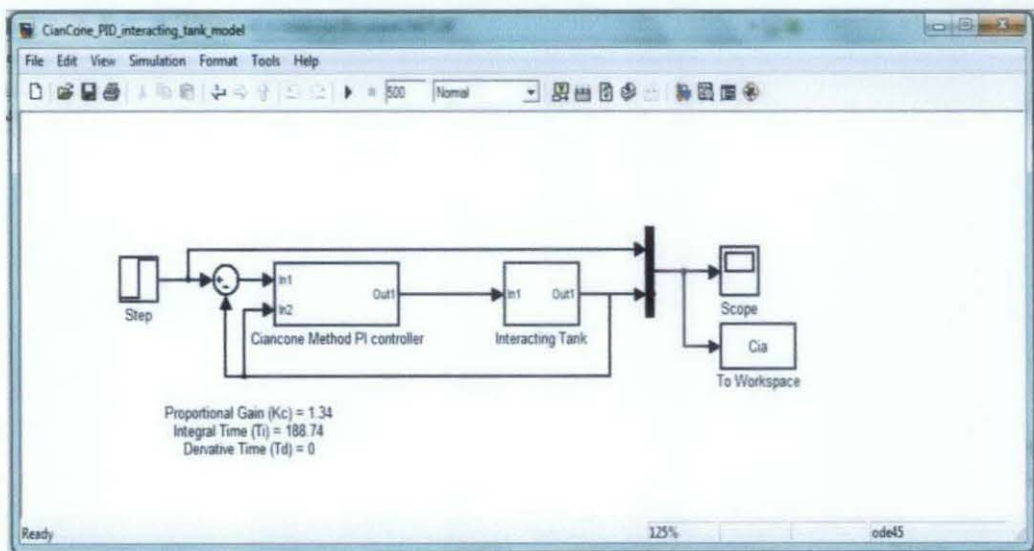


Figure 12 - Ciancone PID Block

The Ciancone block is constructed from the PID formula where:

$$MV(t) = K_c \left[E(t) + \frac{1}{T_i} \int_0^{\infty} E(t') dt' - T_d \frac{dCV(t)}{dt} \right] + I \quad (\text{Equation 9})$$

These adjustable parameters are called **tuning constants**

The Ciancone block consists of the PID formula and the values of K_c , T_i and T_d are added as shown below.

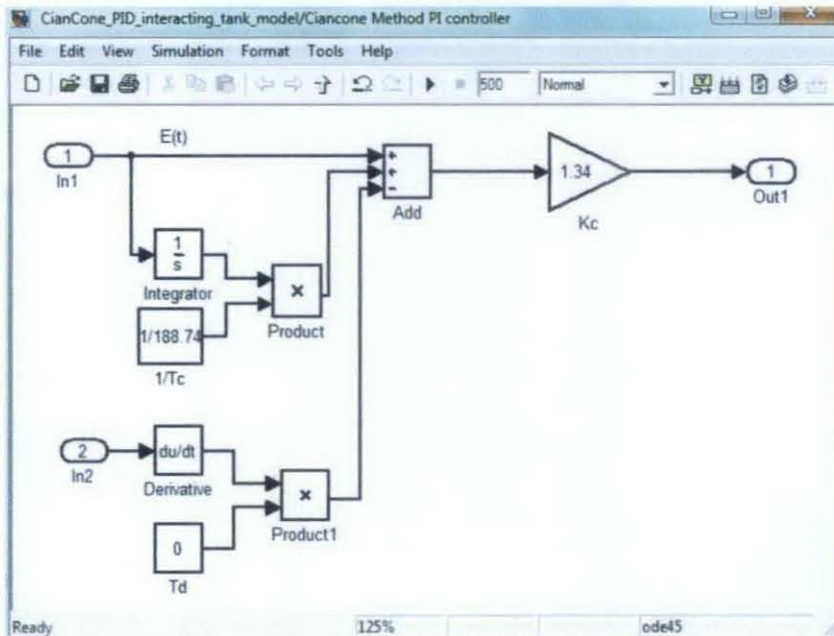


Figure 13 - Ciancone PID Block

The response of the levels with the Ciancone method showed a faster response than the original plant as well as for the input and output flow responses but the input flow experience a small overshoot.

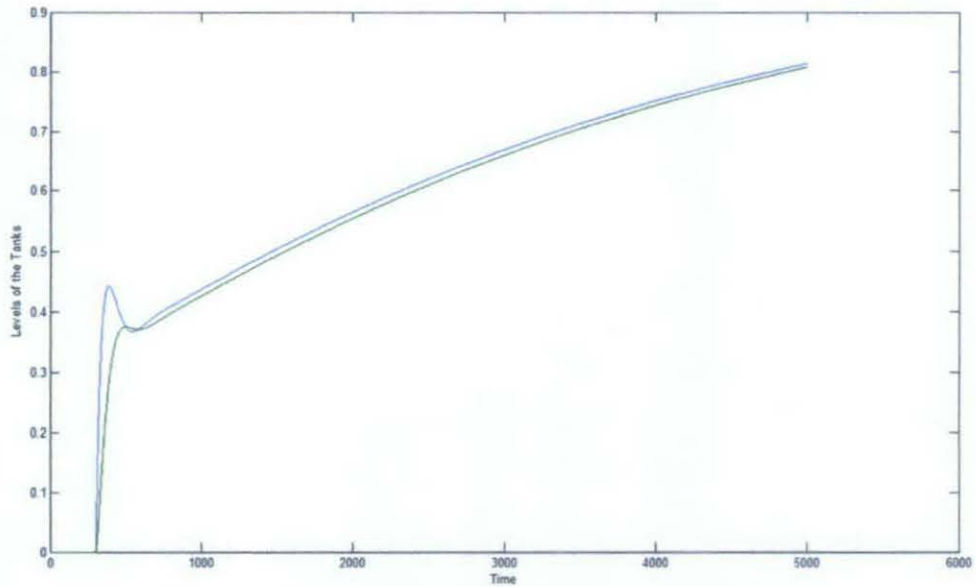


Figure 15 - Level of the tanks using Ciancone controller

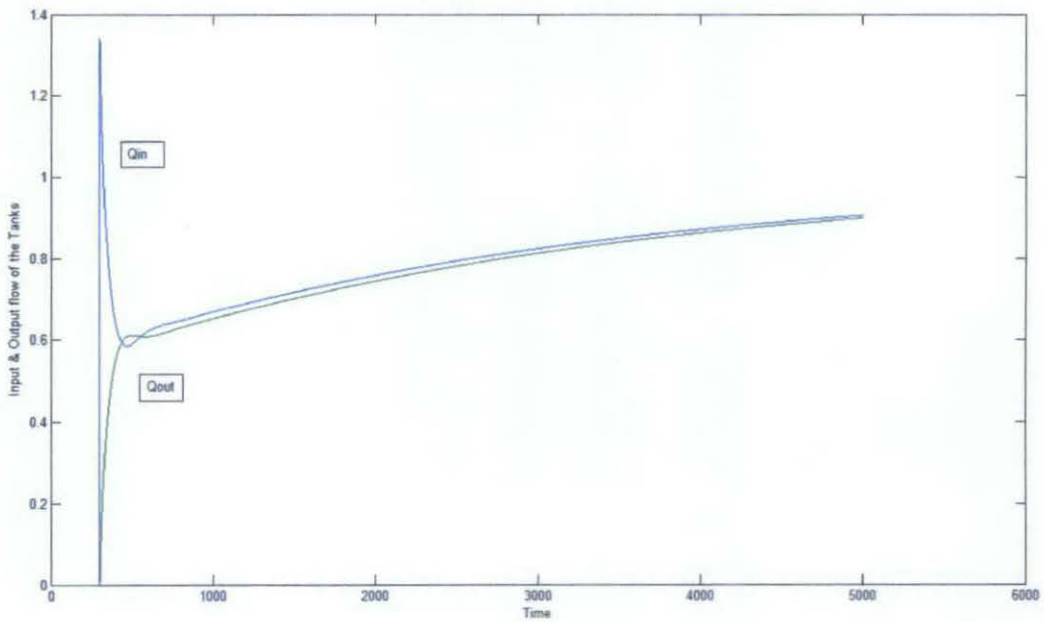


Figure 14 - Input & Output flow of Ciancone method

4.3.2 Cohen Coon Tuning Correlations

The second method which is used to calculate the tuning constants is the Cohen Coon method. The table below shows how to calculate them using the parameters obtained from the process reaction curve.

Table 1 - Cohen Coon Calculations

Controller	K_c	T_I	T_d
P-only	$\frac{1}{K_p} \frac{\tau}{\theta} \left(1 + \frac{\theta}{3\tau} \right)$	—	—
PI	$\frac{1}{K_p} \frac{\tau}{\theta} \left(0.9 + \frac{\theta}{12\tau} \right)$	$\theta \cdot \frac{\left(30 + 3 \frac{\theta}{\tau} \right)}{9 + 20 \frac{\theta}{\tau}}$	—
PID	$\frac{1}{K_p} \frac{\tau}{\theta} \left(\frac{3\theta + 16\tau}{12\tau} \right)$	$\theta \cdot \frac{\left(32 + 6 \frac{\theta}{\tau} \right)}{13 + 8 \frac{\theta}{\tau}}$	$\frac{4\theta}{11 + 2 \frac{\theta}{\tau}}$

Similar to the Ciancone method, a Simulink block was constructed for the P, PI and PID controllers as shown.

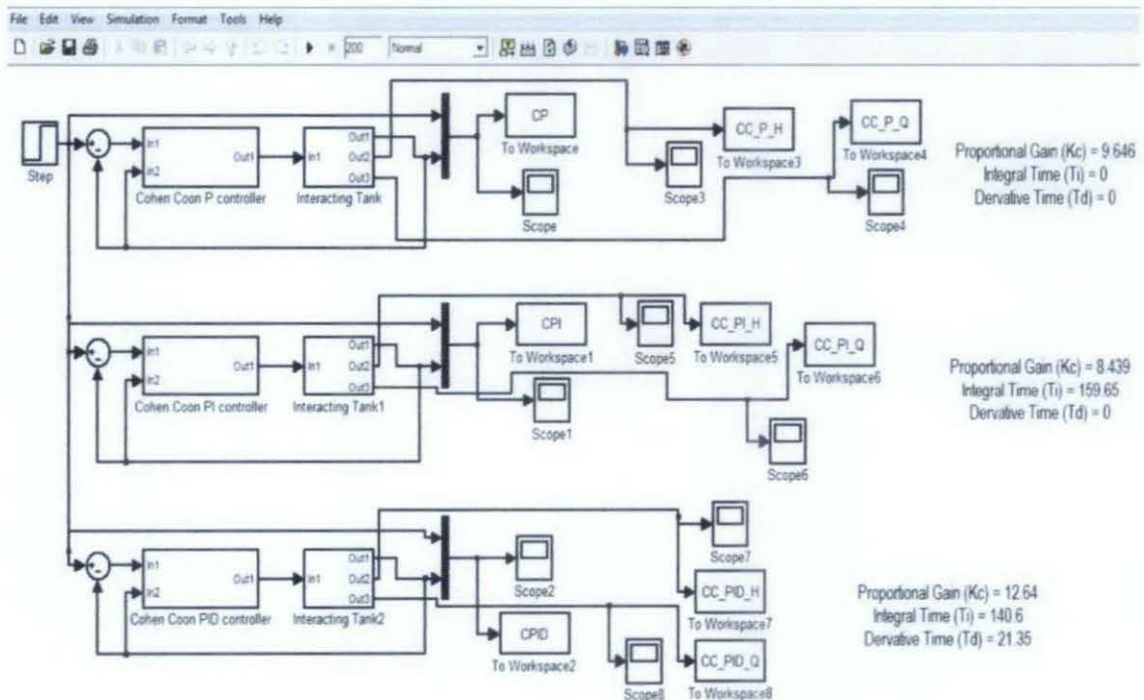


Figure 16 - Simulink Block for Cohen Coon Method

Since the PI and PID controllers showed better responses than the P controller, they will be discussed and analyzed.

