Otto-von-Guericke University

Fakultät für Prozessdynamik und Prozesssynthese
Institut für Automatisierungstechnik (IFAT)

Procees Control: Project Task

Control of a Multivariable Process

Authors:	Enrollment ID:
Darshan Jogani	231922

Friday 2nd July, 2021

Supervisor:
Dr. Ilknur Disli-Kienle
Institut für Automatisierungstechnik (IFAT)
Universitätsplatz 2
39106 Magdeburg, Germany

Tasks

Process modelling

1. Propose a process (Newell and Lee evaporator) to control. Determine the control design aspects of this process.
2. Obtain a mathematical model for this process based on your literature survey and do the degrees of freedom analysis.
3. Find the steady state solution of mathematical model equations (typically an ODE system) in MATLAB.
4. Simulation of dynamic behaviour of the process for some step change scenerios in the system inputs in Matlab as well as in Simulink.

Dynamic Analysis

1. Do numerical linearization of the process model.
2. Do stability analysis of linearized process model.
3. Select the best control structure by computing and comparing available criteria (e.g. RGA analysis) for proposed alternative control structures.

Controller Design, Implementation and Comparison

1. Design suitable controllers for the best control structure by using two different controller tuning methods.
2. Implement the designed controllers and test the performance of controllers on the nonlinear process.
3. Compare the performance of controllers designed by different methods.

Contents

List of Figures

List of Tables

Part I

Process Modelling

Chapter 1

Process Description

1.1 Experimental setup and procedure

An evaporator is generally used to separate solvent from a feed-stream by means of evaporation, mainly in paper or sugar industries.

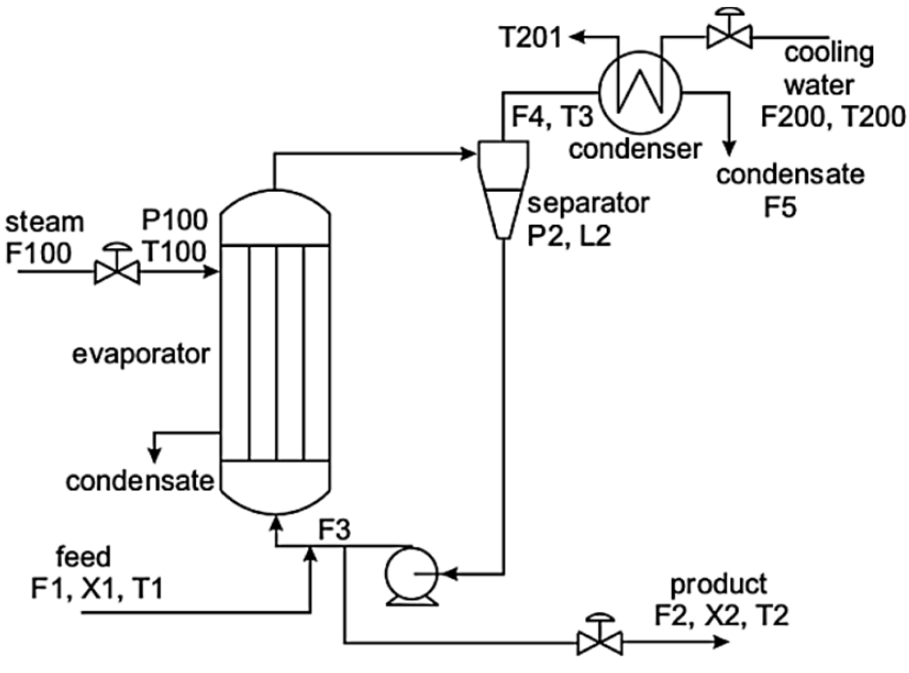

Figure 1.1: A typical evaporation system [1]

As it is shown in the above figure 1.1, an evaporation system with a control valve, flow can be controlled.

A liquid feed is mixed with re-circulationg liquor is fed to an evaporator and for heating medium steam is used. An evaporator is basically a heat exchanger in which heat is transfered to a feed-stream from steam as heating medium, that evaporatesmthe feed and condensate leaves at the bottom. [3]

The re-circulating liquor boils in an evaporator and two phase mixture of evaporates (liquid + vapor) flows to the separator, where liquid and vapor are being separated. Then liquid is pumped back to an evaporator and some of it drawn as a final-product.

Then the separated vapor is fed to a condensor which is againg as heat exchanger. the only differance here is that, heat is transfered from vapor to cooling water and hence, vapor gets condensed anf finally leaves as a condensate, as shown in figure 1.1. [2]

Table 1.1: Operating parameters [2]

Properties	**Value**	**Unit**
Feed flowrate ($F1$)	10.0	kg/min
Product flowrate ($F2$)	2.0	kg/min
Circulating liquor flowrate ($F3$)	50.0	kg/min
Vapor flowrate ($F4$)	8.0	kg/min
Condensate flowrate ($F5$)	8.0	kg/min
Feed composition ($X1$)	5.0	%
Product composition ($X2$)	25.0	%
Feed temperature ($T1$)	40.0	°C
Product temperature ($T1$)	84.6	°C
Vapour temperature ($T2$)	80.6	°C
Separator level ($L2$)	1.0	m
Operating pressure ($P2$)	50.5	kPa
Steam flowrate ($F100$)	9.3	kg/min
Steam temperature ($T100$)	119.9	°C
Steam pressure ($P100$)	194.7	kPa
Heater duty ($Q100$)	339.0	kW
Cooling water flowrate ($F200$)	208.0	kg/min
Cooling water inlet temperature ($T200$)	25.0	°C
Cooling water outlet temperature ($T201$)	43.1	°C
Condensor duty ($Q200$)	307.9	kW

Chapter 2

Problem Defination and Modelling

2.1 Control design aspect

2.1.1 Control objectives

The main control objective is the product quality, in other words keeping the variation in the product quality as small as possible maximiyzes the profitability.
However, evaporater operational safety is also necessary. this requires the pressure (P2) inside evaporator and separator liquid height (L2) needs to be control. Because, if separator overflows, the condensor will get flooded and eventually get damaged, and if level gets too low, separator will get dry and eventually the pump will get damaged. [2]

2.1.2 Input variables

In order to fullfil control objectives, certain variables needs to be manipulated and as a result of that, other variables gets disturbed. [2] [4]

