Neuro-controller for regulating coolant flow of CSTR system

By

SUHAIB MOH'D TAHA YOUSIF

Final Report submitted in partial fulfillment of
The requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronics Engineering)
DEC 2011

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Neuro-controller for regulating coolant flow of CSTR system

By

SUHAIB MOH'D TAHA YOUSIF

Final Report submitted in partial fulfillment of
The requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronics Engineering)
DEC 2011

Approved by

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

(Assoc. Prof. Dr. VIJANTH SAGAYAN ASIRVADAM)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2011

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.

SUHAIB MOH'D TAHA YOUSIF

ABSTRACT

Temperature control is very vital for processes that induce very high temperatures. Thus the need for tight control to prevent undesired production, or reactor burst. Process control is somehow shared between the fields of chemical engineering (more focus to process), and control engineering (more focus to control), and a mid point is to be reached by both sides to achieve the plant and production requirements, and also to maintain the system stable. Two control schemes are to be explored, PID control and neuro-control. The first one is widely used in all of industry sectors, especially process control. The later one is still a field of discovery and exploration, as all focus is there now to make intelligent systems the new substitutes for the conventional control systems. The system under the study is the non-isothermal CSTR with irreversible reaction A→B. The control scheme using the coolant flow showed that it is more efficient in heat removal and maintaining the system stable and under control, at the same time achieving the economic goal of highest productivity. Another scheme to be in touch is the hybrid control, where by different schemes are combined to compensate for each other, reaching an optimal control structure. For Further improvement to the system design, more disturbances to be included, and more complex processing units are to be tested using the controller schemes proposed. Experimental work is also an advantage to verify the simulations of the system and the controllers.

ACKNOWLEDGEMENT

First of all, I would like to express my utmost gratitude to Allah for his uncountable blessings and for giving me the strength to success and finish my final year project.

I would like to express my gratitude to my supervisor, Assoc. Prof. Dr. Vijanth Sagayan Asirvadam, whose support, encouragement, expertise, understanding, and patience, added considerably to my graduate experience during my final year. Since the first day I started till the moment I finished my final version of the final project report.

I would also like to thank my parents and family members for always supporting me and believing in me, not just for this work, but everything in my life and provided me through my entire life and in particular.

I must also acknowledge all the lecturers of Electrical and Electronic Engineering Department for their advices and all my friends in Universiti Teknologi PETRONAS for their continuous help and experience sharing.

In conclusion, I recognize that this project would not have been possible without the good environment offered by Universiti Teknologi PETRONAS, the Department of Electrical and Electronic Engineering.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A = reactant component

aij = constant coefficient in linear ODE

AJ = heat-transfer area of jacket

B = product component

bij = constant coefficient in linear ODE

CSTR = continuous stirred tank reactor

cj = heat capacity of coolant

cp = heat capacity of process

E = activation energy

F = volumetric fresh feed flow rate

Fm = molar fresh feed flow rate

Fj = cooling water flow rate to jacket

k =specific reaction rate

Kc = controller gain

ko = specific reaction rate

Ku = ultimate controller gain

M = molecular weight

Pu = ultimate period

Q = heat-transfer rate to jacket

R = chemical reaction rate

Tj= jacket temperature

Tjo = cooling water supply temperature

To= temperature of feed

T= reactor temperature

U = overall heat-transfer coefficient

Vj= volume of jacket

V= volume of reactor

CA=reactant concentration in reactor

CAo=reactant concentration in fresh feed

R= kinetic pre exponential factor

λ = heat of reaction

F =density of process liquid

Fj=density of coolant

∂M = measurement lag time

∂I =controller integral time

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

Chemicals have emerged as a wide sector in industry, with a broad range of products, such as: polymers and plastic, textiles, petroleum refining and primary metals. The chemical reactor is the prime mover of chemical processes, whereby it takes raw materials in, and converts them into the desirable product or outcome [1].

Chemical Plants consist of various processing units such as: reactors, heat exchangers, distillation columns, absorbers, evaporators, etc. The prime mover and the heart of all processes is the tank reactor. All chemical plants including all its processing units must follow some requirements during their operation, such as: Safety, production specification, environmental regulations, operational constraints and economics [2]. Process control is essential to achieve the mentioned requirements, by means of design and modeling.

The project involves a study on the potential of neuro-controllers as nonlinear controllers, to control a nonlinear process, a non-isothermal continuous stirred tank reactor (CSTR) with an irreversible reaction $A \rightarrow B$. This kind of process is usually controlled by the conventional PID controller, which is a linear controller and has some limitation when the system is hitting some sensitive points at the max or min ranges of operation.

The project will touch on the chemical point of view of the process prior to proceeding with the control scheme, as some characteristics of the system must be defined and explored before attempting to regulate or control any of the parameters.

1.2 PROBLEM STATEMENT

The control of the CSTR has always been a tough task for control engineers. That is due to the CSTR nonlinearity which affects its dynamics, and also has a great effect on the overall process in the plant. The main variable in controlling a CSTR is the temperature [1]. In order to control the temperature of the CSTR, a cooling jacket surrounding the CSTR is filled with a coolant fluid which flow is controlled by a valve. Sometimes the CSTR is not the starting point of the process, and mostly in the middle of the plant, as a mid process. The main variables to control the CSTR are either the flow in the CSTR, or the coolant flow to the cooling jacket [1] [2].

For quite a long time, PID control has been the dominant control method in both academic and industrial sector. But with the rapidly improving technology, new techniques have appeared to surface, such as: neural networks, fuzzy logic, genetic algorithm, particle swarm techniques, etc. Yet still, most of the plants are running on the old PID controllers, which tuning is still carried out using variety of algorithms.

Thus the need for a controller that can be more effective, in order to predict and adapt to unknown disturbances or faults that might occur, due to tear and wear and other caused that come along after long periods of operation.

However, this new will not totally replace the PID, at least not now. So, they can be added to the currently used PID to enhance its performance until the full transition from the PID to the intelligent controller's era.

1.3 OBJECTIVES

This project will cover the modeling and control of the CSTR, using the PID and the neural networks, and will work to achieve the following objectives:

- To study the process of a CSTR.
- To develop a mathematical model for the CSTR.
- To simulate the process of the CSTR using Matlab/Simulink
- To design the PID and the neural network controllers.
- To compare the performance of the two controllers.

1.4 RELEVANCY OF THE PROJECT

The project relevancy comes from following the trend of the intelligent control systems and seeking to replace the current PID controllers, and studying the response of the processes usually controlled by the PID, when an intelligent control is introduced. We also remember the importance of cooling systems in all exothermic processes, and how the failure of the cooling systems led to the disaster in Fukushima nuclear plant in Japan last March, which should upgrade the awareness on control of critical or toxic processes.

CHAPTER 2

LITERATURE REVIEW

2.1 CONTINUOUS STIRRED TANK REACTOR (CSTR)

Fig.1 shows the CSTR, which is a vessel with a mixer or a stirrer, to mix the reactants and the contents of the reactor. Around the vessel is a cooling/heating jacket, depending on the process desired, in our case, the jacket will be used to cool down the CSTR temperature [4]. Few assumptions are made for the CSTR, such as [5]:

- The mold inside the tank is uniformly homogenous all over the tank.
- The product temperature and composition are the same as that in the tank.
- Full mixing in reactor and jacket.

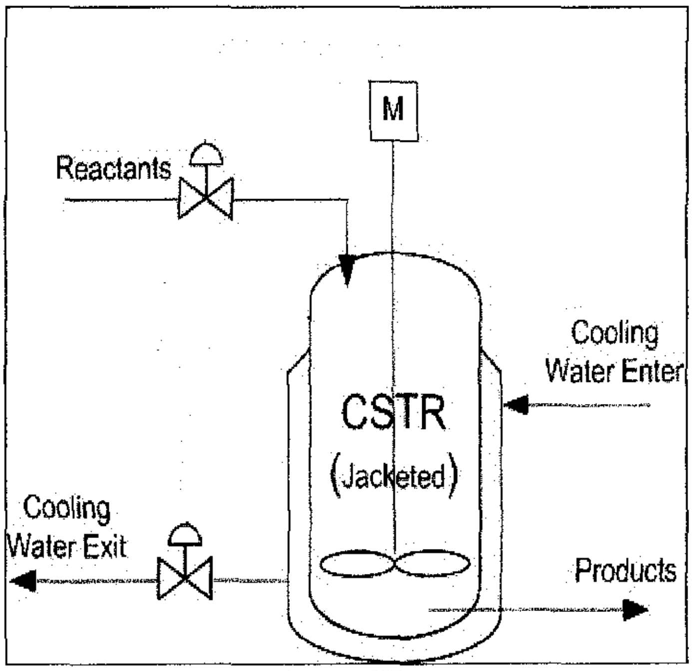


Figure 1: CSTR

The CSTR in use for our case is a non-isothermal CSTR with an exothermic reaction A→B. That means that the system produces that need to be removed by the coolant jacket, and for every mole of reactant A, a mole of product B is produced. The rate of conversion from reactant to products is inversely proportional to concentration. To achieve high conversion, only small concentration is used, and vice versa [6].

The trade-off between economics and controllability comes clear while designing the CSTR. In order to achieve the highest productivity, the system is to be provided with highest temperature, reactor must be of smallest size, and only provide small concentration to get the highest conversion rate [6]. So, a middle point is to be reached, to ensure highest productivity while maintaining system stability.

The system exhibit a nonlinear behavior as shown in the equations below [5] [6]:

a. Reactor component balance:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Ao} - C_A) - k_o C_A e^{-E/RT} \quad \text{.........(eq.1)}$$

b. Reactor energy balance:

$$\frac{dT}{dt} = \frac{F}{V}(T_o - T) + \frac{(\Delta H)}{V\rho c_p} k_o C_A e^{-E/RT} - \frac{UA}{V\rho c_p}(T - T_J) \quad \text{.........(eq.2)}$$

c. Jacket energy balance:

$$\frac{dT}{dt} = \frac{F_J}{V_J}\left(T_{Jo} - T_J\right) + \frac{UA}{V_J \rho_J c_J}(T - T_J) \quad \text{.........(eq.3)}$$

Prior to modeling, the system must be checked for its steady-state temperature, as the system only reach that when the heat produced by the reaction inside the tank is equal to the heat removed by the cooling jacket [6]. That can be shown in the equations below:

➢ Heat production:

$$Q_g = (-\Delta H_r)Vk_oC_Ae^{-E/RT} \quad \text{(eq.4)}$$

➢ Heat removal:

$$Q_r = \frac{UAT+F_J\rho_Jc_JT_{Jo}}{F_J\rho_Jc_J+UA} \quad \text{(eq.5)}$$

The equations eq.4 and eq.6 are to be plotted against the temperature, to obtain the heat plot, and from Fig.2 which resembles the heat plot below, we notice that the system have three steady state points. The choice is for the highest temperature, that is, to get the highest conversion rate [6].