PI Controller

The levels for the PI controller show better response than the Ciancone method and better settling time. The flow graph has also a fast response but Q_{in} experience some overshoot at the beginning. This can be observed through the figures 11 and 12.

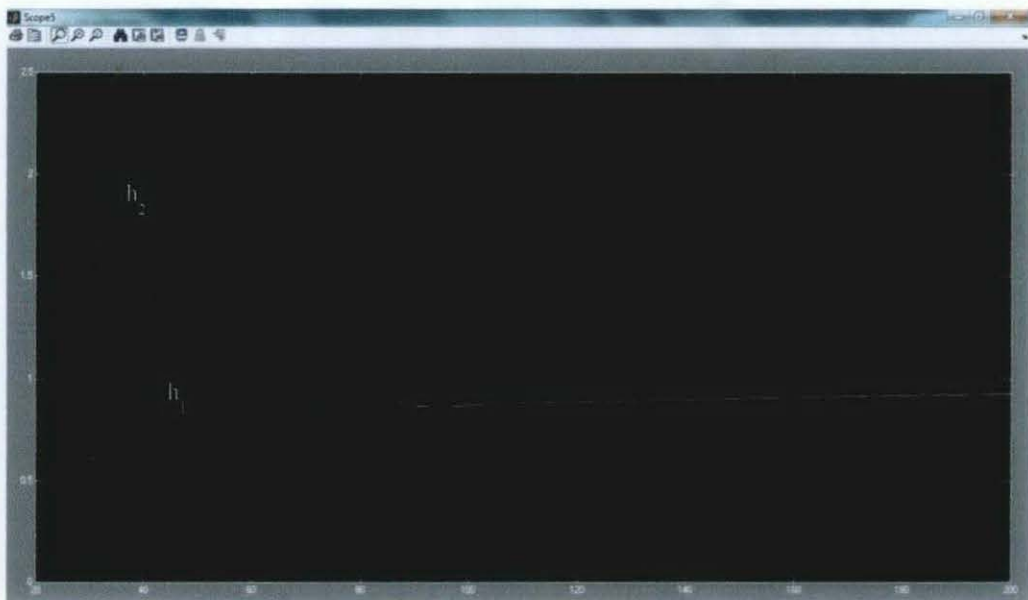


Figure 17 - Level of the tanks of PI controller

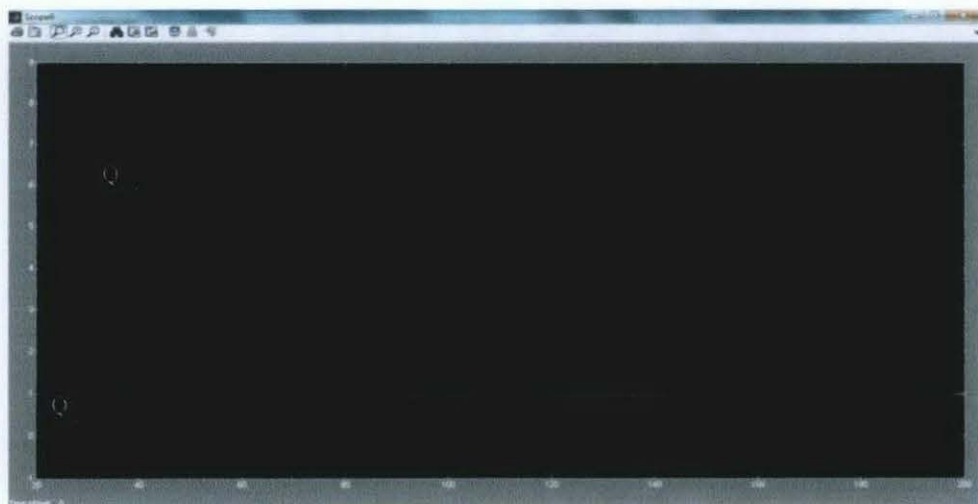


Figure 18 - Flow of the PI Controller

PID Controller

In case of the PID controller, the levels have slower response than the PI controller while the input flow in this controller is very noisy and unstable which is unacceptable. The graphs for the levels and flow are shown as well.



Figure 20 - Levels for PID Controller

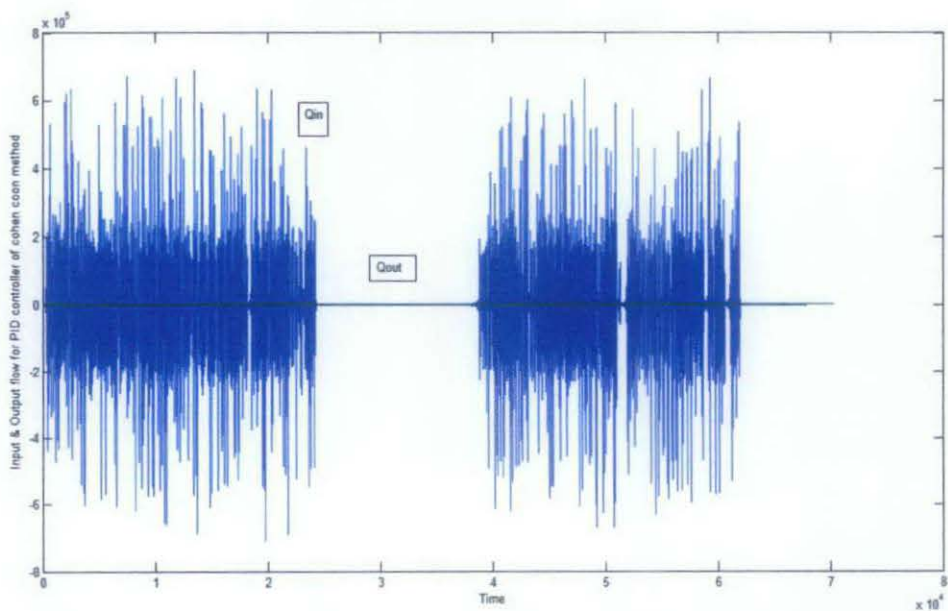


Figure 21 - Input & Output flow for the PID Controller

4.3.3 Ziegler Nicholas Closed Loop Method

The third method which is used is the Ziegler Nicholas. The calculation of the tuning parameters in this case doesn't depend on the process reaction curve like the previous two method, it is derived from the bode plot of the transfer function which is calculated from the coupled tanks.

In the figure below shows the bode plot of the plant. To ensure the stability of the system, we assume the phase degree to be -180 , from that we can calculate the critical frequency (ω_c) as well as the magnitude in decibel (AR_C).

The ultimate gain (K_u) and the ultimate period (P_u) are then calculated using the following formulas:

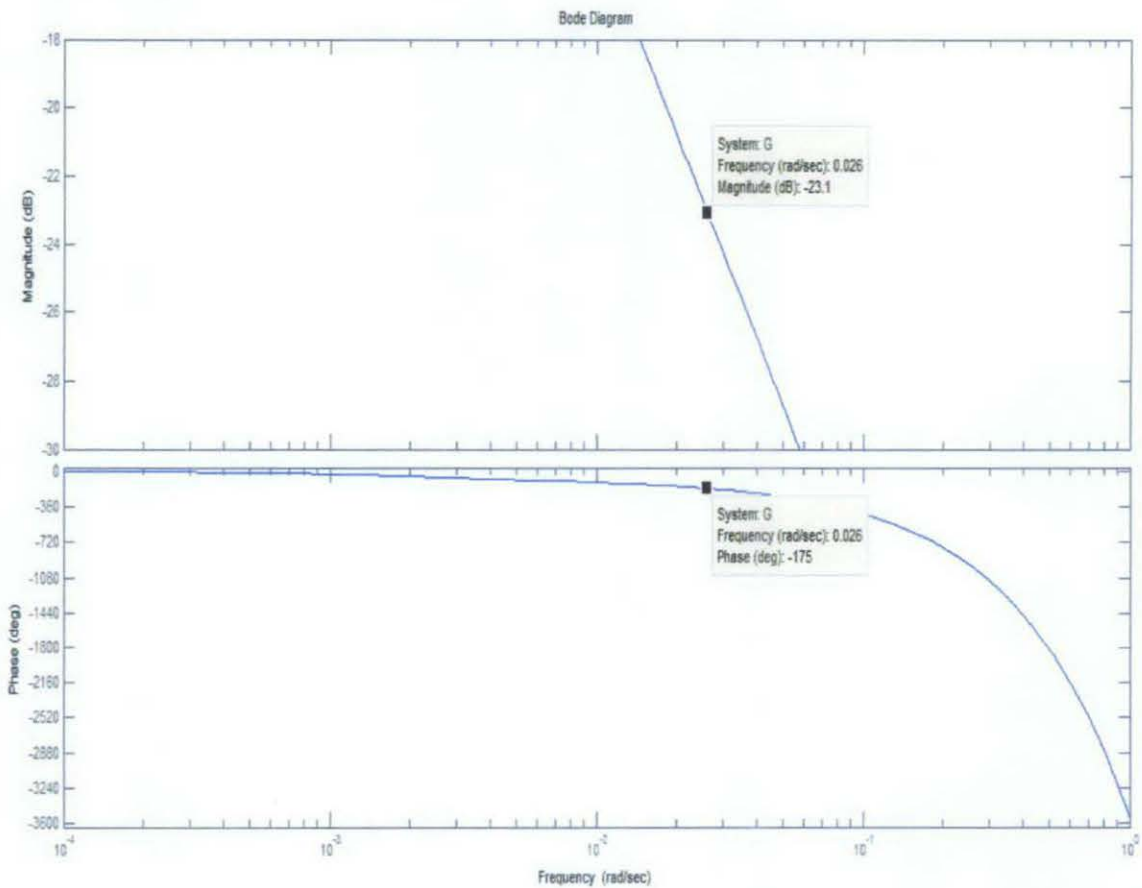


Figure 22 - Bode Plot of the Coupled Tank System

$$K_u = \frac{1}{|G_{ol}(j\omega_c)|} = \frac{1}{AR_c} \quad (\text{Equation 10})$$

$$P_u = \frac{2\pi}{\omega_c} \quad (\text{Equation 11})$$

The P, PI and PID are then calculated using the table below.

The simulink block is then constructed for the three controllers as shown in the previous methods. Since the performance of the PI and PID controllers showed better results than the P controller, they will be discussed only.

PI Controller

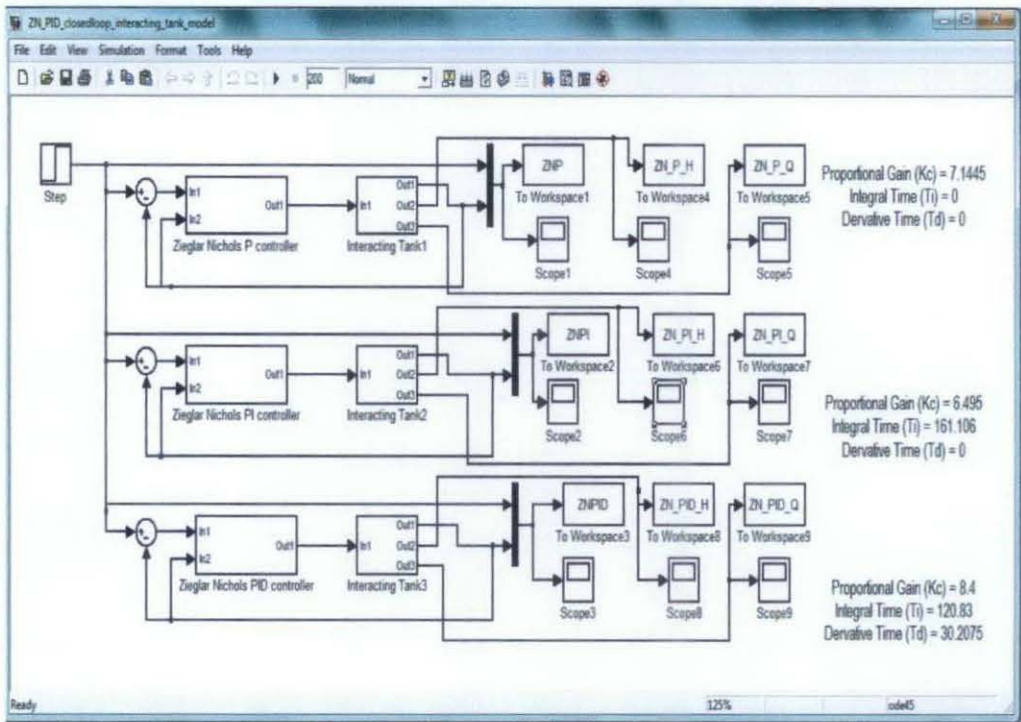


Figure 23 - Ziegler Nichols Closed Loop Method

The graphs for the levels of the PI controller show similarities with the PI of the Cohen Coon method but its responses are slightly better. For the flow, the graphs are typical from the Cohen Coon method.