1. Manipulated variables
   - Product flowrate ($F2$)
   - Steam pressure ($P100$)
   - Cooling water flowrate ($F200$)
2. Disturbed variables
   - Feed flowrate ($F1$)

- Circulating liquor flowrate ($F3$)
- Feed composition ($X1$)
- Feed temperature ($T1$)
- Cooling water inlet temperature ($T200$)

2.1.3 Output variables

1. Liquid level in separator ($L2$)
   - It is directly related to the control objective.
     - If $L2$ is too high: Separator overflows and damages condensor.
     - If $L2$ is too low: Separator runs dry and damages pump.
2. Product composition ($X2$)
3. Operating pressure in the evaporator ($P2$)

2.1.4 Constraints

1. Hard constraints (Input variables)
   - Product flowrate: $(F2 \leq 4) kg/min$
   - Steam flowrate: $(F100 \leq 20) kg/min$
   - Cooling water flowrate: $(F200 \leq 400) kg/min$
   - Operating pressure in the evaporator: $(P2 \leq 100) kPa$
2. Soft constraints (Output variables)
   - Liquid level in separator: $(L2 \leq 2) m$
   - Product composition: $X2 = 25\%$

2.1.5 Operating conditions

It is a Continuous process

1. It must have feed and product streams moving fluid in and out of the process all the time along with long term operation period with constant operating conditions.
2. The system is nonlinear and multivariable system. [4]

2.1.6 Control structure

1. SISO (Single input-Single output)
2. It is possible to use feedforward control is combined with feedback control.

2.2 Mathematical model and the degree of freedom analysis

2.2.1 Model equations

The model for a forced circulation evaporator can be divided into four parts:

The separator

A total mass balance around the separator gives [2]:

$$\rho A \frac{dL2}{dt} = F1 - F2 - F4 \tag{2.1}$$

Where, ρ is the liquid density and A is the cross-sectional area of the separator.

The evaporator

It is modelled by the following equations [2]:

$$M \frac{dX2}{dt} = F1 * X1 - F2 * X2 \tag{2.2}$$

$$C \frac{dP2}{dt} = F4 - F5 \tag{2.3}$$

$$T2 = 0.5616P2 + 0.3126X2 + 48.43 \tag{2.4}$$

$$T3 = 0.507P2 + 55.0 \tag{2.5}$$

$$F4 = \frac{(Q100 - F1Cp(T2 - T1))}{\lambda} \tag{2.6}$$

Where, $F4$ is the vapor flowrate, $F5$ is the condensate flowrate, $T2$ and $T3$ are the product and the vapor temperatures respectively, $Q100$ is the heater duty, and the coefficients have the following values: $M = 20kg, C = 4\frac{kg}{kPa}, Cp = 0.07\frac{kW}{K(kg/min)}, \lambda = 38.5\frac{kW}{(kg/min)}$. [2]

The steam jacket

The Heater Steam Jacket is described by equations [2]:

$$T100 = 0.1538 * P100 + 90.0 \tag{2.7}$$

$$Q100 = 0.16(F1 + F3)(T100 - T2) \tag{2.8}$$

$$F100 = \frac{Q100}{\lambda s} \tag{2.9}$$

Where, $T100$ is the steam temperature, $F100$ is the steam flowrate, and λs is a coefficient with value $\lambda s = 36.6\frac{kW}{(kg/min)}$. Note that $Q100$ is needed as an output to the evaporator. [2]

The condensor

The condenser is also modelled as a set of algebraic equations [2]:

$$Q200 = \frac{UA2(T3 - T200)}{1 + \frac{UA2}{2CpF200}} \tag{2.10}$$

$$T201 = T200 + \frac{Q200}{F200 * Cp} \tag{2.11}$$

$$F5 = \frac{Q200}{\lambda} \tag{2.12}$$

Where, $Q200$ is the condenser duty, $T201$ is the cooling water outlet temperature, and $UA2$ is a coefficient with value $UA2 = 6.84kW/K$. The other coefficients have the same values as above. [2]

2.2.2 The degree of freedom analysis

- Total number of unknown variables, $(NV) = 20$
- Total number of independent equations, $(NE) = 12$
- Degree of freedom [4],

$$DOF = NV - NE \tag{2.13}$$

From equation 2.13, The system has a degree of freedom 8, thus 8 variables are required to get $DOF = 0$ ($\because DOF = 0 \rightarrow$ The system is exactly specifed.).

These variables are: $F2, P100, F200$ (manipulated variables) and $F1, X1, T1, F3, T200$ (disturbance variables) (See section 2.1.2 on page 4).

2.3 Mathamatical Model and Variables

2.3.1 Approaches for modelling

Modelling can be calssified into two types based on approach:

1. **Theoritical process modelling**
   - Based on fundamental laws, using general conversion principles.
2. **Empirical process modelling**
   - Based on experimental data, approximating a model form in terms of a transfer function.

2.3.2 Procedure for process model development

Development of process model is a stepwise process:

- Problem defnition
- Model formulation
- Parameter estimation
- Model validation

It is always important to know that, any model is as accurate and as simple as you defines it. In other words, the accuracy and complexity of any model is depends on how it is defined and used.

2.3.3 Classification of Models

Any model can be classified into 3-types:

1. Lumped parameter vs. distributed parameter
2. Linear vs. nonlinear
3. Deterministic vs. stochastic

2.3.4 Model Equations

In this perticular evaporator system as per the Project Task-1, it is been confirmed that, there are two types of model equations.

Algebraic Equations

1. **The evaporator**

$$T2 = 0.5616P2 + 0.3126X2 + 48.43 \tag{2.14}$$

$$T3 = 0.507P2 + 55.0 \tag{2.15}$$

$$F4 = \frac{(Q100 - F1Cp(T2 - T1))}{\lambda} \tag{2.16}$$

2. **The steam jacket**

$$T100 = 0.1538 * P100 + 90.0 \tag{2.17}$$

$$Q100 = 0.16(F1 + F3)(T100 - T2) \tag{2.18}$$

$$F100 = \frac{Q100}{\lambda s} \tag{2.19}$$

3. **The condensor**

$$Q200 = \frac{UA2(T3 - T200)}{1 + \dfrac{UA2}{2CpF200}} \tag{2.20}$$

$$F5 = \frac{Q200}{\lambda} \tag{2.21}$$

ODEs

1. **Product composition**

$$\frac{dX2}{dt} = \frac{F1 * X1 - F2 * X2}{M} \tag{2.22}$$

2. **Operating pressure**

$$\frac{dP2}{dt} = \frac{F4 - F5}{C} \tag{2.23}$$

3. **Liquid level height in separator**

$$\frac{dL2}{dt} = \frac{F1 - F2 - F4}{\rho A} \tag{2.24}$$

2.3.5 Variables and Operating Parameters

Variables and Operating parameters that is being used to solve this system of equations are listed below. [2]

- $\rho A = 20kg/m$
- $M = 20kg$
- $C = 4\frac{kg}{kPa}$
- $Cp = 0.07\frac{kW}{K(kg/min)}$
- $\lambda = 38.5\frac{kW}{(kg/min)}$
- $\lambda s = 36.6\frac{kW}{(kg/min)}$
- $UA2 = 6.84kW/K$

Chapter 3

Simulation of the Nonlinear Process Model

3.1 Define an ODE function in MATLAB

Writing a MATLAB function for solving the system of ODEs is a cruical part of simulation process.