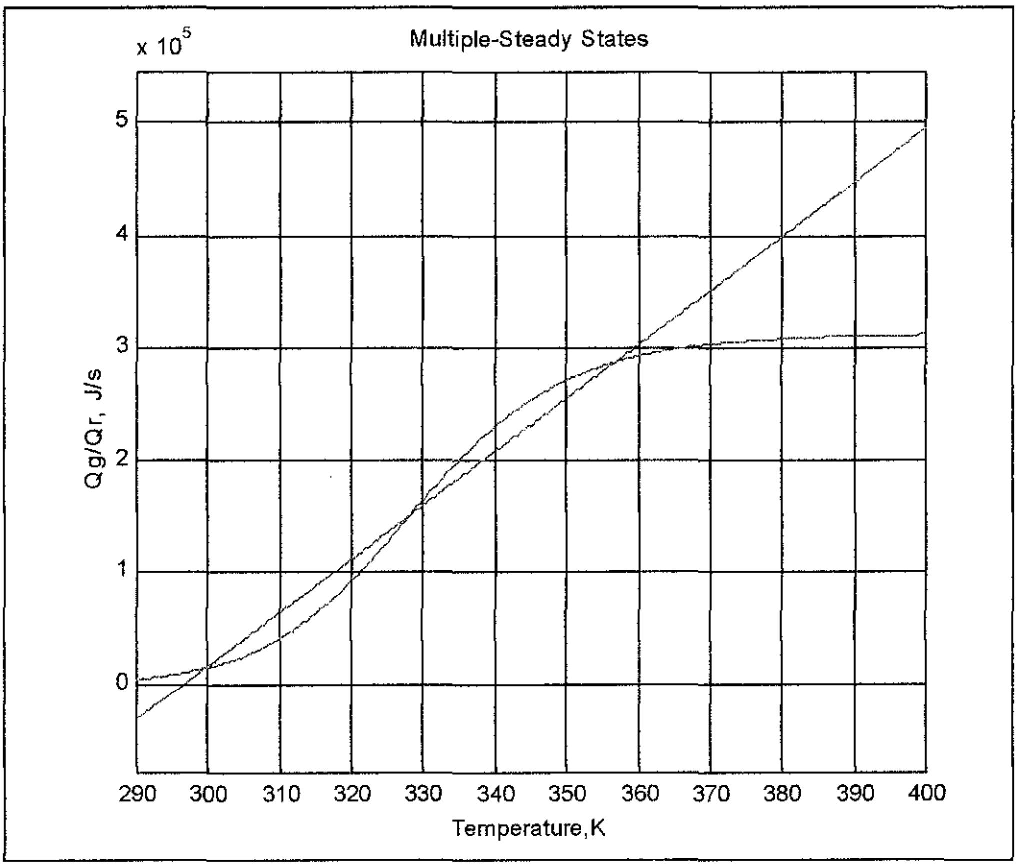


Figure 2: Heat plot with multiple steady states

The three steady state temperatures are:

T1= 299.57K, T2=328.23K, T3=356.81K

The steady state values for the concentration and cooling water temperature can be obtained from the equations below:

$$C_A = \frac{FC_{Ao}}{Vk_o e^{-E/RT} + F} \text{...(eq.6)}$$

$$T_J = \frac{UAT + F_J\rho_J c_J T_{Jo}}{F_J\rho_J c_J + UA} \text{...(eq.7)}$$

The resultant values are: C_A=81.3 mol/m^3, T_J=297.1K.

2.2 PID CONTROLLER

One of the widely used controllers in industry. Where **P** stands for proportional mode, **I** stands for the integral mode, and the **D** stands for the derivative mode. Each mode has got its own properties and usages. The final form of the PID algorithm is:

$$MV(t) = K_p\, e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \text{.....................................(eq.8)}$$

2.2.1 P-only mode

The proportional term, or sometimes called the aggressive mode. It acts aggressively against the error, as if the error increases, the P value will increase, and vice versa. The drawback of P only mode is that it never reaches the set point value. The value of P is also important, for too big P can lead to an oscillatory system [7].

2.2.2 I-only mode

The integral term, also called the persistence mode, tends to correct the error, and seek to reach the set point. It is to be noted that a combination of PI is mostly used, as each mode compensate for the other. Too big I can lead to a phenomena called integrator wind-up [7].

2.2.3 D-only mode

The derivative term, also called the predictive mode, tends to make use to past values of output, and predict the error tendency and direction. A combination of PID is also used in many applications [7].

2.2.4 PID Tuning

Many techniques are being used to tune the PI/PID controller such as: Ziegler-Nichols, Cohen-Coon, Tyreus-Luyben, etc. Each got its own calculations to define the proportional gain, the integral time and the derivative time coefficients [7].

For our case, the technique in use is the ZN method, the table below shows how the ZN method is carried out.

Table 1: Ziegler-Nichols method

Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2K_p / P_u$	-
PID	$0.60K_u$	$2K_p / P_u$	$K_pP_u / 8$

2.3 THE NEURO-CONTROLLER

2.3.1 The Artificial Neural Network

An artificial neural network (ANN) was inspired by the human brain and how it works to solve problems. The interest in ANN started 60 years ago in an attempt to discover how the brain works, and make a system that can somehow follow the way the real brain carry it's processes There is clearly a huge difference between the brain and the digital computers, nevertheless, there are some similarities. The basic processing unit of the ANN is the neuron, which is made in a way to simulate the real neurons in the human brain [8] [9]. In other words, the ANN is a system that learns, from experience, just like the human brain. That is, the system learns from the environment using a process called learning algorithm [8].

There are many applications where the ANN can be applied and used such as [9]:

a. **Signal processing**: applied in adaptive noise cancellation in telephone lines.
b. **Control**: using two modules (emulator, controller), and can be applied in various industrial control applications
c. **Pattern recognition**
d. **Medicine**
e. **Speech production**

In addition to the applications above, the ANN got also many benefits that set it apart from normal programmed networks [8] [9]. They are summarized below:

a. **Nonlinearity**: The neuron itself can be linear or nonlinear. This nonlinearity is useful when the system in use is also nonlinear.
b. **Adaptively**: The learning process helps the ANN to be adaptive to changes in the environment around the ANN, making it a suitable choice for systems with unpredicted disturbances.
c. **Evidential response**
d. **Fault tolerance**

2.3.2 Architecture of neural networks

The neurons connections in ANN define the complexity of the ANN. The way the neurons are interconnected defines the type of the ANN in use. There are three main structures for the ANN based on the type of interconnection [8] [9].

a. **Single layer net**: the simplest form of ANN, consists of an input layer, and an output layer. This net is strictly feedforward or acyclic type, that is, the signal flow in one direction only, from input to output. The notation of single layer refers to the output layer. Input layer of source nodes are not counted, because no computation is done there.

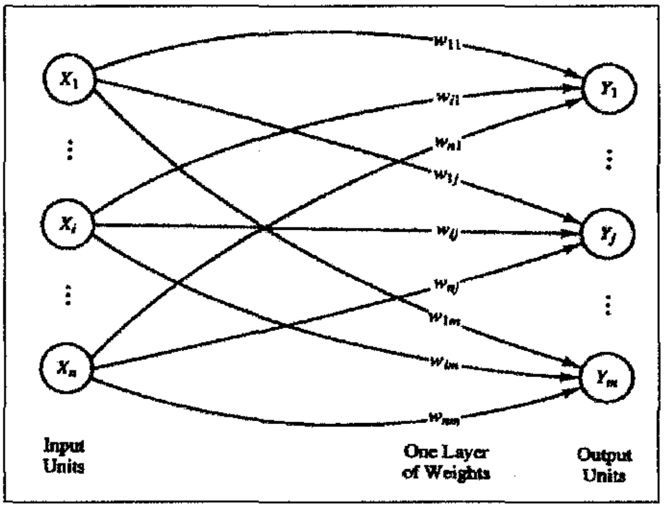


Figure 3: Single layer net

b. **Multilayer net**: distinguished by one or more hidden layers, whose function is to intervene between the input and output in some useful manner. More complicated than single layer, and can solve more complex problems, but require a harder learning process.

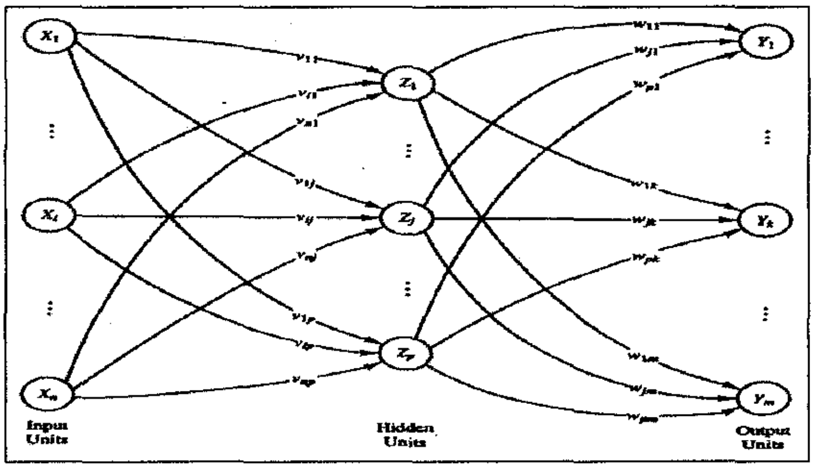


Figure 4: Multilayer net

c. **Recurrent networks**: This type of nets got at least one feedback loop. This feedback got a major effect on the learning capability and performance of the net. In the figure below, the feedback loops contain some time delay units, which result in nonlinear dynamics.

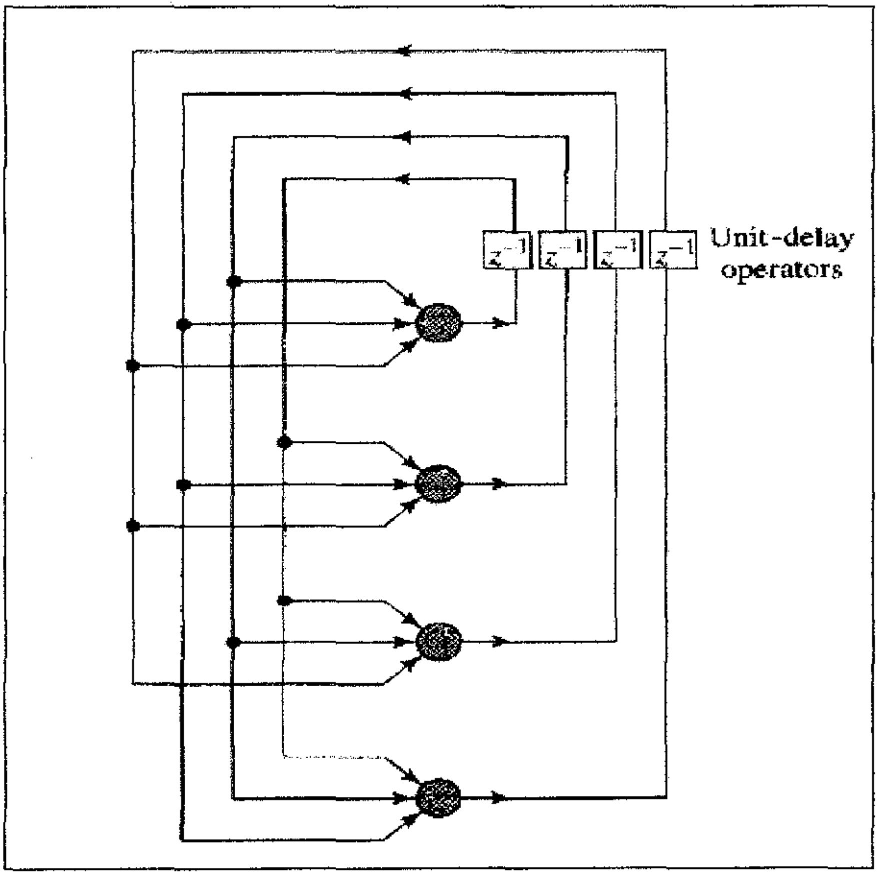


Figure 5: Recurrent network

2.3.3 The learning process

There are basically two type of learning processes:

a. **Supervised training**: the most used training method, by providing the system with an input vector, or array, or a pattern that is associated with a specific output vector. The ability of having storage to recall patterns is called the associative memory. If the output is the same as input it is called auto-associative memory, and if the pattern is different it is called hetero-associative memory [9].