Table 2 - Ziegler Nicholas Closed Loop Calculations

Controller	K_c	T_I	T_d
P-only	$K_u/2$	—	—
PI	$K_u/2.2$	$P_u/1.2$	—
PID	$K_u/1.7$	$P_u/2.0$	$P_u/8$

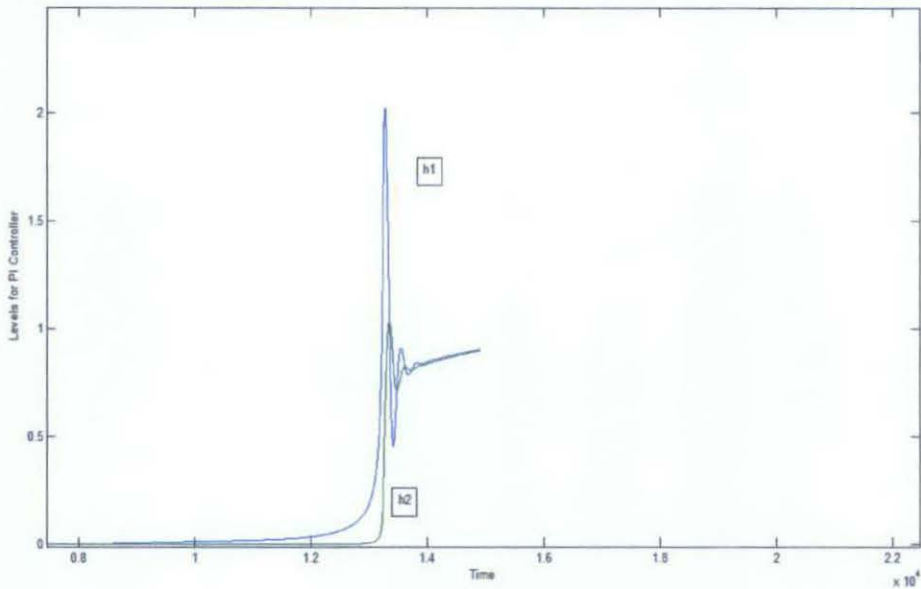


Figure 24 - Levels for the PI controller

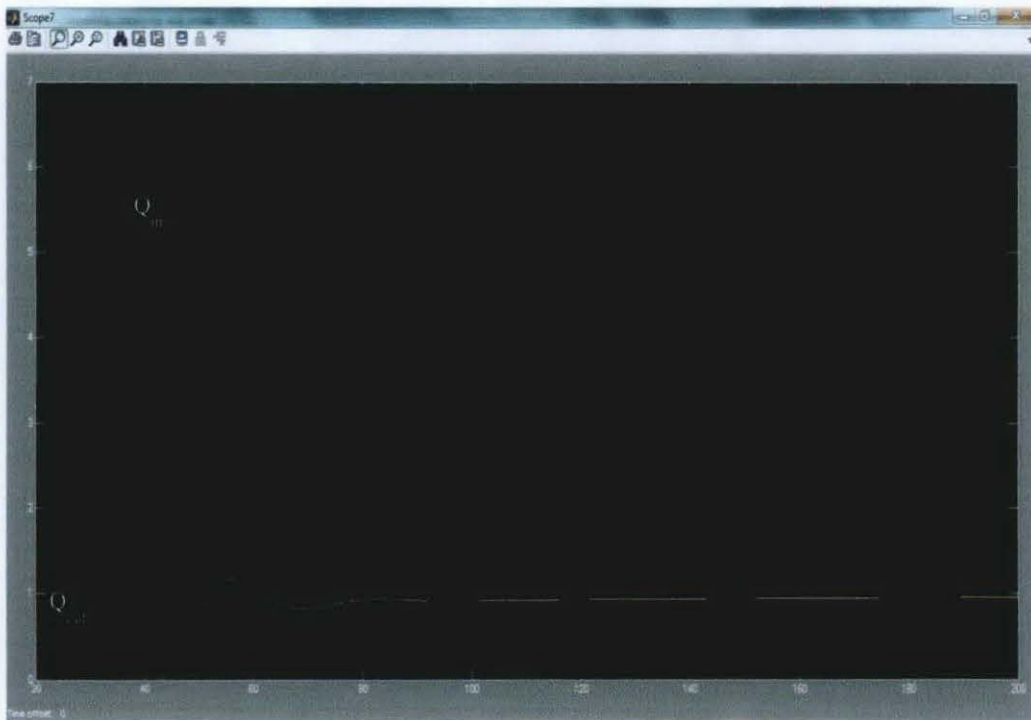


Figure 25 - Flow for PI controller

PID Controller

The PID controller shows the exact typical response with the Cohen Coon method for the levels as well as for the flow.

After discussing the performance of the controllers, the best controller is selected based on fast response, good settling time and low overshoot.

Table 3 - Comparison of performance between the three methods

	Ciancone Correlation Tuning Method	Cohen Coon Tuning Method	Ziegler Nichols Closed Loop Method
PI Controller	Q_{in} has a big overshoot which can damage the valve while the levels & flow keep on increasing with no settling.	H_1 & Q_{in} have a small overshoot at the beginning. The flow has less settling time compared to ZN closed loop method.	Same like Cohen Coon. Slightly better response. The overshoot of H_1 is smaller than Cohen Coon method.
PID Controller	-	Q_{in} is noisy and unstable which is unacceptable.	Same like Cohen Coon method.

The table here summarizes the advantages and disadvantages of the controller used for the three methods. As shown, the Ciancone method has only PI controller where the other methods have PI and PID controllers.

After analyzing table 3, it was found that the PI controller of the Ziegler Nichols closed loop method is best controller with good performance.

h_1 and Q_{in} are observed to experience an overshoot in the controller so fine tuning was applied to the controller to enhance its performance.

4.4 Fine Tuning

Fine tuning is applied the PI controller of the Ziegler Nichols closed loop.

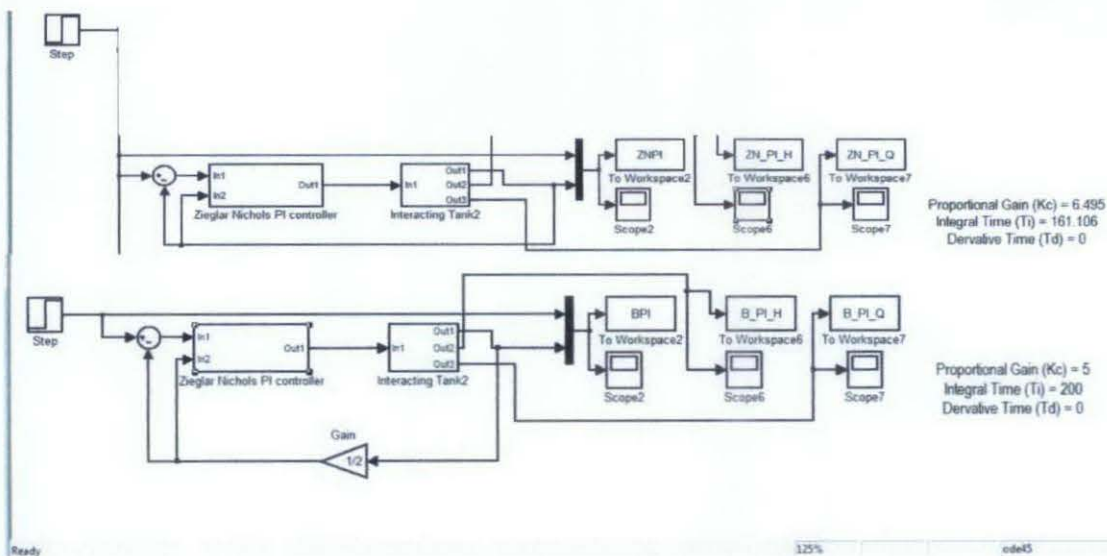


Figure 26 - Comparison between Old & Modified PI Controller using ZN Closed Loop

To overcome the overshoots of h_1 and Q_{in} , the tuning constants values are changed. Since MV is related to Q_{in} based on the PID formula in Equation 1 above, so to decrease Q_{in} MV is decreased and from the equation K_c is also decreased while T_i is increased.

In the figure 25, these changes can be seen where the K_c is decreased and T_i is increased. Also a gain of half has been added to get the maximum tuning from this controller.

A comparison between the levels of the old & modified PI controller has been made. As observed, the overshoot of h_1 has been decreased.

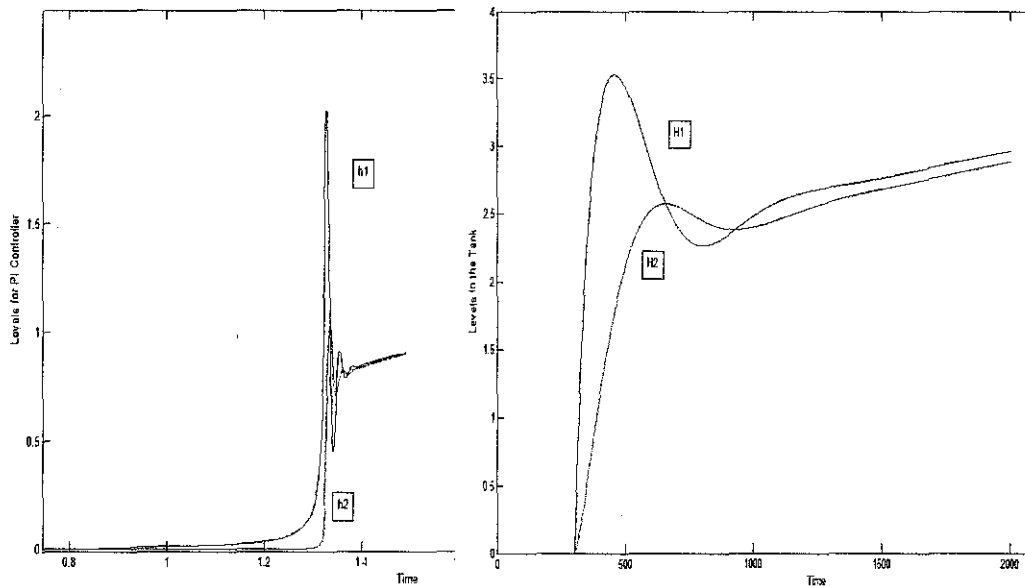


Figure 27 - Comparison between the levels of the Old & Modified PI Controller

In case of the flow, the overshoot of Q_{in} has decreased a bit while the settling time has increased as shown in the figure below [3] [4].