Mainly Lecture notes, guideline videos and Mathworks self study course is used to learn, how to solve the system of ODEs.

This function defination can be found in detail in `evapmod.m` file. (Kindly click on the file name to open it.)

3.2 Steady-state response

Firstly, the initial conditions and range of simulation is defined. Secondly, the ode solver (`ode45`) is being called to solve this system of ODEs using the function that was defined earlier.

Then simulation result is being plot (figure 3.1) in order to observe the response. After several changes in initial conditions, the steady-state is achieved.

After observing the figure 3.1, now another plot is being plotted in which the output should start with the steady-state. This can be observed in figure 3.2. This simulation in MATLAB can be found in detail in `sim-evap.m` file. (Kindly click on the file name to open it.)

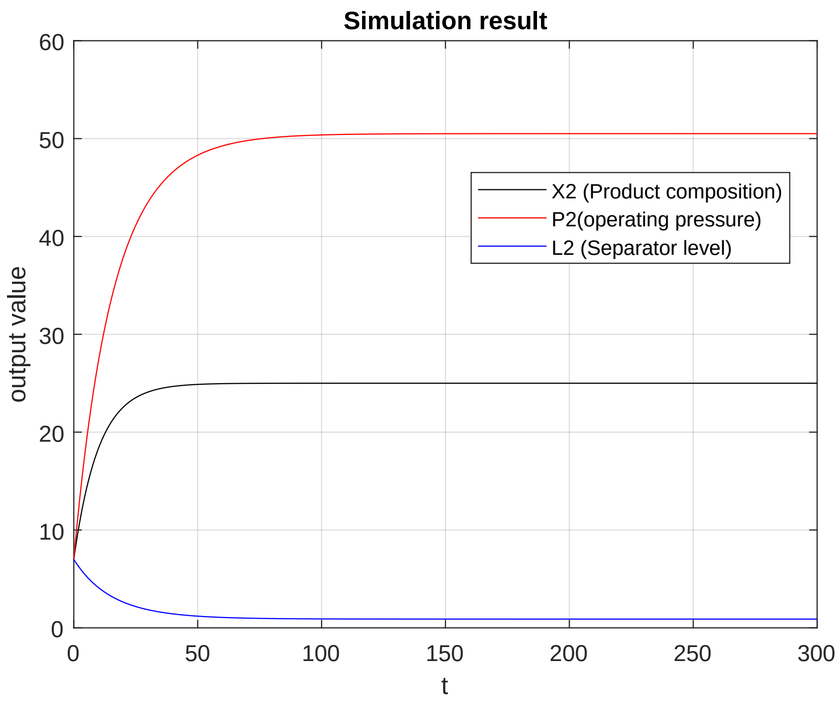

Figure 3.1: Simulation result

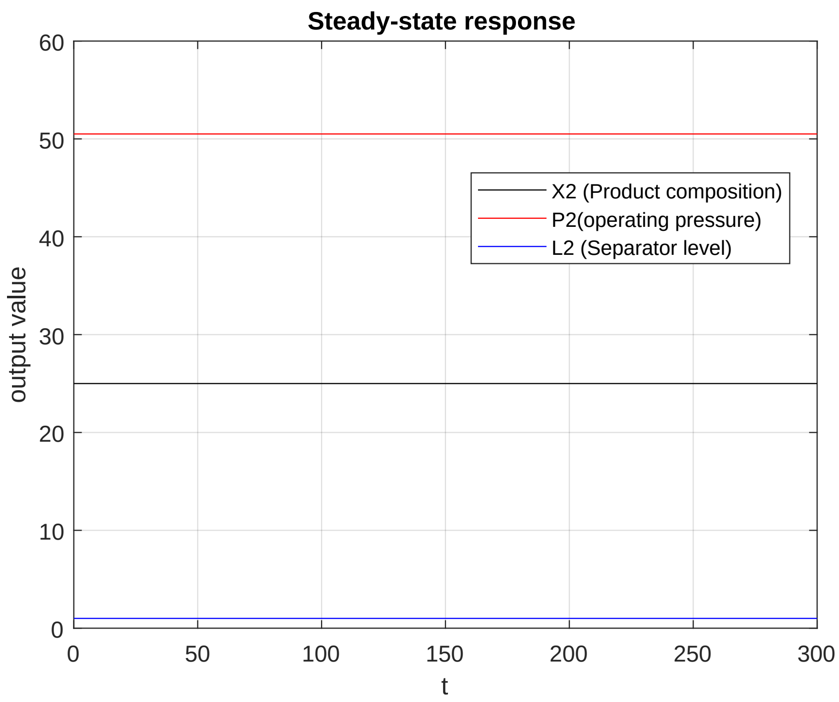

Figure 3.2: Steady-state response

3.3 Dynamic Simulation Response

Now, here some disturbances are created in input variables to see how it affects the steady state response of the entire system. It is described as below.