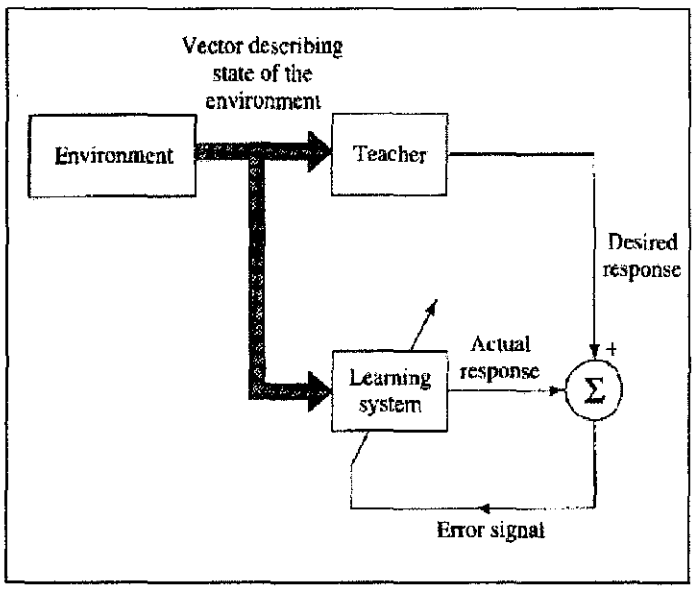


Figure 6: Supervised learning

b. **Unsupervised training**: in this case, the input vector is provided, but the output vector is not specified. Thus, the net assign and adjust the weights in the hidden layers by itself. Such nets are used for clustering [9].

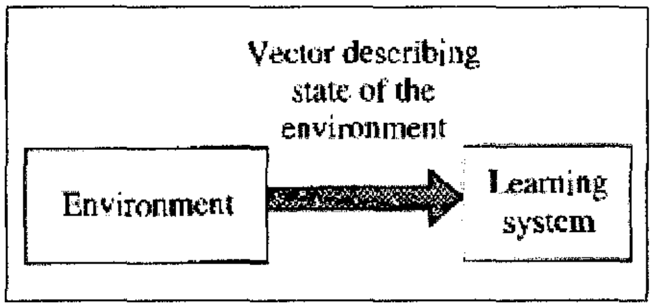


Figure 7: Unsupervised learning

CHAPTER 3

METHODOLOGY

3.1 Project procedure

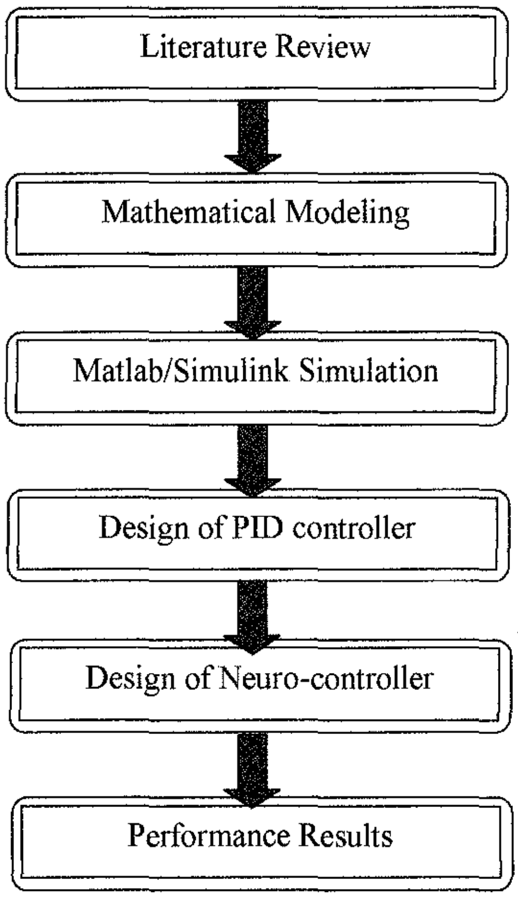
The projected started with an extensive literature review for the CSTR, its process and dynamics, given that it was not covered during previous years of study. Also, the algorithms of PID and its tuning have been covered, in addition to the neural network fundamentals and algorithms.

That was the first stage of the project, followed by constructing and deriving the mathematical model of the CSTR, based on the reactions, component and temperature balances in both the reactor and the jacket.

The third stage was to develop the system model in Matlab/Simulink environment. Followed by the design of a suitable PID structure for the CSTR. This was all covered in FYP within the timeline given later in the gantt chart.

As for FYP2, it was more focused on the design of the neuro-controller. The timeline was not enough to explore the neuro-controller thoroughly, as the CSTR is quite a complex system.

3.2 Project flow chart



3.3 Tools and software

The project was carried out on simulation basis, that is due to the unavailability of a plant with the specifications used in the project, thus the lack of experimental results.

As for the software used, the choice was for Matlab/Simulink. Other softwares could be used to solve for the thermodynamics of the CSTR such as ASPEN and ANSYS, but such results were not within the scope of this study.

Table 2: Gantt chart for FYP1

No.	Deatail/week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Familiarizing with CSTR system															
2	Submission of Extended proposal															
3	Build system model using SimuLink															
4	Design of PID controller															
5	Viva defence															
6	Results analysis															
7	Submission of interim draft report															
8	Submission of interim report															

Table 3: Gantt chart for FYP2

No.	Deatail/week	1	2	3	4	5	6	7		8	9	10	11	12	13	14	15
1	Design of neural controller																
2	Submission of progress report																
3	Results and analysis																
4	Pre-EDX																
5	Submission of Draft report																
6	Submission of Dissertation (soft bound)																
7	Submission of technical paper																
8	Oral presentation																
9	Submission of final report (hard bound)																

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Mathematical Model

The nonlinear model of the CSTR involves dynamic component and energy balances for the reaction liquid and an energy balance for the water in the jacket. Constant holdup in the reactor and jacket and constant physical properties are assumed. Table 3 give values of design parameters used as well as steady-state values of variables under base-case conditions:

Table 4: Parameters values

PARAMETER	Value
Constant parameters	
Feed temperature (T_o)	298K
Fresh feed composition (mole fraction A)	1000 mol/m^3
Activation energy (E)	85,000 (J/mol)
Density of process liquid (ρ)	1000 (Kg/m^3)
Heat capacity of process liquid (c_p)	4180 J/(Kg.K)
Density of coolant liquid (ρ_j)	1000 (Kg/m^3)
Heat capacity of coolant liquid (c_j)	4180 J/(Kg.K)
Volume of Reactor (V)	0.7854 m^3
Reaction constant (k_o)	4.0075×10^{10}/s
Area of reactor (A)	4.7124 m^2
Heat transfer coefficient (U)	100 J/(s. m^2.K)
Heat of reaction (ΔH_r)	-3.135×10^5 J/mol
Parameters varied	
Conversion (X)	95%
Feed flow rate (F)	0.001(m^3/s)
Jacket temp. (T_j)	287 (K)
Cooling water flow rate (F_j)	0.0009075 (m^3/s)

The system model was given in eq., eq.2,and eq.3. But that is only the nonlinear time domain model. In order to apply PID control, te system need to be linearized.

Linearization gives three linear ordinary differential equations:

$$\frac{dC_A}{dt} = a_{11}C_A + a_{12}T + a_{13}T_j + b_{11}F + b_{12}F_j.$$

$$\frac{dT}{dt} = a_{21}C_A + a_{22}T + a_{23}T_j + b_{21}F + b_{22}F_j \quad \text{......................... (eq.9)}$$

$$\frac{dT_j}{dt} = a_{31}C_A + a_{32}T + a_{33}T_j + b_{31}F + b_{32}F_j.$$

The constant a_{ij} and b_{ij} coefficients are given below:

$$a_{11} = -\frac{\bar{F}}{V} - \bar{k}$$

$$a_{12} = -\frac{\overline{C_A}E\bar{k}}{R\bar{T}^2}$$

$$a_{13} = a_{31} = 0$$

$$a_{21} = -\frac{\lambda\bar{k}}{Mc_p}$$

$$a_{22} = -\frac{\overline{C_A}E\bar{k}\lambda}{Mc_pR\bar{T}^2} - \frac{\bar{F}}{V} - \frac{UA_H}{V\rho c_p} \quad \text{.. (eq.10)}$$

$$a_{23} = \frac{UA_H}{V\rho c_p}$$

$$a_{23} = \frac{UA_H}{V_j\rho_j c_j}$$

$$a_{33} = -\frac{\bar{F}_j}{V_j} - \frac{UA_H}{V_j\rho_j c_j}$$

$$b_{11} = \frac{C_{Ao} - \bar{C}_A}{V}$$

$$b_{12} = b_{22} = b_{31} = 0 \quad \text{.. (eq.11)}$$

$$b_{21} = \frac{T_o - \bar{T}}{V}$$

$$b_{32} = \frac{T_{jo} - \bar{T}_j}{V_j}$$

From the equations above, the system model can be mathematically represented in the state space format:

$$\begin{bmatrix} \dot{C}_A \\ \dot{T} \\ \dot{T}_J \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ aa_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} C_A \\ T \\ T_J \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \begin{bmatrix} F \\ F_J \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_A \\ T \\ T_J \end{bmatrix}$$

From there we can obtain the transfer function that relates the coolant flow to the product temperature.

$$G(s) = \frac{T(s)}{F_J(s)} = \frac{-0.002515}{s^3+16.39s^2+60.24s+37.5} \quad \text{(eq.12)}$$

4.2 Controller Design

4.2.1 PID

Before checking the closed loop response, open loop was tested, results are shown below.