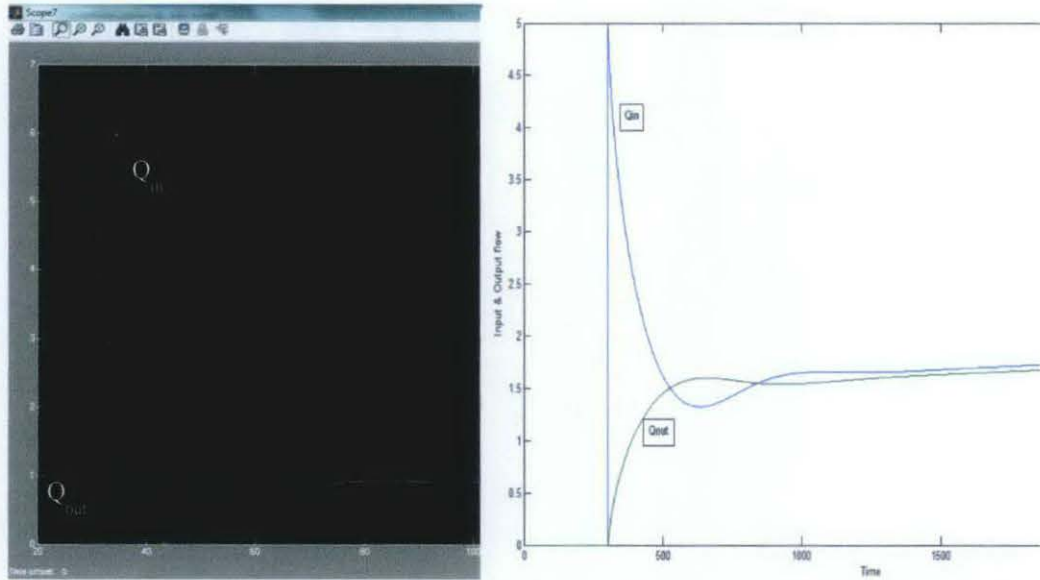


Figure 28 - Comparison between the flows of the Old & Modified PI Controller

4.5 Changing the Output to the Level H_1

All the previous results which were shown earlier, the output was considered as output flow of the second tank of the plant. So to observe how the controller is effective on the plant the output has been changed from the flow output (Q_{out}) to the level of the first tank (H_1) as seen in the figure below.

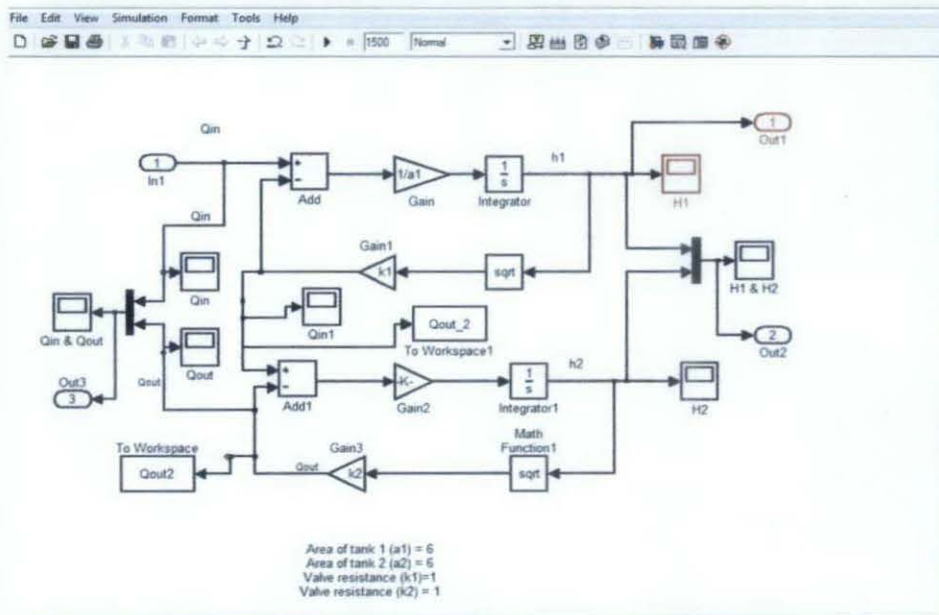


Figure 29 - The plant with the new output (H_1)

The Ziegler Nicholas PI Controller was used with new output different tuning constants (K_c , T_i , T_d) according the new output. The Controller was selected according to the guidelines of controlled modes in table 4.

As shown in the table, the flow and the level always are controlled well by a PI controller and rarely with PD or PID controllers, so because the Ziegler Nicholas PI controller suited best for the output flow then it is assumed the same with the level of the first tank.

Table 4 - Guidelines for Selecting Controller Modes

Control Loop	Controller Mode		
	Proportional	Integral	Derivative
Flow	Always	Usually	Never
Level	Always	Usually	Rarely
Temperature	Always	Usually	Usually
Analytical	Always	Usually	Sometimes
Pressure	Always	Usually	Sometimes

After considering the level as the new output, random variable input functions were used such as sine waves, pulse generators and multiple step variables with different step time and final time as shown in the figure below.

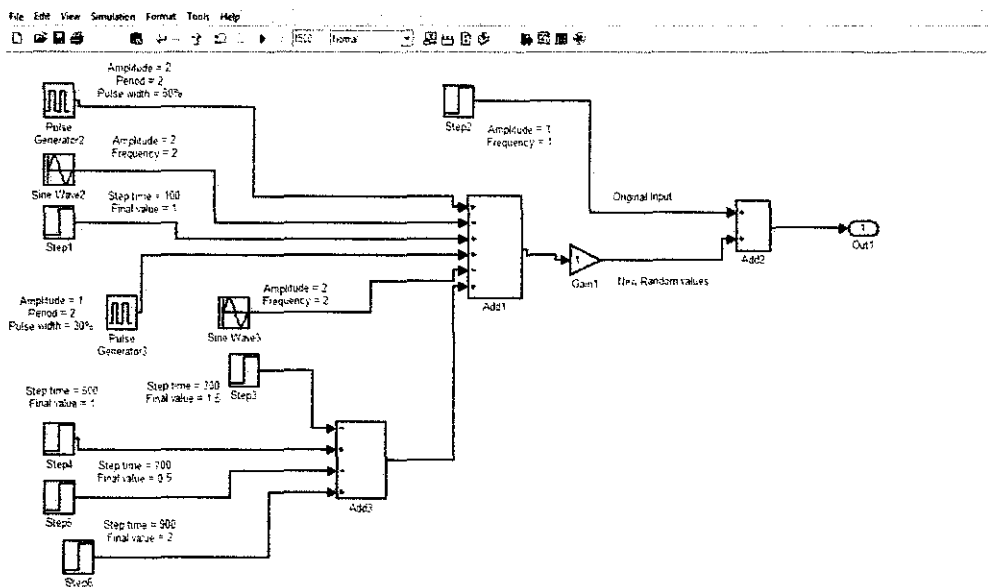


Figure 30 - Random Input Variables block for New Output

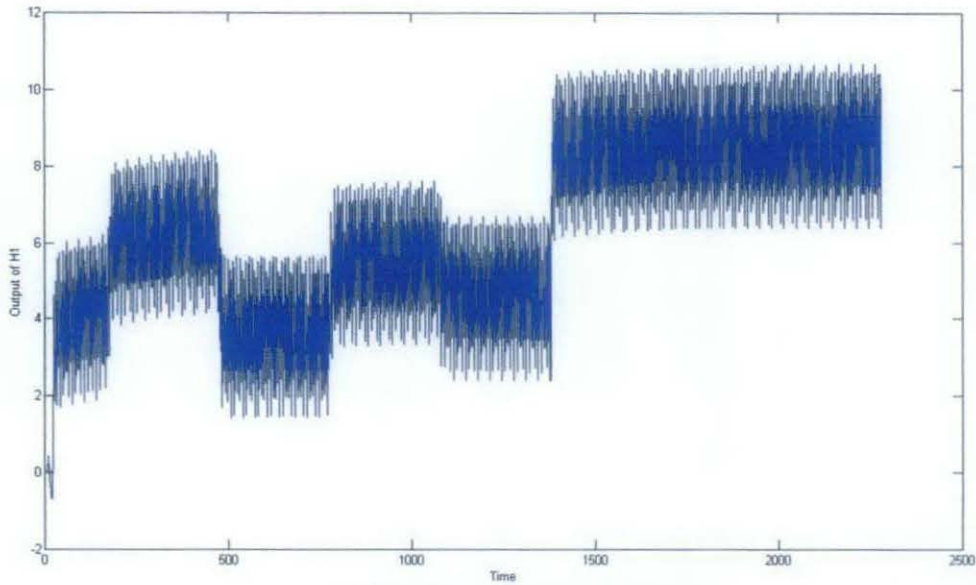


Figure 31 – Input's graph

The graph of the output level after implementing the PI controller is shown in the figure below.

4.6 Designing of ARX model

The ARX model has been chosen from the rest of the models which were mentioned in the literature review. An M-file was made for each ARX (1, 2 &3) to compare and see which one gives more accuracy.

The M-files are used to get the values of alpha's and beta's which are then used in the ARX model. As shown below, the M-file of ARX (1) gives one alpha and one beta while ARX (2) gives two alpha and two beta and so on. The least error among them is ARX (3) with an error equals 0.2697.

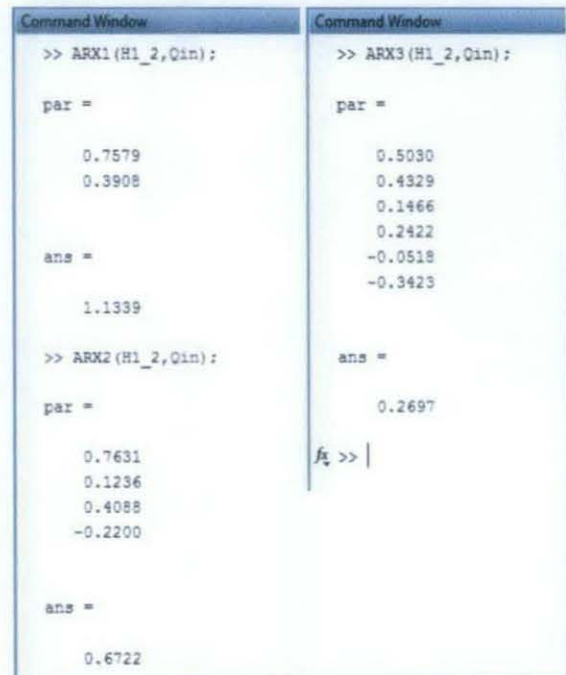


Figure 32 - The values of alphas & betas for different ARXs

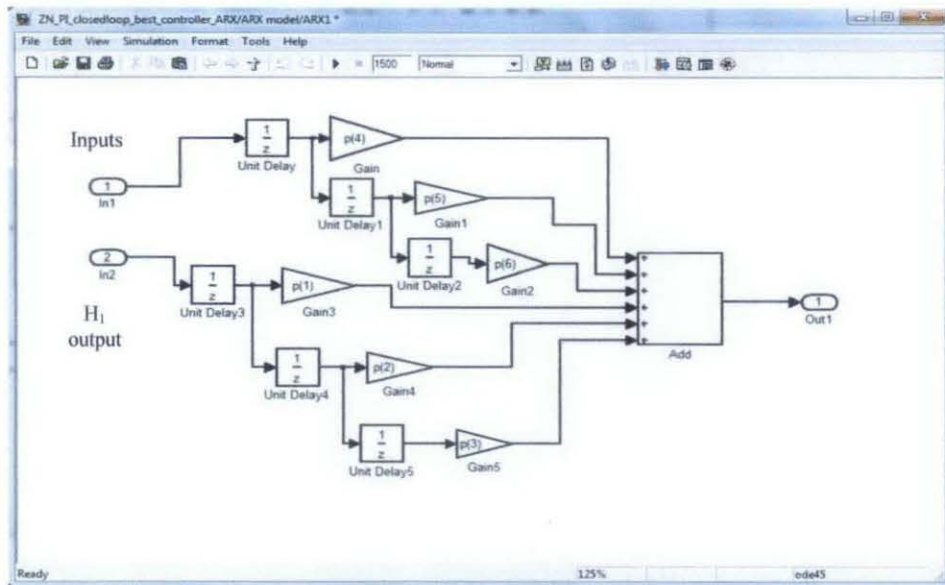


Figure 33 - ARX model

The M-files of the three models can be seen in Appendix [E].