3.3.1 Step Change in Feed Flowrate (F1)

Simulation scenarios : Step change in some disturbances.
a) A step change of $+10\%$ in the feed flow rate, F1
b) A step change of -10% in the feed flow rate, F1

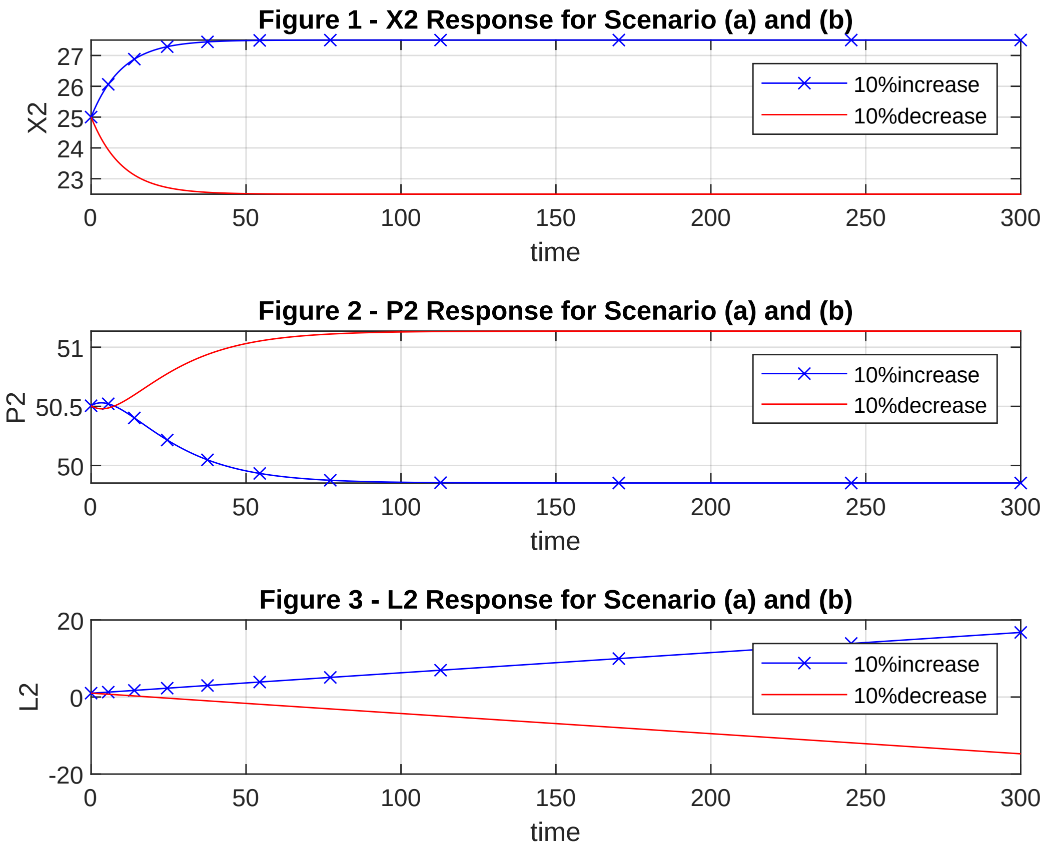

Figure 3.3: Step Change in Feed Flowrate (F1)

3.3.2 Step Change in Feed Composition (X1)

Simulation scenarios : Step change in some disturbances.
c) A step change of $+15\%$ in the feed composition, X1
d) A step change of -15% in the feed composition, X1

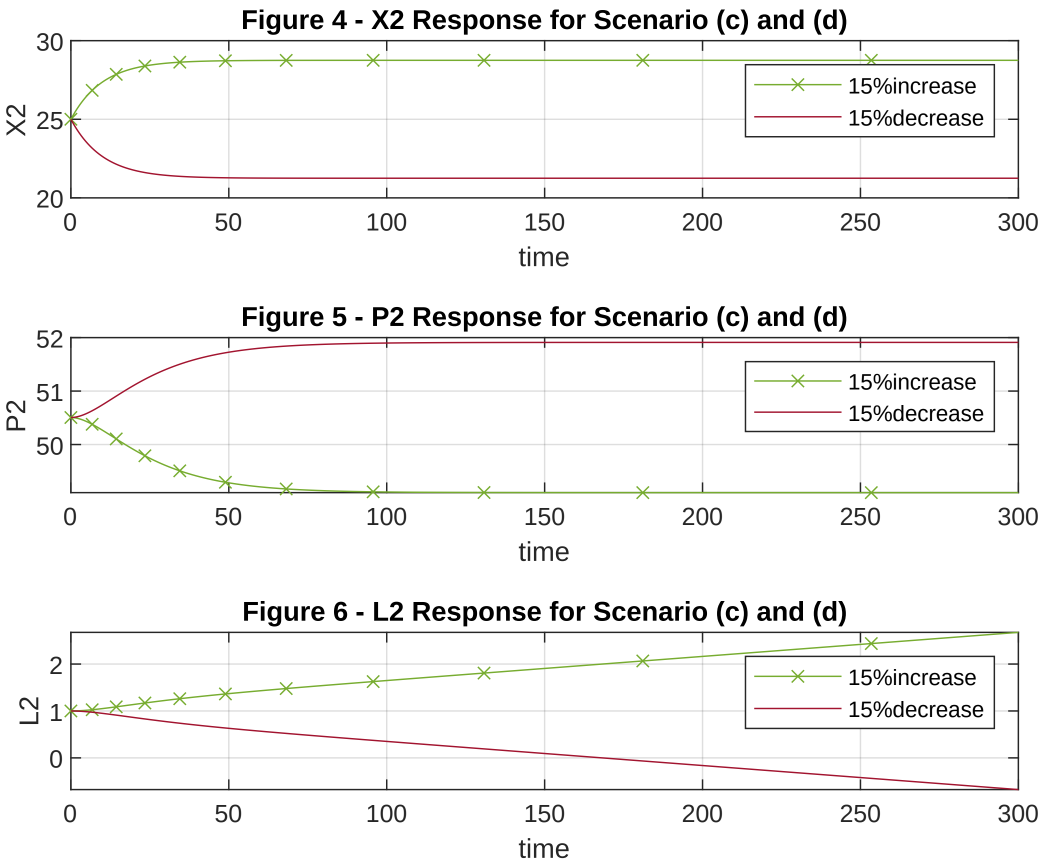

Figure 3.4: Step Change in Feed Composition (X1)

3.3.3 Step Change in Cooling Water Inlet Temperature (T200)

Simulation scenarios : Step change in some disturbances.
e) A step change of +20% in the cooling water inlet temperature, T200
f) A step change of −20% in the cooling water inlet temperature, T200