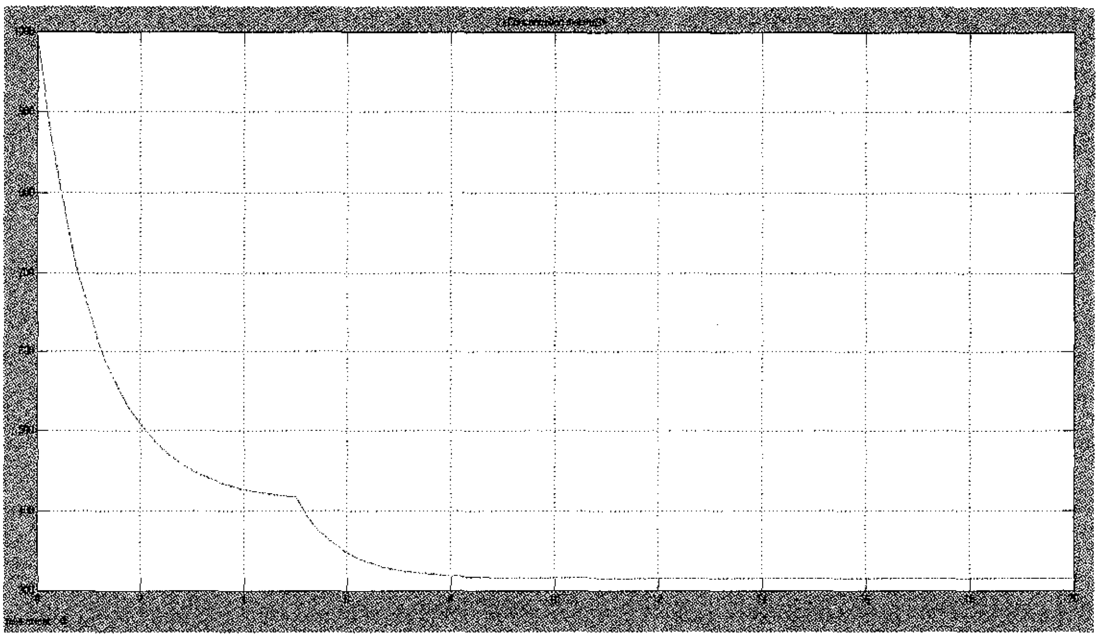


Figure 8: Open Loop Response (Concentration)

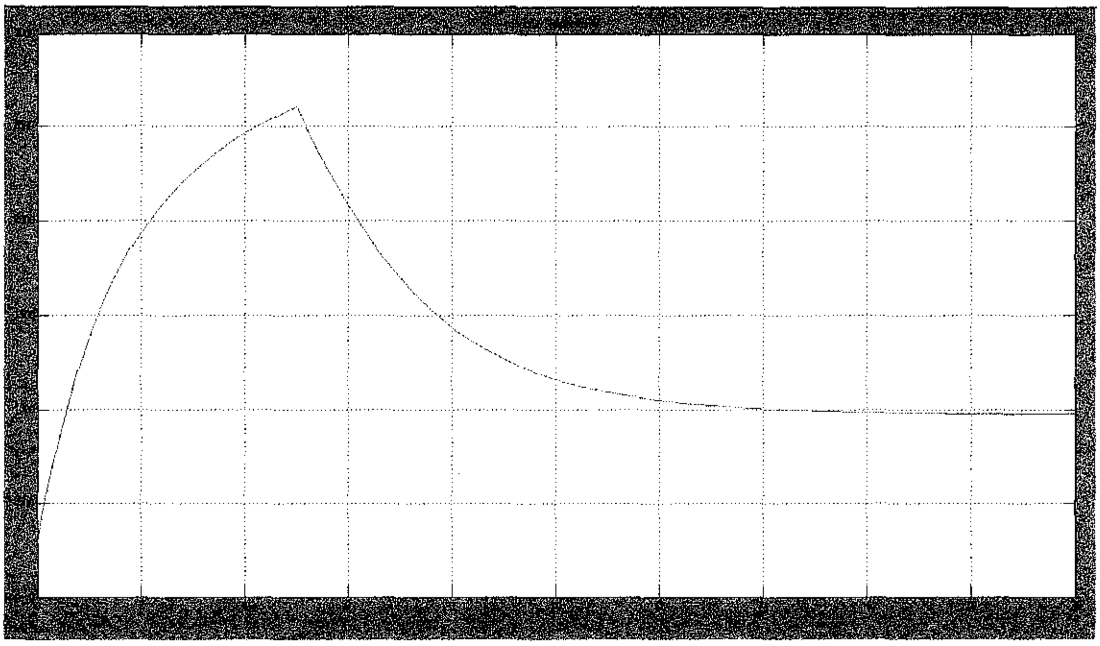


Figure 9: Open Loop Response (Temperature)

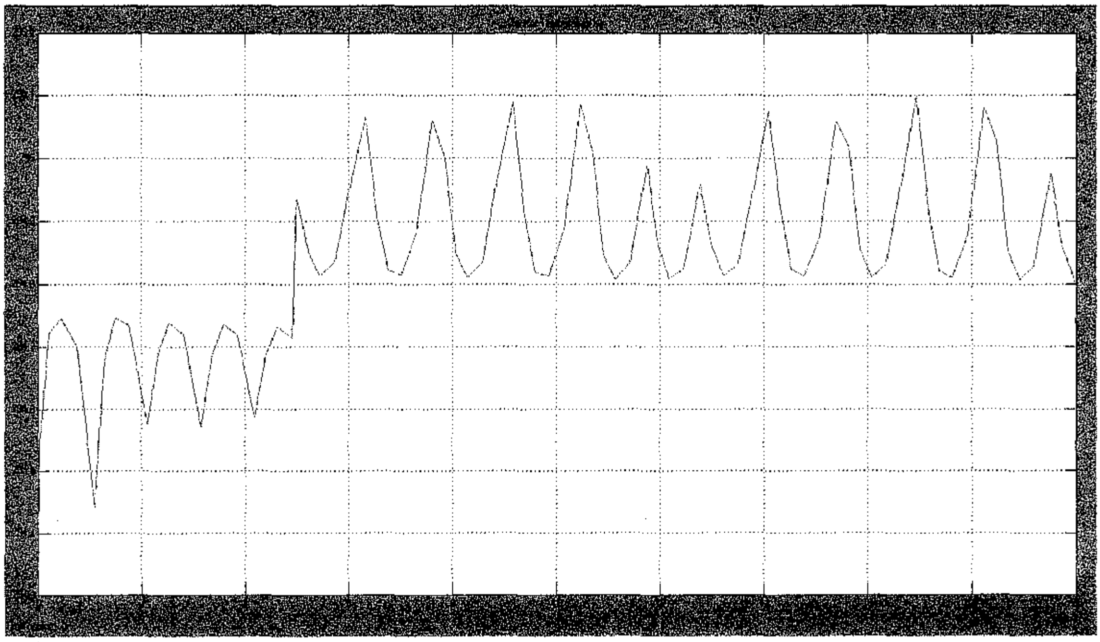


Figure 10: Open Loop Response (Jacket Temperature)

We can see in Fig.8 that the concentration of reactant A is decaying with time, which is true as the reaction takes place, and product B is being produced, but not at the same rate reactant A is decaying. Fig.9 and Fig.10 shows the temperature response of the reactor and the jacket respectively. The reactor temperature rises and then the heat is removed by the cooling fluid in jacket. Jacket temperature goes up and down, yet within a specific range, due to the action of heat removal.

For the controller design, there are many methods such as the well known ZN method. The tuning table was shown earlier in the literature review. Following are the results for the PID controller:

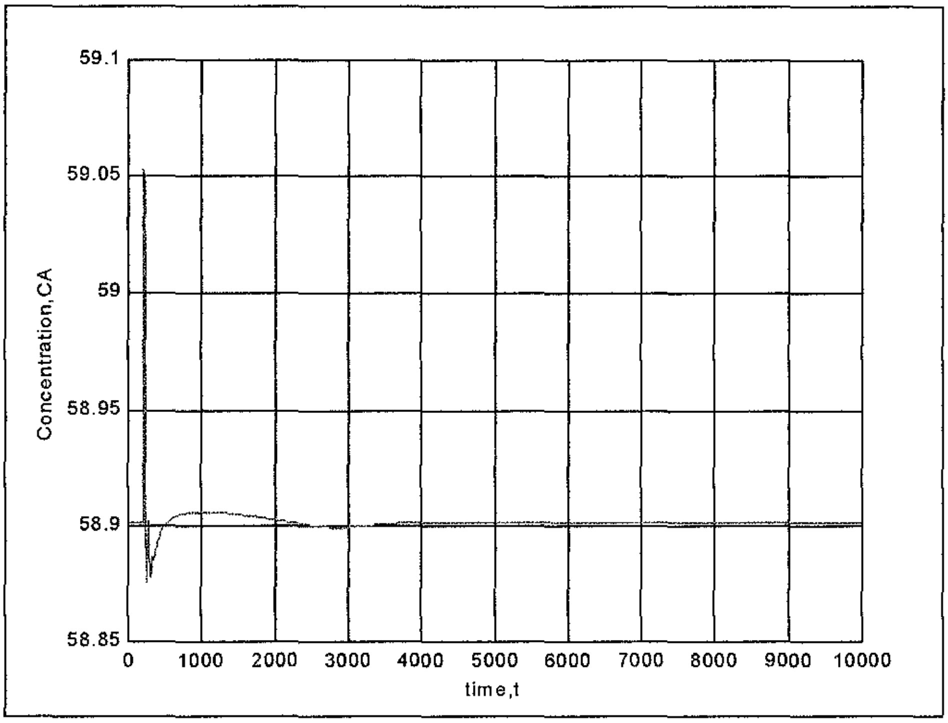


Figure 11: Concentration vs. Time

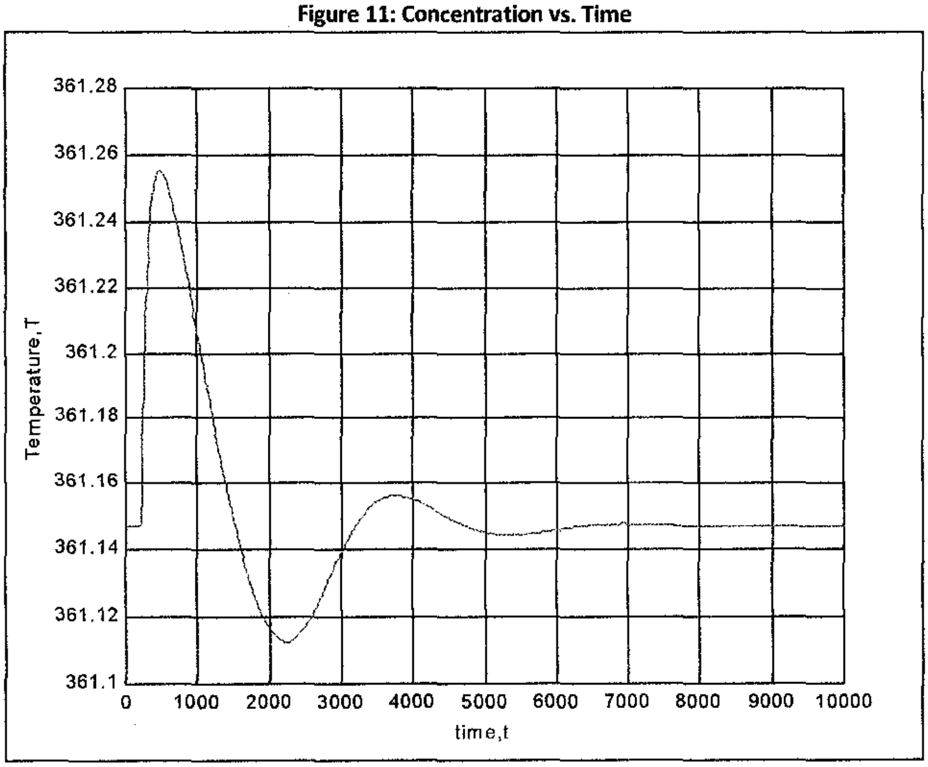


Figure 12: Temperature vs. Time

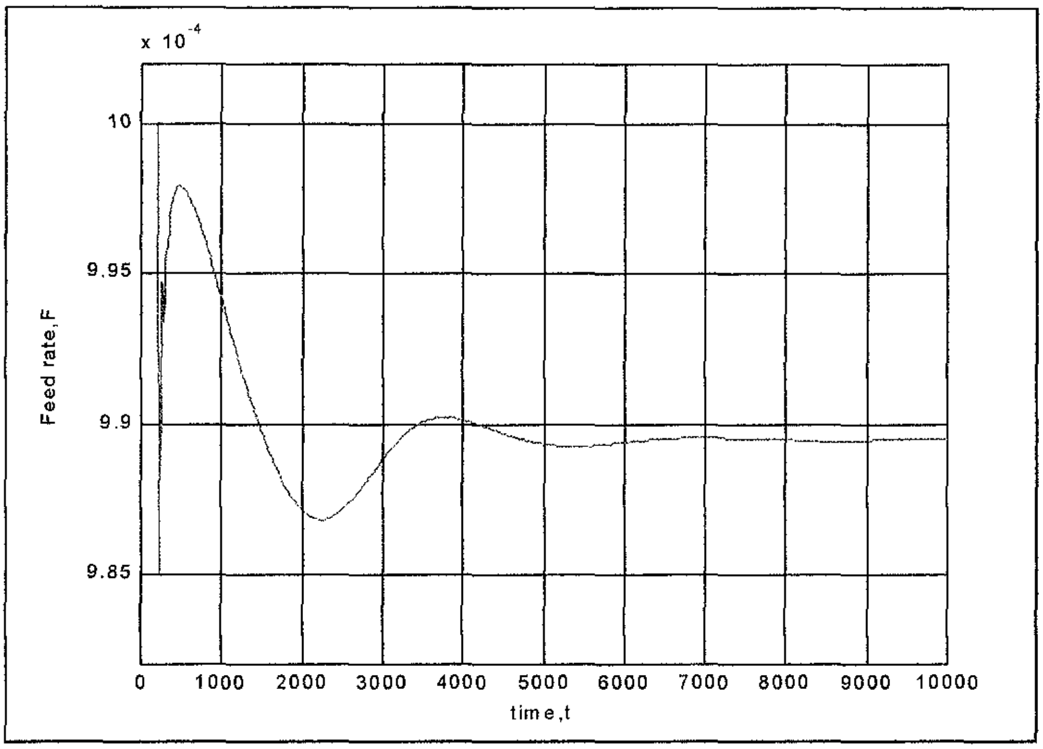


Figure 13: Fresh Feed vs. Time

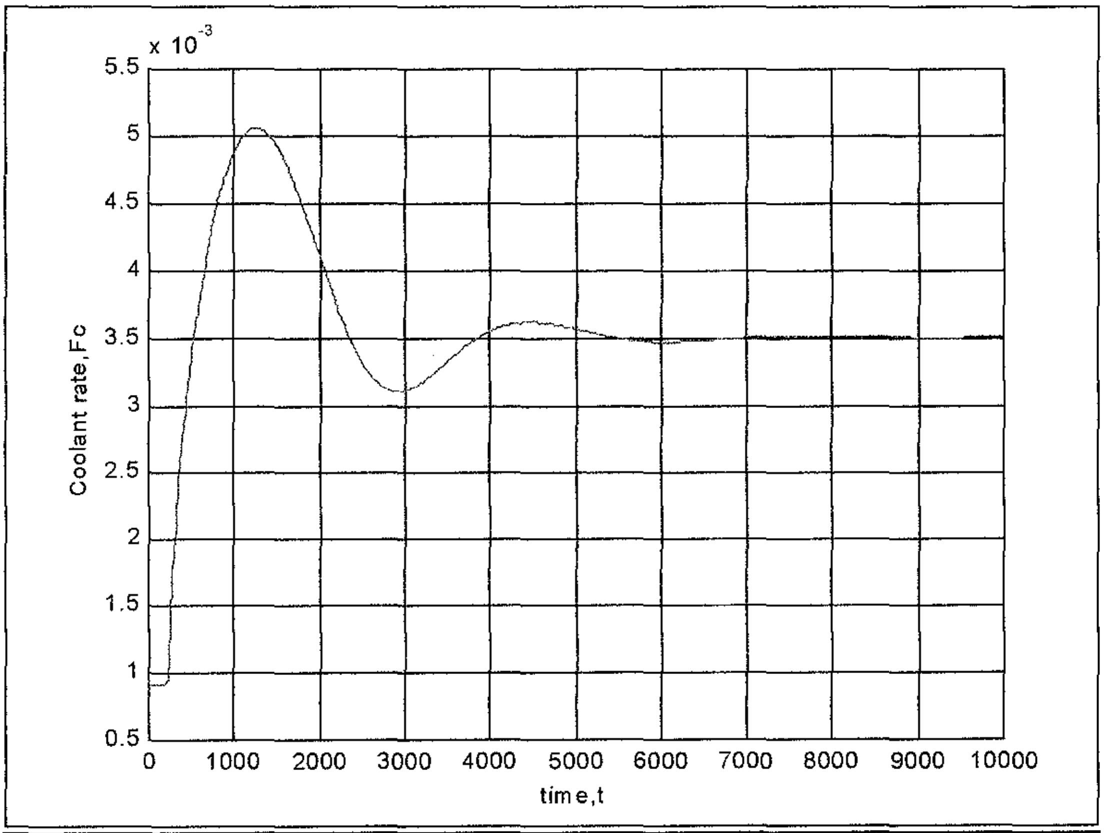


Figure 14: Coolant flow

The system exhibits some overshoots, but still can be returned to set point, that is due to the effect of the integral component in the PID controller algorithm.

Further investigation of the controller response can be done by considering all expected disturbances that might occur, such as fresh feed disturbance, fresh feed temperature, coolant liquid temperature, etc.

All points checked so far are within the equilibrium of the CSTR. Another point outside the equilibrium state was chosen for the PID (500K), and as expected the system went oscillatory and out of control. The effect of nonlinearities is very obvious as shown in the figure below.

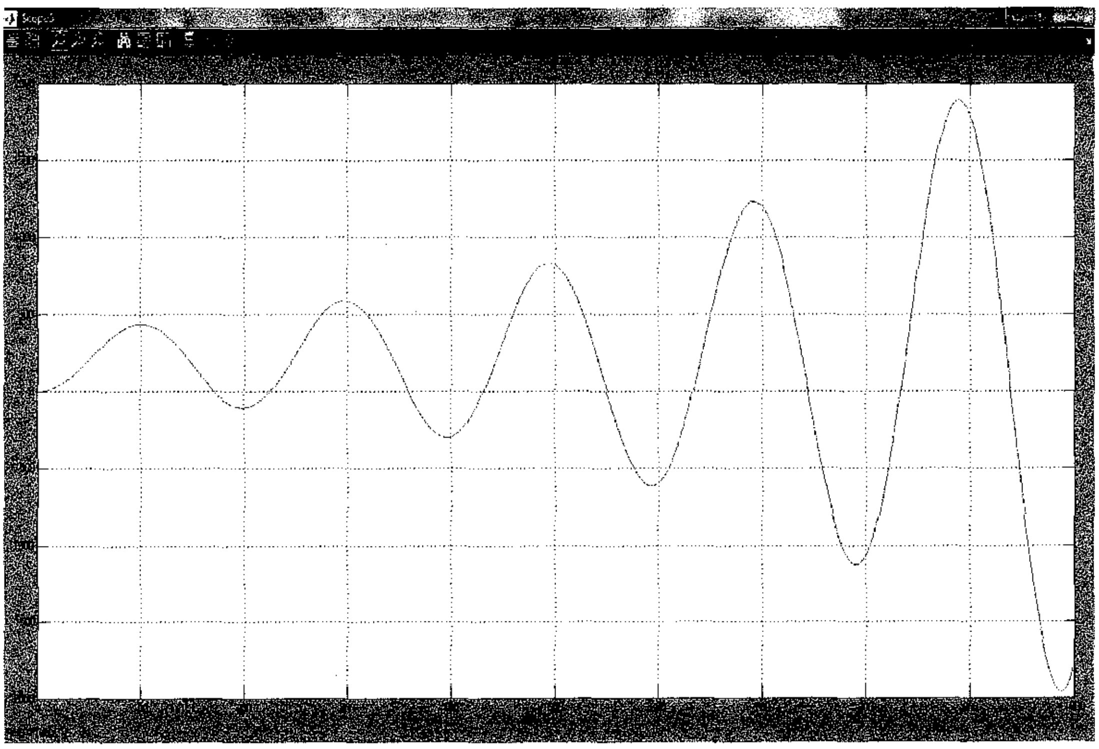


Figure 15: System Out of Control

4.2.2 Neurocontroller (ARX model)

As for the neurocontroller, we decided to use the ARX model. ARX stands for AutoRegression with eXternal input model. ARX model is widely applied linear dynamic model and represented as follows:

$$y[k] + a_1 y[k-1] + a_i y[k-i] + \cdots + a_n y[k-n]$$

$$= b_1 u[k-1] + b_i u[k-i] + \cdots + b_n u[k-n] + e(t) \text{....(eq.13)}$$

Where $y[k]$ and $u[k]$ are autoregressive variable or system output and exogenous variable or system input at time k respectively, and a_i, b_i are coefficients where i= 1,2,3,...,n and n is the system order.

The model is based on the choice:

$$a(d)\, y(t) = b(d)\, u(t) + e(t) \text{......(eq.14)}$$

$$y(t) = \frac{b(d)}{a(d)} u(t) + \frac{1}{a(d)} e(t) \text{......(eq.15)}$$

Linear regression is used to estimate the output of the system, given the input parameters. In this mode we notice that the controller tends to follow the desired output, with less overshoot for one disturbance, but still not containing the second wave of disturbance, and goes above the desired output temperature value. Considering the coolant flow, the output is considerably reasonable, as the coolant flow remained up during the whole period of cooling (heat removal), that is, until all additional heat is removed and the system is back again to the steady state condition.

The realization of disturbances is more valid in experimental environment, where the controller is exposed to real disturbance; therefore the response of the controller would add more credibility to its performance.