The values of ARX (3) were used in the ARX model which was constructed from the equation:

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \alpha_3 Y_{t-3} + \beta_1 U_{t-1} + \beta_2 U_{t-2} + \beta_3 U_{t-3} + \varepsilon$$

The values of 'p' are the values of alpha's and beta's from the ARX (3) M-file. The inputs of the ARX model are input flow (Q_{in}) and the level H_1 without the controller.

A graph is made between the level H_1 of the plant (without the controller), with the PI controller and with the ARX model.

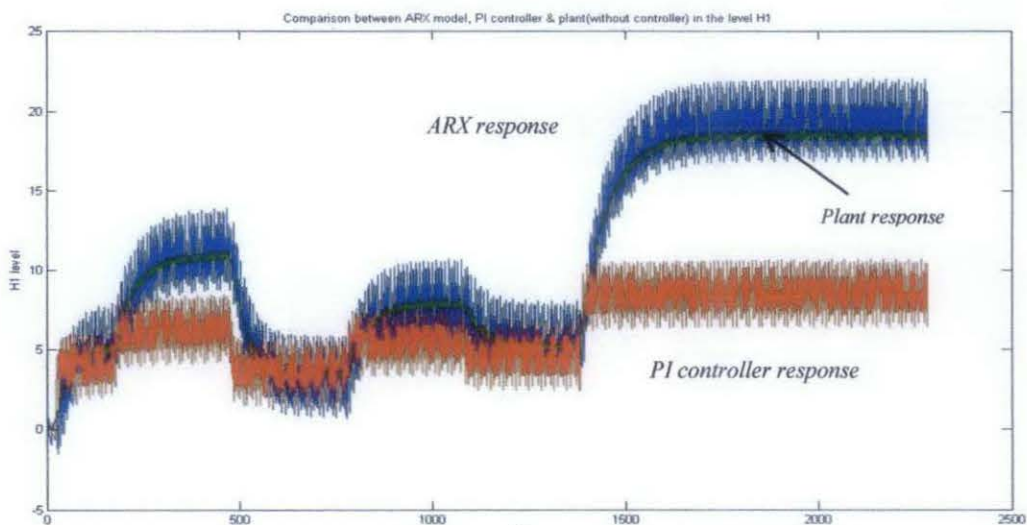


Figure 34 - a graph between ARX model (Blue), PI controller (Red) & plant without controller (Green) in the level H_1

It is clear that the ARX is learning based on the input values. The ARX output is then subtracted from the input flow (Q_{in}) and fed to the PI controller as an input as shown in figure 33.

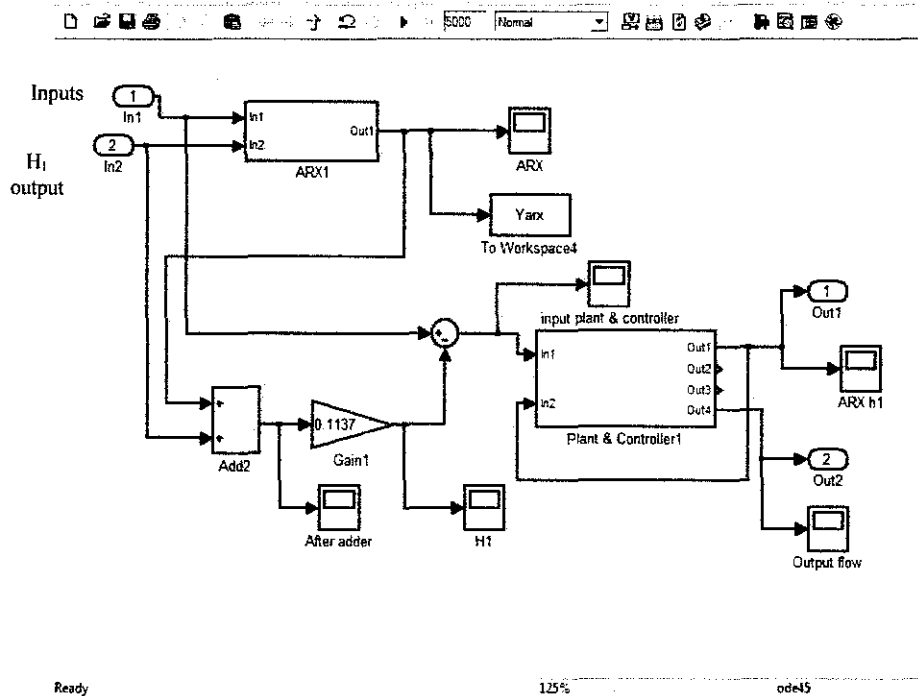


Figure 35 - ARX Model as an Input to the PI controller

The input of the controlled was fine tuned in the form of a gain block as seen in the figure. The value 0.1137 was estimated based on trial and error to make the final level H_1 stable as well as the output flow (Q_{out}) which is the prime objective of the whole project. The gain value is considered a very sensitive value where any other values tend to make the final Output H_1 unstable (either increasing sharply or decreasing).

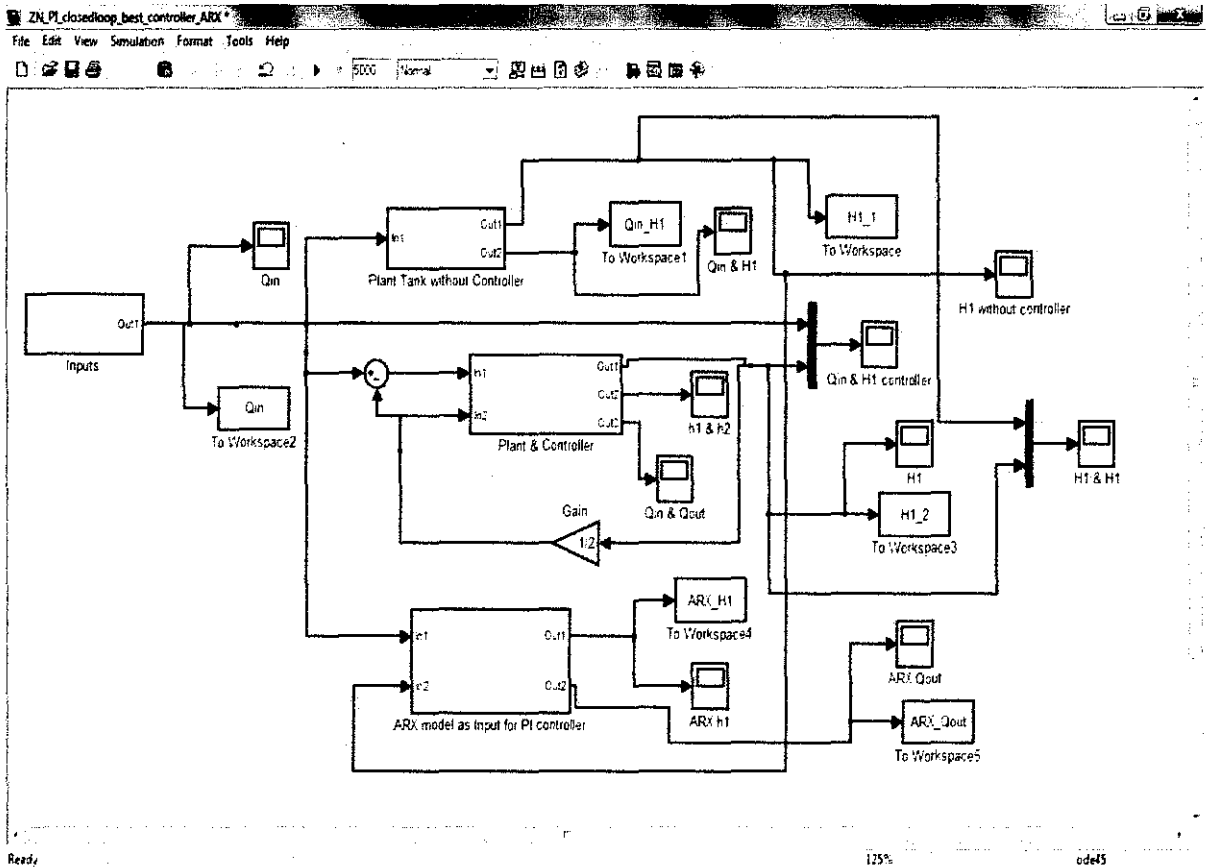


Figure 36 - The Whole System with plant, PI Controller & ARX Model

The ARX model which is shown above is made in a subsystem which is clear in figure 34.

The random inputs which were mentioned earlier are feeding the plant above without any controller as well as the plant with controller and the ARX model where the ARX model is also an input for the same plant with controller.

A comparison is made between the graphs of the level H_1 without controller, H_1 with controller and H_1 of the controller with ARX model as an input (figure 33).

As seen in figure 35, the level in case of PI controller and without controller is stable and doesn't exert rapid changes compared to the level of ARX. It takes some time until it is finally stable.

The stability of the level H_1 from the ARX model has also affected the output flow (Q_{out}) which is the main output. The ARX made also the output flow stable. Another comparison is made between the flow outputs of each system.

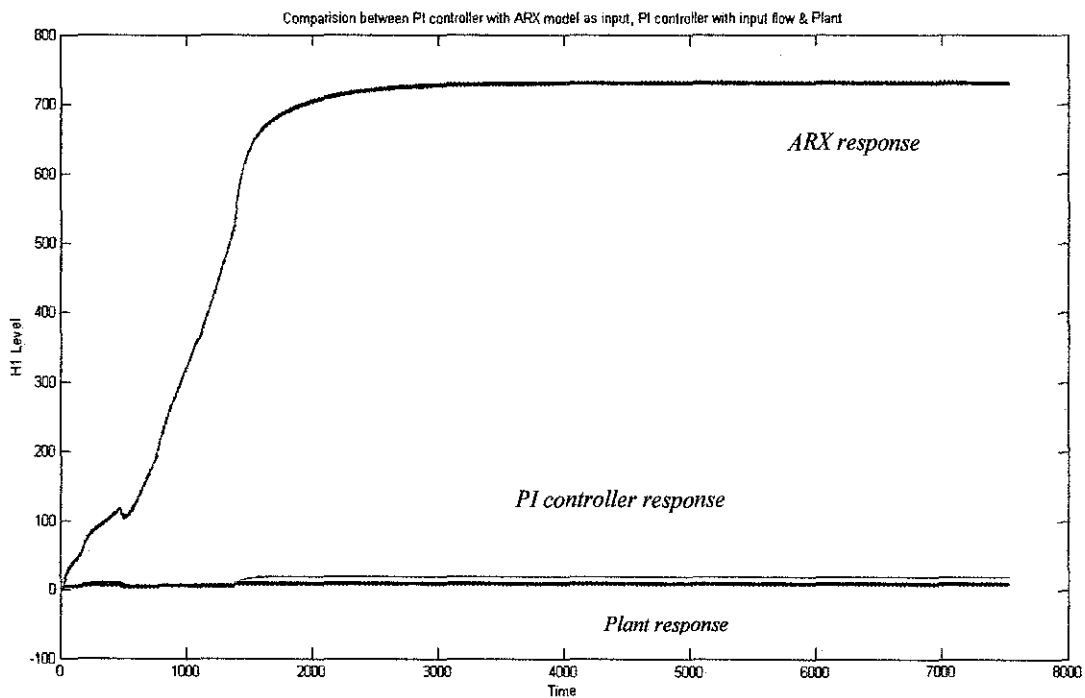


Figure 37 – a graph of level H1 between PI controller with ARX model as input (Blue), PI controller with input flow (Red) & the Plant (Green)

