SIMULATION OF DEBUTANIZER USING MATLAB

By

Emira Farzana binti Ellias
15025
Chemical Engineering

Supervisor:
Dr Nasser bin Ramli

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical)

JANUARY 2015

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

SIMULATION OF DEBUTANIZER USING MATLAB

by

Emira Farzana binti Ellias

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
In partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

____

(Dr Nasser Bin Ramli)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2015

CERTIFICATION OF ORGINALITY

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.

EMIRA FARZANA ELLIAS

ACKNOWLEDGEMENT

Above all, I would like to express my gratitude to my supervisor, Dr. Nasser Bin Ramli for his supervision and guidance throughout this project. I appreciate his support and advice that enable this project to achieve its objectives and be completed successfully within the time frame given. I would also like to thank Department of Chemical Engineering of Universiti Teknologi PETRONAS for allocating this project title for me. With this opportunity, I am able to increase my knowledge, both in theory and practical and most importantly, exploring and learning to appreciate the importance of computer programming in enhancing chemical processes.

Last but not the least, I would like to acknowledge my family and friends and those who has directly or indirectly involved in this project, for their tremendous support and motivation during undertaking this project.

ABSTRACT

A debutanizer is a fractional distillation column used to separate butane from gas during the refining process. In this project a model of a debutanizer is developed for simulation purposes. The model can be used to study the steady state and dynamic behavior of the column. The main tool used for modeling and simulation of the debutanizer is MATLAB. Steady state model involves algebraic equations that will be solved by MATLAB functions while MATLAB toolbox ode45 is applied in dynamic model to obtain the solutions of ordinary differential equations. Simulation study was performed on steady state and dynamic state model in order to understand the principle and behaviour of the debutanizer.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.0 Background of Study

A debutanizer is a certain type of fractional distillation column. It is used to separate butane from gas during the refining process. In order to separate or purify the liquid, distillation is used as a process of heating a liquid to vapor and condensing the vapors back to liquid. Some examples would be to include water that is distilled to purify it and also distill liquor to make it stronger. As occurs in a debutanizer, fractional distillation is the separation of a fraction which is a set of compounds that have a boiling point within a range from the rest of the mixture.

From the petroleum distribution, raw natural gas can be drilled from natural deposits or released as a byproduct. In either case, it is not the nearly pure methane used by consumers. About 20 percent of raw natural gas is various heavier hydrocarbons which are chemical compound s made up of hydrogen and carbon, such as butane, propane and ethane. Oxygen, nitrogen, hydrogen sulfide and trace amounts of the noble gases might be mixed within the gas.

As the natural gas is refined, all these other substances are removed in the production phase. Huge fraction columns which are the industrial towers that are about 2-20 feet across and 20-200 feet high or sometimes more will vaporize the gas in an expansion turbine and later condense it with multiple valve trays. The column might be debutanizer, a depropanizer or a deethanizer, depending on which hydrocarbon is being removed from the natural gas liquids (NGL). The natural gas is pure methane when the process is finished. To make leaks more detectible, traces amount of mercaptan which is the source of the rotten egg smell associated with the natural gas are actually added to the methane.

Different refineries might refine the gas in different orders. However, it is very common that the NGL will first flow through a depropanizer to remove heavier propane from the mix and the through the debutanizer to siphon off the butane. There are other refineries that use a debutanizer to remove a mix of butane and propane (C_3/C_4 mix) from the NGL and then use a depropanizer to separate butane and propane from each other.

Updating debutanizers for greater efficiency would be both profitable and eco-friendly. Normal petroleum refinery operation allows the light hydrocarbon gasses to be dissolved into the petroleum. Siphoning off these gases gives the oil company another energy source to market but makes the petroleum more efficient by decreasing plugging and fouling of burner tips. (Hubbard, 2014)

1.1 Problem Statement

The main task of this project is to model and simulate a debutanizer where butane is separated from natural gas using MATLAB software. A MATLAB program will be developed to model and simulate a debutanizer. The project attempts to expand and modify the program in order to broaden its application.

1.2 Objective & Scope of Study

The main objective of the research is to develop a model of a debutanizer using MATLAB program. The program is then used to study its steady state and dynamic behavior of for further understanding. The characteristics and performance of the column can be revealed by observing the steady state and dynamics of the system.

This project provides an opportunity to explore the features of MATLAB software and its importance in engineering study apart from deriving the mathematical model of the column especially in modeling and simulation of process equipment.

This project will be based on data collected from a chosen refinery in Malaysia which is PETRONAS Penapisan (Terengganu) Sdn Bhd : PP(T)SB. LPG is one of the light and top products yield from the refinery's distillation tower. The focus of this project is the debutanizer column in the Crude Distillation Unit (CDU) of Kerteh Refinery 1.

The task of modeling and simulation is given individually and to be completed within 28 weeks (2 semesters). Therefore, the scope of this project has to be clearly defined and should be feasible within the time period. The task is concentrated on observing the behavior of the system using MATLAB software.

For this project, there are main aims in modeling and simulation using MATLAB were identified as below:

1. Mathematical modeling of a debutanizer distillation column.
2. Programming models into MATLAB with appropriate functions.
3. Gaining optimum performance of debutanizer column and maximize LPG production from Crude Distillation Unit (CDU)

CHAPTER 2

LITREATURE REVIEW AND THEORY

2.0 LITREATURE REVIEW

In the recent years, debutanizer has been gaining a growing attention in both industrial and a research sector as the process is economically favourable in many cases. Debutanizer, a fractional distillation equipment, is used to separate butane and lighter components from natural gas liquids. There were not many published on experimentally validated models for debutanizer. The research on this area is still quite new compare to other separation process.

Some of the literature I found is a rigorous model for the simulation of the steady state behavior of the distillation column by (Sohail Rasool Lone 2013). A simulated case study on the debutanizer for advance process control by Morten Bakke (2008) also explains more on the dynamic state of debutanizer. These journal topics are discussed more below.

2.0.1 Modeling and Simulation for Distillation Column of Steady State Behaviour

In this journal by Sohail Rasool Lone (2013), MESH equations which represent the behaviour of the distillation column is used in MATLAB. Methanol is used as feed. The effect of feed condition and the feed composition on the steady state behavior is studied. The mathematical model presented consists of a set of algebraic equations for the column. Assumptions made comply well with the mathematical models. 'Fsolve' solver was used in MATLAB to solve the non-linear equations.

2.0.2 A Simulated Debutanizer Case Study

The main objective of this journal is to control a distillation column (debutanizer). This journal only shows the simulation of the control but does not model the debutanizer. MATLAB is not used in this journal. Instead it uses a Honeywell simulation tool called UniSim. However, this journal stress the focus more on the control theory instead of the tools used for simulation.(Bakke & Skogestad)

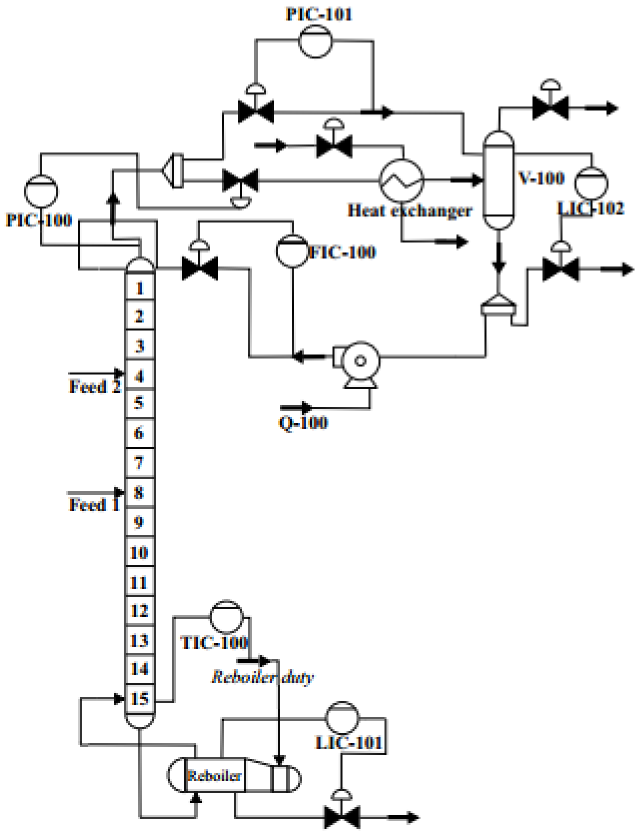


Figure 2.1 : Process Flow Diagram of Debutanizer

2.1 Dynamic Modelling

Dynamic models are used to predict how a process and its controls respond to various upsets as a function of time. They can be used to evaluate equipment conditions and control schemes and to determine the reliability and safety of a design before capital is committed to the project. For grassroots and revamp projects, dynamic simulation can be used to accurately assess transient conditions that determine process design temperatures and pressures. In many cases, unnecessary capital expenditures can be avoided using dynamic simulation.

Dynamic simulation during process design leads to benefits during plant start-up. Expensive field changes, which impact schedule, can often be minimized if the equipment and control system is validated using dynamic simulation. Start-up and shutdown sequences can be tested using dynamic simulation.

Dynamic simulation also provides controller-tuning parameters for use during start-up. In many cases, accurate controller settings can prevent expensive shutdowns and accelerate plant start-up. Dynamic simulation models used for process design are not based on transfer functions as normally found in operator training simulators, but on fundamental engineering principles and actual physical equations governing the process.

When used for process design, dynamic simulation models include:

- Equipment models that include mass and energy inventory from differential balances
- Rigorous thermodynamics based on property correlations, equations of state, and steam tables

- Actual piping, valve, distillation tray, and equipment hydraulics for incompressible, compressible, and critical flow.

These models are so detailed that the results can influence engineering design decisions and ensure a realistic prediction of the process and the control system's interaction to assess control system stability.

2.2 PID CONTROLLER

A proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller attempts to correct the error between a measured process variable and a desired setpoint by calculating and then outputting a corrective action that can adjust the process accordingly and rapidly, to keep the error minimal.

The PID contxoller calculation (algorithm) involves three separate parameters; the proportional, the integral and derivative values. The proportional value determines the reaction to the ~t error, the integral value determines the reaction based on the sum of recent errors, and the der/vat/~ value determines the reaction based on the rate at which the ~ has been changing. The weight~ sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.

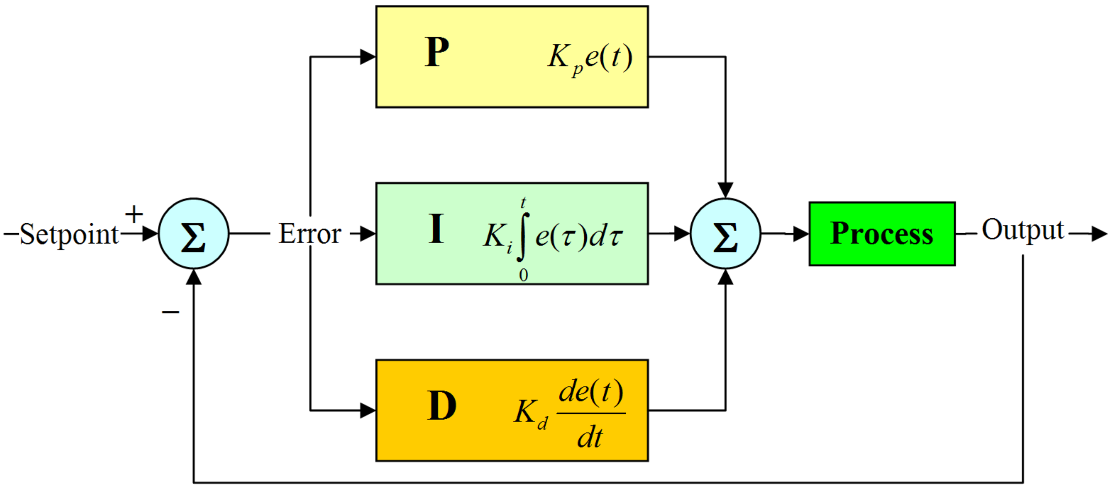


Figure 2.2 : A block diagram of a PID controller

By tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

Some applications may require using only one or two modes to provide the appropriate system control. This is achieved by setting the gain of undesired control outputs to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are particularly common, since derivative action is very sensitive to measurement noise, and the absence of an integral value may prevent the system from reaching its target value due to the control action.

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). Hence:

$$MV(t) = P_{out} + I_{out} + D_{out}$$

where Pout,/out, and Dour are the contributions to the output from the PID controller from each of the three terms, as defined below.

2.2.1 Proportional term

The proportional term (sometimes called gain) makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain.

The proportional term is given by:

$$P_{out} = K_p e(t)$$

Where

- Pout : Proportional term of output
- Kp: Proportional gain, a tuning parameter
- e : Error = SP- PV
- t : Time of instantaneous time (the present)

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, asmall gain results in a small output response to a large input error, and a less responsive (or sensitive) controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

In the absence of disturbances, pure proportional control will not settle at its target value, but will retain a steady state error that is a function of the proportional gain and the process gain. Despite the steady-state offset, both tuning theory and industrial practice indicate that it is the proportional term that should contribute the bulk of the output change.

2.2.2 Integral term

The contribution from the integral term (sometimes called reset) is proportional to both the magnitude of the error and the duration of the error. Summing the instantaneous error over time (integrating the error) gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain and added to the controller output. The magnitude of the contribution of the integral term to the overall control action is determined by the integral gain, Ki.

The integral term is given by:

$$I_{out} = K_i \int_0^t e(\tau) d\tau$$

Where

- Iout : Integral term of output
- Ki : Integral gain, a tuning parameter
- e : Error = SP- PV
- t : Time or instantaneous time (the present)
- τ : A dummy integration variable

The integral term (when added to the proportional term) accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a proportional only controller. However, since the integral term is responding to accumulated errors from the past, it can cause the present value to overshoot the set point value (cross over the set point and then create a deviation in the other direction). For further notes regarding integral gain tuning and controller stability, see the section on loop tuning.

2.3.3 Derivative terms

The rate of change of the process error is calculated by determining the slope of the error over time (i.e., its first derivative with respect to time) and multiplying this rate of change by the derivative gain Ka. The magnitude of the contribution of the derivative term (sometimes called rate) to the overall control action is termed the derivative gain,

The derivative term is given by:

$$D_{out} = K_d \frac{de}{dt}(t)$$

Where

- Dout : Derivative term of output
- Kd : Derivative gain, a tuning parameter
- e : Error = SP- PV
- t : Time or instantaneous time (the present)

The derivative term slows the rate of change of the controller output and this effect is most noticeable close to the controller set point. Hence, derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability. However, differentiation of a signal amplifies noise and thus this term in the controller is highly sensitive to noise in the error term, and can cause a process to become unstable if the noise and the derivative gain are sufficiently large.

2.3 INTERNAL MODEL CONTROL (IMC)

Internal Model Control (IMC) was developed by Morari and coworkers (Garcia and Morari, 1982; Rivera et al., 1986). The IMC method is based on an assumed process model and leads to analytical expressions for the controller settings. The IMC approach has the advantage that it allows model uncertainty and tradeoff between performance and robustness to be considered in a more systematic fashion.

The IMC method is based on the simplified block diagram shown in Figure 2.2:

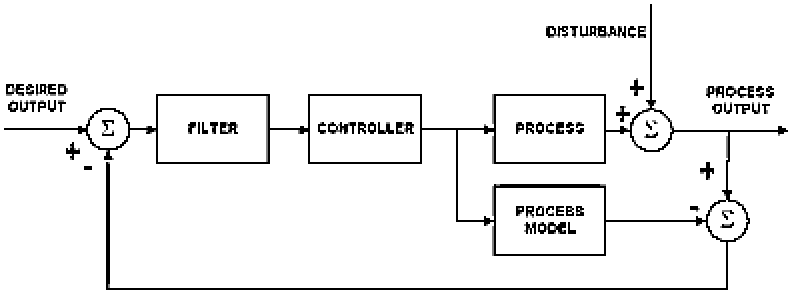


Figure 2.3 : Internal Model Control Scheme

From Figure 2.3, below are the following conventions to describe the blocks in the system:

- controller - Gc(s)
- process - Gp(s)
- internal model - Gpm(s)
- distmbance - d(s)
- digmbance transfer function - D(s)

The Figure 2.3 shows the standard linear IMC scheme where the process model Gpm(s) plays an explicit role in the control structure. This structure has some advantages over convectional feedback loop structures. For the nominal case Gp(s) = Gpm(s), for instance, the feedback is only affected by the disturbance D(s) such that the system is effectively open loop and hence no stability problems can arise. This control structure also depicts that if the process Gp(s) is stable, which is true for most industrial processes, the closed loop will be stable for any stable controller Gc(s). Thus, the controller Gc(s) can simply be designed as a feedforward controller in the IMC scheme.

From the IMC scheme depicted in Figure 2.2 above, the feedback signal is represented as follows:

$$d'(s) = [Gp(s) - Gpm(s)]U(s) + D(s)$$

As said above, if the model is an exact representation of the process then d'(s) is simply a measure of the disturbance. If there exist no disturbance, then d'(s) is simply a measure in difference in behaviour between the process and its model.

The closed loop transfer function of the IMC scheme can be seen modelled as below:

$$Y(s)Z\frac{R(s)G_c(s)G_p(s)H\left[1\vartheta G_c(s)G_{pm}(s)\right]D(s)}{1H\left[G_p(s)\vartheta G_{pm}(s)\right]G_c(s)}$$

The above transfer function will be shown to exist in the next section on transfer functions. From this closed loop analysis, we can see that if we design a controller such that Gc(s) = Gpm(s)$^{-1}$ where the process model is an exact representation of the process, then the design will yield good set point tracking and disturbance rejection.

The controller is then detuned for robustness to account for a possible plant model mismatch. This is done by augmenting the controller with a low-pass filter to reduce the loop gain for high frequencies. This idea also counteracts the effects of model inversion, as the pure inverse of the model is not physically realizable. The inversion often process model may also lead to unstable controllers in case of unstable zeros in the model.
(Source: Donald A. Mohutsiwa (2006), PID controller tuning using Internal Model Control)

2.4 OVERVIEW OF LPG

Liquefied Petroleum Gas (LPG) is a non-renewable source of energy. It is a byproduct which extracted from crude oil and natural gas. The main composition of LPG is a mixture of hydrocarbon gases containing three or four carbon atoms. The normal components of LPG are propane (C_3H_8) and butane (C_4H_{10}). Small concentrations of other hydrocarbons may also be present like propylene (C_3H_6) and butylenes (C_4H_8).

LPG is used as a fuel in heating appliances and vehicles. It replaces chlorofluorocarbons, as an aerosol propellant to reduce ozone layer damaged.

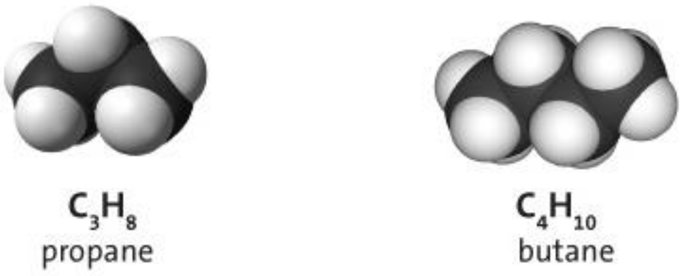


Figure 2.4 : Molecule of Propane and Butane

LPG is a vapor at atmospheric pressure and normal ambient temperatures which can be liquefied by compression at ambient temperature to enhance the molecular distribution. It is one of the cleanest fuels available and can be easily condensed, packaged, stored and utilized, which makes it an ideal energy source for a wide range of applications. In Malaysia, it is used extensively as fuel for domestic cooking, agricultural, heating and drying process and vehicle as well as for industrial applications.

In fact, LPG can be liquefied at relatively low pressure approximately at 2.2 bar which approaching 2.17 atm pressure; it facilitates the storage of large amount in spherical tank or cylindrical tank. This is because LPG in liquid state is 250 times denser than when in the gas state. For an ideal gas, the general equation of state is as the following:

$$PV=RT$$

Where, P is absolute pressure; V is molal volume; T is temperature and R is gas constant

In low pressure case, gases will depart slightly from ideality, a compressibility factor,z is then introduced into the equation of state. At high pressure, the gas has low molal volume which slightly less than the expected volume for ideal gas. Therefore, to liquefy the gas, increment of pressure is required. The LPG density will be higher at liquid state since it's volume reduced and mass remains constant.

LPG is highly flammable and must therefore be stored away from sources of ignition and in a well-ventilated area, so that any leak can disperse safely. A special chemical; mercaptan, is added to give distinctive to LPG; an unpleasant smell so that a leak can be detected. The concentration of the chemical is such that an LPG leak can be smelled when the concentration is well below the lower limit of flammability.

The composition chart in Figure 2.5 below outlines the typical components of LPG as supplied via Gas Malaysia's distribution system.

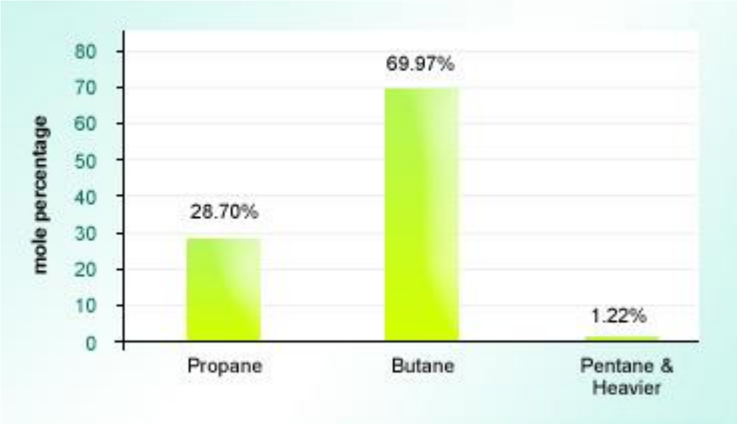


Figure 2.5: Chemical Composition of LPG Supplied by Gas Malaysia Sdn. Bhd

2.6 PRODUCTION OF LPG IN PETRONAS PENAPIS~J~I (TERENGGANU) SDN. BHD.

LPG Recovery Unit is designed to process the unstabilized product from Crude Tower together with unstabilized LPG from Reformer Unit. The designed capacity of mixed LPG is 1180 BPSD. Crude Distillation Unit (CDU), Catalytic Reforming Unit (CRU) and Condensate Fractionation Unit (CFU) are the sources of throughput for LPG Recovery Unit. The fractionation section of LPG Recovery Unit consists of a Deethanizer column and Debutanizer column. The function of this fractionation section is to recover light gaseous and LPG from the overhead distillate before producing Light Naphtha. The light gases mainly ethane (C2) from Deethanizer is muted to Refinery Fuel Gas System. Mixed LPG which is a mixture of propane and butane are sent to LPG storage.