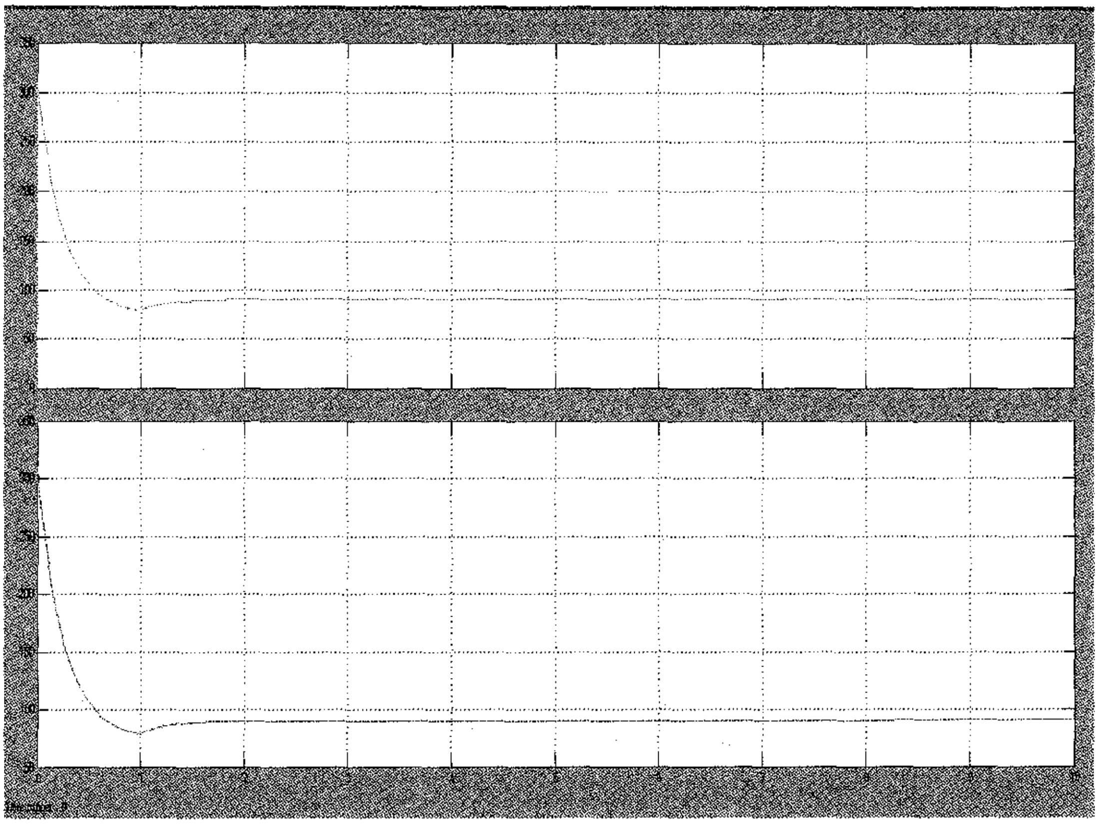


Figure 16: Neuro-controller response (controller output and system output)

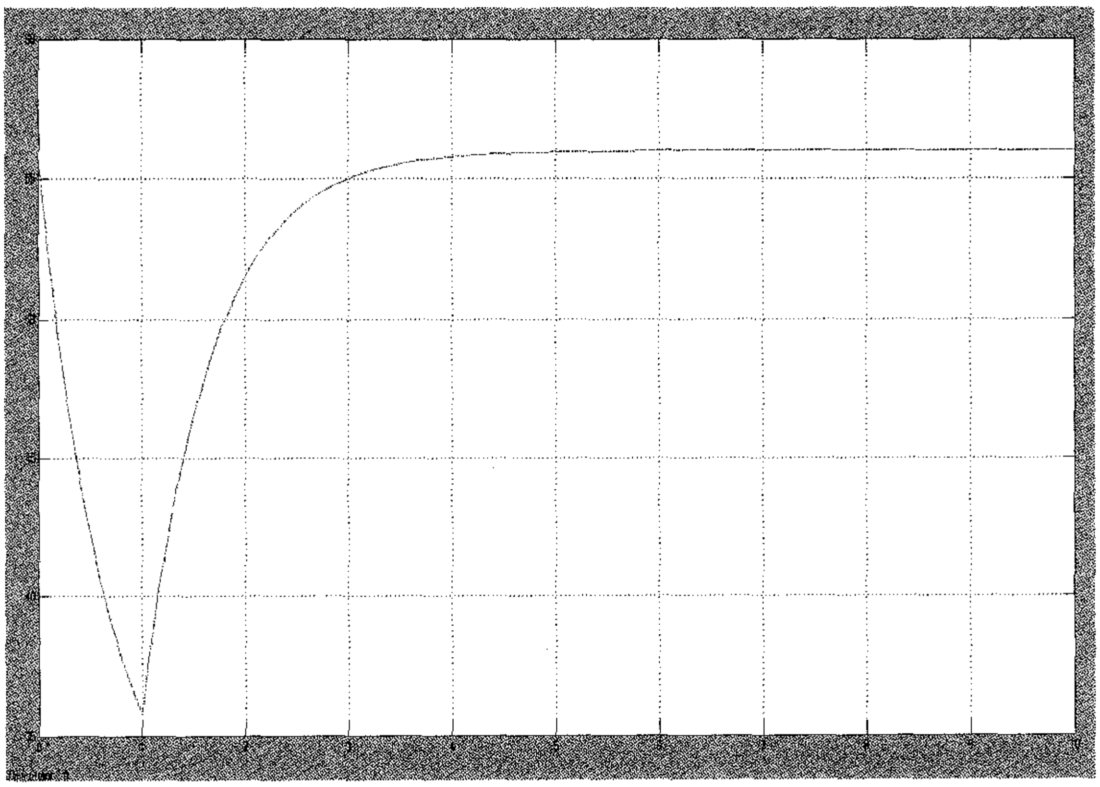


Figure 17: Neuro-controller (concentration)

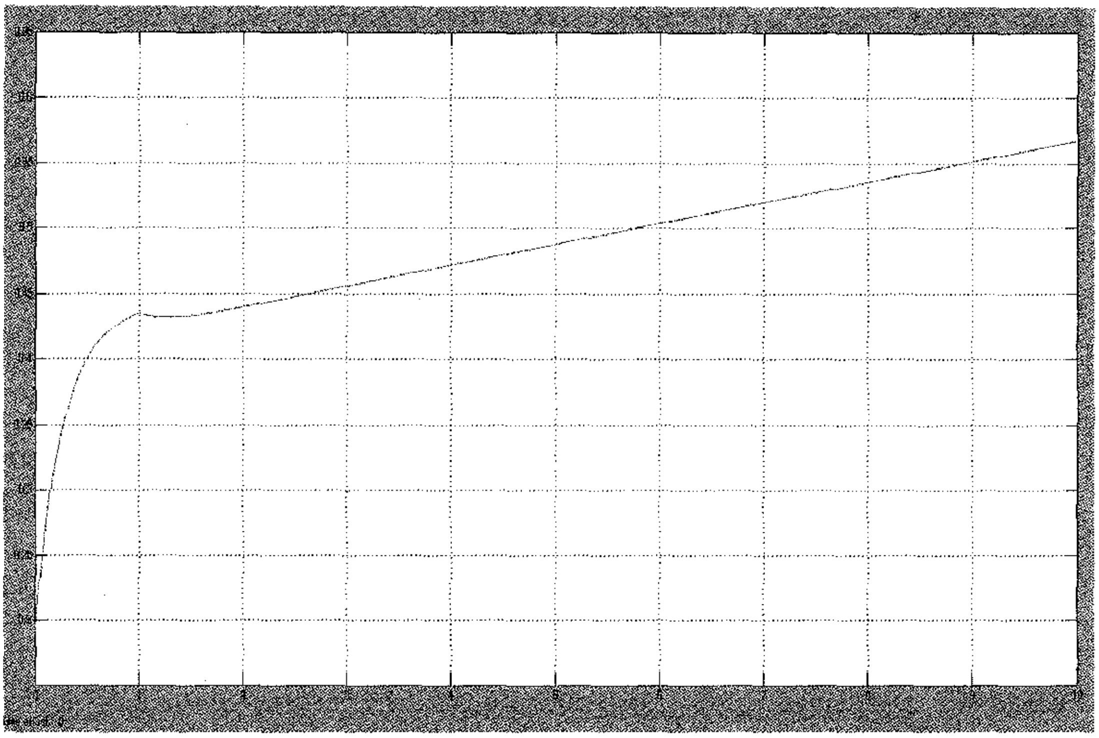


Figure 18: Neuro-controller (coolant flow)

It is clear from the figures above how temperature control is very stable using the neuro-controller, which was designed using the Recursive Least Squares algorithm (check Appendix 3). The concentration is also within acceptable range, also due to temperature maintained stable. In Fig.18 we notice and increasing flow rate with time, which is normal considering the action of heat removal, but if we consider for the valve action, such increment might drive the valve to malfunction, as the opening limit is exceeded. Valve specs should be considered as well.

The last figure below shows a hybrid system, whereby the output of the nuero-controller is fed back to the PID controller. In this case, the estimated values of the system output are fed to the PID controller prior to the reaction taking place, and the controllers can take action prior to the disturbance anticipated.

It is to be noted that only one vector was used to train the system (learning process), because only one disturbance was assumed (feed temperature). In case more disturbances were assumed, more complex algorithm is to be used for the learning process.