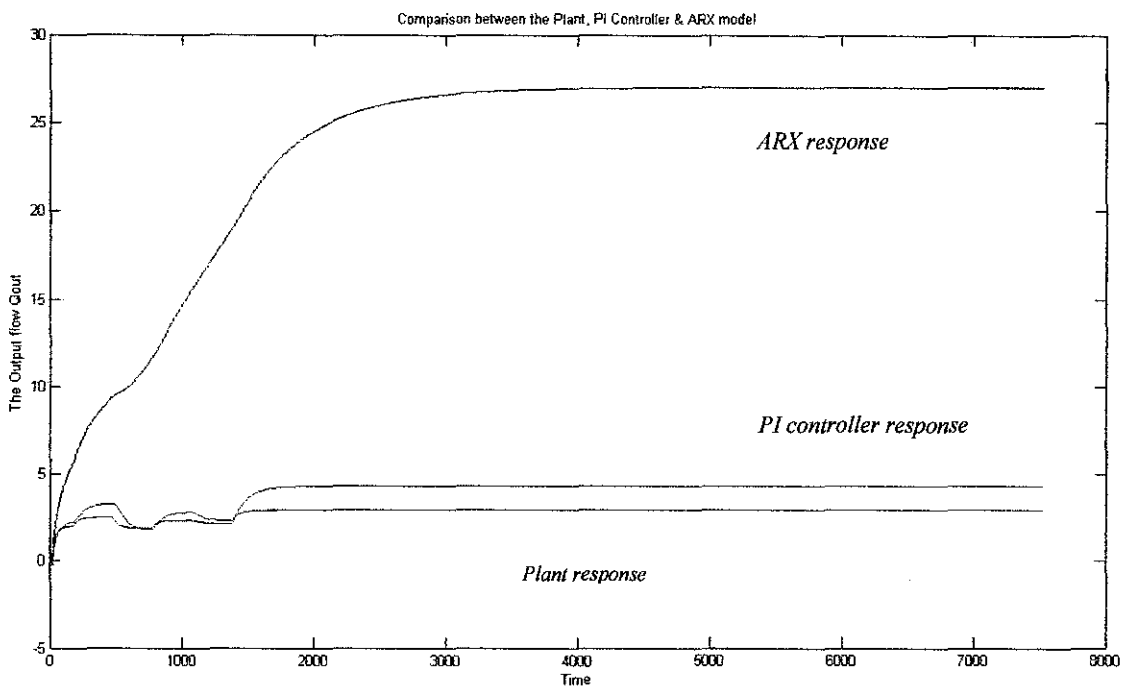


Figure 38 – a graph of Output flow (Q_{out}) between PI controller with ARX model as input (Blue), PI controller with input flow (Red) & the Plant (Green)

The output flow follows the same path of the level H_1 with the three systems.

4.7 Training of Neural Network

The neural network has been used to compare between its performance and the PI controller. To train a network on “MATLAB”, the command “nftool” is entered at the command window and the neural network fitting tool window will come out as shown in the figure below.

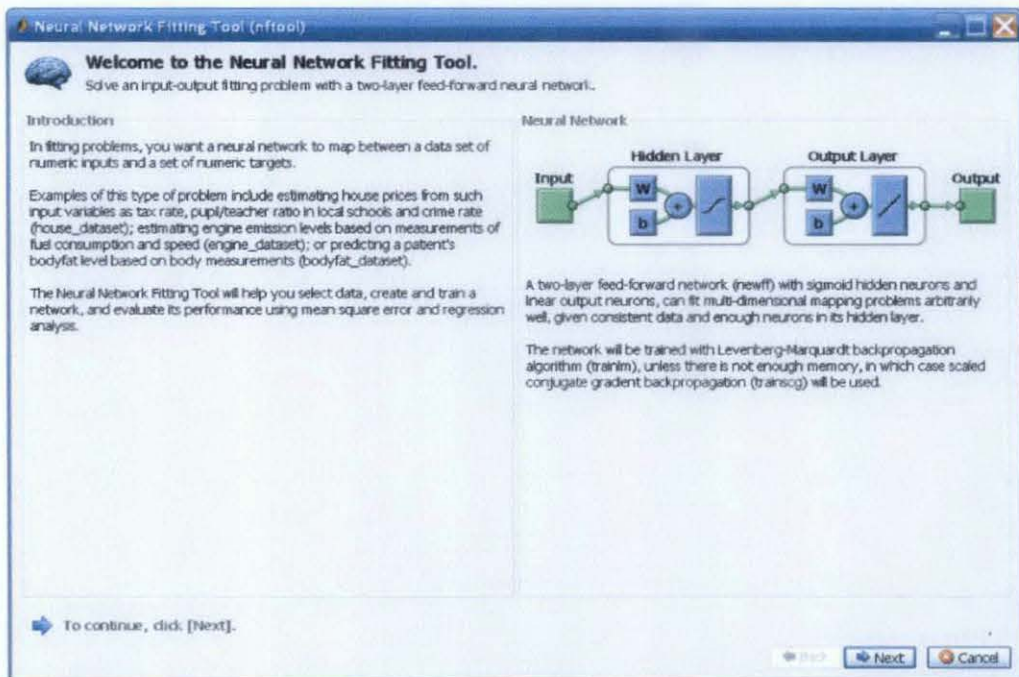


Figure 39 - Neural Network fitting tool window

The “Next” button is clicked and the number of inputs and targets are chosen which are the input flow (Q_{in}) and the output flow (Q_{out}) from the PI controller. Since the range of values is 7000 samples, only 500 samples were used in order to get good performance from the network. Other sample numbers can also be used as well. This can be seen in figure 40.

The network size is then set. The number of hidden layers neurons is chosen here. In this case, it is set by default to 20. If the network showed low accuracy after training, this number can be adjusted.

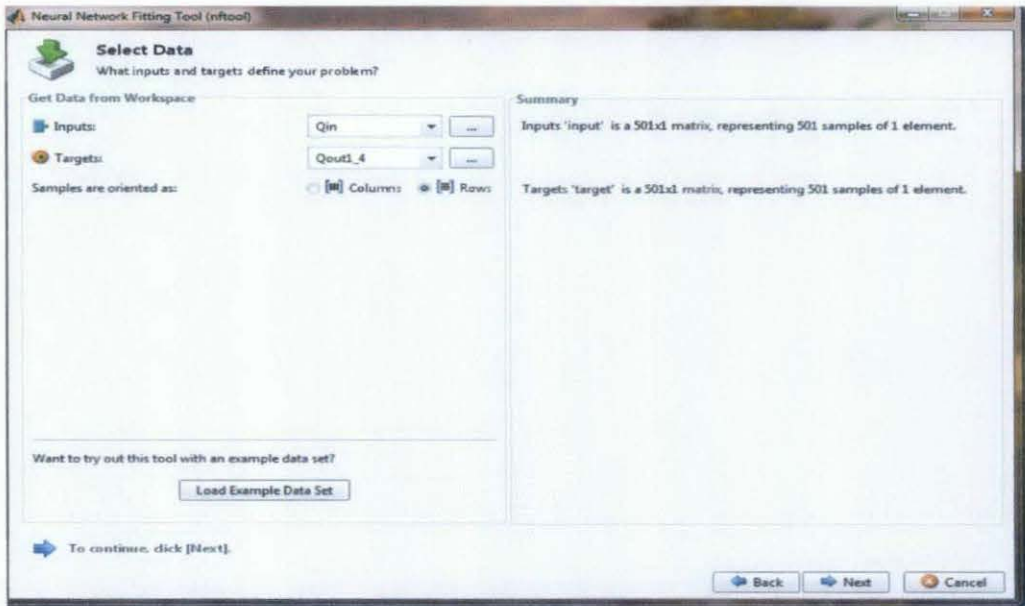


Figure 41 - Selecting the input & target data

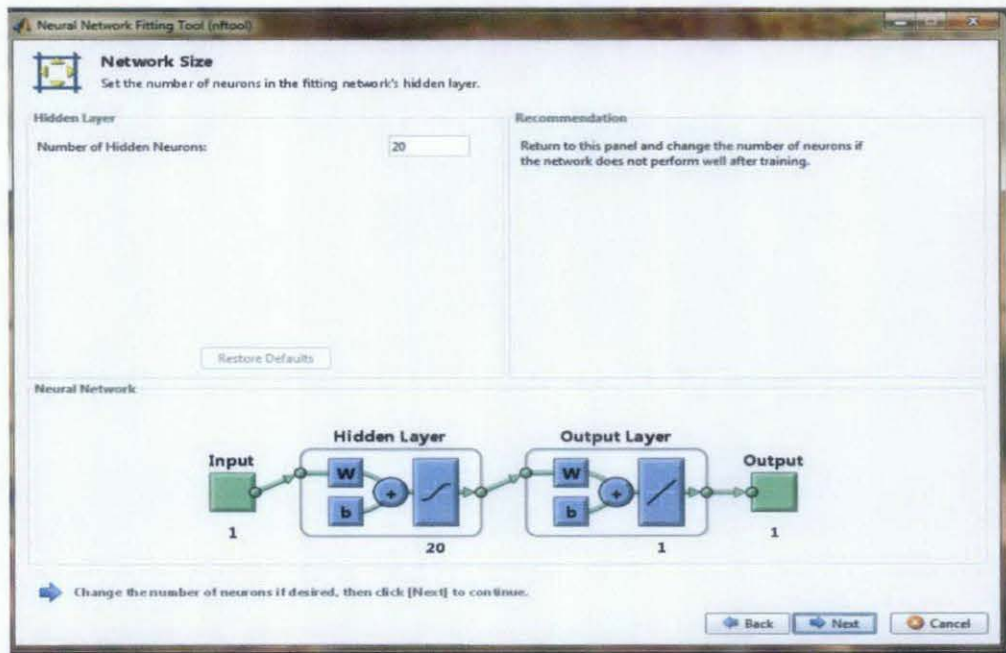


Figure 40 - Selecting the neural network size

Next the network is trained using Levenberg-Marquardt backpropagation method as seen in figure 43 then the training panel is opened showing all the details of the training. The panel shows the algorithms used progress of the training and

the plots. As seen in figure 43, the training stops when the validation error increased to 6 iterations which occurred at iteration 11.

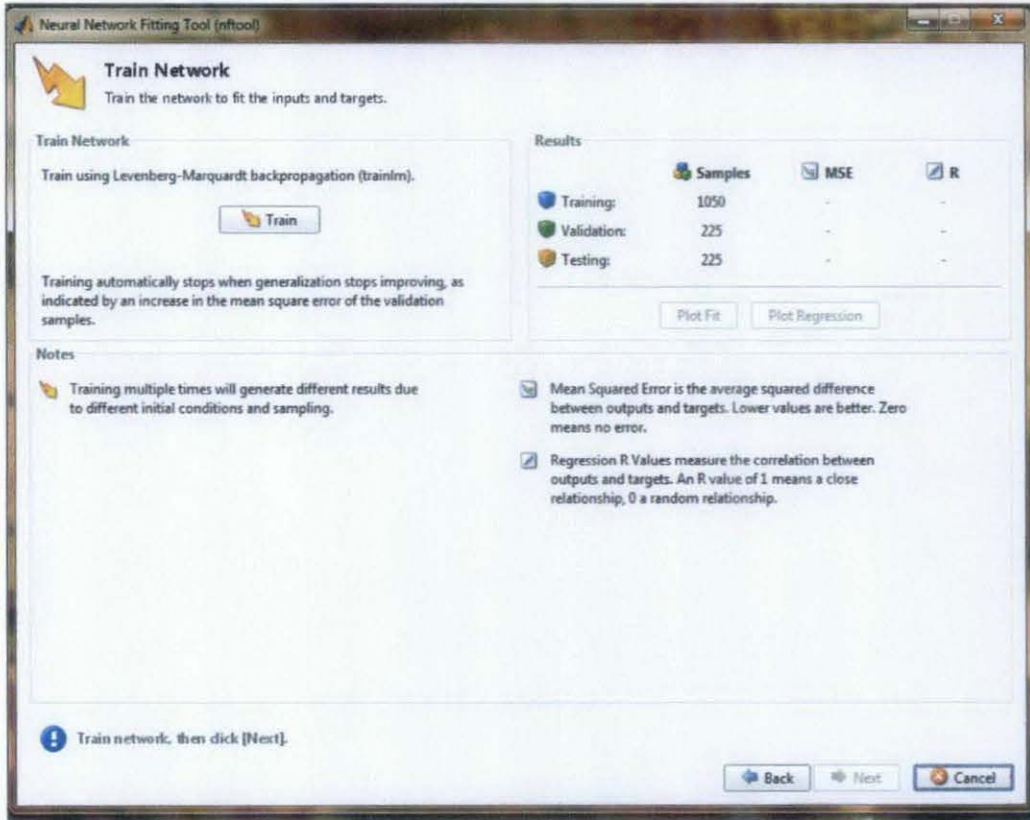


Figure 42 - Training the network

The performance plot can be seen in figure 44. A plot of the training errors, validation errors and test errors can be seen in the performance, the results here are reasonable as:

- The final mean square error is small.
- The test set error and validation set error have similar characteristics.