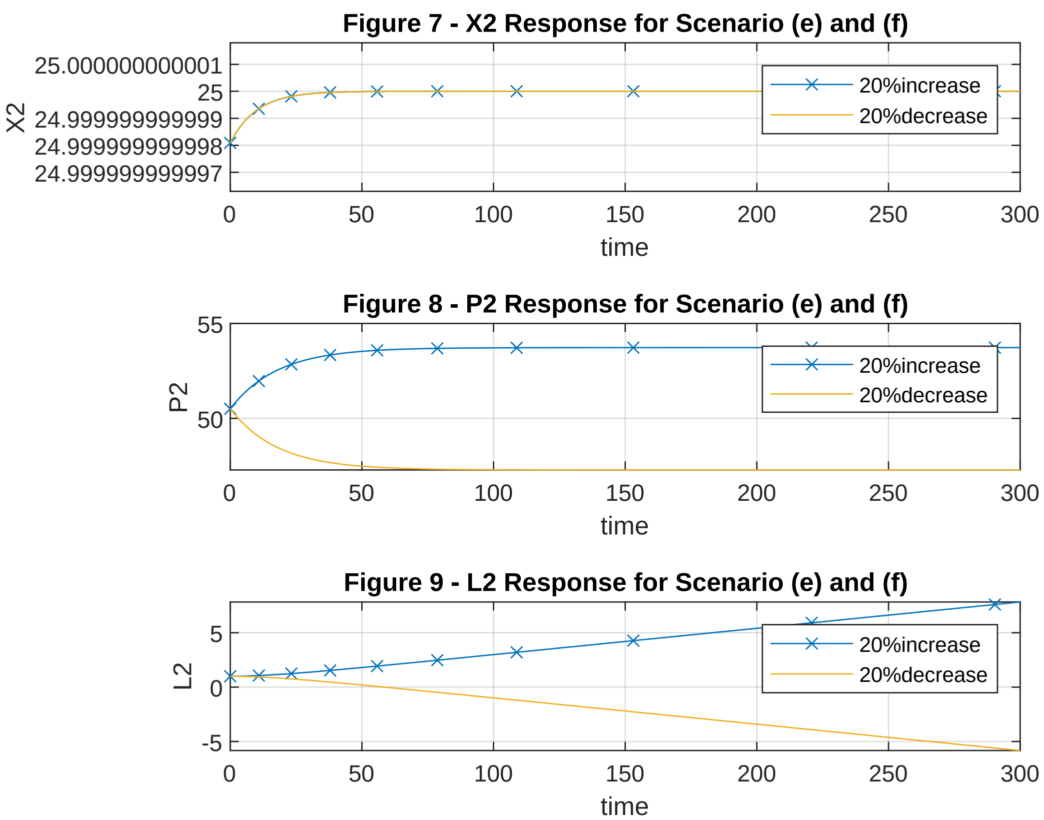

Figure 3.5: Step Change in Cooling Water Inlet Temperature (T200)

Part II

Dynamic Analysis

Chapter 4

Linearization

4.1 Process Model in Simulink

For process modelling, the S-function is establised for different cases and then ised it in simulink for simulation.

4.1.1 S-Function

Why does the S-Function (system-function) is used? [5]

- S-Function provides the powerful mechanism for extending the capabilities of Simulink.
- In other words, S-Function is very powerful link between MATLAB and Simulink. It can be used to instruct the Simulink process model for it to how to work and operate the model.
- S-Function makes the simulation work easier and short.

4.1.2 Simulation using S-Function in Simulink

Here, complete simulink model is showed which can be further used for the linear analysis.

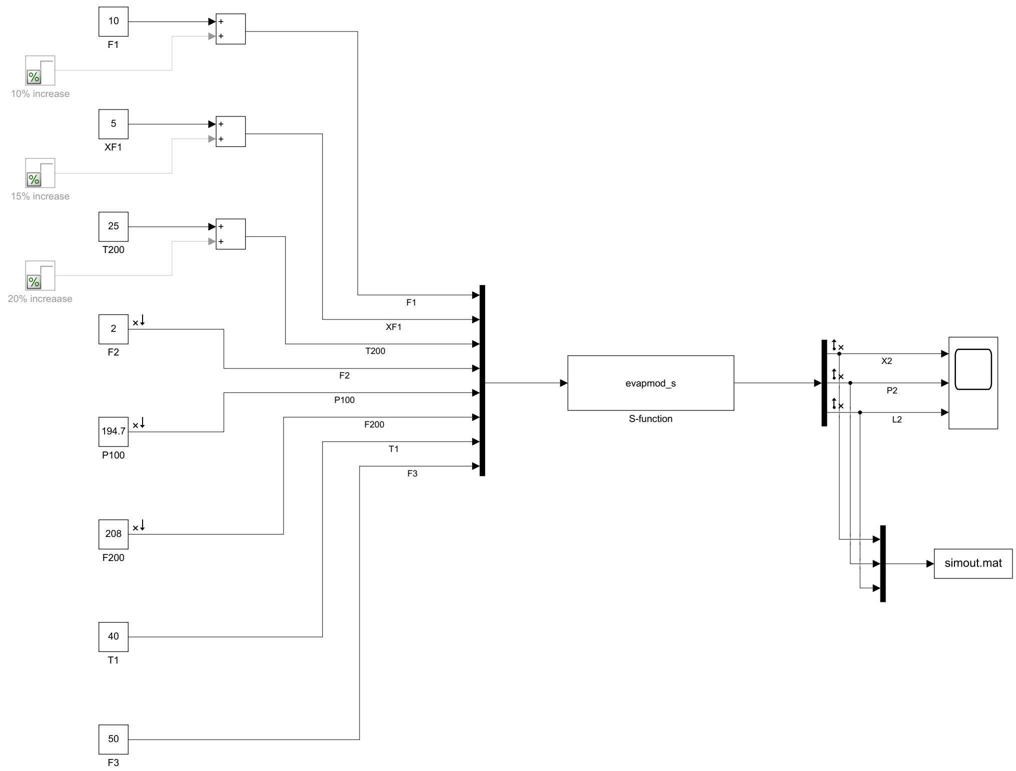

Figure 4.1: Process model in Sumilink

4.2 Numerical Linearization of the Process Model

After successful simulation of the process model in Simulink using S-Function, the next step is to linerize the process model. Since, the system is non-linear, it needs to be linearize, for simplicity and to check the stability of an entire system.

Firstly, Input and output points are assigned in process model that was created in Simulink. The numerical linearization is done based on these I/O points that was assigned earlier in process model. Now, the linearization of the model is done by using 'Model Linearizer' that is part of control systems inside Simulink. After the linearization is done, one set of data is generated which contins the linearized process model data which are further used in stability analysis.