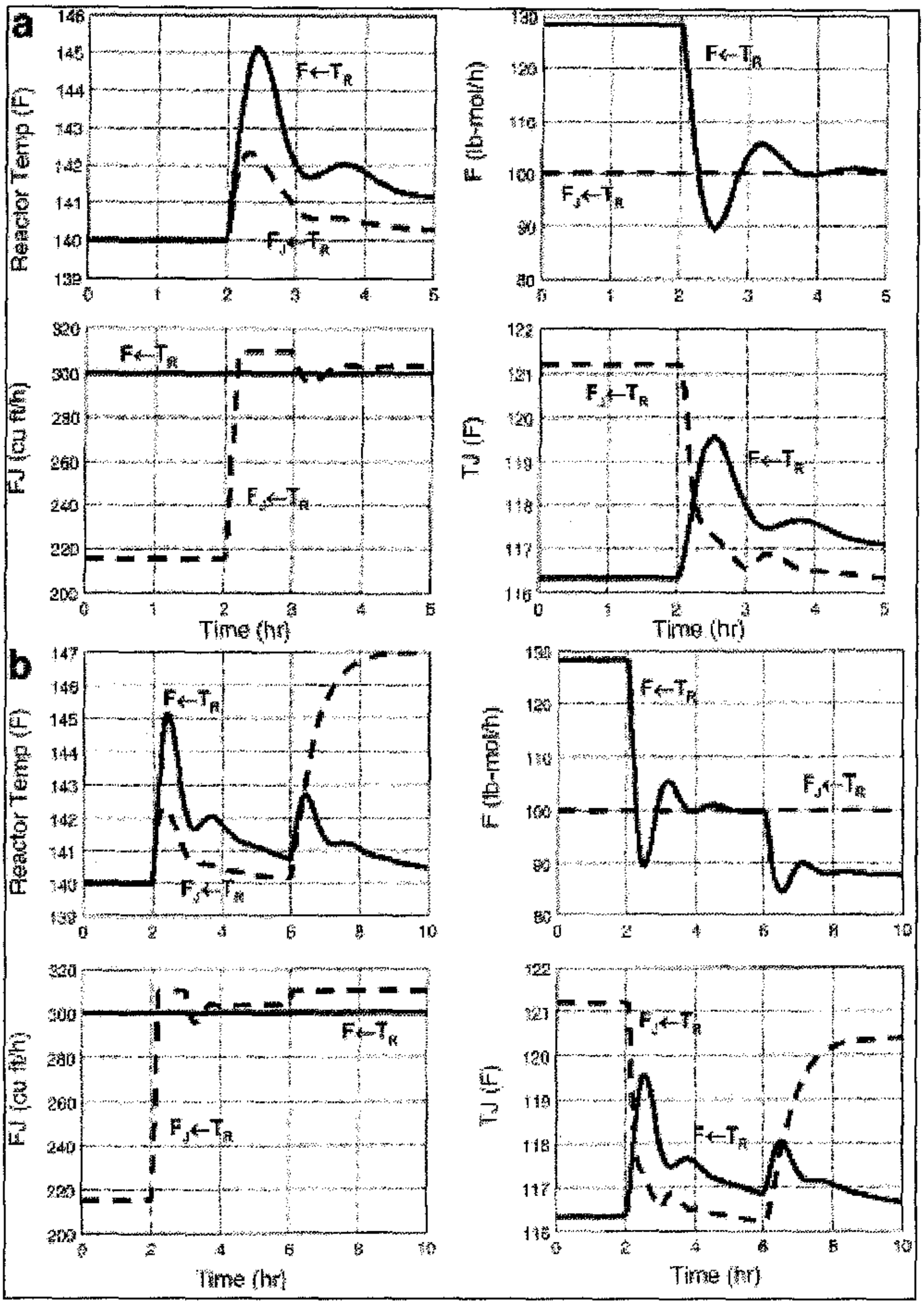


Figure 19: (A) Hybrid Control ; one To disturbance. (B) Comparison of control structure; two T0 disturbances.

CHAPTER 5
CONCLUSION

The control of coolant flow have been explored, via both the PID and neuro-controller. Both controllers achieved the target set point, and met the objectives of the project. The constraint of heat removal was not focused on, since it was assumed within the coolant flow control.

Manipulating the coolant flow guarantees the economic value of the product that is, achieving the highest productivity. Moreover, in case heat removed is so high to the point of steaming the coolant liquid, it can be used to heat other processes as in heat exchangers, instead of disposing the water. The previous point was due to the fact that there are no stand alone processing units, as they usually integrate and interact with various of units on a plant. Thus, the need to consider other units, and their effect on the overall plant operation.

The neural network is quite the promising field, and is still subject to improvement and more exploration. Following the trend of automating all operation, thus cutting cost by decreasing the dependency on labor. Recently, all researches in the field of intelligent measurement and control are focused on such techniques like ANN, fuzzy logic, etc. which might lead to current control schemes as in PID and PLC to be replaced by these intelligent methods, since they provide more accuracy and reliability, and do not require maintenance.

CHAPTER 6
RECOMMENDATIONS

First of all, more disturbances to be included and taken into account. For this project, only the feed temperature was taken as disturbance, the rest of parameters were assumed constant. Other disturbances can be a change in feed concentration, change in jacket temperature or change in fresh feed flow. Moreover, the action of the valve should be accounted for as well, since we are controlling the flow via a valve, we cannot exceed the valve max and min opening.

Second, we can also add more processing units such as: heat exchangers, tanks in series, separators, etc. Because usually, we do not see a standalone processing unit in actual plants. Rather, they interact in so many ways, and different material phases are to be considered. Adding more units means raising system's complexity level, making it a more hectic task to control the whole plant. To make it easier, each unit is taken alone, then we check to see which units are mostly related, or affect each other directly. From there, we can build up the control system to control the whole plant or facility.

Third, other simulation tools such as ASPEN and ANSYS can be utilized, because they account for some properties and specifications in the system (from chemical processing point of view), still beneficial to improve the controllability, with more emphasis on the dynamical behavior. Plus they provide for visual on the system, so it is easier to picture how the process takes place in reality.

Forth, neurocontrol was the proposed scheme, and other schemes can be implemented as well, such as: fuzzy logic, genetic algorithm, etc. the simulation results need to be verified with some experimental work, which will add validity to the results obtained in this project. Also other types of ANN can be used, depending on the complexity of the system structure proposed.

REFERENCES

[1] Schmidt, L. D. (1998). *The Engineering of Chemical Reactions.* New York: Oxford University Press, Inc.

[2] Stephanopoulos, G. (1984). *Chemical Process Control: An Introduction to Theory and Practice*. New Jersey: Prentice Hall.

[3] Malar, R. and Thyagarajan, T. (2009). *Artificial Neural Networks Based Modeling and Control of Continuous Stirred Tank Reactor.* American J. of Engineering and Applied Sciences 2 (1): (pp. 229-235).

[4] Fogler, H. S. (2006). *Elements of Chemical Reaction Engineering 3rd Ed.* (pp. 10-11). New Jersey: Prentice Hall

[5] Bequette, B. W. (May 2002). *Behavior of a CSTR with a Recirculating Jacket Heat Transfer System.* Preceding of the American Control Conference.

[6] Luyben, W. L. (2009). *Chemical Reactor Design and Control.* (pp. 19-20). New Jersey: Wiley-Interscience.

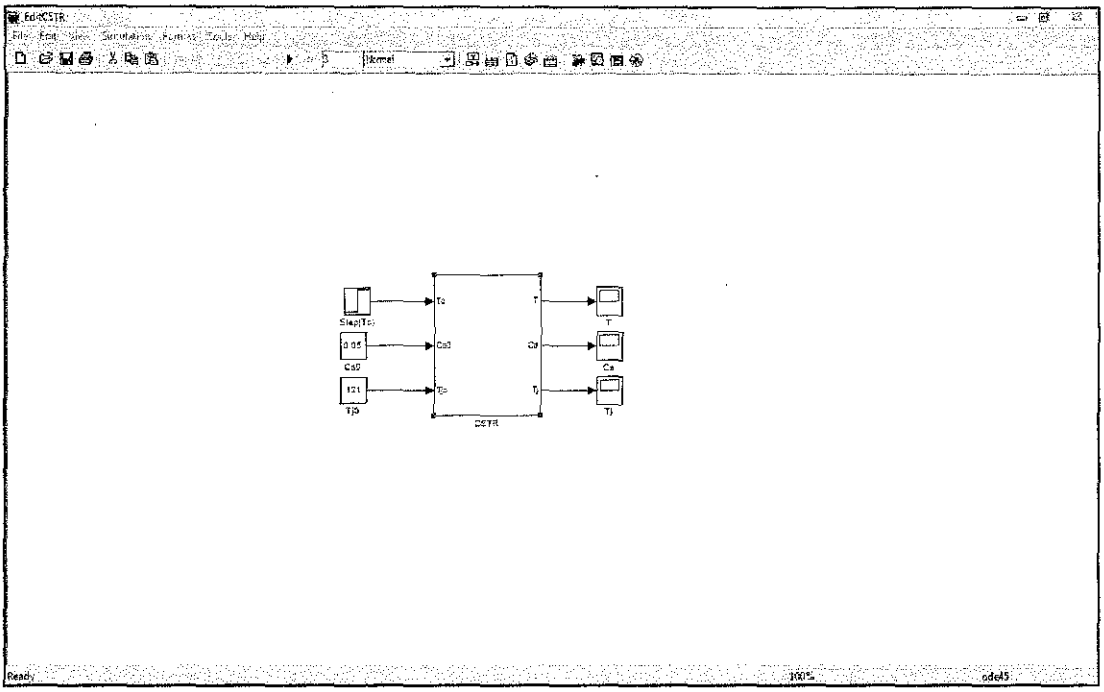
[7] Visioli, A. (2006). *Practical PID Control.* (pp. 3-6). London: Springer Verlag.

[8] Haykin, S. (1999). *Neural networks, a comprehensive foundation.* (2nd Ed., pp. 1-5). New Jersey: Prentice Hall.

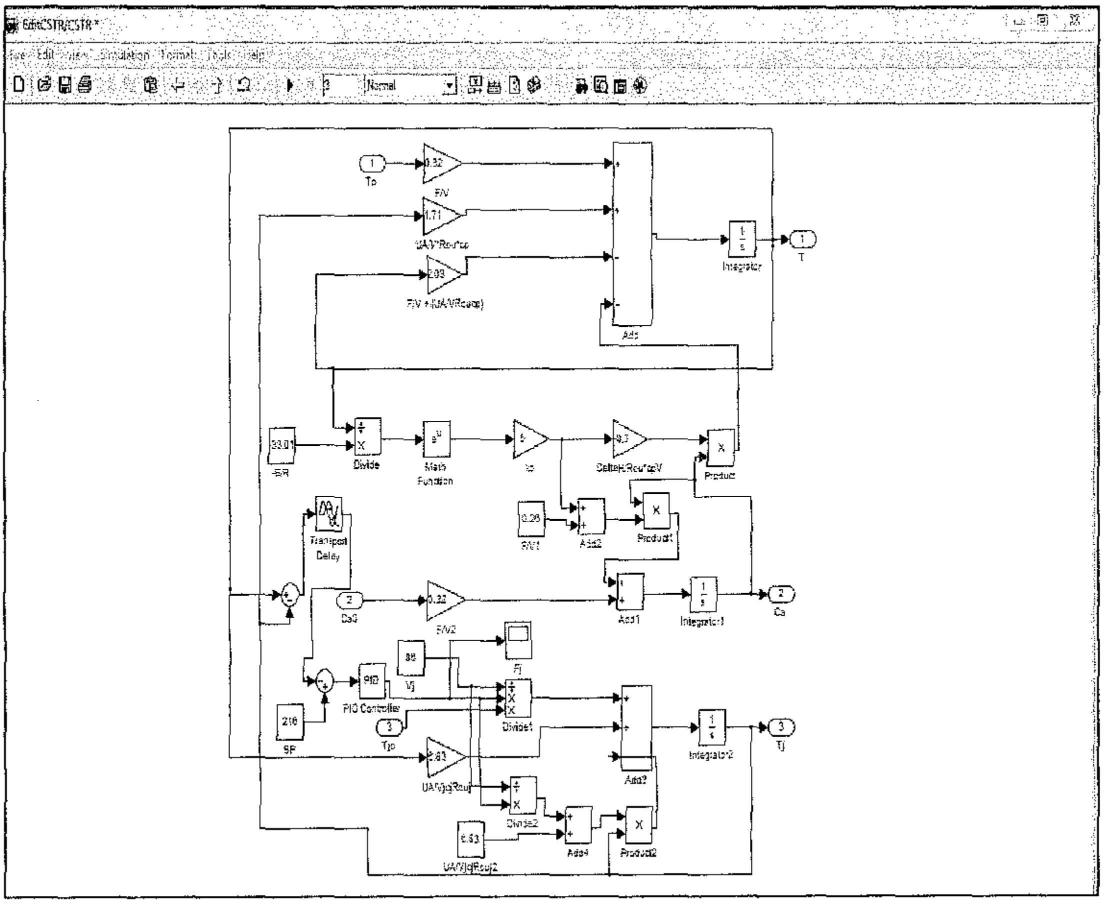
[9] Fausett, L. V. (1994). *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications.* (pp. 3-5). New Jersey: Prentice Hall.