2.7 LPG PRODUCTION FROM CRUDE DISTILLATION UNIT (CDIY)

The throughput of Debutanizer column is the bottom product of Deethanizer. The Debutanizer column is equipped with 35 valve type trays and one liquid pass. The low boiling point components rise up the tower in contact with the internal reflux.

In contrast, the high boiling point components, which are the heavier components, flow down in contact with the vapor produced in the Debutanizer reboiler which provided to the reboiler bottom level. Thus, hydrocarbon feed to the Debutanizer will be fractionated to the lighter components; mixture of C3 and C4 as the overhead product and heavier components as the bottom product of the column.

The Debutanizer condenser condensed the overhead vapor. Part of the stream may by pass from the condenser in order to control the overhead pressure. The condensed overhead is collected inside the Debutanizer receiver drum. The Debutanizer overhead system is set and controlled at 8.0 kg/cm2 (7.85 bars) by the Debutanizer overhead pressure control valve which has two split range controllers. While if the pressure low, the other valve will be opened and by passed part of the overhead gas from the Debutanizer condenser. Meanwhile, if the pressure of the system increases too much, manual valve should be opened and routed to the flare. In case of emergency loss of fuel gas system, the off gas will be routed to the Refinery Fuel Gas System by opening the tie in block valves.

Part of the condensed hydrocarbon collected is pumped by the Debutanizer reflux pump to the Debutanizer top as reflux. The reflux flow rate is controlled by the Debutanizer overhead reflux control valve. The balanced can be routed to flare the stock if the LPG product is off specification.

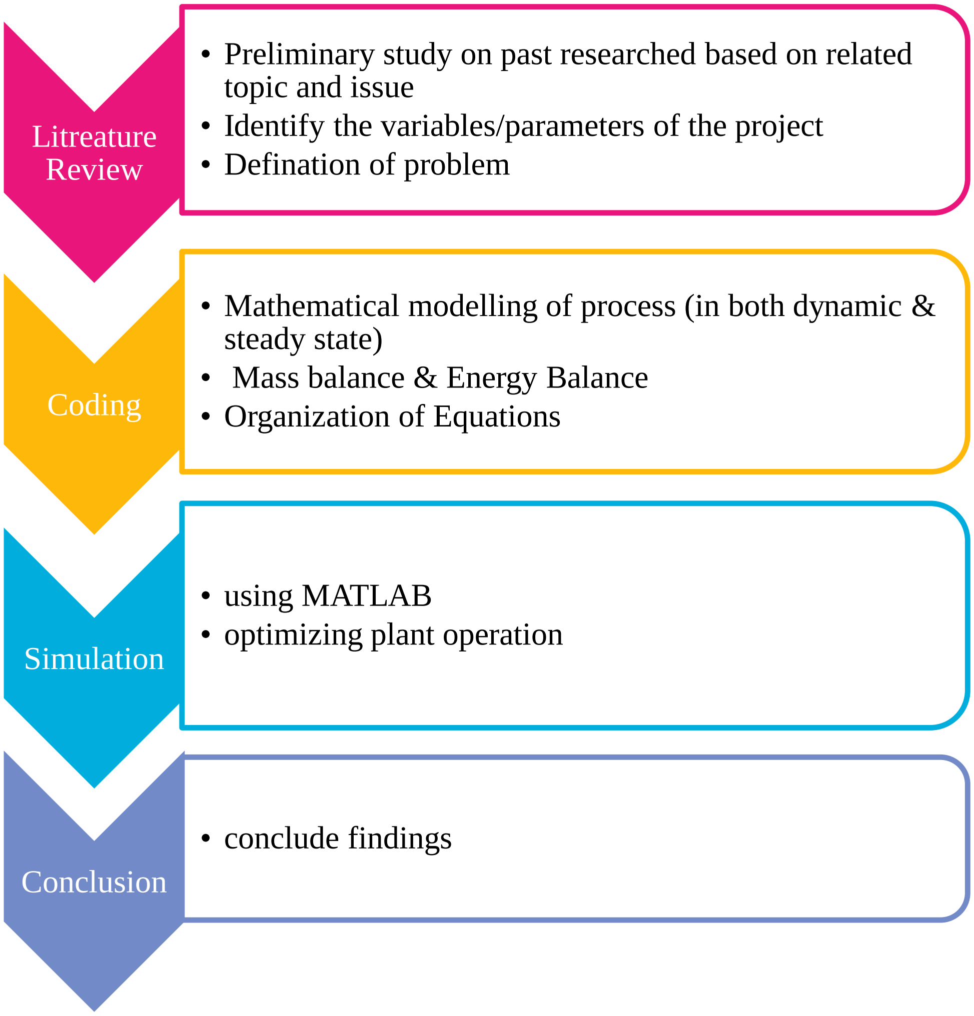
The Debutanizer reboiler is equipped to the Debutanizer bottom section which is heated by the Kerosene pump around system. The Debutanizer reboiler temperature control valve controls the reboiler temperature. While the Debutanizer bottom level controller controls the bottom product level.

Bottom product is then muted via Light Naphtha rundown cooler to Light Straight Run Naphtha rundown tank or to slop if off specification. Light Straight Run Naphtha (LSRN) rundown rate is measured by LSRN flow meter.

CHAPTER 3

METHODOLOGY

3.1 Project Flow Chart



3.2 Data Collection

After the problem is clearly defined, the relevant data is identified and gathered. Simulation using MATLAB will be performed in order to know which data is needed. Data is collected from supervisor with the help of an engineer working at PETRONAS Penapisan (Terengganu) Sdn. Bhd. (PP(T)SB).

3.2.1 Debutanizer Column (C-110)

Number of trays of the column	35
Feed tray – stage number	23
Type of tray used	Valve
Column diameter	1.3m
Column length	23.95m
Type of Condenser	Partial
Feed mass flow rate	44 106 kg/hr
Feed temperature	113 C
Feed pressure	823.8 kPa
Overhead vapor mass flow rate	11 286 kg/hr
Overhead liquid mass flow rate	5040 kg/hr
Pressure condenser	823.8 kPa
Pressure Reboiler	853.2 kPa

Table 3.1 : Debutanizer column plant data

3.2.2 Composition in the feed in mass fraction including components in the feed

Liquefied Petroleum Gas (LPG) and LSRN

Composition	Mass Fraction
Propane	0,037
i-Butane	0,093
n-Butane	0,062
i-Pentane	0,082
n-Pentane	0,110

3.2.3 Operational Parameters

3.2.3.1 TC-110

Mode	Auto
Action	Reverse
SP	140.7 C
OP	52.00%
Kc	250
Ti	1.33 minutes
Td	0.333 minutes
PV Minimum	125.15 C
PV Maximum	145.55 C

3.2.3.2 FIC-123

Mode	Auto
Action	Reverse
SP	19.37m3/h
OP	74.2%
Kc	0.1
Ti	0.5 minutes
Td	-
PV Minimum	19.37 m3/h
PV Maximum	56.40 m3/h

3.2.3.3 PC-109

Mode	Auto
Action	Reverse
SP	823.8kPa
OP	25.30
Kc	0.5
Ti	0.7 minutes
Td	-
PV Minimum	552.6 kPa
PV Maximum	903.58 kPa

3.2.3.4 LC-111

Mode	Manual
Action	Reverse
SP	60.65 %
OP	60.00 %
Kc	0.45
Ti	11 minutes
Td	-
PV Minimum	45.4 %
PV Maximum	75.76 %

3.2.3.5 LC-112

Mode	Manual
Action	Reverse
SP	50.00%
OP	60.00%
Kc	0.25
Ti	15 minutes
Td	-
PV Minimum	25%
PV Maximum	73%

3.2.3.6 FIC-126

Mode	Cascade
Action	Reverse
SP	8.8206 m3/h
OP	54.30%
Kc	0.2
Ti	0.2 minutes
Td	-
PV Minimum	0.00 m
PV Maximum	15.8

3.2.3.7 FC-121

Mode	Manual
Action	Reverse
SP	515.9 m3/h
OP	50.00
Kc	0.028
Ti	0.417 minutes
Td	-
PV Minimum	379 m3/h
PV Maximum	721 m3/h

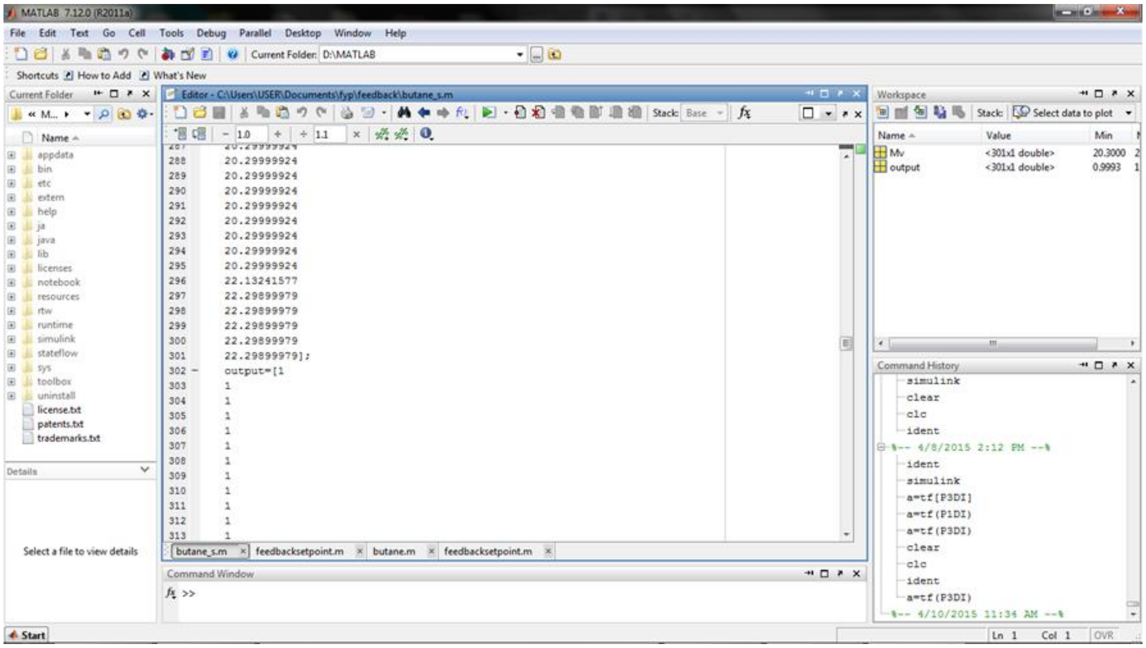
3.3 On-Line Controller Tuning

3.3.1 Step Test Method

After the process has reached steady state (at least approximately), the controller is placed in the manual mode. Then a step change in the controller output (e.g 52 to 82%) is introduced. The controller settings are based on the closed |oop response.

3.3.1.1 Process model parameter estimation using MATLAB

1. Insert the step change data (process variables, PV and controller output, OP) in M-File.



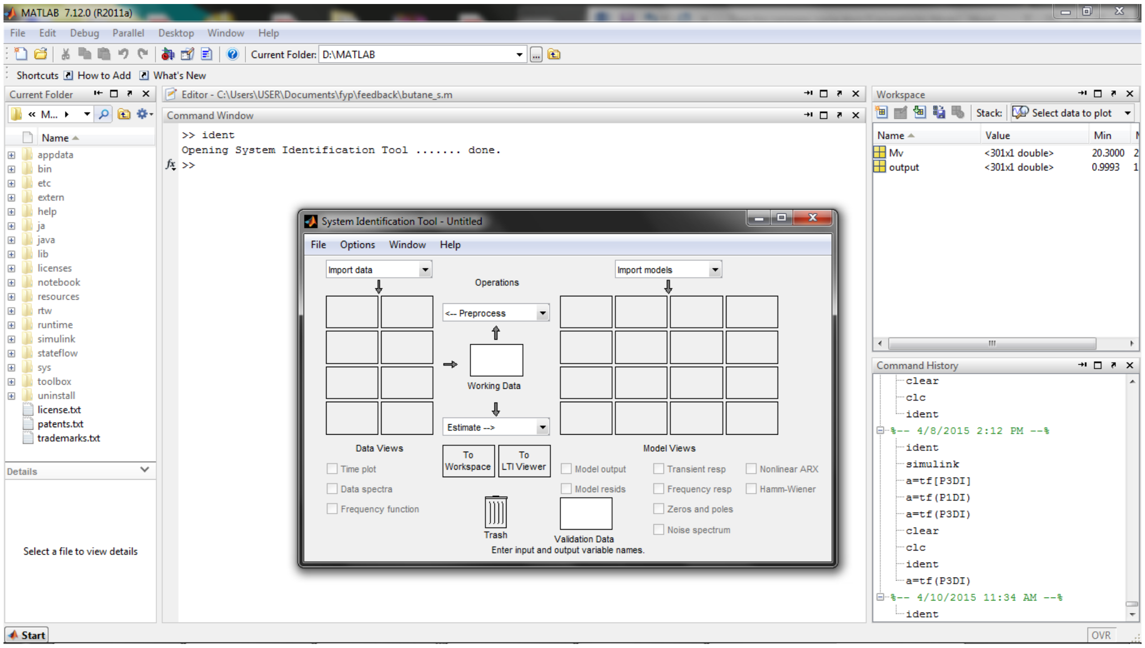
```
20.29999924
20.29999924
20.29999924
20.29999924
20.29999924
20.29999924
20.29999924
20.29999924
22.13241577
22.29899979
22.29899979
22.29899979
22.29899979
22.29899979];
output=[1
1
1
1
1
1
1
1
1
1
1
1
```

```
>>
```

Name	Value	Min
Mv	<301x1 double>	20.3000
output	<301x1 double>	0.9993

```
simulink
clear
clc
ident
%-- 4/8/2015 2:12 PM --%
ident
simulink
a=tf[P3DI]
a=tf(P1DI)
a=tf(P3DI)
clear
clc
ident
a=tf(P3DI)
%-- 4/10/2015 11:34 AM --%
```

2. Open Ident

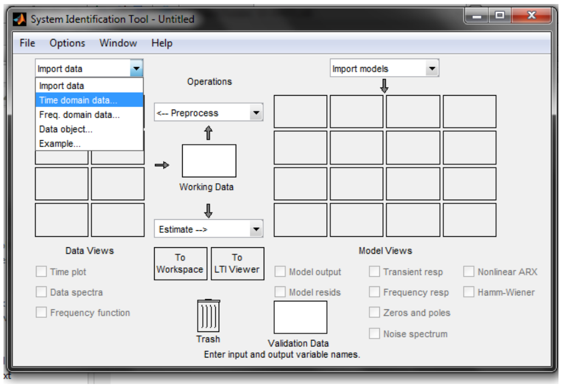


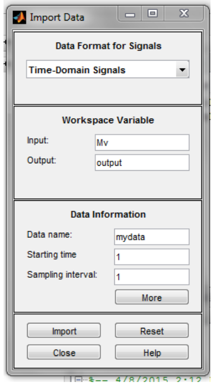
```
>> ident
Opening System Identification Tool ....... done.
>>
```

Name	Value	Min
Mv	<301x1 double>	20.3000
output	<301x1 double>	0.9993

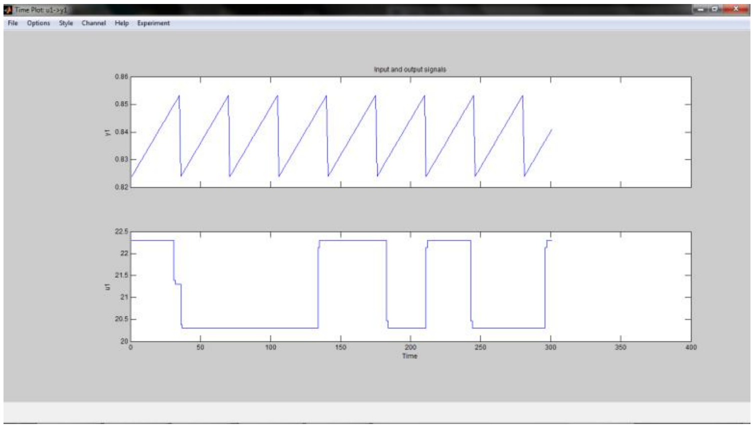
```
clear
clc
ident
%-- 4/8/2015 2:12 PM --%
ident
simulink
a=tf[P3DI]
a=tf(P1DI)
a=tf(P3DI)
clear
clc
ident
a=tf(P3DI)
%-- 4/10/2015 11:34 AM --%
ident
```

3. Set time domain data

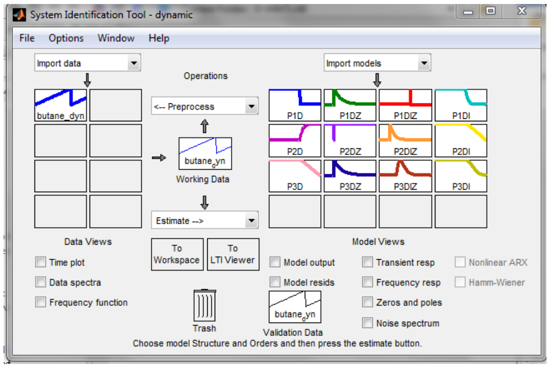




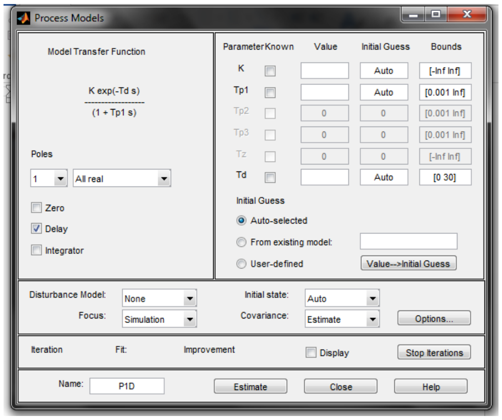
Input: Mv
Output: output
Data name: mydata
Starting time: 1
Sampling interval: 1



4. Estimate process models



Choose model Structure and Orders and then press the estimate button.



$$\frac{K \exp(-Td\ s)}{(1 + Tp1\ s)}$$

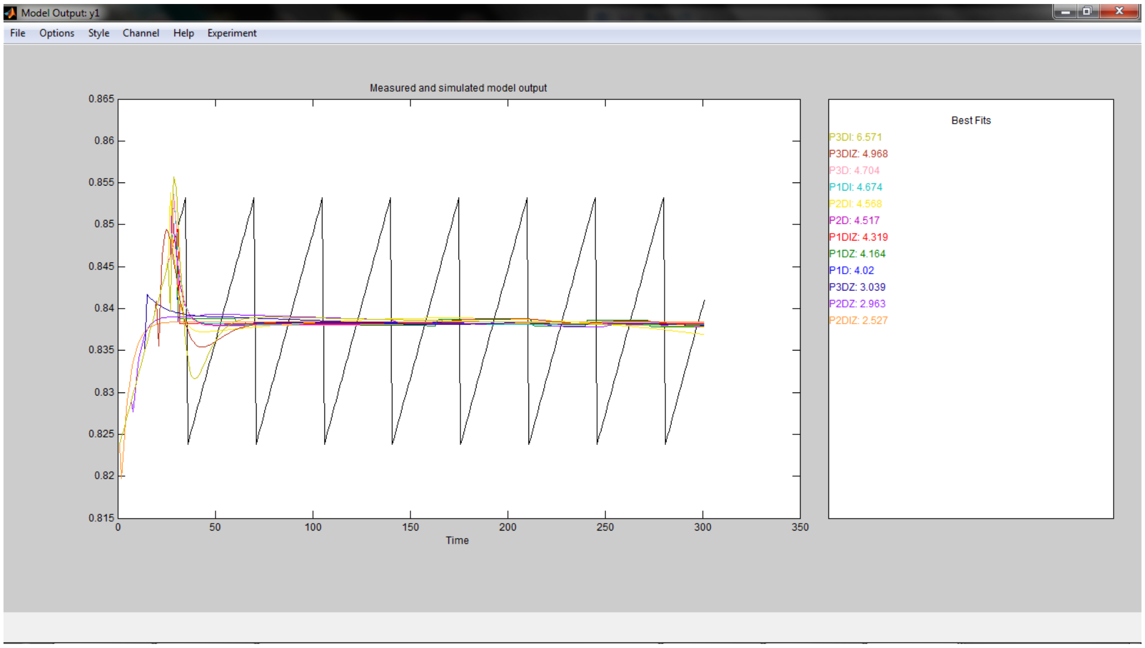
Poles: 1, All real

☐ Zero
☑ Delay
☐ Integrator

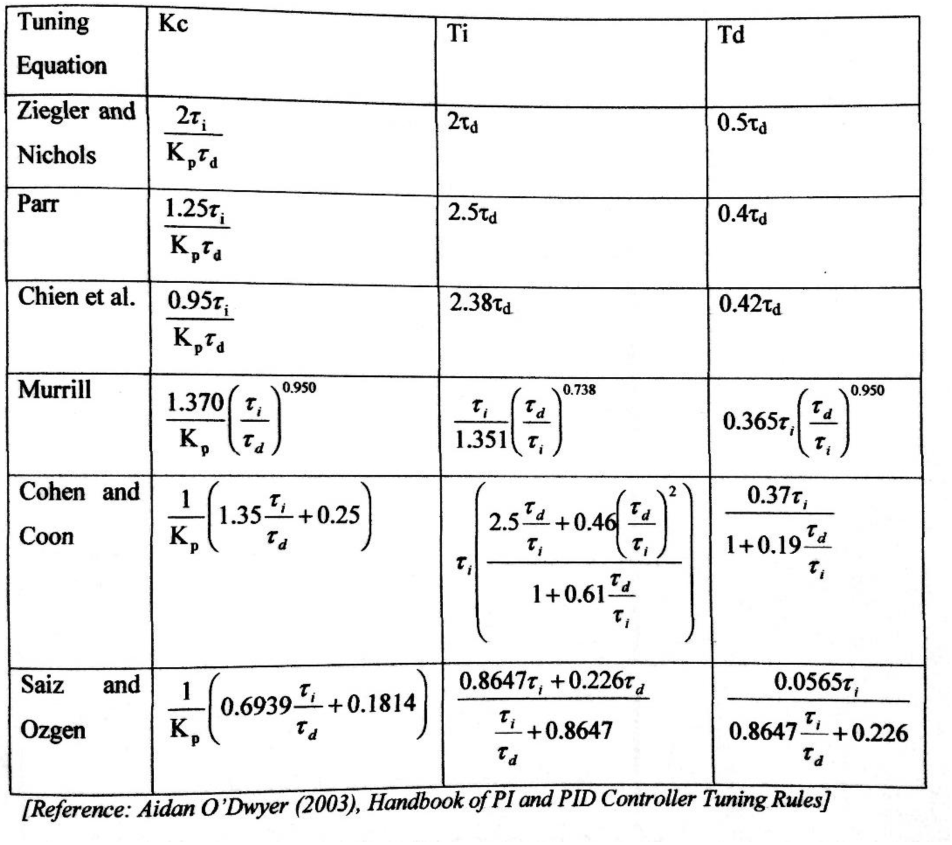
Parameter	Known	Value	Initial Guess	Bounds
K	☐		Auto	[-Inf Inf]
Tp1	☐		Auto	[0.001 Inf]
Tp2	☐	0	0	[0.001 Inf]
Tp3	☐	0	0	[0.001 Inf]
Tz	☐	0	0	[-Inf Inf]
Td	☐		Auto	[0 30]