Chapter 5

Stability Analysis

Stability of the linearized system can be defined from the values of poles and zeros. To further simplify this, after the linearization, a set of data is generated which are then transfered to a workspace, and it cointains number of matrices (i.e A, B, C, D), which again depends on the system itself.

To get the idea about a stability of the system, matrix A is important. Next the eigenvalues of matrix A is extracted, which is also represents the poles of the linearized system.

In order for a system to be stabilize, it needs to have following conditions: [4]

- No imaginary values. (Y-axis data should be zero)
- Negative real values. (X-axis data should have negative value)

Let's just understand it using pole-zero map for the linearized system. Eigenvalues of matrix A in linearized data are:

$$\begin{bmatrix} -0.1000 \\ -0.0558 \\ 0 \end{bmatrix}$$

These eigenvalues can also be seen in pole-zero map. (see figure 5.1)

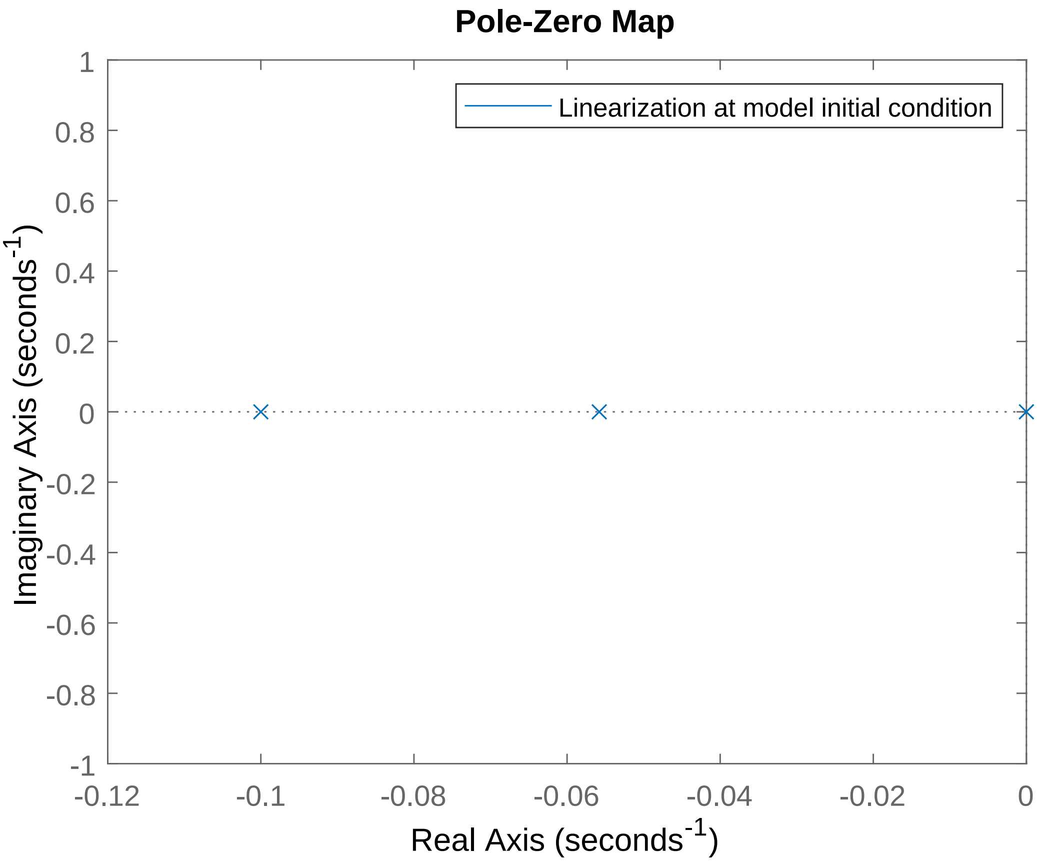

Figure 5.1: Pole-Zero Map for the linearized system

Now that the system is leariued, its time to determine the SISO pair for the control of the process.

Chapter 6

Control Structure Selection

6.1 Control Loop Interaction and General Pairing Problems

Most industrial processes are multivariable. They have multiple input variables which affect multiple output variables.

It is common practice to implement decentralized controllers, the principle is to design multiple SISO controllers. Because of process interections one SISO control loop affects the other controll loops in terms of stability and performance of the controlled system (control loop interaction).

Before starting to design decentralized controllers for MIMO process, understand the interactions among the control loops and select the manipulated-controlled variable (I/O) pairing that best suit the control problem.[4]

6.1.1 General pairing problems

In most cases, a fully coupled multivariable system of gain matrix is used to control the process. That makes things more complicated for pairing process.

For any multivariable process ($m = P$),

$$\begin{pmatrix} U_1(s) \\ \vdots \\ U_m(s) \end{pmatrix} = \begin{pmatrix} G_{C,11}(s) & \cdots & G_{C,1p}(s) \\ \vdots & \ddots & \vdots \\ G_{C,m1}(s) & \cdots & G_{C,mp}(s) \end{pmatrix} \begin{pmatrix} E_1(s) \\ \vdots \\ E_p(s) \end{pmatrix} \tag{6.1}$$

Now the problems with Input-Output pairing are that,

- Which output U_i should be paired with which input E_i to form the SISO control loop?
- It is always necessary that, $m = p$ (i.e square gain matrix).
- Select the pairings such that, the effect of interactions is minimized.

There's a solution to overcome these compexities, which is called Relative Gain Array (RGA) analysis. It is a method to predict the extent of interaction amoung the control loops when multiple SISO loops are used.