APPENDICES

Appendix A

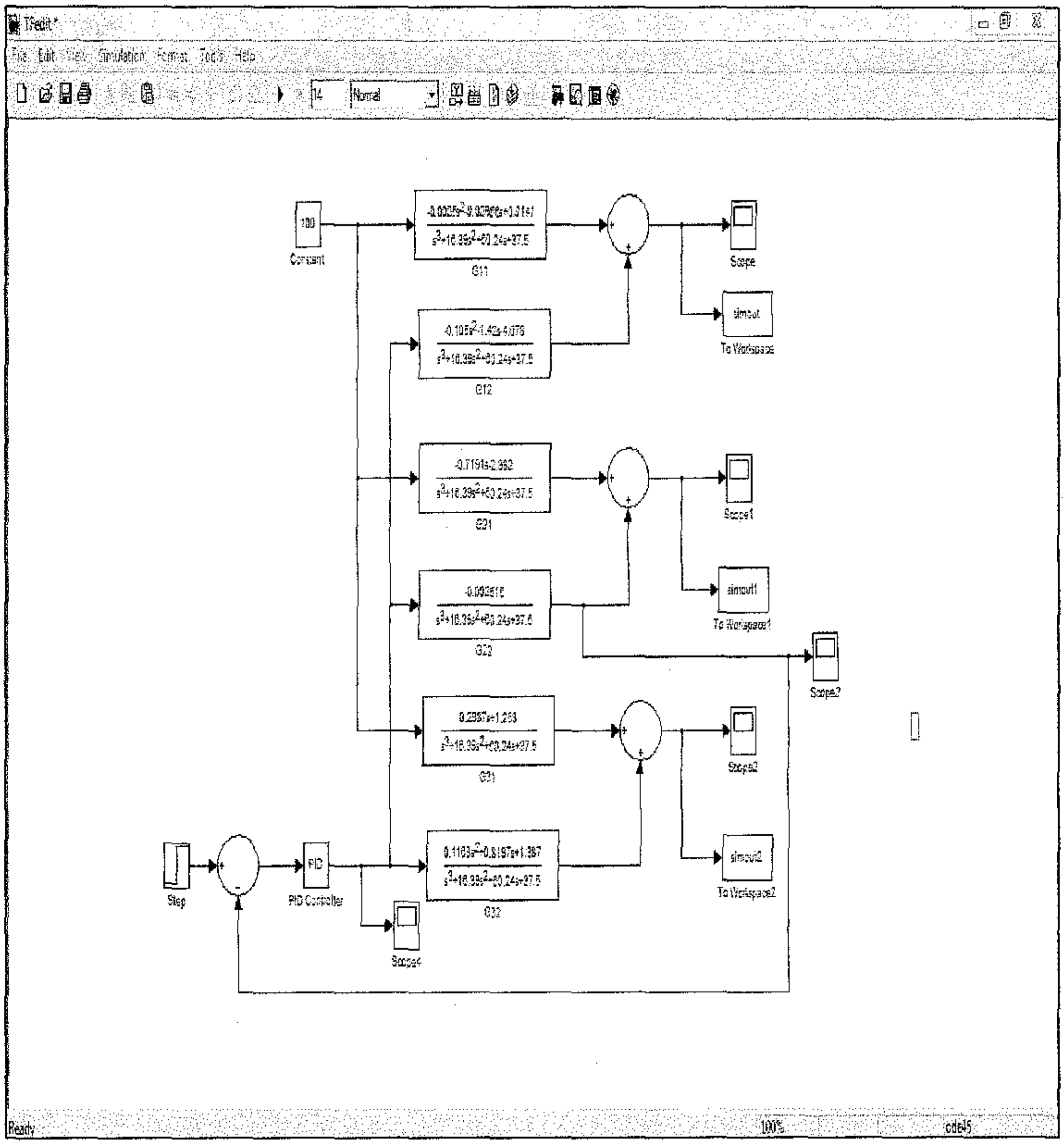


Step(To)
Ca0
Tj0
CSTR
T
Ca
Tj



F/V
Add
Integrator
Divide
Math Function
Product
Product1
Add2
Add1
Integrator1
Transport Delay
PID Controller
SP
Divide1
Add3
Integrator2
Divide2
Add4
Product2

Appendix B



Ready
100%
Constant
Scope
simout
To Workspace
Scope1
simout1
To Workspace1
Step
PID
PID Controller
Scope4

Appendix C

```
function [sys,x0,str,ts] = SfunRLS(t,x,u,flag,...
                                   samTime,Morder,cfac,Ts,erridx)



global count;
global st_y;
global st_x;
global lag;
global max_st;
global Ystore;
global Xstore;
global Inp;
global theta;
global Pmat;
global er;

switch flag,

case 0,
    [sys,x0,str,ts] = mdlInitializeSizes(samTime,Morder);

    if (size(Morder,2) ==  3)
       st_y = Morder(1);
       st_x = Morder(2);
       lag =  Morder(3);
    else
       error('Wrong Morder row vector dimension ');
    end

    if ( (st_y < 0 | st_x <= 0) | lag <= 0)
         error('Vector element should be positive');
    end

    if ( cfac < 0.9 | cfac > 1 )
         error(' Large or incorrect forgeting factor input' );
    end

er = 0;
max_st = max(st_y,st_x);
Ystore = ones(1,(max_st+1)+(lag-1));
Xstore = ones(1,(max_st+1)+(lag-1));
Pmat = eye((st_y+st_x));
Inp = zeros((st_y+st_x),1);
theta = ones((st_y+st_x),1);

count =0;

case 2,
```

```
     sys = mdlUpdate(t,x,u,st_y,st_x,lag,max_st,cfac);

case 3,
    sys = mdlOutputs(t,x,u);

case 9,
    sys = [];

  otherwise
    error(['unhandled flag = ',num2str(flag)]);
end


function [sys,x0,str,ts]=mdlInitializeSizes(samTime,Morder)

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = Morder(1)+Morder(2)+3;
sizes.NumOutputs     = Morder(1)+Morder(2)+3;
sizes.NumInputs      = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

x0  = 0;
str = [];
ts  = [samTime  0]; % Sample period




%
function sys = mdlUpdate(t,x,u,st_y,st_x,lag,max_st,cfac);

global Xstore;
global Ystore;
global Pmat;
global Inp;
global theta;
global count;
global er;

Ystore(2:(max_st+1)+(lag-1)) = Ystore(1:(max_st)+(lag-1));
Xstore(2:(max_st+1)+(lag-1)) = Xstore(1:(max_st)+(lag-1));


if (count >= (max_st+1 + lag-1))

  % start_index = lag+1;
  % End_index = st_? -1;

  Inp = [ Ystore((lag+1):(lag+1)+st_y-1)  Xstore((lag+1):(lag+1)
+st_x-1) ]';
  Pmat = (Pmat- (Pmat*Inp*Inp'*Pmat)/(cfac + Inp'*Pmat*Inp))/cfac;
  er = u(1) - Inp'*theta;
  theta = theta + Pmat*Inp*er;
  yEst = theta'*Inp;
```

```
  LSQE = win_SQE(theta,u(1),Inp);
else
   count = count +1;
   yEst = theta'*Inp;
   LSQE = 1;
end


Ystore(1) = u(1);
Xstore(1) = u(2);
sys = [theta ; u(1); yEst;LSQE];


function sys = mdlOutputs(t,x,u)
sys = x;




function [sys,x0,str,ts] = SfunLMS(t,x,u,flag,...
                                   samTime,Morder,cfac)




global count;
global st_y;
global st_x;
global lag;
global max_st;
global Ystore;
global Xstore;
global Inp;
global theta;
global Pmat;
global er;
global conR;
global Lrate;

switch flag,

case 0,
    [sys,x0,str,ts] = mdlInitializeSizes(samTime,Morder);

    if (size(Morder,2) ==  3)
       st_y = Morder(1);
       st_x = Morder(2);
       lag =  Morder(3);
    else
       error('Wrong Morder row vector dimension ');
    end

    if ( (st_y <= 0 | st_x <= 0) | lag <= 0)
         error('Vector element should be positive');
    end

    if ( cfac < 0.9 | cfac > 1 )
         error(' Large or incorrect forgeting factor input' );
    end
```

```
er = 0;
conR = 0;
max_st = max(st_y,st_x);
Ystore = ones(1,(max_st+1)+(lag-1));
Xstore = ones(1,(max_st+1)+(lag-1));
Pmat = eye((st_y+st_x));
Inp  = ones((st_y+st_x),1);
Lrate = ones((st_y+st_x),1);

theta = ones((st_y+st_x),1);

count =0;

case 2,

     sys = mdlUpdate(t,x,u,st_y,st_x,lag,max_st,cfac);

case 3,
    sys = mdlOutputs(t,x,u);

case 9,
    sys = [];

  otherwise
    error(['unhandled flag = ',num2str(flag)]);
end


function [sys,x0,str,ts]=mdlInitializeSizes(samTime,Morder)

sizes = simsizes;
sizes.NumContStates   = 0;
sizes.NumDiscStates   = Morder(1)+Morder(2)+3;
sizes.NumOutputs      = Morder(1)+Morder(2)+3;
sizes.NumInputs       = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

x0  = 0;
str = [];
ts  = [samTime  0]; % Sample period




%
function sys = mdlUpdate(t,x,u,st_y,st_x,lag,max_st,cfac);

global Xstore;
global Ystore;
global Pmat;
global Inp;
global theta;
global count;
global er;
```

```
global conR;
global Lrate;

Ystore(2:(max_st+1)+(lag-1)) = Ystore(1:(max_st)+(lag-1));
Xstore(2:(max_st+1)+(lag-1)) = Xstore(1:(max_st)+(lag-1));


if (count >= (max_st+1 + lag-1))

  % start_index = lag+1;
  % End_index = st_? -1;

   Inp = [ Ystore((lag+1):(lag+1)+st_y-1)  Xstore((lag+1):(lag+1)
+st_x-1) ]';
   er = u(1) - Inp'*theta;
   conR =  cfac*conR + Inp'*Pmat*Inp;
   Lrate = Inp/conR;
   theta = theta + er*Lrate;
   yEst = theta'*Inp;
   LSQE = win_SQE(theta,u(1),Inp);
else
   count = count +1;
   yEst = 1;
   LSQE = 1;
end


Ystore(1) = u(1);
Xstore(1) = u(2);
sys = [theta ; u(1); yEst;LSQE];

function sys = mdlOutputs(t,x,u)
sys = x;
```