Initial Guess
● Auto-selected
○ From existing model:
○ User-defined

Disturbance Model: None
Focus: Simulation
Initial state: Auto
Covariance: Estimate
☐ Display
Name: P1D



3.4 PID Controller Settings using various tuning equations



Tuning Equation	Kc	Ti	Td
Ziegler and Nichols	$\frac{2\tau_i}{K_p\tau_d}$	$2\tau_d$	$0.5\tau_d$
Parr	$\frac{1.25\tau_i}{K_p\tau_d}$	$2.5\tau_d$	$0.4\tau_d$
Chien et al.	$\frac{0.95\tau_i}{K_p\tau_d}$	$2.38\tau_d$	$0.42\tau_d$
Murrill	$\frac{1.370}{K_p}\left(\frac{\tau_i}{\tau_d}\right)^{0.950}$	$\frac{\tau_i}{1.351}\left(\frac{\tau_d}{\tau_i}\right)^{0.738}$	$0.365\tau_i\left(\frac{\tau_d}{\tau_i}\right)^{0.950}$
Cohen and Coon	$\frac{1}{K_p}\left(1.35\frac{\tau_i}{\tau_d}+0.25\right)$	$\tau_i\left(\frac{2.5\frac{\tau_d}{\tau_i}+0.46\left(\frac{\tau_d}{\tau_i}\right)^2}{1+0.61\frac{\tau_d}{\tau_i}}\right)$	$\frac{0.37\tau_i}{1+0.19\frac{\tau_d}{\tau_i}}$
Saiz and Ozgen	$\frac{1}{K_p}\left(0.6939\frac{\tau_i}{\tau_d}+0.1814\right)$	$\frac{0.8647\tau_i+0.226\tau_d}{\frac{\tau_i}{\tau_d}+0.8647}$	$\frac{0.0565\tau_i}{0.8647\frac{\tau_i}{\tau_d}+0.226}$

[Reference: Aidan O'Dwyer (2003), Handbook of PI and PID Controller Tuning Rules]

CHAPTER 4

RESULTS AND DISCUSSION

4.1. The Model of Debutanizer Column

The debutanizer column will be employed for the separation of an four-component hydrocarbon mixture. The rigorous process model consists of a set of ordinary dlfferential equations (ODEs) coupled with algebraic equations/correlations The ODEs are obtained from mass and energy balances around each plate of the distillation column. The algebraic equations/ correlations are used to predict the thermodynamic and physical properties, plate hydraulics, and actual vapour-phase compositions.

The following assumptions have been made in the development of the process model:

- The molar vapour holdup is negligible compared to the molar liquid holdup.
- The liquid and vapour leaving each plate are in thermal equilibrium.
- The definition of Murphree plate efficiency applies for each plate.
- The liquid is perfectly mixed on each plate.
- No subcooling is considered in the total condenser.
- Coolant and steam dynamics in the condenser and reboiler respectively are neglected.
- Molal liquid holdup varies in each tray (including reflux drum and column base) but the holdups in reflux drum and column base are constant in volume.
- The column operates with the top (Pt) and base (PB) pressures of 102.98 and 122.70 psia, respectively. The tray-to-tray pressure varies linearly according to the following form of equation:

$$P_n = P_B - \frac{n(P_B - P_T)}{n_T + 1}$$

Where Pn is the pressure in the nth tray and nT represents the total number of trays (here 35).

- Liquid hydraulics are calculated from the Francis weir formula.
- Vapour-liquid equilbria (VLE) and enthalpy are calculated based on the Soave-Redlich-Kwong (SRK) equation of state.
- The Muller iteration method based on second degree equation is used in the buble-point calculations.

4.2 Material and Energy Balance Equations

For all column trays including a reboiler and a reflux drum, and for a component i= 1,.., Nc, the model ordinary differential equations representing total continuity (one per tray), component continuity (Nc-1 per tray) and energy balance (one per tray) can be obtained. However for this simulation we will focus only on :

Reboiler-Column Base System

Total Continuity:

$$m_B = L_1 - V_B - B$$

Component Continuity:

$$m_B x_{B,i} = L_1 x_{1,i} - V_B y_{B,i} - B x_{B,i}$$

Dynamic State:

$$\frac{dx_{B,i}}{dt} = \frac{1}{m}\left[L_1 x_{1,,i} + V_B y_{B,i} - B x_{B,i}\right]$$

Bottom Tray (subscript ‘1’)

Total Continuity:

$$m_1 = L_2 + V_B - L_1 - V_1$$

Component Continuity:

$$m_1 x_{1,i} = L_2 x_{2,i} + V_B y_{B,i} - L_1 x_{1,i} - V_1 y_{1,i}$$

Dynamic State :

$$\frac{dx_{1,i}}{dt} = \frac{1}{m}\left[L_2 x_{2,i} + V_B y_{B,i} - L_1 x_{1,i} - V_1 y_{1,i}\right]$$

Nth Tray (subscript ‘n’ where to and to)

Total Continuity :

$$m_n = L_{n+1} + V_{n-1} - L_n - V_n$$

Component Continuity :

$$m_n x_{n,i} = L_{n+1} x_{n+1,i} + V_{n-1} y_{n-1,i} - L_n x_{n,i} - V_n y_{n,i}$$

Dynamic State :

$$\frac{dx_{n,i}}{dt} = \frac{1}{m}\left[L_{n+1} x_{n+1,i} + V_{n-1} y_{n-1,i} - L_n x_{n,i} - V_n y_{n.i}\right]$$

Feed Tray (subscript ‘n$_F$’)

Total Continuity:

$$m_{nF} = L_{nF+1} + F^L + V_{nF+1} - L_{nF} - V_{nF}$$

Component Continuity:

$$m_{nF} x_{nF,i} = L_{nF+1} x_{nF+1,i} + F^L x_{F,i} + V_{nF-1} y_{nF-1,i} - L_{nF} x_{nF,i} - V_{nF} y_{nF,i}$$

Dynamic State :

$$\frac{dx_{nF,i}}{dt} = \frac{1}{M}\left[L_{nF+1} x_{nF+1,i} + F^L x_{F,i} + V_{nF-1} y_{nF-1,i} - L_{nF} x_{nF,i}\right]$$

Above Feed Tray (subscript ‘’nF+1’)

Total Continuity:

$$m_{nF+1} = L_{nF+2} + F^V + V_{nF} - L_{nF+1} - V_{nF+1}$$

Component Continuity:

$$m_{nF+1} x_{nF+1,i} = L_{nF+2} x_{nF+2,i} + F^V y_{F,i} + V_{nF} y_{nF,i} - L_{nF+1} x_{nF+1} - V_{nF+1} y_{nF+1,i}$$

Dynamic State :

$$\frac{dx_{nF+1,i}}{dt} = \frac{1}{m}\left[L_{nF+2} x_{nF+2} + V_{nF} y_{nF,i} - L_{nF+1} x_{nF+1,i} - V_{nF+1} y_{nF+1,i}\right]$$

Top Tray (subscript ’nT’)

Total Continuity:

$$m_{n\tau} = R + V_{n\tau-1} - L_{n\tau} - V_{n\tau}$$

Component Continuity:

$$m_{n\tau}x_{n\tau,i} = Rx_{D,i} + V_{n\tau-1}y_{n\tau-1,i} - L_{n\tau}x_{n\tau,i} - V_{n\tau}y_{n\tau,i}$$

Dynamic State:

$$\frac{dx_{n\tau,i}}{dt} = \frac{1}{m}\left[Rx_{D,i} + V_{n\tau-1}y_{n\tau-1,i} - L_{n\tau}x_{n\tau,i} - V_{n\tau}y_{n\tau,i}\right]$$

Condenser-Reflux Drum System (subscript ‘D’)

Total Continuity:

$$m_D = V_{n\tau} - R - D$$

Component Continuity:

$$m_D x_{D,i} = V_{nT}y_{nT,i} - (R + D)x_{D,i}$$

Dynamic State:

$$\frac{dx_{D,i}}{dt} = \frac{1}{m}\left[V_{nT}y_{nT,i} - (R + D)x_{D,i}\right]$$

In the above distillation modeling equations:

$x_{n,i}$ = the mole fraction of component i in liquid stream leaving nth tray

$y_{n,i}$ = the mole fraction of component i in a vapour stream leaving nth tray

$x_{F,i}$ = the mole fraction of component i in the liquid feed,

$y_{F,i}$ = the mole fraction of component i in the vapour feed,

$x_{D,i}$ = the mole fraction of component i in the distillate.

$x_{B,i}$ = the mole fraction of component i in the bottom product,

L_n = the liquid flow rate leaving nth tray (lbmol/h),

V_n = the vapour flow rate leaving nth tray 0bmol/h),

$x_{D,i}$ = the mole fraction of component i in the distillate.

V_n = the vapour flow rate leaving nth tray

R = the reflux flow rate (lbmol/h),

D = the distillate flow rate (lbmol/h),

V_B = the vapour boil-up rate (lbmol/h),

B = the bottom product flow rate (lbmol/h),

F^L = the flow rate of the liquid feed (lbmol/h),

F^V = the flow rate of the vapour feed (lbmol/h),

m_n = the liquid holdup on the nth tray (lbmol),

m_D = the liquid holdup in the reflux drum (lbmol),

m_B = the liquid holdup in the reboiler-column base system (lbmol)

Q_R = the heat input to the reboiler (Btu/h),

N_c = the number of components (here 4) and

n_F = the feed tray.

4.3 Tray Holdup Dynamics

Distefano (1968) has reported a procedure for tray holdup calculations in ditillation columns. This approach formulates based on the assumption of constant volume holdup on the plates. Therefore, the molal holdup on nth plate is given by: where, is the constant volumetric liquid holdup (volume) on any plate and represent the average density (mass/volume) and average molecular weight, respectively, of the liquid stream on nth plate.

In real-time distillation processes, the volumetric liquid holdups (or heights) in the reflux drum and column base are held almost constant by implementing conventional level controllers (proportional) to the manipulation of distillate and bottom product flow rates, respectively. But this is not the ease for the internal trays. Distefano (1968) has suggested to use the Equation for all trays of a batch distillation column except the still-pot. Again Luyben (1990) preferred to simulate the total continuity equations for the calculation of liquid holdups on the internal trays of a multicomponent continuous distillation column. But he considered constant volumetric liquid holdups in the condenser-accumulator as well as in the reboiler - column base system. In order to predict the holdup dynamics in the condenser-reflux drum and reboiler column base systems, the molal holdups may be represented according to the approach explained above as:

For Condenser-Reflux Drum System

$$m_D = \frac{mv_D \rho_{LD}}{MW_D}$$

For Reboiler-Column Base System

$$m_B = \frac{mv_B \rho_{LB}}{MW_D}$$

Where, subscripts 'D' and 'B' refer, respectively to the distillate and bottoms. For this project, debutanizer column Equations will be used for the computations respectively and internal tray holdups will be calculated simulating the total continuity equations.

4.4 PID Controller Settings using Various Tuning Equations

Error is then calculated for each tuning to choose the best performance tuning. The error data is tabulated below:

No	Type of Tuning	Error Value Calculation
1	IAE	7.645×10^{-6}
2	ITAE	3.112×10^{-5}
3	IMC	0.0018
4	HA	1.2775×10^{-5}
5	ISE	6.5215×10^{-6}
6	Cohen Coon	2.4481×10^{-7}
7	Ziegler and Nichols (1942) Model Method 2	9.9733×10^{-7}
8	Hazebroek and Van der Waerden(1950),Model Method 2	4.5997×10^{-7}
9	Chien(1952), Servo, Model: Method 2, 0% overshoot	3.6639×10^{-6}
10	Chien(1952), Servo, Model: Method 2, 20% overshoot	7.3022×10^{-6}
11	Cohen and Coon (1953), Model: Method 2	8.2070×10^{-6}
12	Two Constraints Method- Wolfe (1951), Model: Method 3	4.6099×10^{-6}
13	Two Constraints Criterion- Murrill (1967), Model: Method 4	5.2716×10^{-6}
14	McMillan (1994), Model: Method 4	2.8349×10^{-5}
15	St. Clair (1997), Model: Method 4	4.0526×10^{-6}
16	Shinskey (2000), (2001) Model: Method 2	2.3709×10^{-6}
17	Hay (1998) Servo Tuning 1, Model: Method 2	2.2114×10^{-6}

18	Hay (1998) Servo Tuning 2, Model: Method 2	1.4495×10^{-6}
19	Minimum IAE- Rovira et al. (1969), Model: Method 4	7.6450×10^{-6}
20	Minimum IAE- Marlin (1995), Model: Method 1	2.6216×10^{-5}
21	Minimum IAE- Smith and Corripio (1997), Model: Method 1	7.3022×10^{-6}
22	Minimum IAE- Hwang (1995), Model: Method 26	8.1822×10^{-4}
23	Minimum ISE- Zhuang and Atherton (1993), Model: Method 1	0.0030
24	Minimum ISE- Khan and Lehman (1996), Model: Method 1	7.3155×10^{-6}
25	Minimum ITAE- Rovira et al. (1969), Model: Method 4	1.4230×10^{-6}
26	Minimum ISTSE- Zhuang and Atherton (1993), Model: Method 1	3.0857×10^{-6}
27	Minimum ISTES- Zhuang and Atherton (1993), Model: Method 1	7.0588×10^{-7}

Table 4.1 : Error for each tuning done

The tuning with the least error value is the best tuning method for debutanizer. From the table above, we can see that tuning with smallest value of error is the Cohen Coon Tuning. Therefore we can say that Cohen Coon tuning is the best tuning method for the debutanizer column.

4.4.1 Close Loop Cohen Coon Response-Setpoint

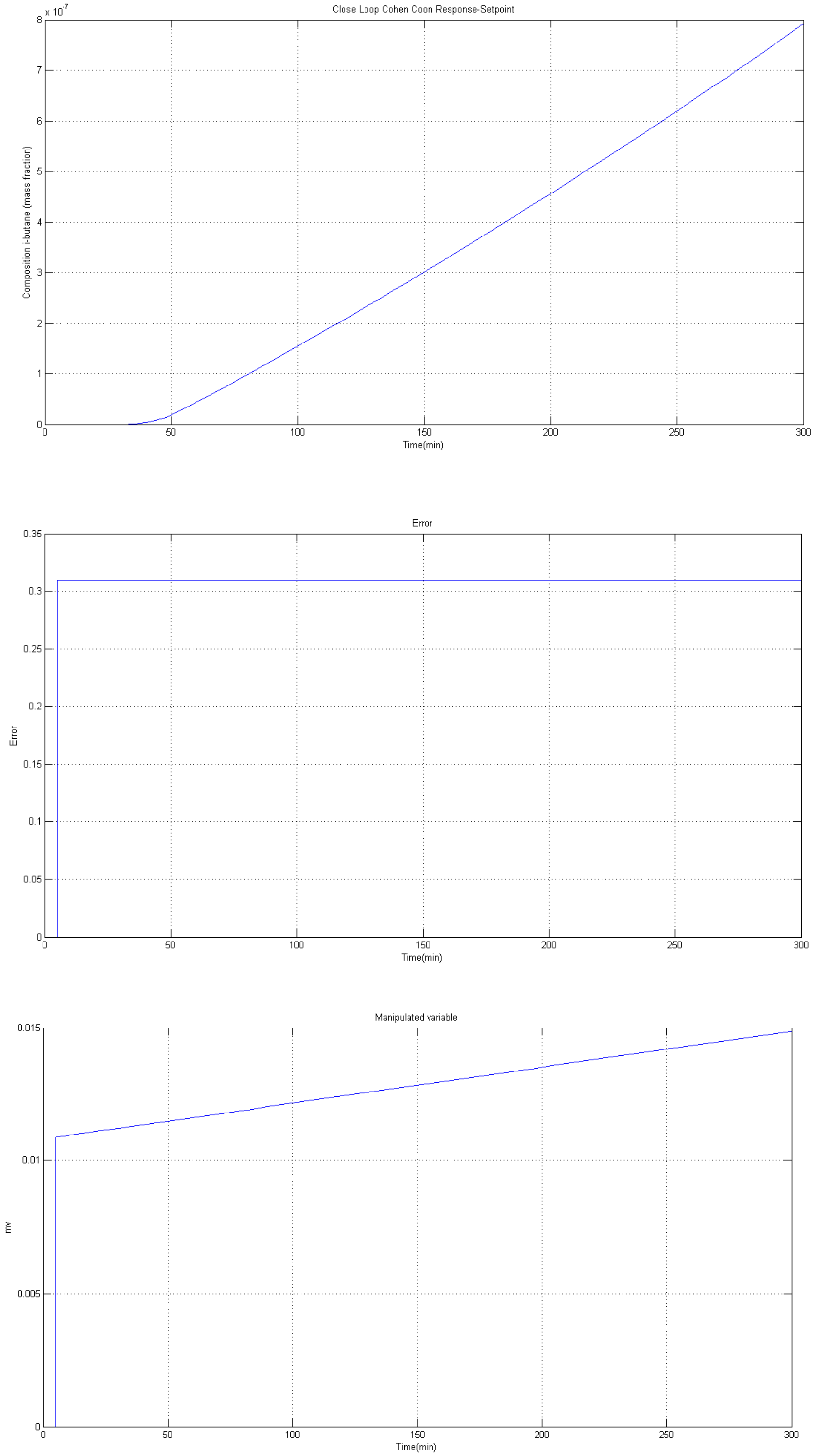


Figure 4.1 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Cohen Coon Response-Setpoint

Figure 4.1 represent the composition of Butane, Error and Manipulated Variable value after Closed Loop Cohen Coon Response-Setpoint tuning. The composition increases proportionally with time when using the closed loop Cohen Coon Tuning. Error increases to a point after a while and maintains the same value throughout the time run. The manipulated variable has a value after a few minits and increases proportionally after.

CHAPTER 5

CONCLUSION

This project is mainly about modelling a debutanizer column in order to optimize the performance of the column and to identify the best tuning method for the column. Debutanizer Column in Crude Distillation unit (CDU) of Kerteh Refinery-1 (KR-1) has been chosen as a model for this project.

From the results and discussion, it is concluded that different tuning methods would give different results on the behavior of the response. The optimum response has been chosen considering the behaviour of the response and the value of error calculation.

All the research and findings obtained will be used to improve the overall performance of the plant as well as to improve the quality of the product and maximize profitability. The successful outcome of this project will be a great helping hand for industrial application.

REFERENCES

Bakke, M., & Skogestad, S. A simulated debutanizer case study for teaching advanced process control.

Hubbard, M. (2014, 3rd October 2014). What is a Debutanizer? , from http://www.wisegeek.com/what-is-a-debutanizer.htm

Sohail Rasool Lone , S. A. A. (2013). Modeling and Simulation of a Distillation Column using MATLAB. *International Journal of Engineering Research and Science & Technology, 2*.