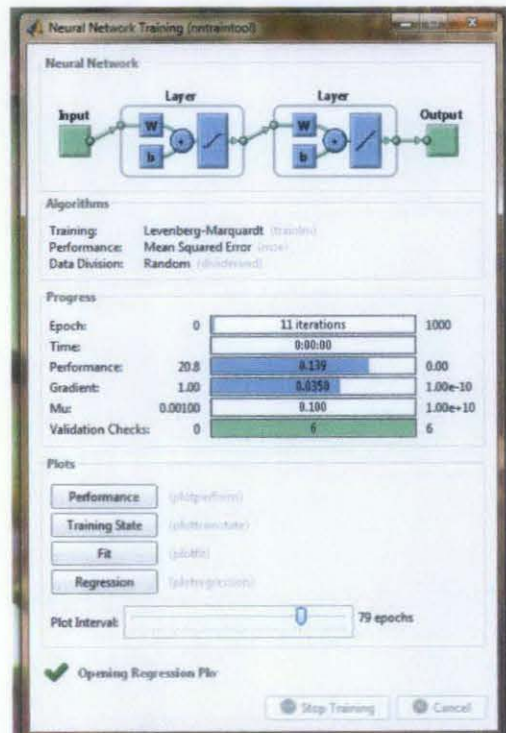


Figure 43 – The Training Panel

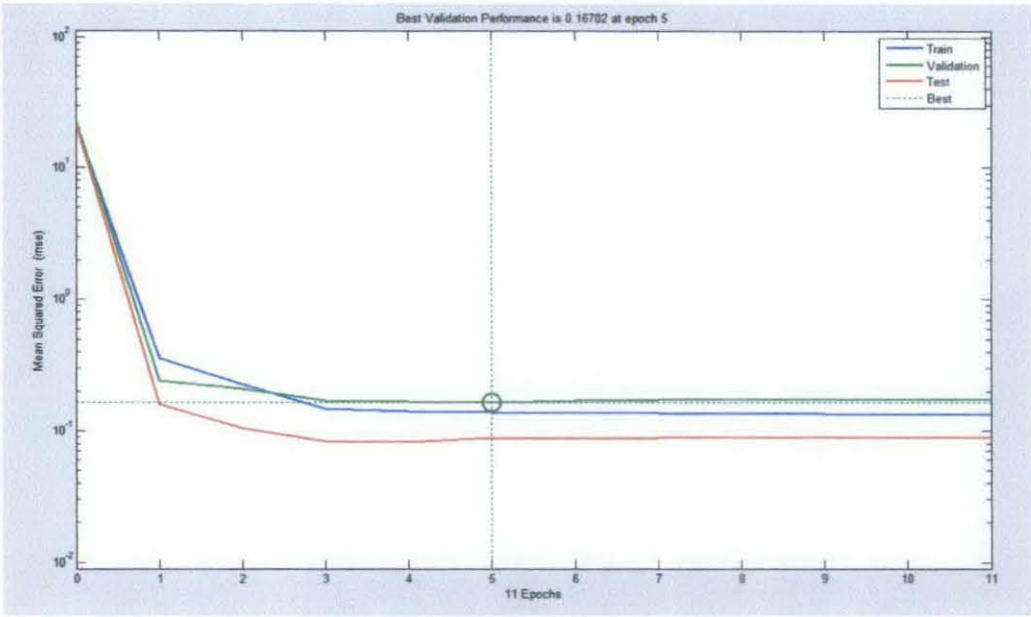


Figure 44 - Performance curve

When observing the regression curve, the regression should be more than 0.8 or 80% but in it was managed to get only 49%. The regression shows how accurate the training is. To increase the accuracy of the network:

- The initial network weights should be reset.
- The number of hidden neurons should be increased.
- The number of the training vectors increased.

These initiatives were made to enhance the accuracy and also the range of the input and target samples have been also changed but these attempts couldn't push the regression value beyond 49%.

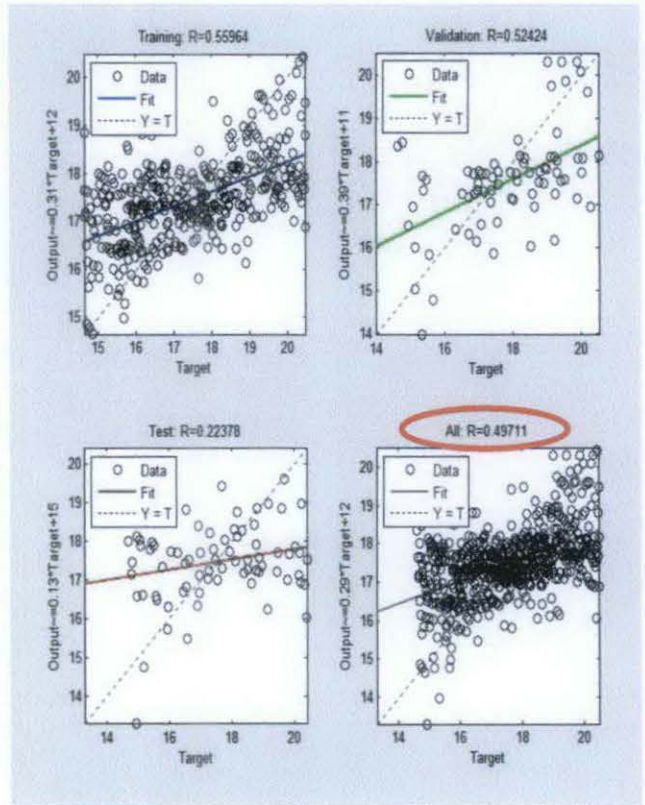


Figure 45 - Regression values

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Based on the latest's results, the "ARX" model showed a good response as the output flow (Q_{out}) stabilizes but it diverted far away from the set point. This can be a good start to use an intelligent controller in the process industry but not totally replace the PI and PID controllers. Usually, Industries try to focus more on how to improve the performance of PI controllers as they are more reliable and may not consider intelligent controllers. For the neural network if the accuracy increased to reach at least 85% it can be a reliable and be used in the field instead of the PID controllers.

During this year, a lot have been learnt about the level tanks systems and how to implement a PI controller using Simulink/Matlab. There are a lot problems related to tanks system in process industries. This kind of problems is concerned with stability of the plant. Control engineers should understand how control system works and how to solve the flow and level problems.

A model for a coupled tank system has been designed and several controllers have been tested (P, PI or PID controllers) and calculated by three different methods. The best controlled undergo fine tuning to get the best performance.

Then, the output has been changed to the level of the first tank H_1 as it is affected more easily than the output flow. Also random input variables have been used instead of a simple step function. After that, the ARX modeling was considered as an intelligent controller for the plant instead of PI controller and several comparisons were made to show their performances against each other. Finally a

training is made to a neural network to mimic the PI controllers in performance but its accuracy showed 49% only.

It must be mentioned that resources on this project were very limited either on online papers or in textbooks; most of the work was under the supervision of the supervisor especially the ARX modeling part. The controller calculations were based on a project which was made in “Plant Process Control Systems” subject during final year.

5.2 Recommendation

It is recommended to make M-files for the calculations of the process reaction curve, Ciancone calculation, Cohen Coon calculation as their calculation is time consuming. It is also recommended if several outputs were considered.

When selecting the level H_1 as an output, it was assumed that the best controller the same as the output flow (Q_{out}) based on Table 4. Comparing the performance of the controllers and choosing the best one will definitely affect the ARX modeling and show better learning as well as better results.

When training the neural network, several methods attempted but failed due to dimensional errors in the input samples. Some neural networks require the input to be in two dimensions form and this was a problem as the inputs are in row form. It is recommended to use other methods in training and compare the regression results.

Last but not least, it is recommended to apply these results on a prototype to compare between the modeling of the MATLAB with the practical results and compare the differences between the margins of errors between the two.

REFERENCES

[1] Elke Laubwald, *Coupled Tank Systems 1*, www.control-systems-principles.co.uk/

[2] "NeuroSolutions",
<http://www.nd.com/neurosolutions/products/ns/nnandnsvideo.html>

[3] "NeuroSolutions",
<http://www.nd.com/neurosolutions/products/ns/whatisNN.html>

[4] Process Control: Designing Processes and Control Systems for Dynamic Performance 2nd Ed, Marlin, McGraw-Hill, 2000

[5] Matlab software, Help guide.

[6] Martin T. Hagan, *Neural Networks for Control*.

[7] Martin T. Hagan & Howard Demuth, *Neural Networks Toolbox 6, User's Guid*, <http://www.mathworks.com>

[8] *NN predictive Control*,
<http://www.mathworks.com/help/toolbox/nnet/control3.html>

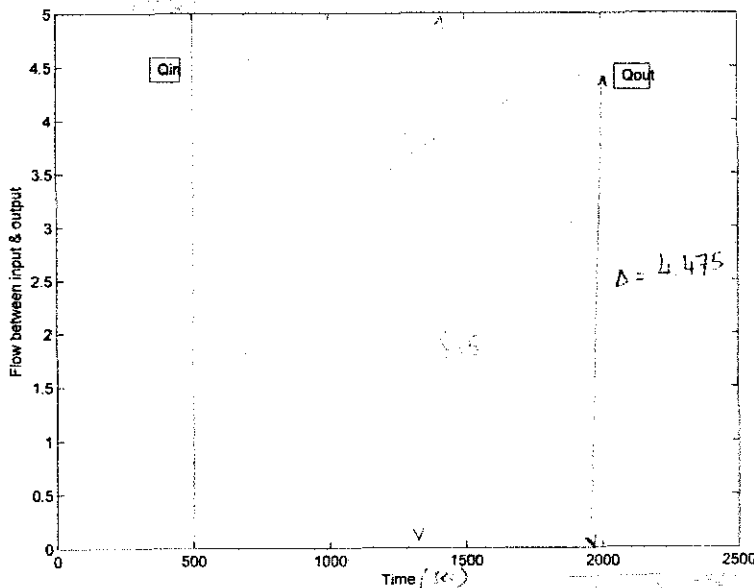
APPENDICES

APPENDIX A – CALCULATIONS OF THE PROCESS REACTION CURVE PARAMETERS OF THE OUTPUT FLOW (Q_{OUT})

The following is the process reaction curve with the calculations of process gain (K_c), dead time (θ) and time constant (τ).