6.2 RGA Analysis

RGA is the dimensionless measure of interaction between inputs and outputs. Defination of Relative Gain Array (RGA) analysis in simple word is given by, the relative gain λ_{ij} between input j and output i,

$$\begin{aligned}\lambda_{ij} &= \frac{\text{gain between input j and output i with all other loops open}}{\text{gain between input j and output i with all other loops cloased}}\\ &= \frac{\text{gain between input j and output i with all other inputs constant}}{\text{gain between input j and output i with all other outputs constant}}\\ &= \frac{\left(\frac{\partial y_i}{\partial u_j}\right)_{U_k, k\neq j}}{\left(\frac{\partial y_i}{\partial u_j}\right)_{y_k, k\neq i}}\end{aligned} \tag{6.2}$$

The relative gain array (RGA) is a matrix that contains the individual gains as elements:

$$\Lambda = (\lambda_{ij}) \tag{6.3}$$

The gain between input j and output i with all other inputs constant is the appropriate element of the steady-state process gain:

$$\left(\frac{\partial y_i}{\partial u_j}\right)_{U_k, k\neq j} = G_{ij}(s=0) = K_{ij} \tag{6.4}$$

For any desired output vector, y, we can find the corresponding input vector $\underline{u}$ from,

$$\underline{u} = G^{-1}(0)y = K^{-1}y \tag{6.5}$$

and with,

$$\hat{G}(0) = G^{-1}(0) = K^{-1} \tag{6.6}$$

we obtain,

$$\underline{u} = \hat{G}(0)\underline{y} = \hat{K}^{-1}\underline{y} \tag{6.7}$$

The relationship between input j and output i with all other outputs constant is:

$$\left(\frac{\partial y_i}{\partial u_j}\right)_{U_k, k \neq j} = \hat{G}_{ij}(s=0) = \hat{K}_{ij} \tag{6.8}$$

The relative gain between input and output is:

$$\Lambda = G(0).*\left(G^{-1}(0)\right)^T \tag{6.9}$$

where $.*$ $\rightarrow$ elementwise matrix multiplication.

6.2.1 RGA Analysis of 3x3 MIMO System

First of all, the linearization data of manipulated variables and controlled variables are extracted in previous task-3a. Now using that data, RGA analysis is being done in series of steps as follows:

- **Step 1: Compute the linear system matrices, A, B, C, D:**

$$A = \begin{bmatrix} -0.1000 & 0 & 0 \\ -0.0209 & -0.0558 & 0 \\ -0.0042 & 0.0075 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -1.2500 & 0 & 0 \\ 0 & -0.0018 & 0.0096 \\ -0.0500 & 0 & -0.0019 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- **Step 2: Compute the gain matrix (G(0)) for the 3x3 MIMO system:**

$$G(0) = -C*(A)^{-1}*B + D$$

  Since, 3rd column of matrix A is zero, normal inverse can not be found. So, insted Moore-Penrose Pseudoinverse of Matrix A is being found.

- **Step 3: Computation of RGA based on G(0):**

$$RGA = \begin{bmatrix} 0.9998 & -0.0000 & 0.0002 \\ 0.0002 & 0.0334 & 0.9665 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{pmatrix} X2 \\ P2 \\ L2 \end{pmatrix} = \begin{pmatrix} 0.9998 & -0.0000 & 0.0002 \\ 0.0002 & 0.0334 & 0.9665 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} F2 \\ P100 \\ F200 \end{pmatrix} \tag{6.10}$$

- **Check the sum of row elements and column elements of RGA:**

$$sum\_row = \begin{bmatrix} 1.0000 \\ 1.0000 \\ 0 \end{bmatrix}$$

$$sum\_column = \begin{bmatrix} 1.0000 & 0.0334 & 0.9666 \end{bmatrix}$$

6.2.2 General Properties of RGA

Relative Gain Array (RGA) has some general properties that needs to be satisfied in order for RGA analysis to be successful.

1. The sum of elements in one row and one column equal to 1.
2. The RGA is scaling independent.
3. Change of rows or columns of gain matrix (G) corresponds to change of rows and columns of RGA.
4. If gain matrix is triangular, then RGA is identity matrix.
5. Systems with large RGA values are ill-conditioned. That means, such systems are difficult to control with decenttralized control.

6.2.3 Reducing the System of MIMO

Since, in 3x3 MIMO system, 3rd eigenvalue is 0, due to that RGA analysis is not successful. (see equation 6.10)

One must reduce this 3x3 MIMO system to 2x2 MIMO. But, for doing that controlled variable, liquid level (L2) should be controlled using level controller. Liquid level must always be controlled as it was stated in task-1 that, if too high or too low liquid level may cause some serious problem in the process. But to remove liquid level variable from the system, consiqutive manipulated varible must also be selected. For selecting a suitable manipulated variable to control the liquid level certain guidelines must be followed,[4]

- Choose the manipulation that has a direct and fast effect on a controlled variable.
- choose the coupling so that, there is a little dead time between every manipulation and the corrosponding controlled variable.
- Select the coupling so that, the interaction of the control loop is minimal.

As from the above guidelines, differential equation as well as the process diagram it is clear that, liquid level (L2) can directly, fast and efficiently be controlled by manipulation in product flowrate (F2).

So, this system of matrices must be reduced by eleminating the 3rd row and column, convert the entire MIMO system in to 2x2 MIMO system.

6.2.4 RGA Analysis of 2x2 MIMO System

First of all, the linearization data of manipulated variables and controlled variables are extracted in previous 3x3 MIMO RGA analysis. Now reducing the 3x3 matrices into 2x2 matrices and using that data, RGA analysis is being done in series of steps as follows:

- **Step 1: Compute and reduce the linear system matrices, A, B, C, D into 2x2 matrices:**

$$A = \begin{bmatrix} -0.1000 & 0 \\ -0.0209 & -0.0558 \end{bmatrix}$$

$$B = \begin{bmatrix} -1.2500 & 0 \\ 0 & -0.0018 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- **Step 2: Compute the gain matrix (G(0)) for the 3x3 MIMO system:**

$$
\begin{aligned}
G(0) &= -C * (A)^{-1} * B + D \\
&= \begin{bmatrix} -12.5000 & 0 \\ 4.6839 & -0.0328 \end{bmatrix}
\end{aligned}
$$

- **Step 3: Computation of RGA based on G(0):**

$$
RGA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
$$

$$
\begin{pmatrix} X2 \\ P2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P100 \\ F200 \end{pmatrix} \tag{6.11}
$$

- **Check the sum of row elements and column elements of RGA:**

$$
sum\_row = \begin{bmatrix} 1 \\ 1 \end{bmatrix}
$$

$$
sum\_column = \begin{bmatrix} 1 & 1 \end{bmatrix}
$$

Now that the RGA fulfills the all properties, one must go further to find the Niederlinski Index (NI).

- **Step 4: Computation of NI based G(0):**

$$
\begin{aligned}
NI &= \frac{|G(0)|}{\prod diag(G(0))} \\
&= 1
\end{aligned}
$$

6.2.5 Interpreting the RGA Elements

1. $\lambda_{ij} = 1$
   - Perfect decoupling exists.
   - Ideal pairing.
2. $\lambda_{ij} = 0$
   - No effect of input on output.
   - Do not pair.