Donald A. Mohutsiwa (2006), *PID controller tuning using Internal Model Control*

William L. Luyben 1990, *Process Modelling, Simulation and Control for Chemical Engineers, 2nd edition McGraw Hill*

APPENDICES

Other Tuning Graphs

4.4.2 Closed Loop IAE Response-Setpoint

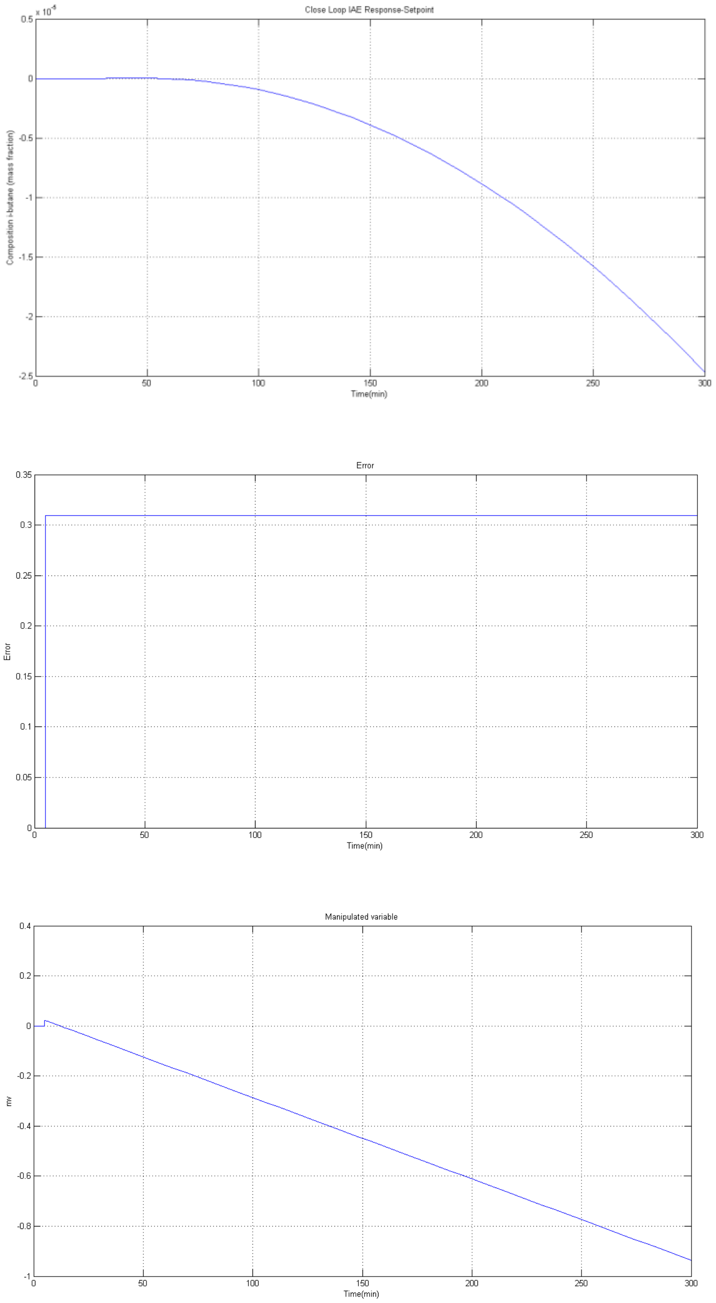
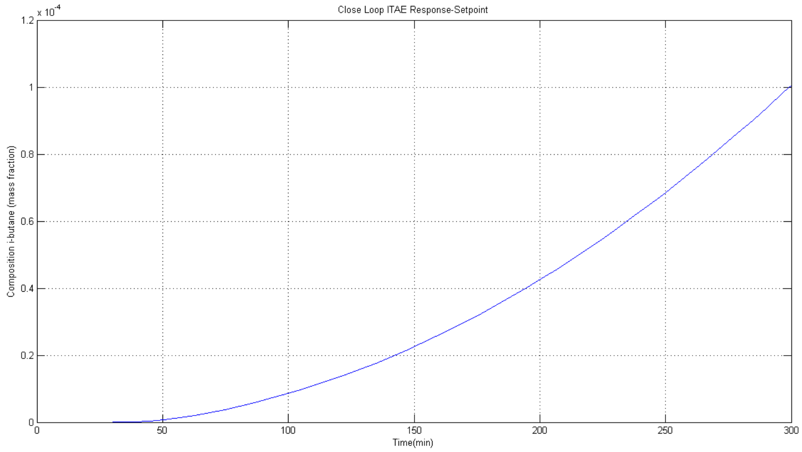
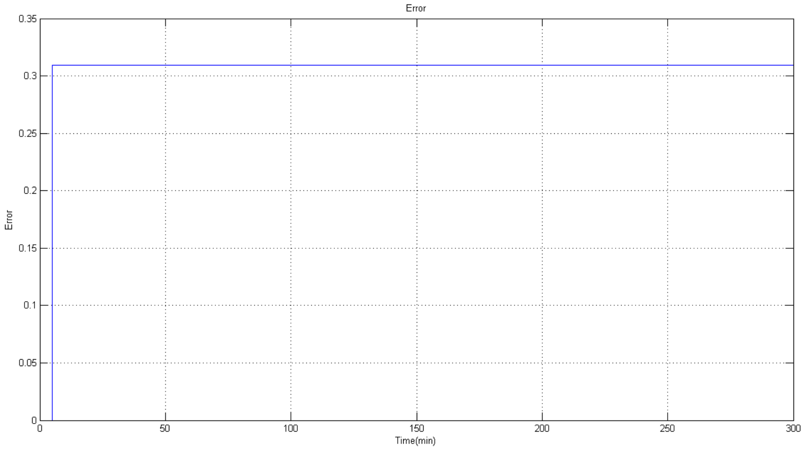


Figure 4.1 : Composition of Butane, Error and Manipulated Variable value after Closed Loop IAE Response-Setpoint

4.4.3 Close Loop ITAE Response-Setpoint





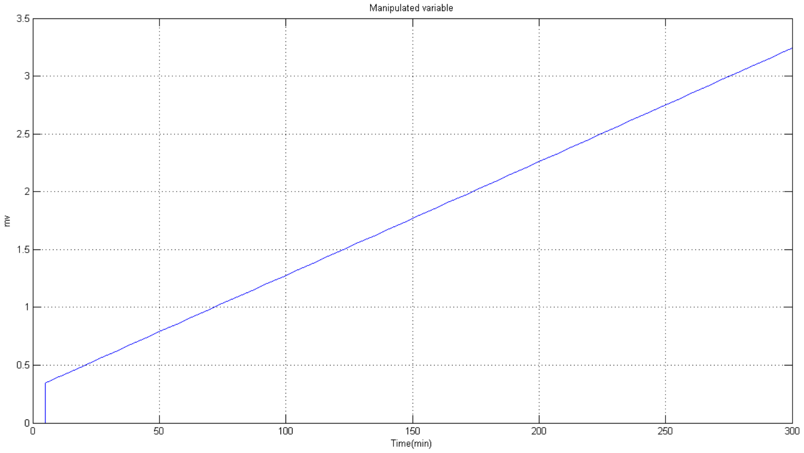
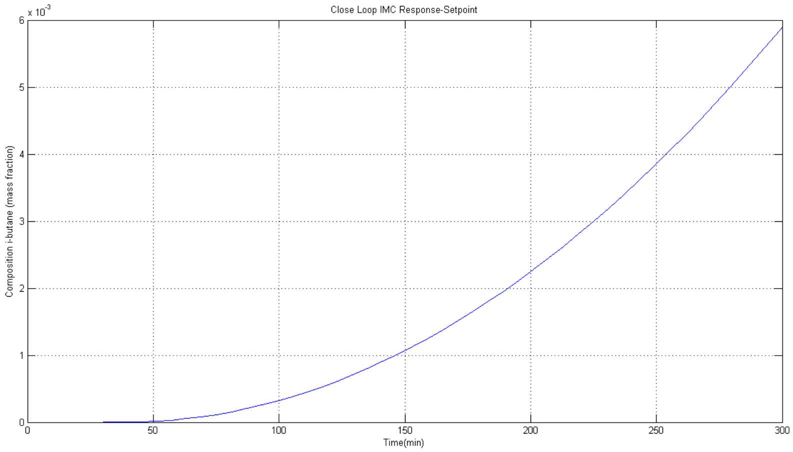
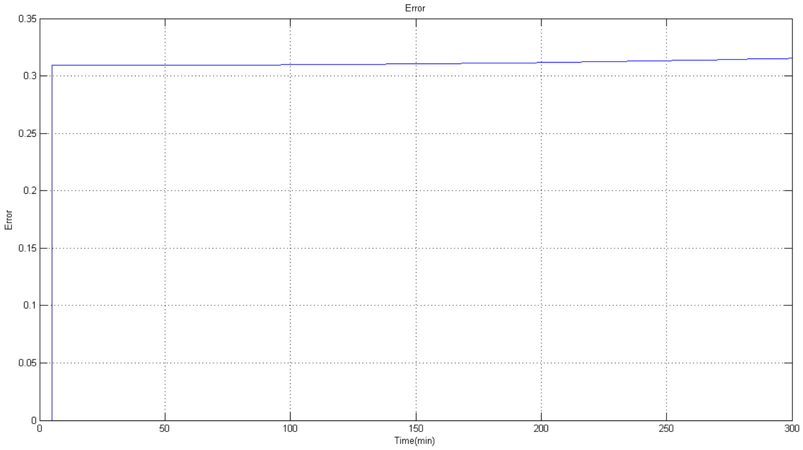


Figure 4.2 : Composition of Butane, Error and Manipulated Variable value after Closed Loop ITAE Response-Setpoint

4.4.4 Closed Loop IMC Response-Setpoint





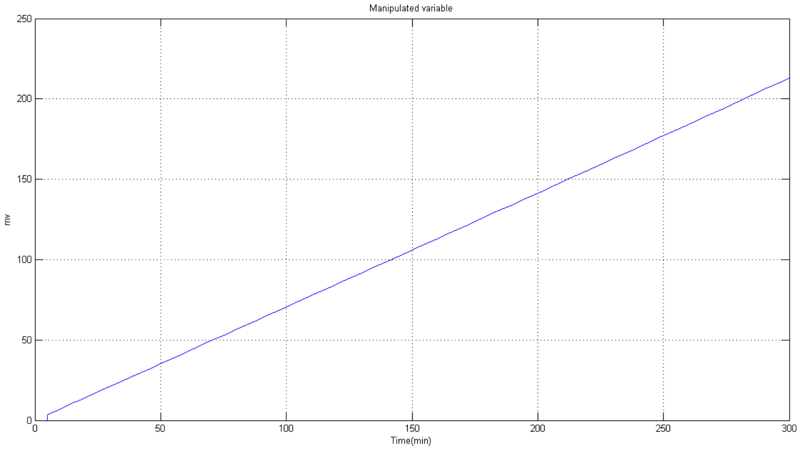
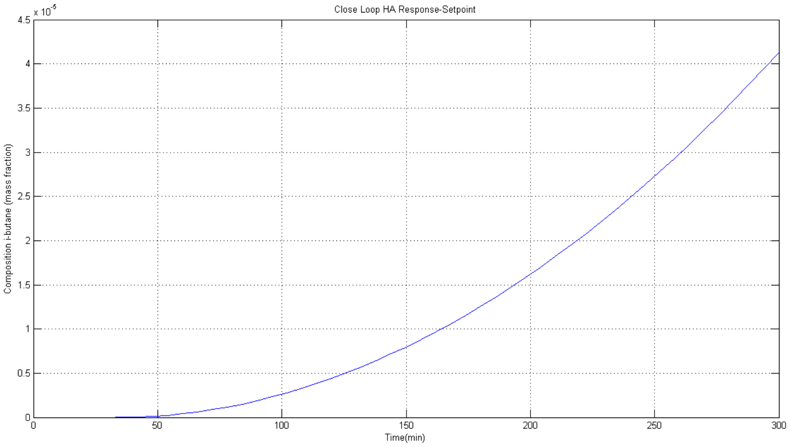
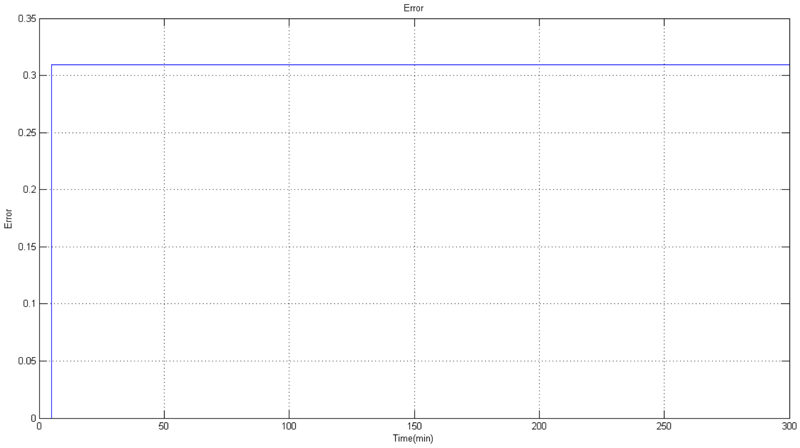


Figure 4.3 : Composition of Butane, Error and Manipulated Variable value after Closed Loop IMC Response-Setpoint

4.4.5 Close Loop HA Response-Setpoint





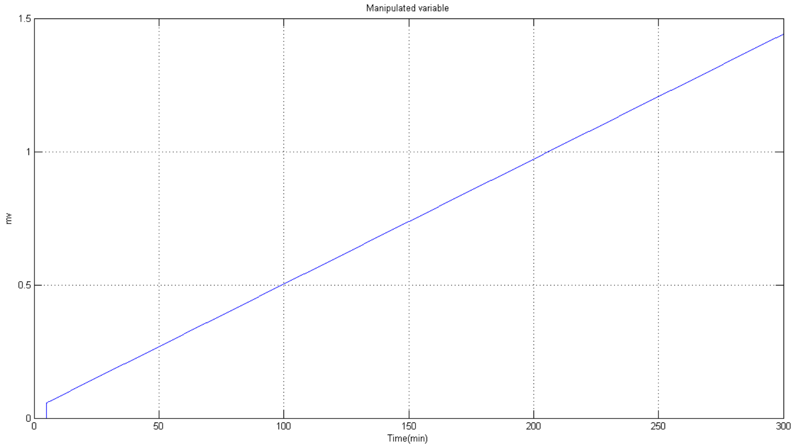


Figure 4.4 : Composition of Butane, Error and Manipulated Variable value after Closed Loop HA Response-Setpoint

4.4.6 Close Loop ISE Response-Setpoint

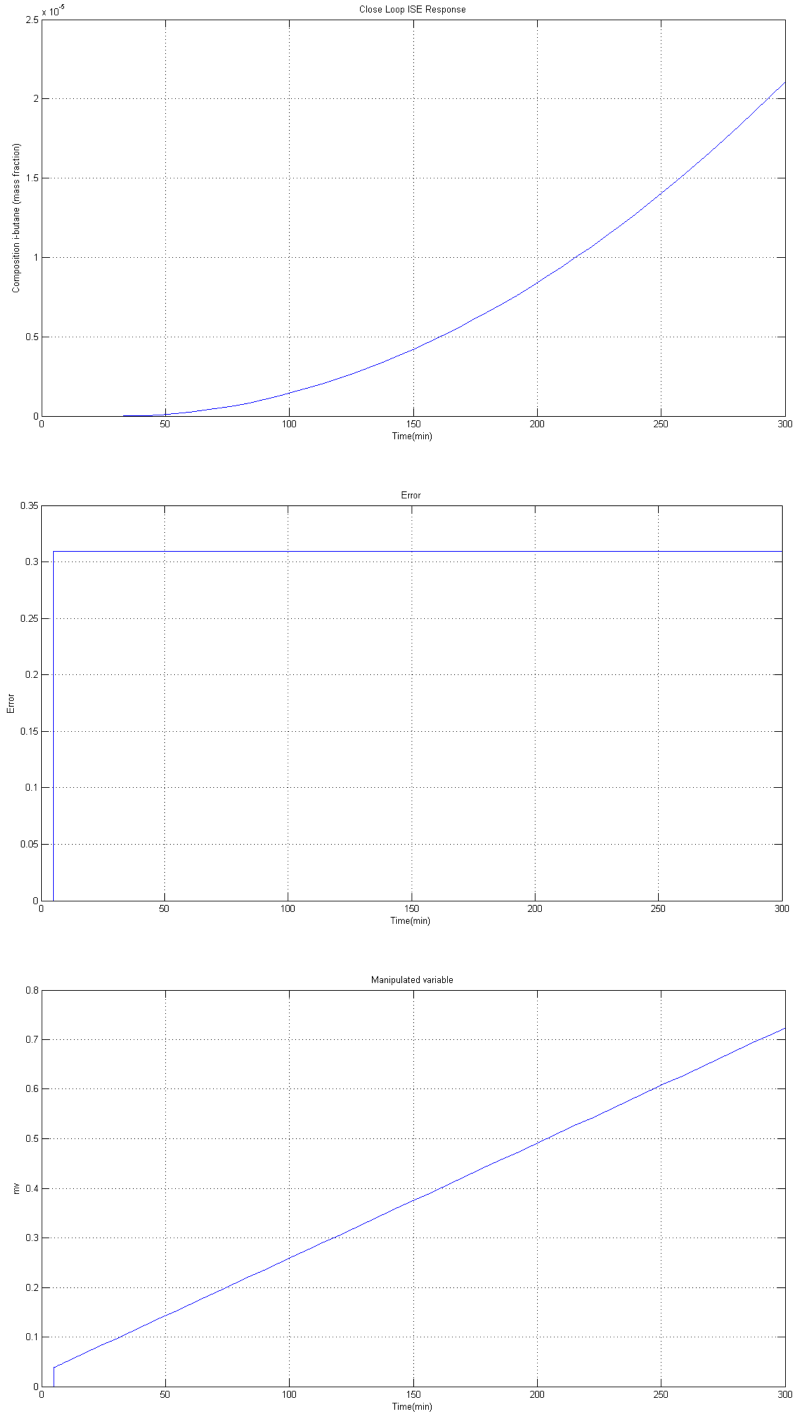
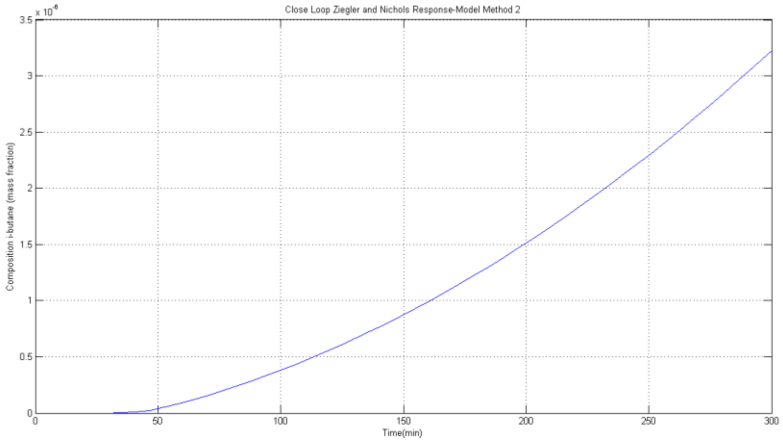
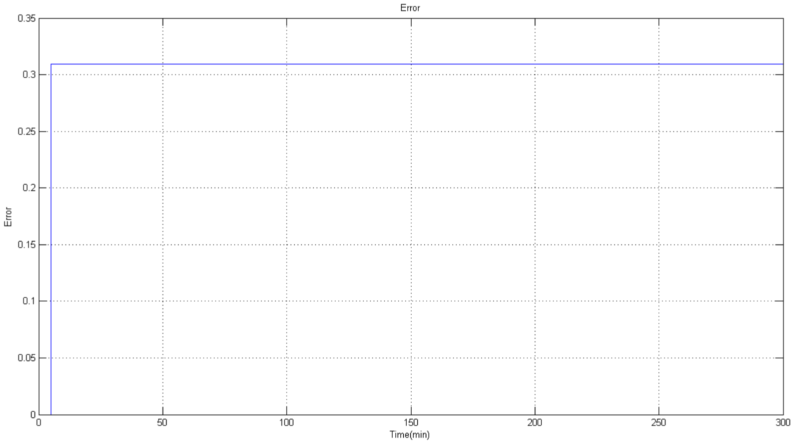


Figure 4.5 : Composition of Butane, Error and Manipulated Variable value after Closed Loop ISE Response-Setpoint

4.4.7 Close Loop Ziegler and Nichols Response





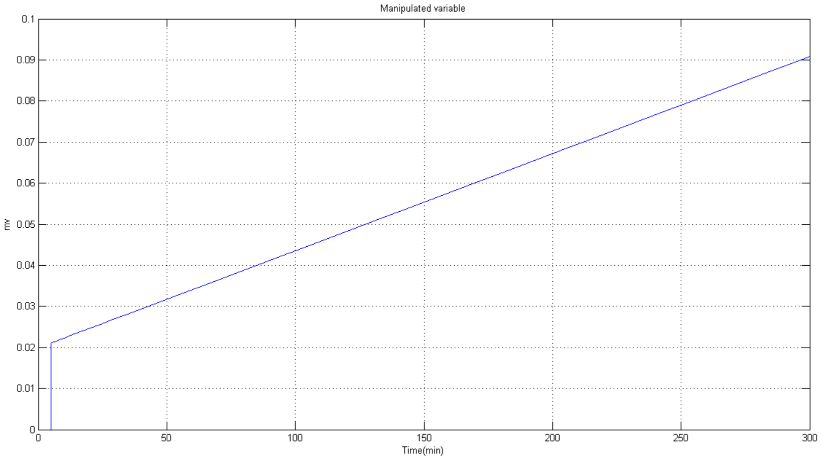
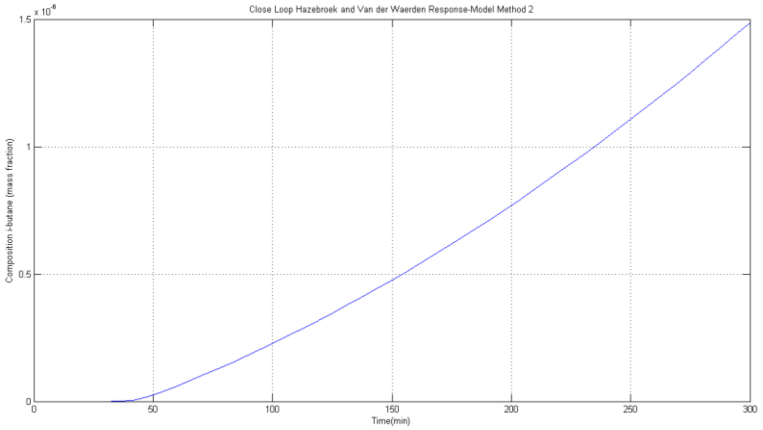
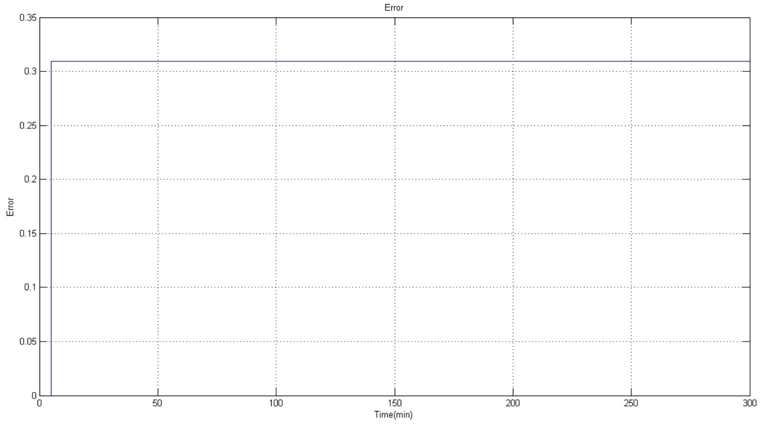


Figure 4.7 : Composition of Butane, Error and Manipulated Variable value after Close Loop Ziegler and Nichols Response.