$$\delta = 5 \quad K_p \text{ (process gain)} = \frac{\Delta}{\delta} = \frac{4.475}{5} = 0.895$$

$$\begin{aligned} t_{28\%} &= 4.475 \times 0.28 = 1.253 \\ \text{from graph } t_{28\%} &= 608 - 500 = 106 \\ t_{63\%} &= 4.475 \times 0.63 = 2.811 \\ \text{from graph } t_{63\%} &= 938 - 500 = 438 \\ \tau &= 1.5(t_{63\%} - t_{28\%}) = 1.5(438 - 106) = 498 \end{aligned}$$



$$\text{(dead time)} \theta = t_{63\%} - \tau = 438 - 498 = 60$$

$$\begin{aligned} \text{Max. execution time } \Delta t &\leq 0.25(\theta + \tau) \\ &\leq 0.25(60 + 498) \\ &= 27.9 \end{aligned}$$

$$\theta' = \theta + \frac{\Delta t}{2} = 60 + \frac{27.9}{2} = 73.95$$

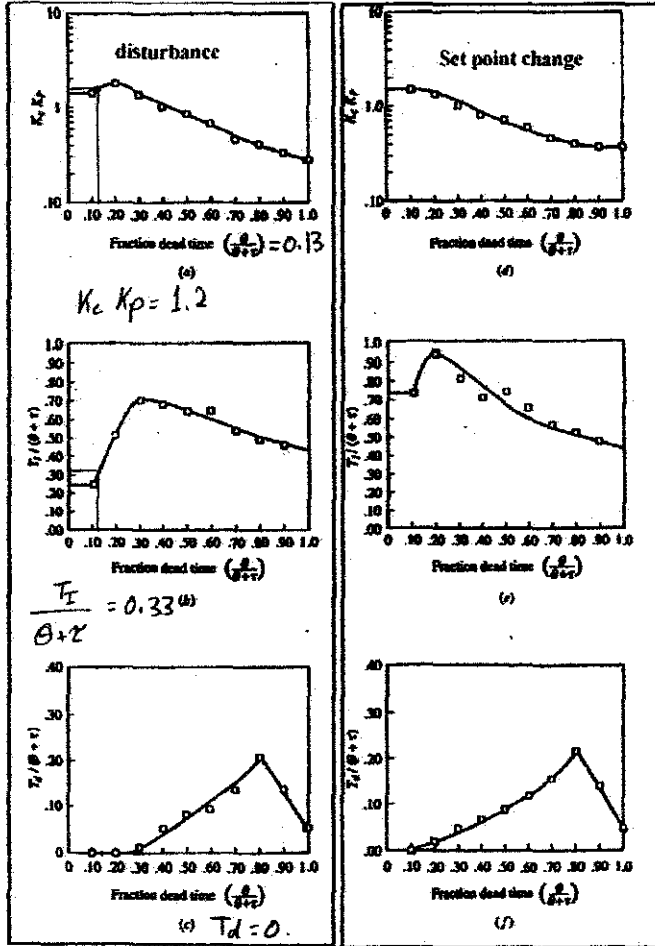
$$\text{modified dead time} = \frac{\theta'}{\delta} = \frac{73.95}{5} = 14.79 \approx 15$$

APPENDIX B – CALCULATIONS OF CIANCONE CONTROLLER

EAB4233/EEB5213

APPENDIX I

CIANCONE CORRELATIONS FOR DIMENSIONLESS TUNING CONSTANTS (PID ALGORITHM)



For disturbance response: (a) control system gain, (b) integral time, (c) derivative time.
 For set point response: (d) control system gain, (e) integral time, (f) derivative time.

Clan Cone

APPENDIX B: Tabulation of Results

Table B.1: Results for Process Reaction Curve (either Method I or Method II can be used)

Measurement	Value
Change in perturbation / MV, σ	5
Change in output / PV, Δ	4.475
Maximum slope, S	
Apparent dead time, θ	60
Calculations	Value
Steady State Process Gain, $K_p = \Delta / \sigma$	0.895
Apparent time constant, $\tau = \Delta / S$ or $\tau = 1.5(t_{0.63\sigma} - t_{0.22\sigma})$	498
*Fraction dead time, $\frac{\theta}{\tau}$	0.13

* The fraction dead time mentioned here is slightly different than one being used for Ciancone correlations.

Table B.2: PID Controller Parameters

Tuning Parameters:	P-only	PI	PID	PD
Proportional Gain, K_c		1.34		
Integral Time, T_i (minutes/repeat)		188.74		
Derivative Time, T_D (minutes/repeat)		0		

APPENDIX C - CALCULATIONS COHEN COON TUNING CORRELEATIONS

Controller	K_c	T_i	T_d
P-only	$\frac{1}{K_p} \frac{\tau}{\theta} \left(1 + \frac{\theta}{3\tau}\right)$	—	—
PI	$\frac{1}{K_p} \frac{\tau}{\theta} \left(0.9 + \frac{\theta}{12\tau}\right)$	$\theta \frac{\left(30 + 3\frac{\theta}{\tau}\right)}{9 + 20\frac{\theta}{\tau}}$	—
PID	$\frac{1}{K_p} \frac{\tau}{\theta} \left(\frac{3\theta + 16\tau}{12\tau}\right)$	$\theta \frac{\left(32 + 6\frac{\theta}{\tau}\right)}{13 + 8\frac{\theta}{\tau}}$	$\frac{4\theta}{11 + 2\frac{\theta}{\tau}}$

No. _____
Date: _____

Subject: _____

Cohen Coon Tuning Correlations (open loop)

P-only

$$K_c = \frac{1}{0.895} \left(\frac{498}{60}\right) \left(1 + \frac{60}{3(498)}\right)$$

$$= 9.2737 (1.04016) = 9.646$$

PI

$$K_c = \frac{1}{0.895} \left(\frac{498}{60} \right) \left(0.9 + \frac{60}{12(498)} \right) = 9.2737(0.91) \\ = 8.439$$

$$T_I = 60 \left[\frac{30 + 3 \left(\frac{60}{498} \right)}{9 + 20 \left(\frac{60}{498} \right)} \right] = 60 \left[\frac{30.36}{11.4} \right] = 159.65$$

PID

$$K_c = \frac{1}{0.895} \left(\frac{498}{60} \right) \left[\frac{3(60) + 16(498)}{12(498)} \right] = 9.2737 [1.3635] \\ = 12.64$$

$$T_I = 60 \left[\frac{32 + 6 \left(\frac{60}{498} \right)}{13 + 8 \left(\frac{60}{498} \right)} \right] = 60 \left[\frac{32.7229}{13.964} \right] \\ = 140.6$$

$$T_d = \frac{4(60)}{11 + 2 \left(\frac{60}{498} \right)} = 21.35$$

 www.controlstationary.com

Tuning Parameters:	P-only	PI	PID
Proportional Gain, K_c	9.646	8.439	12.64
Integral Time, T_I (minutes/repeat)	-	159.65	140.6
Derivative Time, T_D (minutes/repeat)	-	-	21.35

APPENDIX D – CALCULATIONS ZIEGLAR NICHOLS CLOSED LOOP TUNING METHOD CALCULATIONS

Ziegler Nichols closed loop Tuning method

for the system to be stable, The phase should be -180

$\omega_c = 0.026$

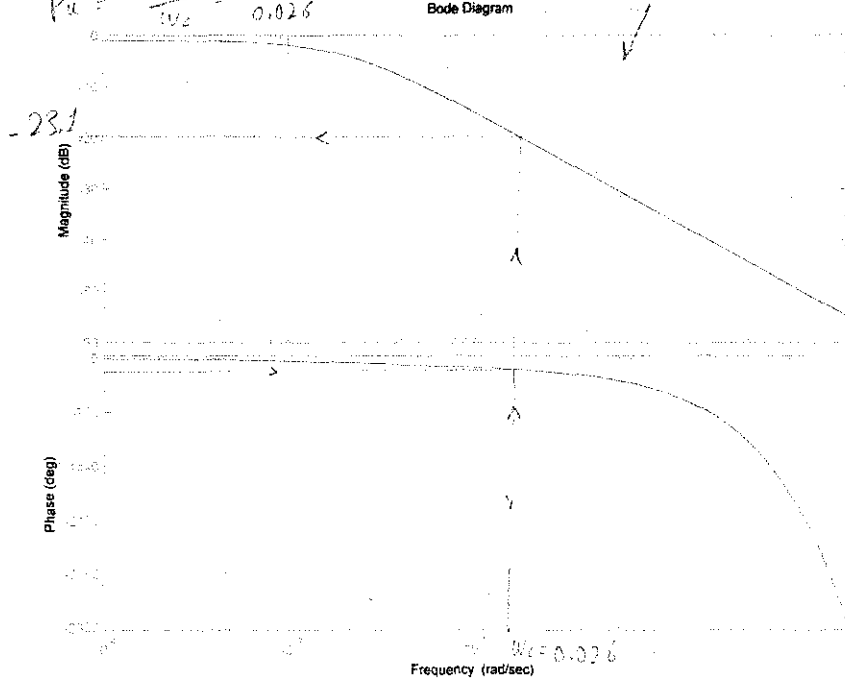
$K_{cr} = \frac{1}{10^{(-23.1/20)}} = 14.289$

$P_u = \frac{2\pi}{\omega_c} = \frac{2\pi}{0.026} = 241.66$

$\rightarrow G = \frac{1}{10} [10.857] [4.81] s$

$\rightarrow G$ is Delay 60s

\rightarrow Bode ω



Controller	K_c	T_i	T_d
P	$\frac{K_u}{2} = 7.1445$	-	-
PI	$\frac{K_u}{2.2} = 6.495$	$\frac{P_u}{2} = 120.83$	-
PID	$\frac{K_u}{1.7} = 8.4$	$\frac{P_u}{2} = 120.83$	$\frac{P_u}{8} = 30.2075$

APPENDIX E – M-FILES

The M-files which were used for ARX modeling can be seen viewed here.

1. ARX (1) M-file

```
function Yt= ARX1(H1_2,Qin)
Yt=H1_2(2:2280);
Yt_1=H1_2(1:2279);
Ut=Qin(2:2278);
Ut_1=Qin(1:2279);
X=[Yt_1 Ut_1];
par=pinv(X)*Yt
Ypred=X*par;
plot([Yt Ypred]);
err=(Yt-Ypred);
err.^2;
mean(err.^2)
```

2. ARX (2) M-file

```
function Yt= ARX2(H1_1,Qin)
Yt=H1_1(3:2280);
Yt_1=H1_1(2:2279);
Yt_2=H1_1(1:2278);
Ut=Qin(3:2280);
Ut_1=Qin(2:2279);
Ut_2=Qin(1:2278);
X=[Yt_1 Yt_2 Ut_1 Ut_2];
par=pinv(X)*Yt
Ypred=X*par;
plot([Yt Ypred]);
err=(Yt-Ypred);
err.^2;
mean(err.^2)
```

3. ARX (3) M-file

```
function par= ARX3(H1_2,Qin)
Yt=H1_2(4:2280);
Yt_1=H1_2(3:2279);
Yt_2=H1_2(2:2278);
Yt_3=H1_2(1:2277);
Ut=Qin(4:2280);
Ut_1=Qin(3:2279);
Ut_2=Qin(2:2278);
Ut_3=Qin(1:2277);
X=[Yt_1 Yt_2 Yt_3 Ut_1 Ut_2 Ut_3];
par=pinv(X)*Yt
Ypred=X*par;
plot([Yt Ypred]);
err=(Yt-Ypred);
err.^2;
mean(err.^2)
```

The ARX (3) is one which is used during the calculation as the error is minimum compared to ARX (1) and ARX (2).

APPENDIX F – CD OF THE PROJECT

A CD is attached with this report with the entire Simulink blocks and M-files which were made to acquire these results and graphs. Also this can be downloaded online from this link

http://hotfile.com/dl/81742832/9891850/MLP_sim_Mf.rar.html

APPENDIX [G] – Gantt Chart