4.4.8 Close Loop Hazerbroek and Van der Waarden Response





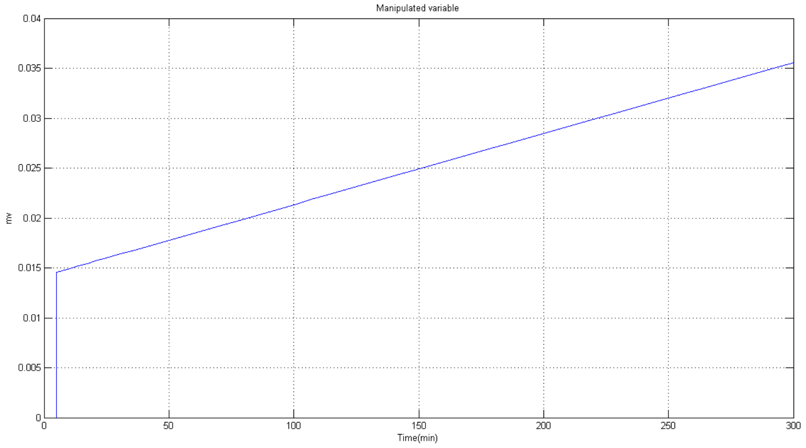


Figure 4.8 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Hazerbroek and Van der Waarden Response

4.4.9 Close Loop Chien (1952)- Servo,Model: Method 2, 0% overshoot

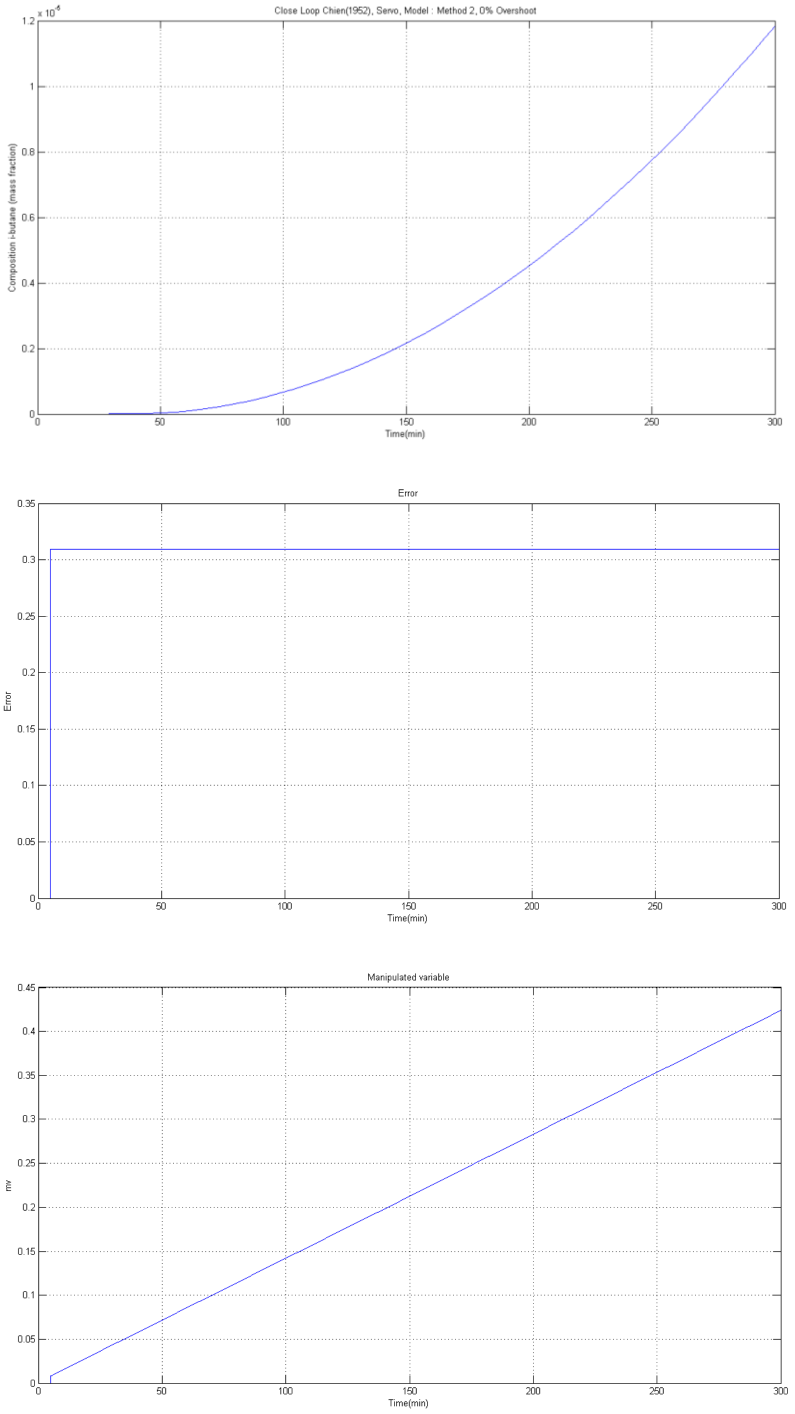


Figure 4.9 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Chien (1952)- Servo,Model: Method 2, 0% overshoot

4.4.10 Close Loop Chien (1952) Servo, Model 2, 20% Overshoot

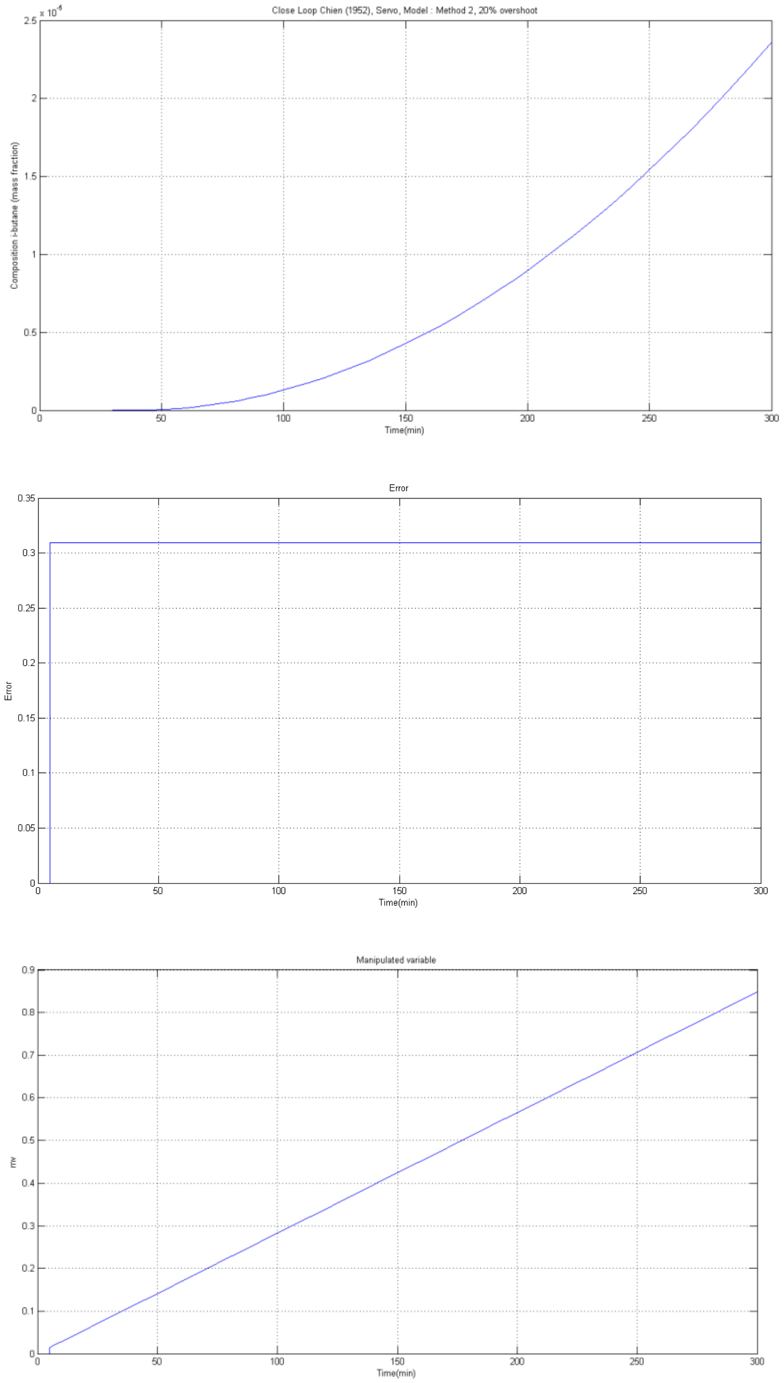


Figure 4.10 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Chien (1952) Servo, Model 2, 20% Overshoot

4.4.11 Close Loop Cohen and Coon (1953), Model Method 2

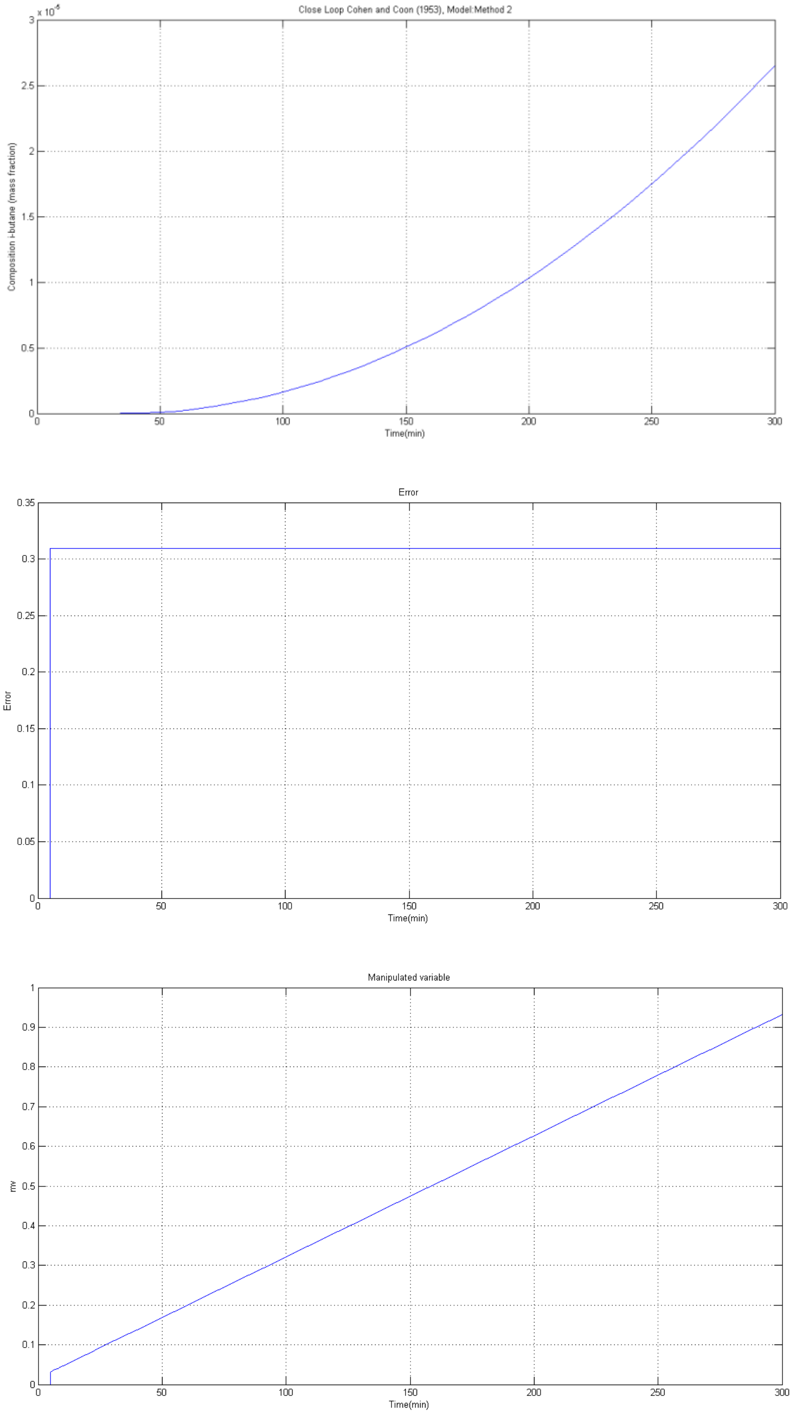
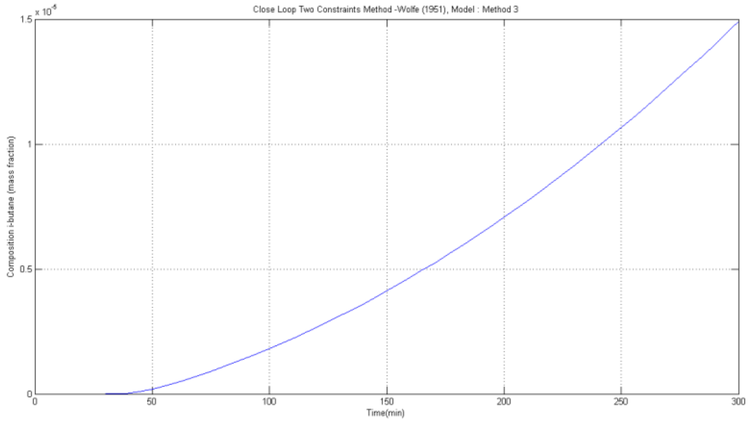
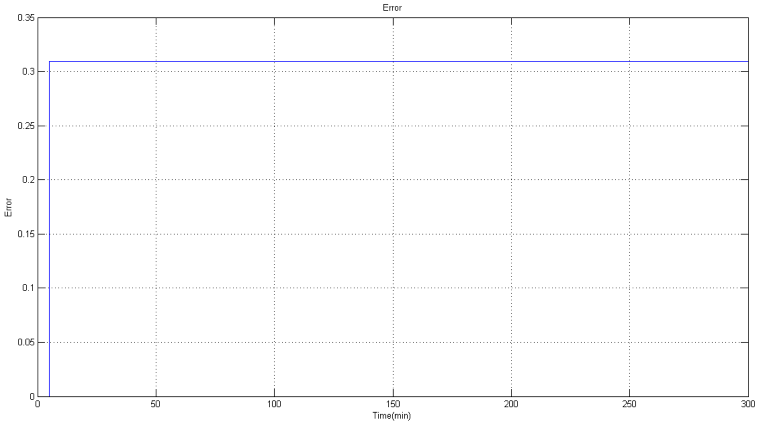


Figure 4.11 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Cohen and Coon (1953)-Method : Model 2

4.4.12 Close Loop Two Constraints Method – Wolfe (1951), Model : Method 3





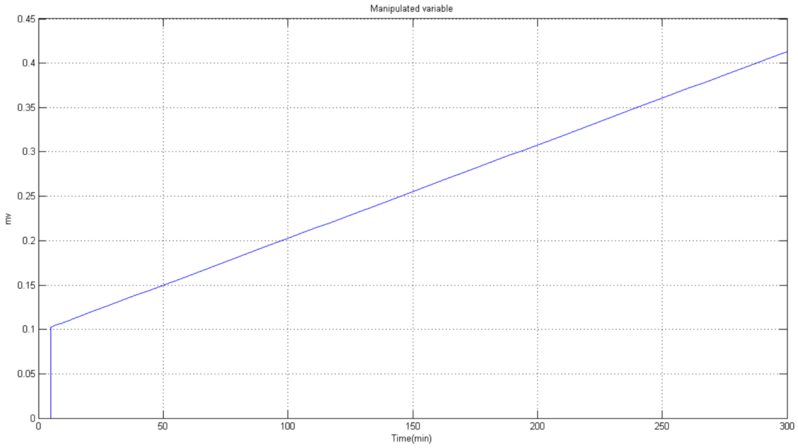
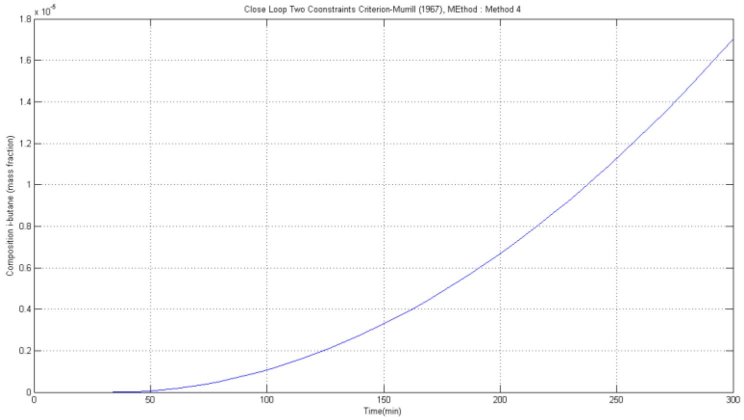
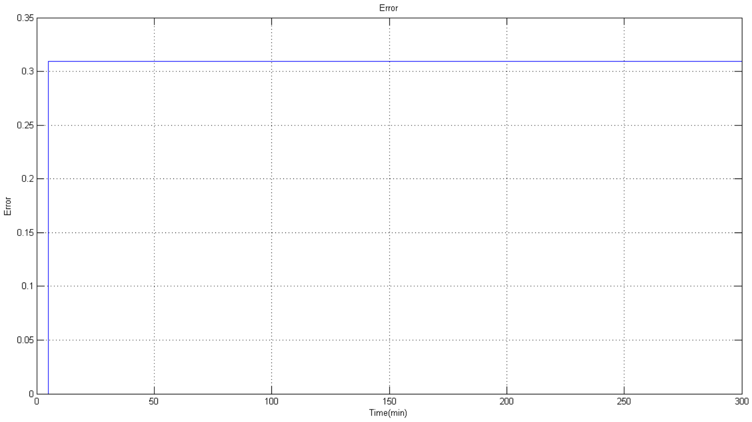


Figure 4.12 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Two Constraints Method – Wolfe (1951), Model : Method 3

4.4.13 Close Loop Two Constraints Criterion-Murill (1967) Method : Method 4





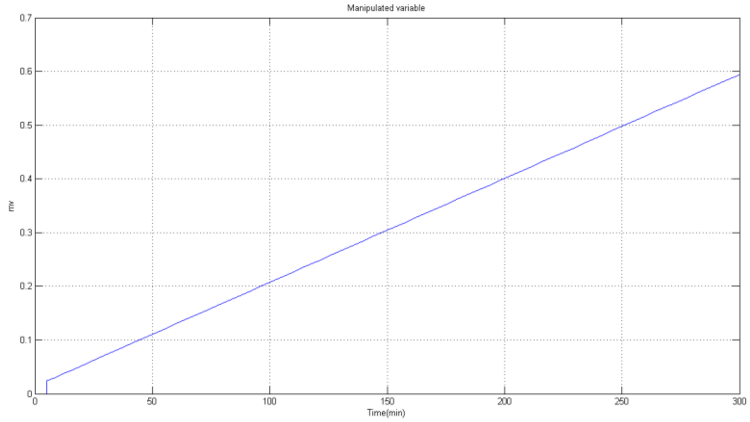
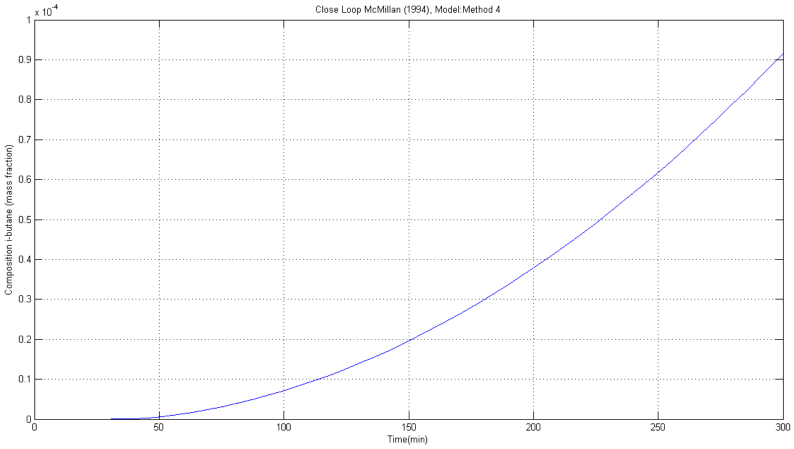
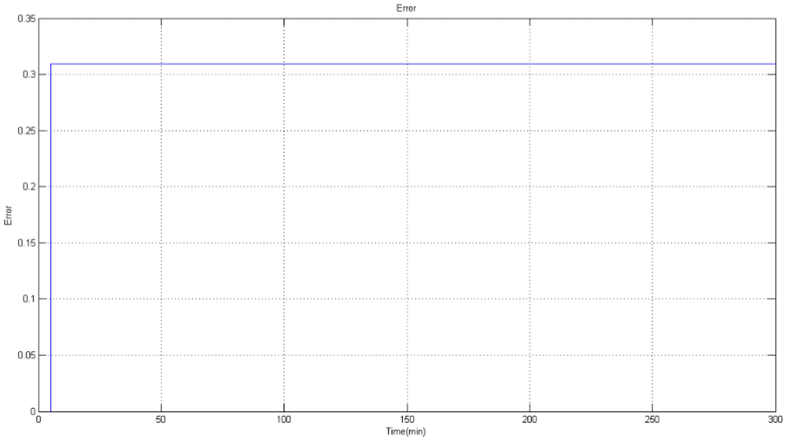


Figure 4.13 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Two Constraints Criterion-Murill (1967) Method : Method 4

4.4.14 Close Loop McMillan (1994) Model : Method 4





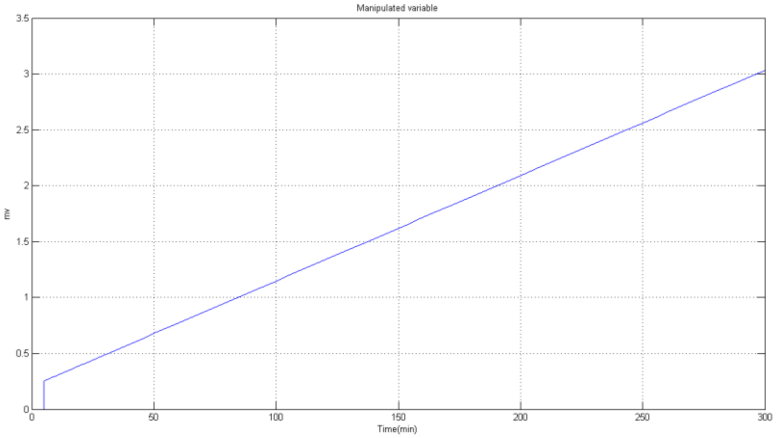


Figure 4.14 : Composition of Butane, Error and Manipulated Variable value after Closed Loop McMillan (1994) Model : Method 4

4.4.15 Close Loop St. Clair (1997), Model : Method 4

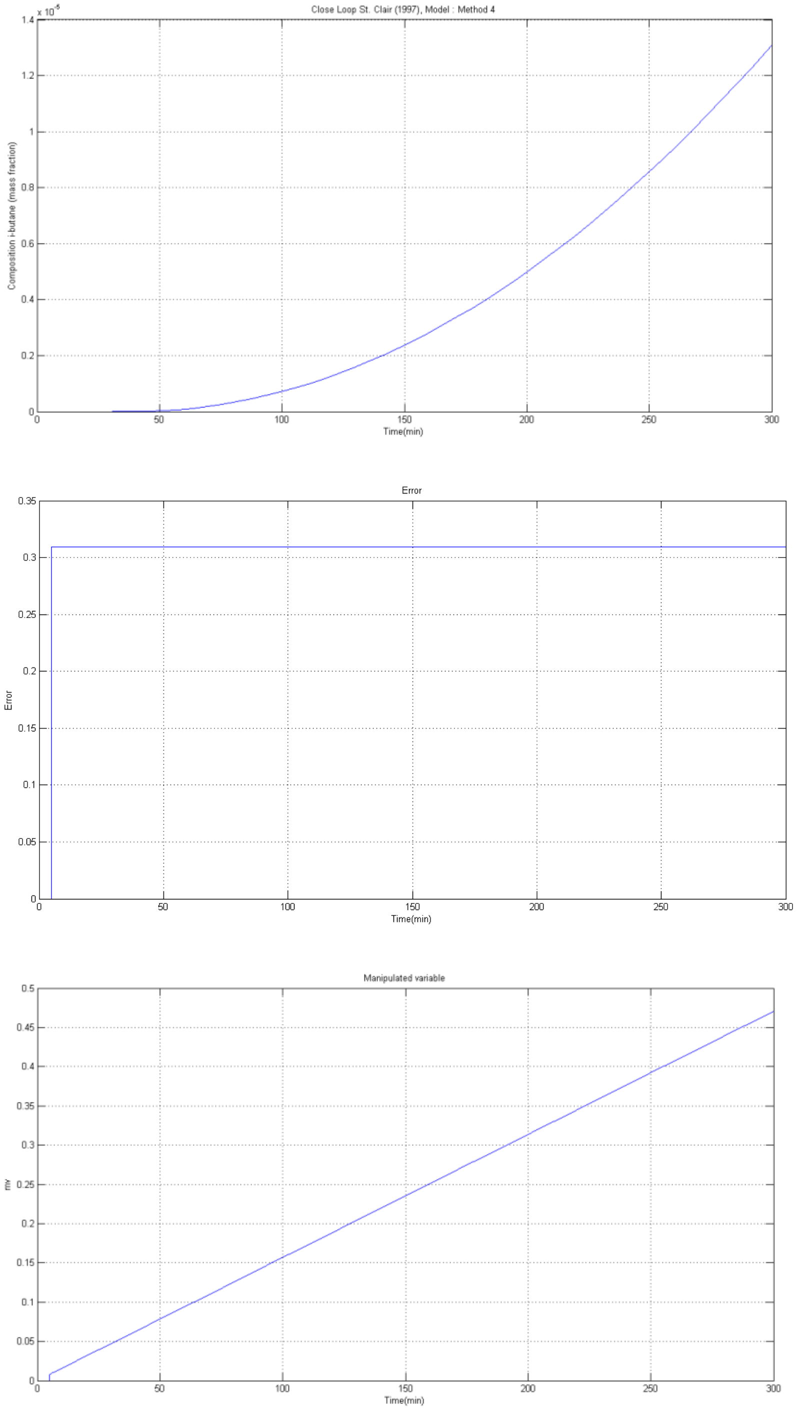
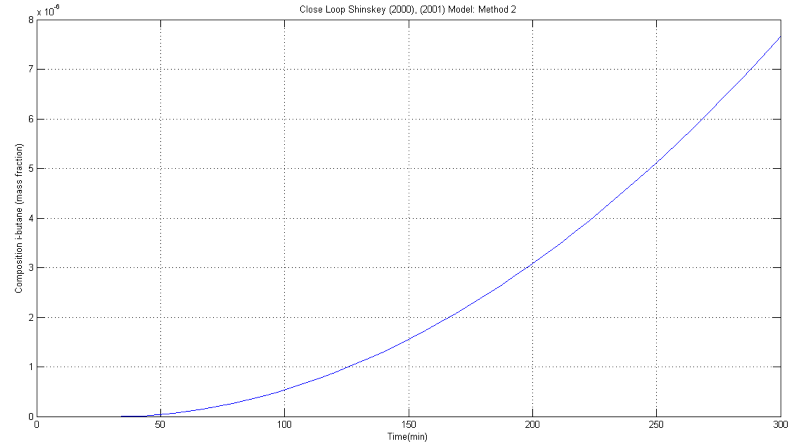
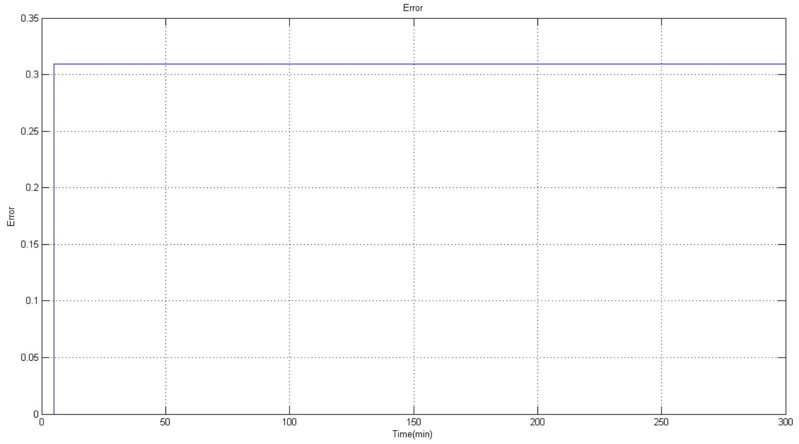


Figure 4.15 : Composition of Butane, Error and Manipulated Variable value after Closed Loop St. Clair (1997), Model : Method 4

4.4.16 Close Loop Shinskey (2000), (2001) Model : Method 2





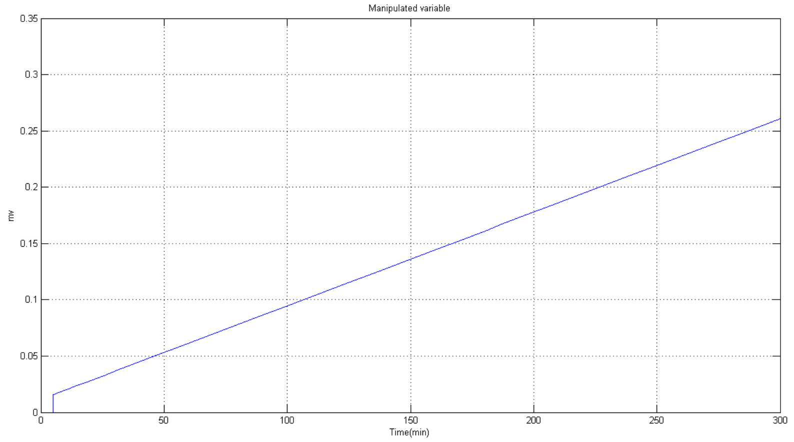
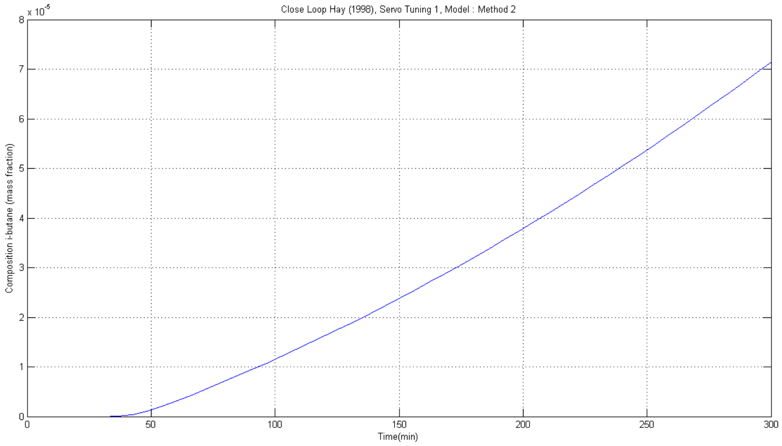
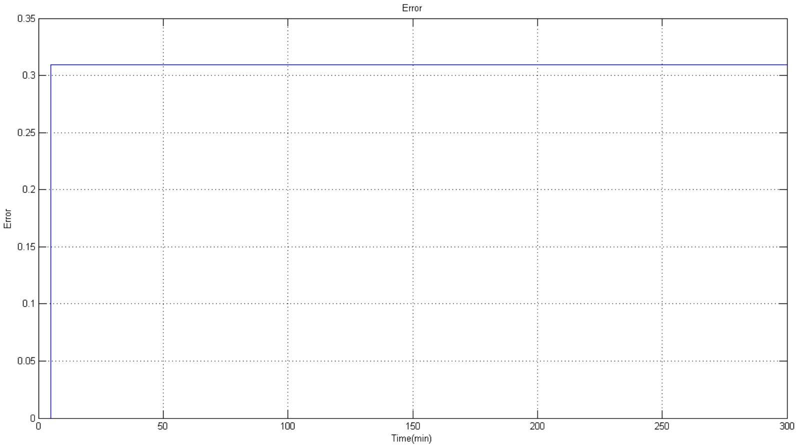


Figure 4.16 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Shinskey (2000), (2001) Model : Method 2

4.4.17 Close Loop Hay (1998), Servo Tuning 1, Model : Method 2





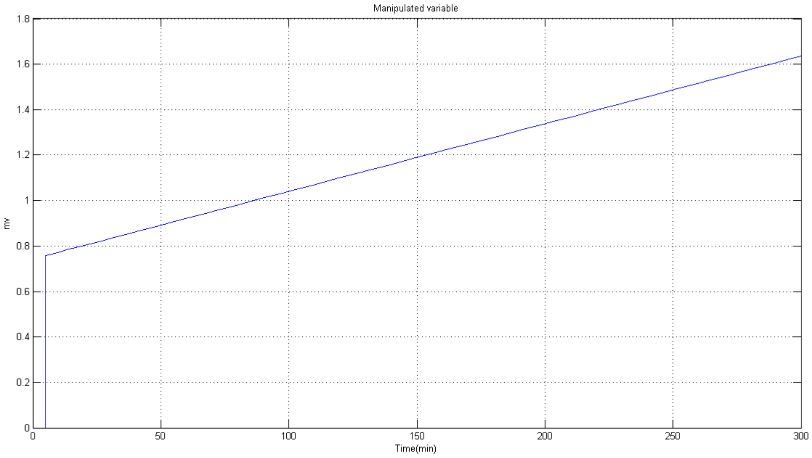
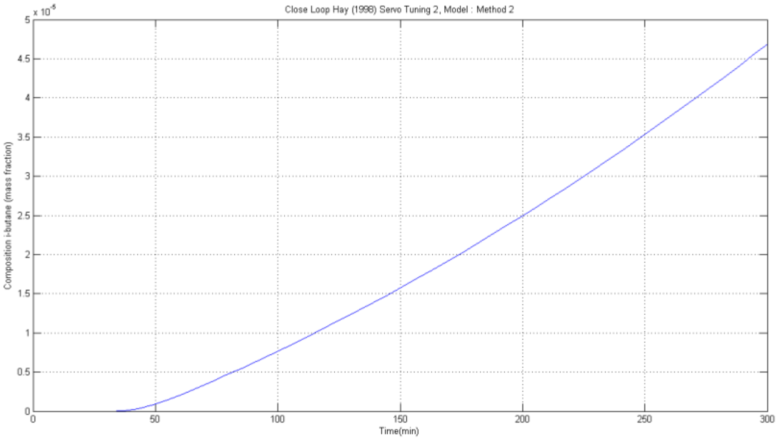
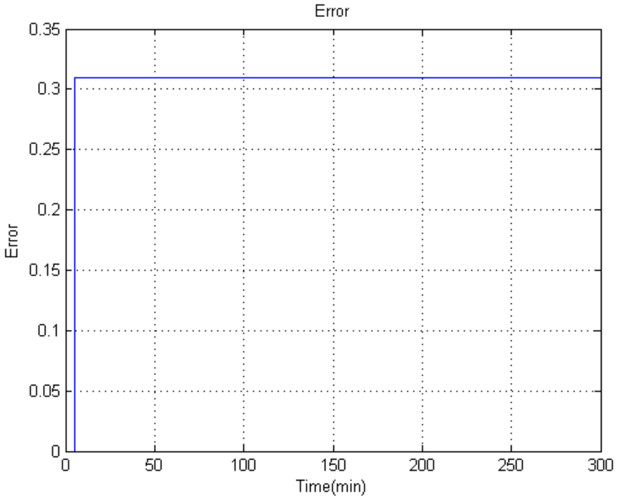


Figure 4.17 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Hay (1998), Servo Tuning 1, Model : Method 2

4.4.18 Close Loop Hay (1998) Servo Tuning 2, Model : Method 2





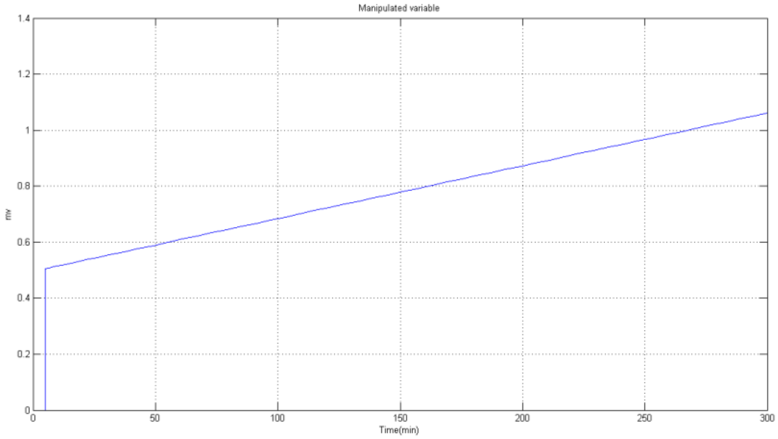


Figure 4.18 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Hay (1998) Servo Tuning 2, Model : Method 2

4.4.19 Close Loop Minimum IAE-Rovira et. Al.(1969), Model : Method 4

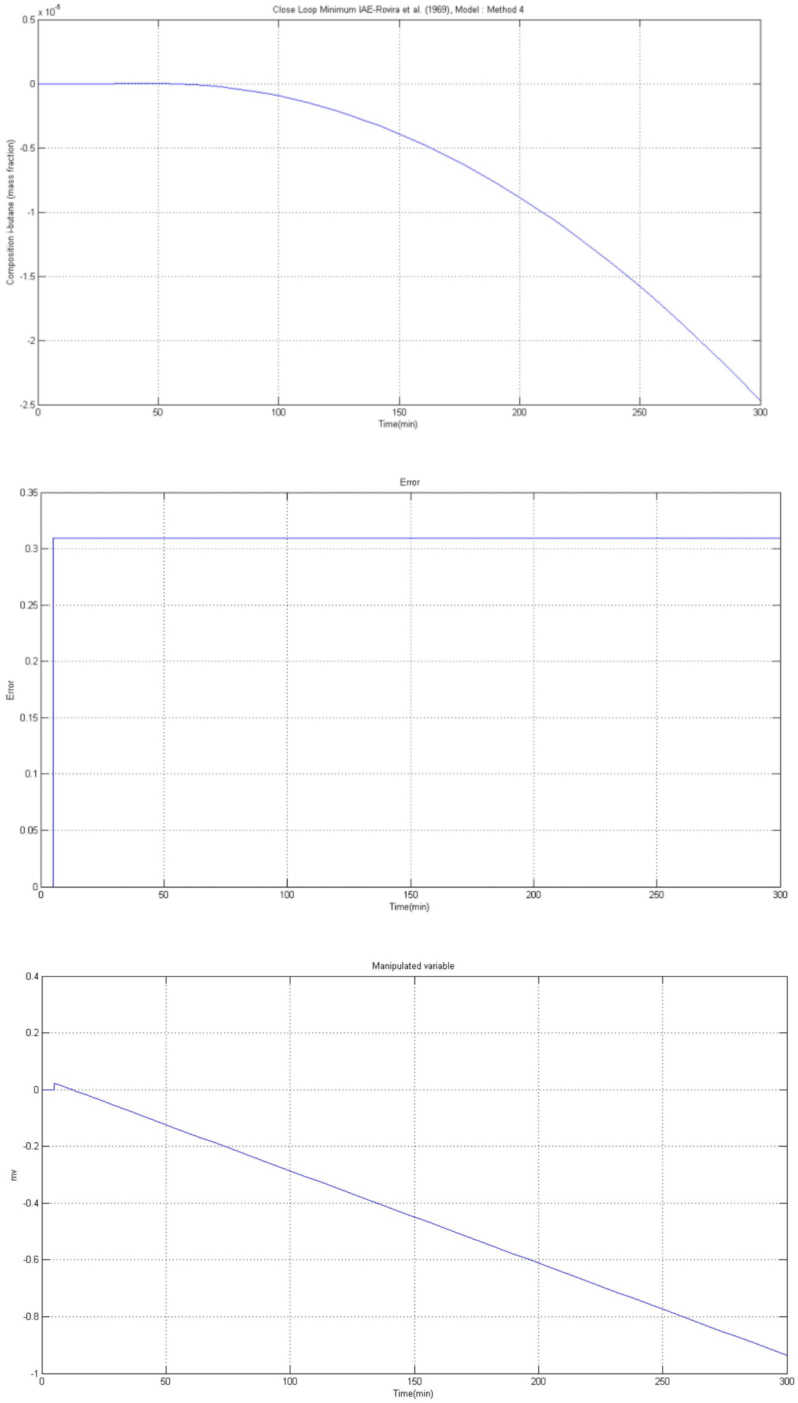


Figure 4.19 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Minimum IAE-Rovira et. Al.(1969), Model : Method 4

4.4.20 Close Loop Minimum IAE-Marlin(1995)-Model: Method 1

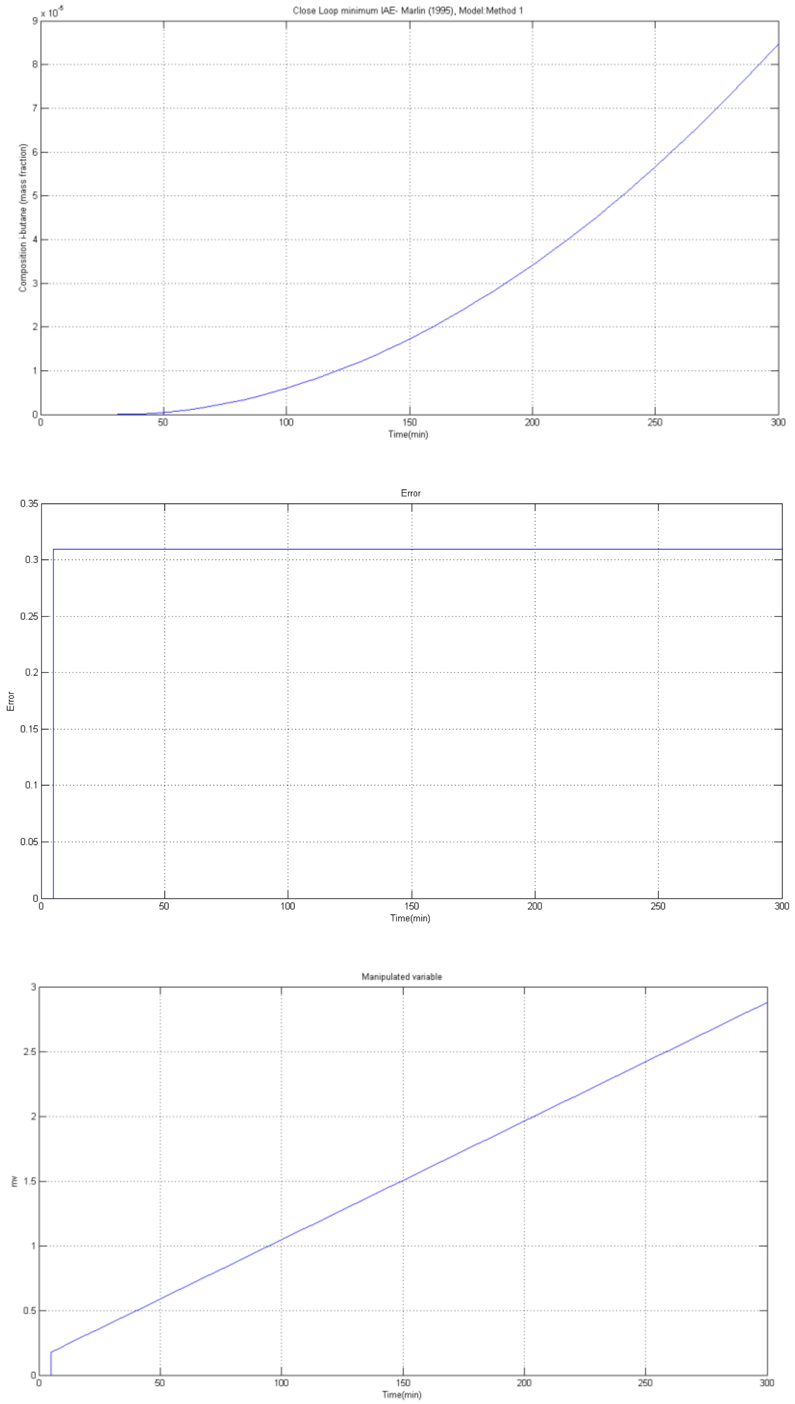


Figure 4.20 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Minimum IAE-Marlin(1995)-Model: Method 1

4.4.21 **Close Loop Minimum IAE- Smith and Corripio, Model : Method 1**

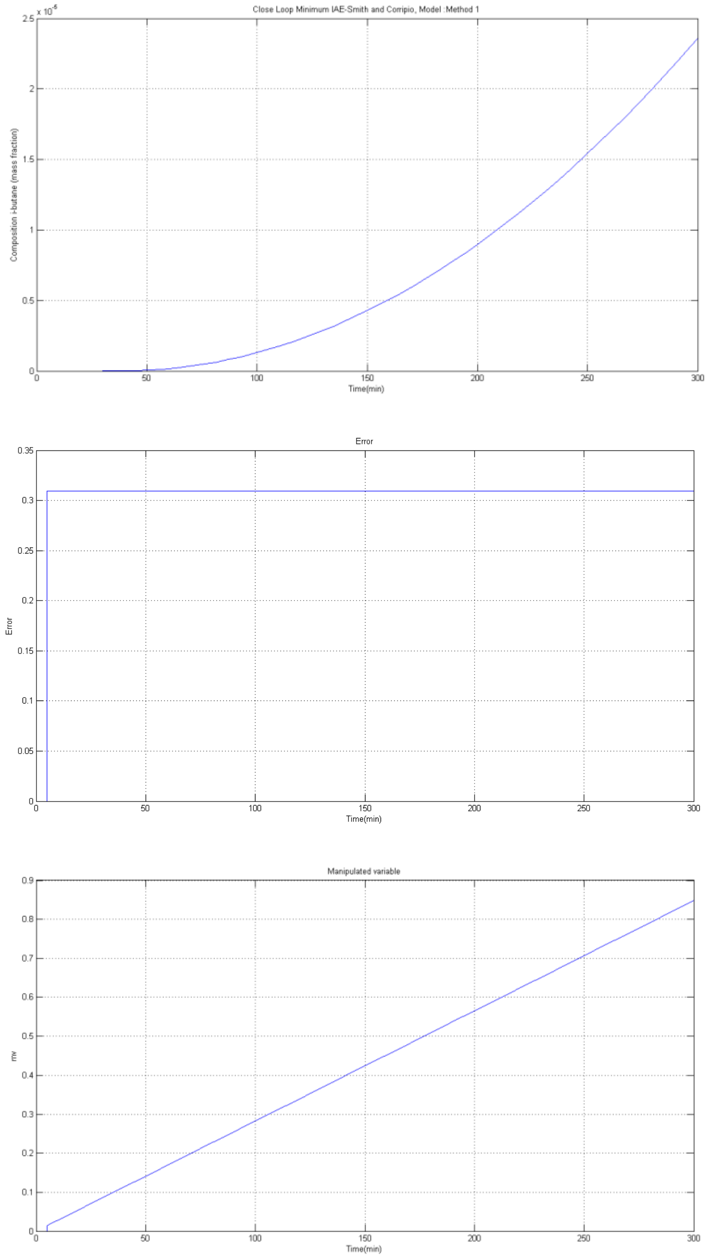


Figure 4.21 : Composition of Butane, Error and Manipulated Variable value after Closed Loop Minimum IAE- Smith and Corripio, Model : Method 1

M-Files Codings

1. DynamicState.m

```
% A program developed by
% Emira Farzana Ellias
% Universiti Teknologi PETRONAS
% Bandar Seri Iskandar
% Perak

% Declaration of some variables


global k1ik1
global k2
global DIST_PAR
global stepk1
global tstepk1
global xA xB xC xD
global yA yB yC yD

% General Data Input for Column

disp('Please Enter Distillation Column Parameters');

D=input('Distillate withdrawal rate (mol/s):');
DIST_PAR(1)=input('Number of stages :');
DIST_PAR(2)=input('Reactant A Feed stage:');
DIST_PAR(3)=input('Reactant B Feed stage: ');
DIST_PAR(4)=input('Feed A flowmte (tool/s):');
DIST_PAR(5)=input ('Feed B flowrate (mol/s):');
DIST_PAR(6)=input ('Feed A purity (mole traction):');
DIST_PAR(7)=input ('Feed B purity (mole fraction):');
DIST_PAR(8)=input ('Feed A quality:');
DIST_PAR(9)=input ('Feed B quality:');
DIST_PAR(10)=input ('Reflux flowrate (mol/s):');
DIST_PAR(11)=(DIST_PAR(10)+DIST_PAR(4)*DIST_PAR(8)+DIST_PAR(5)*DIST_PAR(9))-
(DIST_PAR(4)+DIST_PAR(5)-D);
DIST_PAR(24)=input ('Overall liquid hold-up in column (mol.s/m3):');
DIST_PAR(25)=input('Distillate molar hold-up (mol):');
DIST_PAR(26)=input('Bottom molar hold-up (mol):');
DIST_PAR(35)=input('Condensor Pressure (bar):');
```

```
DIST_PAR(36)=input('Reboiler Pressure (bar):');
%Antomds Equation
disp('Please,inSett Ant~in~ Equatio~ Con~anU for Reactant A, Reactant B, Product C and Product D');

disp('For Reactant A');
DIST_PAR(12)=input('Constant A:');
DIST_PAR(13)=input('Constant B:');
DIST_PAR(14)=input('Constant C:');

disp('For Reactant B');
DIST_PAR(15)=input ('Constant A:');
DIST_PAR(16)=input ('Constant B :');
DIST_PAR(17)=input ('Constant C :');

disp('For Product C');
DIST_PAR(18)=input ('Constant A:');
DIST_PAR(19)=input ('Constant B:');
DIST_PAR(20)=input ('Constant C :');

disp('For Product D');
DIST_PAR(21)=input ('Constant: A:');
DIST_PAR(22)=input ('Constant B:');
DIST_PAR(23)=input ('Constant C :');


% nur~ber of stages % teed A stage % feed B stage % teed A 11oxvrate % teed B flow~'ate % teed A purity' % iced B purity % teed A qua~6t'y'' % teed B quali~t3' % rctlux t, lov,a'at.e % vapor ftowrate % liquid ho|dup % condenser pressure % rebokler pressure

dist = input ('Do you want to introduce a disturbance into the process? Please Enter 1 for YES or 2 for NO:');

% Assignment of Variables

ns = DIST_PAR(1);
nfA = DIST_PAR(2);
nfB = DIST_PAR(3);
feedAi = DIST_PAR(4);
feedBi = DIST_PAR(5);
zfeedAi = DIST_PAR(6);
zfeedBi = DIST_PAR(7);

qfA = DIST_PAR(8);
qfB = DIST_PAR(9);
refluxi = DIST_PAR(10);
vapori = DIST_PAR(11);
mt = DIST_PAR(24);
md = DIST_PAR(25);
```

```
mb = DIST_PAR(26);

Pc = DIST_PAR(35);
Pr = DIST_PAR(36);

Aa=DIST_PAR(12)
Ba=DIST_PAR(13)
Ca=DIST_PAR(14)

Ab=DIST_PAR(15)
Bb=DIST_PAR(16)
Cb=DIST_PAR(17)

Ac=DIST_PAR(18)
Bc=DIST_PAR(19)
Cc=DIST_PAR(20)

Ad=DIST_PAR(21)
Bd=DIST_PAR(22)
Cd=DIST_PAR(23)


% To Introduce diturbance into the system

if dist==1;
   disp('Please insert magnitude step disturbance and the time of step change for desired parameter:');
   disp('For other parameters, please enter 0 to indicate no step change for the particular parameter:');
   DIST_PAR(27)=input('Magnitude of step in reflux:');
   DIST_PAR(28)=input('Time of reflux step change:');
   DIST_PAR(29)=input('Magnitude of step in vapor:');
   DIST_PAR(30)=input('Time of vapor step change:');
   DIST_PAR(31)=input('Magnitude of feed A composition change:');
   DIST_PAR(32)=input('Time of feed A composition change:');
   DIST_PAR(33)=input('Magnitude of feed flow A change:');
   DIST_PAR(34)=input('Time of feed flow A change:');
   DIST_PAR(37)=input('Magnitude of feed  B composition change:');
   DIST_PAR(38)=input('Time of feed B composition change:');
   DIST_PAR(39)=input('Magnitude of feed B flow change:');
   DIST_PAR(40)=input('Time of feed flow B change :');

   stepk1=input('Magnitude of change of forward reaction rate:');
   tstepk1=input('Time of forward reaction rate change :');

   disp('Please wait, solving for the dynamic-state stage composition');
   disp('of reactant A, reactant B, product C and product D');

% Zero the funtion vector
```

```
xA=zeros(ns,1);
xB=zeros(ns,1);
xC=zeros(ns,1);
xD=zeros(ns,1);

% Linear Variation of Pressure at Each Column Stage

P(1)=Pc;
for i=2:ns-1;
   P(i)=P(i-1)+(Pr-Pc)/(ns-1);
end
P(ns)=Pr;

% Subroutine for Column Tray Temperature
% To calculate vapor composition from Raoult's Law
% Assume initial guess temperature of 50oC


      T=50;

   while abs(net)>0.02;
      pA(i)=10^(Aa-(Ba/(T+Ca)))/760;
      pB(i)=10^(Ab-(Bb/(T+Cb)))/760;
      pC(i)=10^(Ac-(Bc/(T(i)+Cc)))/760;
      pD(i)=10^(Ad-(Bd/(T(i)+Cd)))/760;

      yA1(i)=pA(i)*xA(i)/P(i);
      yB1(i)=pB(i)*xB(i)/P(i);
      yC1(i)=pC(i)*xC(i)/P(i);

      xD(i)=1-xA(i)-xB(i)-xC(i);

      yD1(i)=pD(i)*xD(i)/P(i);

      sum_y=yA1(i)+yB1(i)+yC1(i)+yD1(i);
      net=1-sum_y;
      T(i)=T(i)+0.001;
   end

   yA(i)=yA1(i);
   yB(i)=yB1(i);
   yC(i)=yC1(i);
   yD(i)=yD1(i);
```

```
% Solving ordinaiy differential equation by Runge-Kutta 4th and 5th order method
% Initial condition of component composition

for i=1:ns;
    xA0(i)=xA(i)
    xB0(i)=xB(i)
    xC0(i)=xC(i)
    xD0(i)=xD(i)
end

% Time range of change

tspan = [0 100];

[t, xA]=ode45('dist_dynA',tspan,xA0);
[t, xB]=ode45('dist_dynB',tspan,xB0);
[t, xC]=ode45('dist_dynC',tspan,xC0);
[t, xD]=ode45('dist_dynD',tspan,xD0);

% No disturbance in system

else

DIST_PAR(27)=0;
DIST_PAR(28)=0;
DIST_PAR(29)=0;
DIST_PAR(30)=0;
DIST_PAR(31)=0;
DIST_PAR(32)=0;
DIST_PAR(33)=0;
DIST_PAR(34)=0;

% Zero the function vector

xA=zeros(ns, 1);
xB=zeros(ns, 1);
xC=zeros(ns, 1);
xD=zeros(ns, 1);

% Linear Variation of Pressure at Each Column Stage

P(1)=Pc;
for i=2:ns-1;
    P(i)=P(i-1)+(Pr-Pc)/(ns-1)
end

% Subroutine for Column Tray Temperature
```

```
% To calculate vapor composition from Raoult's law


    T=50;

   net=1;
while abs(net)>0.02;
pA(i) = 10^(Aa - (Ba/(T + Ca)))/760;
pB(i) = 10^(Ab - (Bb/(T + Cb)))/760;
pC(i) = 10^(Ac - (Bc/(T + Cc)))/760;
pD(i) = 10^(Ad - (Bd/(T + Cd)))/760;

yA1(i) = pA(i)*xA(i)/P(i);
yB1(i) = pB(i)*xB(i)/P(i);
yC1(i) = pC(i)*xC(i)/P(i);

xD(i) = 1-xA(i)-xB(i)-xC(i);
yD1(i) = pD(i)*xD(i)/P(i);

sum_y = yA1(i)+yB1(i)+yC1(i)+yD1(i);
net = 1- sum_y;

end
yA(i) = yA1(i);
yB(i) = yB1(i);
yC(i) = yC1(i);
yD(i) = yD1(i);
end

% Solving ordinary differential equation by Runge-Kutta 4th and 5th order method
% Initial condition of composition range

for i=1:ns;
    xA0(i)=xA(i);
    xB0(i)=xB(i);
    xC0(i)=xC(i);
    xD0(i)=xD(i);
end

% Time range of step ch.~mge

tspan = [0 100];

[t, xA]=ode45('dist_dynA',tspan,xA0);
[t, xB]=ode45('dist_dynB',tspan,xB0);
[t, xC]=ode45('dist_dynC',tspan,xC0);
[t, xD]=ode45('dist__dynD',tspan,xD0);
```

```
            % Graph Plotting of Result

            plot(t,xD(:,4)), xlabel('Time'), ylabel('Composition of Methyl Acetate'),...
            title('Disturbance',0),...
axis([0 100 0.5 1.0]);

            % Function file for Reactant A

            function xAdot = dist_dynA(t,xA)

            % Declaration of some variables

            global DIST_PAR
            global k1i kl
            global k2
            global stepk1
            global tstepk1
            global xB xC xD
            global yA yB yC yD

            % Assignment of Values to parameters

            ns = DIST_PAR(1);
            nfA = DIST_PAR(2);
            nfB = DIST_PAR(3);

            feedAi = DIST_PAR(4);
            feedBi = DIST_PAR(5);
            zfeedAi = DIST_PAR(6);
            zfeedBi = DIST_PAR(7);

            qfA = DIST_PAR(8);
            qfB = DIST_PAR(9);
            refluxi = DIST_PAR(10);
            vapori = DIST_PAR(11);

            mt = DIST_PAR(24);
            md = DIST_PAR(25);
            mb = DIST_PAR(26);

            Pc = DIST_PAR(35);
            Pr = DIST_PAR(36);

            if length(DIST_PAR)==40;
              stepr = DIST_PAR(27);
```

```
    tstepr = DIST_PAR(28);
    stepv = DIST_PAR(29);
    tstepv = DIST_PAR(30);
    stepzfA= DIST_PAR(31);
    tstepzfA = DIST_PAR(32);
    stepfA = DIST_PAR(33);
    tstepfA =DIST_PAR(34);
    stepzfB = DIST_PAR(37);
    tstevzfB = DIST_PAR(38);
    stepfl3 = DIST_PAR(39);
    tstepfB =DIST_PAR(40);
else

stepr=0;
tstepr=0;
stepv=0;
tstepv=0;

stepzfA=0;
tstepzfA=O;
stepfA=0;
tstepfA=0;

stepzfB=0;
tstepzfB=0;
stepfB=0;
tstepfB=0;

end

% Check for disturbances in Reflux

if t<tstepr;
    reflux=refluxi;

else

    reflux=refluxi+stepr;

end

% Check the disturbances vapor boil-up

if t<tstepv;
    vapor=vapori;
else
    vapor=vapori+stepv;
```

```
end

% Check for disturbances in Feed A Composition

if t<tstepzfA;
   zfeedA = zfeedAi;
else

   zfeedA = zfeedAi + stepzfA;
end

% Check for disturbances in Feed B Composition

if t<tstepzfl3;
   zfeedB = zfeedBi;
else
   zfeedB = zfeedBi + stepzfB;
end

% Check for disturbances in Feed A Flowrate

if t<tstepfA;
   feedA=feedAi;
else
   feedA=feedAi+stepfA;
end

% Check for disturbances in Feed B Flowrate

if t<tstepfB;
   feedB=feedBi;
else
   feedB=feedBi+stepfB;
end

% Check the disturbances in Rate of Forward Reaction

if t<tstepkl;
   kl=kli;
else
   kl=kli+stepkl;
end

% Rectifying and stripping section liquid flowrates
lr = reflux;
lsl = reflux+feedA*qfA;
Is = reflux+feedA*qfA+feedB*qtB;
```

```
% Rectitifying and stripping section vapor flowrates

vs = vapor;
vr=vs+feedA*(1-qfA)+feedB*(1-qfB);

% Distillate. and bottoms rates

dist = vr - reflux;
lbot = ls - vs;

% Zeros the function vector

xAdot=zeros(ns,1);

% Material balances

% Overhead receiver

xAdot(1)=(l/md)*(vr*yA(2)-(dist+reflux)*xA(1));

% Rectifiying (top)section

for i=2:nfA-1;
    xAdot(i)=(1/mt)*(lr*xA(i-1)+vr*yA(i+l)-lr*xA(i)-vr*yA(i));
end

%Feed A

xAdot(nfA)=(1/mt)*(lr*xA(nfA-1)+vr*yA(nfA+l)-ls1*xA(nfA)-vr*yA(nfA)+feedA);

% Ratio of fo~'ard reaction rate to reverse reaction rate

K=k1/k2;

% extractive section
for i=nfA+1:nfA+4;
    xAdot(i)=(1/mt)*(ls1*xA(i-1)+vr*yA(i+1)-ls1*xA(i)-vr*yA(i));
end

% Intermediate Section

for i=nfA+5:nfB-1;
    xAdot(i)=(1/mt)*(ls1*xA(i-1)-ls1*xA(i)+vr*yA(i+1)-vr*yA(i))+(k1*(xC(i)*xD(i)/K-xA*xB(i)));
end

% Feed B stage reaction
```

```
    xAdot(nfB)=(1/mt)*(ls1*xA(nfB-l)-ls*xA(nfB)+vs*yA(nfB+1)-vr*yA(nfB))+(k1*(xC(nfB)*xD(nfB)/K-
    xA(nfB)*xB(nfB)));

    % Stripping (bottom) section

    % Reaction in the Stripping Section

    xAdot(nfB+1)=(1/mt)*(ls*xA(nfB)-ls*xA(nfB+1)+vs*yA(nfB+2)-vs*yA(nfB+1))+(k1*(xC(nfB+1)*xD(nfB+1)/K-
    xA(nfB+l)*xB(nfB+l)));

    % Normal stripping section

    for i=nfB+2:ns-1;
    xAdot(i)=(l/mt)*(ls*xA(i-1)+vs*yA(i+l)-ls*xA(i)-vs*yA(i));
    end

    % Reboiler

xAdot(ns)=(l/mb)*(ls*xA(ns-l)-lbot*xA(ns)-vs*yA(ns));

    % Function file of Reactant B

    function xBdot = dist_dynB(t, xB)

    % Declaration of some variables

    global DIST_PAR

    global kli kl
    global k2
    global stepkl
    global tstepkl
    global xA xC xD
    global yA yB yC yD

    % Assignment of values to parameters

    ns = DIST_PAR(l);
    nfA = DIST_PAR(2);
    nfB = DIST_PAR(3);
    feedAi = DIST_PAR(4);
    feedBi = DIST_PAR(5);
    zfeedAi = DIST_PAR(6);
    zfeedBi = DIST_PAR(7);

    qfA = DIST_PAR(8);
```

```
qfB = DIST_PAR(9);
refluxi = DIST_PAR(10);
vapori = DIST_PAR(11 );
mt = DIST_PAR(24);
md = DIST_PAR(25);
mb = DIST_PAR(26);

Pc = DIST_PAR(35);
Pr = DIST_PAR(36);

if Iength(DIST_PAR)==40;
    stepr = DIST_PAR(27);
    tstepr = DIST_PAR(28);
    stepv = DIST_PAR(29);
    tstepv = DIST_PAR(30);
    stepzfA= DIST_PAR(31);
    tstepzfA = DIST_PAR(32);
    stepfA = DIST_PAR(33);
    tstepfA =DIST_PAR(34);
    stepzfB = DIST_PAR(37);
    tstepzfB = DIST_PAR(38);
    stepfB = DIST_PAR(39);
    tstepfB =DIST_PAR(40);
else
    stepr=0; tstepr=0; stepv=0; tstepv=0;
    stepzfA=0; tstepzfA=0; stepfA=0; tstepfA=0;
    stepzfB=0; tstepzfB=0; stepfB=0; tstepfB=0;
end

% Check the disturbances in Reflux

if t<tstepr;
    reflux=refluxi;
else
    reflux=refluxi+stepr;
end

% Check the disturbances Vapor boil-up
if t<tstepv;
    vapor=vapori;
else
    vapor=vapori+stepv;
end

% Check for disturbances in Feed A Composition

if t<tstepzfA;
```

```
    zfeedA = zfeedAi;
else
    zfeedA = zfeedAi + stepzfA;
end

% Check for disturbances in. Feed B Composition

if t<tstepzfB;
    zfeedB = zfeedBi;
else
    zfeedB = zfeedBi + stepzfB;
end

% Check for disturbances in Feed A Flowrate

if t<tstepfA;
    feedA=feedAi;
else
    feedA=feedAi+stepfA;
end

% Check for disturbances in Feed B Flowmte

if t<tstepfB;
    feedB=feedBi;
else
    feedB=feedBi+stepfB;
end

% Check the disturbances in Rate of Forward Reaction

if t<tstepk1;
    kl=kli;
else
    k1 =k1i+stepk1;
end

% Rectitifying and stripping section liquid flowrates

lr = reflux;
ls1 = reflux+feedA*qfA;
ls = reflux+feedA*qfA+feedB*qfB;

% Rectitifying ,and stripping section vapor flowrates

vs = vapor;
vr = vs + feedA*(1-qfA) + feedB*(1-qfB);
```

```
% Distillate and bottoms rates

dist = vr - reflux;
lbot = ls - vs;

% Zeros the function vector

xBdot=zeros(ns,1);

% Material balances

% Overhead receiver

xBdot(1)=(1/md)*(vr*yB(2)-(dist+reflux)*xB(1));

% Rectifying (top) section

for i=2:nfA-1;
   xBdot(i)=(1/mt)*(lr*xB(i-1)+vr*yB(i+l)-lr*xB(i)-vr*yB(i));
end

%Feed A

xBdot(nfA)=(1/mt)*(lr*xB(nfA-1)+vr*yB(nfA+1)-ls1*xB(nfA)-vr*yB(nfA));

% Ratio of forward reaction rate to reverse reaction rate

K=kl/k2;

%extractive section

for i=nfA+ 1 :nfA+4;
   xBdot(i)=(1/mt)*(ls1*xB(i-1)+vr*yB(i+l)-lsl*xB(i)-vr*yB(i));
end

% Ratio of forward reactiort rate to reverse reaction rote

K=kl/k2;

% Intermediate Section

for i=nfA+5 :nfB-1;
xBdot(i)=(1/mt)*(ls1*xB(i-1)-ls1*xB(i)+vr*yB(i+ 1)-vr*yB(i))+(kl*(xC(i)*xD(i)/K-xA(i)*xB(i)));
end

% Feed B stage reaction
```

```
xBdot(nfB)=(l/mt)*(feedB *zfeedB+ls1*xB(nfB-1)-ls*xB(nfB)+vs*yB(nfB+1)-vr*yB(nfB))+(kl*(xC(nfB)*xD(nfB)/K-xA(nfB)*xB(nfB)));

% Stripping (bottom) section

% Normal stripping section

for i=nfB+2:ns-1;
  xBdot(i)=(l/mt)*(ls*xB(i-1)+vs*yB(i+l)-ls*xB(i)-vs*yB(i));
end

% Reboiler

xBdot(ns)=(1/mb)*(ls*xB(ns-1)-lbot*xB(ns)-vs*yB(ns));
% ~mction file for Product C

function xCdot = dist_dynC(t, xC)

% Declaration of some variables

global DIST_PAR

global kli kl
global k2
global stepkl
global tstepkl
global xA xB xD
global yA yB yC yD

% Assignment of values to parameters

ns = DIST_PAR(1);
nfA = DIST_PAR(2);
nfB = DIST_PAR(3);
feedAi = DIST_PAR(4);
feedBi = DIST_PAR(5);
zfeedAi = DIST_PAR(6);
zfeedBi = DIST_PAR(7);

qfA = DIST_PAR(8);
qfB = DIST_PAR(9);
refluxi = DIST_PAR(10);
vapori = DIST_PAR(11);
mt = DIST_PAR(24);
md = DIST_PAR(25);
mb = DIST_PAR(26);
```

```
Pc = DIST_PAR(35);
Pr = DIST_PAR(36);

mt = DIST_PAR(24);

md = DIST_PAR(25);
mb = DIST_PAR(26);

if length(DIST_PAR)==40;

stepr = DIST_PAR(27);
tstepr = DIST_PAR(28);
stepv = DIST_PAR(29);
tstepv = DIST_PAR(30);
stepzfA= DIST_PAR(31);
tstepzfA = DIST_PAR(32);
stepfA = DIST_PAR(33);
tstepfA =DIST_PAR(34);

stepzfB = DIST_PAR(37);
tstepztB = DIST_PAR(38);
stepm = DIST_PAR(39);
tstepm =DIST_PAR(40);
else
stepr=0; tstepr=0; stepv=0; tstepv=0;
stepzfA=0; tstepzfA=0; stepfA=0; tstepfA=0;
stepzfB=0; tstepzfB=0; stepfB=0; tstepfB=0;
end

% Check for disturbances in. Reflttx

if t<tstepr;
   reflux=refluxi;
else
   reflux=refluxi+stepr;
end

% Check for disturbances Vapor boil-up

if t<tstepv;
   vapor=vapori;
else
   vapor=vapori+stepv;
end

% Check for disturbances in Feed A Composition
```

```
if t<tstepzfA;
    zfeedA = zfeedAi;
else
    zfeedA = zfeedAi + stepzfA;
end

% Check for disturbances in :Feed B Composition

if t<tstepzfB;
    zfeedB = zfeedBi;
else
    zfeedB = zfeedBi + stepzfB;
end

% Check fbr disturbances in Feed A Flowrate

if t<tstepfA;
    feedA=feedAi;
else
    feedA=feedAi+stepfA;
end

% Cheek tbr disturbances in Feed B Flowrate

if t<tstepfB;
    feedB=feedBi;
else
    feedB=feedBi+stepfB;
end

% Check for disturbances in Rate of Forward Reaction

if t<tstepkl;
    kl=kli;
else
    kl=kli+stepkl;
end

% Rectifying and stripping section liquid flowrates

lr = reflux;
lsl = reflux+feedA*qfA;
ls = reflux+feedA*qfA+feedB*qfB;

% Recti~'ing and stripping section vapor fiowrates
```

```
vs = vapor;

vr = vs + feedA*(l-qfA) + feedB*(l-qfB);

% Distillate and bottoms rates

dist = vr - reflux;
lbot = Is - vs;

% Zeros the function vector

xCdot=zcms(ns, 1);

% Material balances

xCdot(1)=(1/md)*(vr*yC(2)-(dist+rcflux)*xC(1));

% R.ecti~'ing (~p)section

for i=2 :nfA-1;
    xCdot(i)=(1/mt) *(lr*xC(i-1)+vr*yC(i+1)-lr*xC(i)-vr*yC(i));
end

%Feed A

xCdot(nfA)=(1/mt)*(lr*xC(nfA-1)+vr*yC(nfA+1)-ls1*xC(nfA)-vr*yC(nfA));

% Ratio of fbrward reaction rate to reverse reaction rate

K=k1/k2

%extractive section

for i=nfA+ 1 :nfA+4;
    xCdot(i)=(1/mt)*(ls1*xC(i-1)+vr*yC(i+1)-ls1*xC(i)-vr*yC(i));
end

% Ratio of forward reaction rate to reverse reaction rate

K=kl/k2;

% Intermediate Section

for i=nfA+ 1 :ntB- 1;
xCdot(i)=(1/rnt)*(ls1*xC(i-t)-ls1*xC(i)+vr*yC(i+1)-vr*yC(i))+feedA*xA(i)+(kl(xA(i)*xB(i)-xC(i)*xD(i)/k));
end
```

```
% Feed B stage reaction

xCdot(nfB)=(1/rnt)*(feedB*zfeedB+ls1*xC(nfB-1)-ls*xC(nfB)+vs*yC(nfB+1)-vr*yC(nfB))+(kl*(xA(nfB)*xB(nfB)-xC(nfB)*xD(nfB)/K));

% Stripping (bottom) section

% Reaction ill the Stripping Section

xCdot(nfB+ 1 )=(1/mt)*(ls*xC(nfB)-ls*xC(nfB+ 1 )+vr*yC(nfB+2)-vr*yC(nfB+ 1 ))+(1:1 *(xA(nfB+ 1)*xB(nfB+ 1)-xC(nfB+l)*xD(nfB+l)/K));

% Normal stripping section
for i=nfB+2:ns-1;
xCdot(i)= (1/mt)*(ls*xC(i-1)+vs *yC(i+ 1)-ls*xC(i)-vs*yC(i));

end
% Reboiler
xCAot(m)=(1/rnb)*(ls*xC(ns-1)-lbot*xC(ns)-vs*yC(ns));
% Function file the Product D

function xDdot = dist_dynD(t, xD)

% Declaration of some variables

global DIST_PAR
global kli kl
global k2
global stepkl
global tstepkl
global xA xB xC
global yA yB yC yD

% Assigmnent ofvalu.es to variables

ns = DIST_PAR(1);
nfA = DIST_PAR(2);
nfB = DIST_PAR(3);
feedAi = DIST_PAR(4);
feedBi = DIST_PAR(5);
zfeedAi = DIST_PAR(6);
zfeedBi = DIST_PAR(7);

qfA = DIST_PAR(8);
qfB = DIST_PAR(9);
refluxi = DIST_PAR(10);
vapori = DIST_PAR(11);
```

```
mt = DIST_PAR(24);
md = DIST_PAR(25);
mb = DIST_PAR(26);

Pc = DIST_PAR(35);
Pr = DIST_PAR(36);

if length(DIST_PAR)==40;
    stepr = DIST_PAR(27);
    tstepr = DIST_PAR(28);
    stepv = DIST_PAR(29);
    tstepv = DIST_PAR(30);
    stepzfA= DIST_PAR(31);
    tstepzfA = DIST_PAR(32);
    stepfA = DIST_PAR(33);
    tstepfA = DIST_PAR(34);
    stepzfB = DIST_PAR(37);
    tstepzfB = DIST_PAR(38);
    stepfB = DIST_PAR(39);
    tstepfB = DIST_PAR(40);

else
    stepr=0; tstepr=0; stepv=0; tstepv=0;
    stepzfA=0; tstepzfA=0; stepfA=0; tstepfA=0;
    stepzfB=0; tstepzfB=0; stepfB=0; tstepfB=0;
end

% Check for disturbances in Reflux

if t<tstepr;
    reflux=refluxi;
else
    reflux=refluxi+stepr;
end

% Check for disturbances Vapor boil-up

if t<tstepv;
    vapor=vapori;
else
    vapor=vapori+stepv;
end

% Check for disturbances in Feed A Composmon

if t<tstepzfA;
    zfeedA=zfeedAi;
```

```
else
    zfeedA=zfeedAi+stepzfA;
end

% Check for disturbances in Feed B Composition

if t<tstepzfB;
    zfeedB=zfeedBi;
else
   zfeedB=zfeedBi+stepzfB;
end

% Check for dism~ in Feed A Flowrat~

if t<tstepfA;
    feedA=feedAi;
else
 feedA=feedAi+stepfA;
end

% Check for disturbances in Fe~d B Flowrate

if t<tstepfB;
    feedB=feedBi;
else
    feedB=feedBi+stepfB;
end

% Check for disturbanc~ in Rate of Forward Reaction

if t<tstepkl;
    kl=kli;
else
    kl=kli+stepkl;
end

% Rectifying and stripping section liquid flowrates

lr = reflux;
ls1 = reflux+feedA*qfA;
Is = reflux+feedA*qfA+feedB,qfl3;

% Rectit~ing ,and stripping section vapor flowrates

vs = vapor;

vr = vs + feedA*(1-qfA) + feedB*(l-qtB);
```

```
% Distillate and bottoms rates

dist = vr - reflux;
lbot = ls - vs;

% Zeros the function vector

xDdot=zeros(ns, 1);

% Material balances

% Overhead r~eiver

xDdot( 1 )=( 1/md)*(vr*yD(2)-(dist+reflux)*xD( 1 ));

% Rectifying (top)section

for i=2:nfA-1;
xDdot(i)=(1/mt)*(lr*xD(i-1)+vr*yD(i+l)-lr*xD(i)-vr*yD(i));
end

%Feed A

xDdot(nfA)=( 1/mt)*(lr*xD(nfA- 1 )+vr*yD(nfA+ 1 )-ls1*xD(nfA)-vr*yD(nfA));

% Ratio of forward reaction rate to reverse reaction rate

K=kl/k2;

%extractive section

for i=nfA+ 1 :nfA+4;
xDdot(i)=(1/mt)*(ls1*xD(i-1)+vr*yD(i+1)-ls1*xD(i)-vr*yD(i));
end

% Ratio of forward reaction rote to revel~e reaction rate

K=k1/k2

% Intermediate Section

for i=nfA+ 1 :nfB-1;
xDdot(i)=(1/mt)*(ls1*xD(i-1)-ls1*xD(i)+vr*yD(i+1)-vr*yD(i))+feedA*xA(i)+(kl*(xA(i)*xB(i)-xC(i)*xD(i)/K));
end

% Feed B stage reaction
```

```
xDdot(nfB)=(l/mt)*(feedB*zfeedB+lsl *xD(nfB-1)-ls*xD(nfB)+vs*yD(nfB+l)-vr*yD(nfB))+(kl*(xA(ntB)*xB(nfB)-xC(nfB)*xD(nfB)));

% Stripping (bottom) section

% Reaction in the Stripping Section

xDdot(nfB+ 1)=(1/mt)*(ls*xD(nfB)-ls*xD(nfB+ 1)+vs*yD(nfB+2)-vs*yD(nfB+l))+(kl*(xA(nfB+l)*xB(ntB+l);

% Normal stripping section

for i=nfB+2:ns- 1;
xDdot(i)=(1/mt)*(ls*xD(i'l)+vs*yD(i+ 1)-ls*xD(i)-vs*yD(i));
end

% Reboiler
xDdot=(l/mb)*(ls*xD(ns'l)'lb°t*xD(ns)'vs*yD(ns));
```

2. Feedbacksetpoint.m

```
%For Setpoint
%Step
K1=2.45;T1=4.9755;d1=26.805;

%IAE
Kc=(0.758/K1)*((d1/T1)^-0.861);Ti=T1/(1.02-0.323*(d1/T1));Td=0;
sim('feedback');
figure(1);
plot(time,T3);
title('Close Loop IAE Response-Setpoint');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(2);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(3);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%ITAE
Kc=(0.586/K1)*((d1/T1)^0.916);Ti=T1/(1.03-0.165*(d1/T1));Td=0;
sim('feedback');
figure(4);
plot(time,T3);
title('Close Loop ITAE Response-Setpoint');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(5);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(6);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;


%IMC
Kc=(1/K1)*(T1/T1+d1);Ti=T1;Td=0;
sim('feedback');
figure(7);
plot(time,T3);
title('Close Loop IMC Response-Setpoint');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(8);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(9);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%HA
Kc=(0.14/K1)+(0.28*T1/d1*K1);Ti=0.33*d1+(6.8*d1*T1/(10*d1+T1));Td=0;
sim('feedback');
figure(10);
plot(time,T3);
title('Close Loop HA Response-Setpoint');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(11);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(12);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;



%ISE
Kc=(1.495/K1)*(d1/T1)^(-0.945);Ti=(T1/(1.101)*(d1/T1)^0.771);Td=0;
sim('feedback');
figure(13);
plot(time,T3);
title('Close Loop ISE Response');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(14);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(15);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%Cohen Coon
Kc=(T1/(K1*d1))*(0.9+d1)/(12*T1);Ti=d1*(30+(3*d1/T1)/(9+20*d1)/T1);Td=0;

sim('feedback');
figure(16);
plot(time,T3);
title('Close Loop Cohen Coon Response-Setpoint');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(17);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(18);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;



%Ziegler and Nichols(1942),Model Method 2
Kc=((0.9*T1)/(K1*d1));Ti=d1*3.33;Td=0;
sim('feedback');
figure(19);
plot(time,T3);
title('Close Loop Ziegler and Nichols Response-Model Method 2');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(20);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(21);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%Hazebroek and Van der Waerden(1950),Model Method 2
Kc=(0.621386*T1)/(K1*d1);Ti=d1*7.62715178;Td=0;
sim('feedback');
figure(22);
plot(time,T3);
title('Close Loop Hazebroek and Van der Waerden Response-Model Method 2');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(23);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(24);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;


%Chien(1952), Servo, Model: Method 2, 0% overshoot
Kc=(0.35*T1)/(K1*d1);Ti=1.17*T1;Td=0;
sim('feedback');
figure(25);
plot(time,T3);
title('Close Loop Chien(1952), Servo, Model : Method 2, 0% Overshoot');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(26);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(27);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%Chien(1952), Servo, Model: Method 2, 20% overshoot
Kc=(0.6*T1)/(K1*d1);Ti=T1;Td=0;
sim('feedback');
figure(28);
plot(time,T3);
title('Close Loop Chien (1952), Servo, Model : Method 2, 20% overshoot');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(29);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(30);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;



%Cohen and Coon (1953), Model: Method 2
Kc=(1/K1)*(0.9*(T1/d1)+0.083);Ti=T1*((3.33*(d1/T1)+0.31*(d1/T1)^2)/(1+2.22*(d1/T1)));Td=0;
sim('feedback');
figure(31);
plot(time,T3);
title('Close Loop Cohen and Coon (1953), Model:Method 2');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(32);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(33);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%Two Constraints Method- Wolfe (1951), Model: Method 3
Kc=(4.37450102*T1)/(K1*d1);Ti=3.63221264*d1;Td=0;
sim('feedback');
figure(34);
plot(time,T3);
title('Close Loop Two Constraints Method -Wolfe (1951), Model : Method 3');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(35);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(36);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```

```
%Two Constraints Criterion- Murrill (1967), Model: Method 4
Kc=(0.928/K1)*((T1/d1)^0.946);Ti=(T1/1.078)*((d1/T1)^0.583);Td=0;
sim('feedback');
figure(37);
plot(time,T3);
title('Close Loop Two Coonstraints Criterion-Murrill (1967), MEthod : Method 4');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(38);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(39);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
```

```
grid on;



%McMillan (1994), Model: Method 4
Kc=K1/3;Ti=d1;Td=0;
sim('feedback');
figure(40);
plot(time,T3);
title('Close Loop McMillan (1994), Model:Method 4');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(41);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(42);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;



%St. Clair (1997), Model: Method 4
Kc=(0.333*T1)/(K1*d1);Ti=T1;Td=0;
sim('feedback');
figure(43);
plot(time,T3);
title('Close Loop St. Clair (1997), Model : Method 4');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(44);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(45);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
```

```
ylabel('mv');
grid on;



%Shinskey (2000), (2001) Model: Method 2
Kc=(0.667*T1)/(K1*d1);Ti=3.78*T1;Td=0;
sim('feedback');
figure(46);
plot(time,T3);
title('Close Loop Shinskey (2000), (2001) Model: Method 2');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(47);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(48);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;


%Hay (1998) Servo Tuning 1, Model: Method 2
Kc=(6/K1);Ti=9.5*d1;Td=0;
sim('feedback');
figure(49);
plot(time,T3);
title('Close Loop Hay (1998), Servo Tuning 1, Model : Method 2');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(50);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(51);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
```

```
ylabel('mv');
grid on;



%Hay (1998) Servo Tuning 2, Model: Method 2
Kc=(4/K1);Ti=10*d1;Td=0;
sim('feedback');
figure(52);
plot(time,T3);
title('Close Loop Hay (1998) Servo Tuning 2, Model : Method 2');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(53);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(54);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;



%Servo Tuning
%Minimum IAE- Rovira et al. (1969), Model: Method 4
Kc=(0.758/K1)*((T1/d1)^0.861);Ti=T1/(1.020-0.323*(d1/T1));Td=0;
sim('feedback');
figure(55);
plot(time,T3);
title('Close Loop Minimum IAE-Rovira et al. (1969), Model : Method 4');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(56);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(57);
plot(time,Mv);
```

```
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;


%Minimum IAE- Marlin (1995), Model: Method 1
Kc=1.4/K1;Ti=0.72*d1;Td=0;
sim('feedback');
figure(58);
plot(time,T3);
title('Close Loop minimum IAE- Marlin (1995), Model:Method 1');
xlabel('Time(min)');
ylabel('Composition i-butane (mass fraction)');
grid on;
figure(59);
plot(time,Error);
title('Error')
xlabel('Time(min)');
ylabel('Error');
grid on;
figure(60);
plot(time,Mv);
title('Manipulated variable');
xlabel('Time(min)');
ylabel('mv');
grid on;
```