3. $0 < \lambda_{ij} < 1$
   - Interaction exists.
   - $\lambda_{ij} < 0.5$: Indirect coupling is dominant.
   - $\lambda_{ij} > 0.5$: Direct coupling is dominant.
   - Avoid pairing for $\lambda_{ij} \leq 0.5$.
4. $\lambda_{ij} > 1$
   - Interaction exists.
   - Difficult to control.
   - Do not pair whenever λ_{ij} have high value.
5. $\lambda_{ij} < 0$
   - Open loop and close loop gain have opposite signs.
   - Critical situation.
   - Never pair whenever λ_{ij} have negative value.

Use of RGA to Deteremine Loop Pairings

Rule 1: Prefer the pairing with RGA elements close to 1.

- It is desirable that, $\lambda_{ij} = 1$, but not necessary for all the elements.
- It is required that, $\lambda_{ij} \approx 1$.

Rule 2: Avoid pairings corresponding to $\lambda_{ij} < 0$.

- The sighn of the steady-state gain between input and output may change.
- This will yield instability.

Rule 3: Any loop pairings with negative Niederlinski Index ($NI < 0$) is not acceptable.

- In order to avoid unstable closed loop system with reasonable RGA pairings, the Niederlinski Index is used additionally.

Part III

Controller Design and Implementation

Chapter 7

Controller Design

After performing RGA analysis on the linearization data which is generated in Simulink with the help of model linearizer, SISO loop pairings are decided for the controller designing.

1. Liquid level (L2) must be controlled by manipulation in the product flowrate (F2) by using simple level controller (P-controller).
2. Product composition (X2) must be controlled by manipulation in the steam pressure (P100). (see equation 6.11)
3. Operating pressure (P2) must be controlled by manipulation in the cooling water flowrate (F200) inside the condensor. (see equation 6.11)

7.1 Decentralized Control

Decentralized control tries to control multivariable processes by means of SISO control loops.

Advantages: easy implementation and tuning, each design is reduced to a SISO problem.

Disadvantage: works only well if no strong couplings occur in the plant. Snce, as seen in previous task that, strong coupling exist between manipulated and controlled variables.[4]

Decentralized control design procedure in two steps:

1. **First step:** the choice of pairings.

- Select good parings of inputs and outputs, such that the effect of interactions (couplings) is minimized.
- As it was already done in previous task, the choice of pairings are as follows:
  (a) F2-L2 I/O pair
  (b) P100-X2 I/O pair
  (c) F200-P2 I/O pair

2. **Second step:** the design (tuning) of each controller based on pairing loops. There are three approaches available to follow:

   (a) **Fully coordinated design** of decentralized control systems: All SISO controllers are designed simultaneously based on the complete model.
     - It requires the use of an optimization method.
     - It is not not commonly used in practice.
       - Very challenging design approach.
       - Detailed dynamic model is required.
     - It can be used when centralized control is difficult for geographical reasons.

   (b) **Independent design** of decentralized control systems:
     - Each SISO controller is designed based only on the corresponding process diagonal element.
     - Because of process interaction, the MIMO model may give,
       - Poor performance
       - Instability
     - Results may be improved by 'detuning' some of the SISO controllers.
     - It can be used when the integrity is desired, when the loops can operate independentaly without instability.

   (c) **Sequential design** of decentralized control systems:
     - Controllers are designed sequentially with the previously designed controller implemented.
     - Advantages:
       - It can be used for interactive problems where the independent approach does not work.
       - Closed loop stability.

- Disadvantages:
  - The order in which the loops are closed has to be determined.
  - Stability of individual loops cannot be guaranteed when inner loops fail (integrity).

Since, this evaporator system has interections with other loops. (i.e. change in one variable may cause an effect on more than one outputs) So, for easy and effective controller tuning **sequential design approach is selected in this case**.

7.2 Feedback Control Design

Basic control structure:

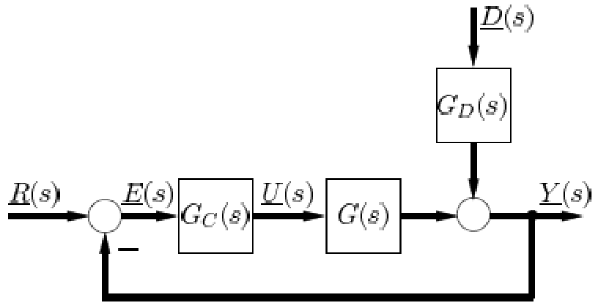

Figure 7.1: Basic control structure

Control law: $u(t) = G_C\big(e(t)\big)$
The most common function form of G_c is the three-mode controller, in other words proportional-integral-derivative control law (PID controller).

There are three form of PID control.

1. Parallel form:

$$G_C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D \cdot s$$

2. Ideal (Time constant) form:

$$G_C(s) = \frac{U(s)}{E(s)} = K_P\Big(1 + \frac{1}{T_I \cdot s} + T_D \cdot s\Big)$$

where, $T_I = \dfrac{K_P}{K_I}$ and $T_D = \dfrac{K_D}{K_P}$

3. Series form:

$$G_C(s) = \frac{U(s)}{E(s)} = K'_P\Big(1 + \frac{1}{T'_I \cdot s}\Big)\big(1 + T'_D \cdot s\big)$$

In this perticular case of controller tuning of an evaporator system only **parallel form is used**.

7.2.1 Effects of P, I, and D actions

1. **Proportional action:**
   - Proportional to the control error.
   - Based on the current value os the control error.
   - Advantage: produces a small manipulated variable whrn the control error is small. Hence, excessive control efforts are avoided.
   - Disadvantage: there might be some steady-state error.
2. **Integral action:**
   - Proportional to the integral of an error.
   - Related to the past values of the control error.
   - Advantage: the steady-state error is always zero.
   - Disadvantage: the integral term 'winds up' when the actuator/process saturates. (i.e. more overshoot)
3. **Derivative action:**
   - Proportional to the derivative of the rate of change of the error.
   - Related to the predicted future values of the control error.
   - Advantage: large amount of error is avoided. Hence, it has the potential in improving the control performance.
   - Disadvantage: sensitivity to measurement noise.

Combinations that can be used for controller tunings are P, I, PI, PD, PID.

Many control problems can be solved using the PI-controller. However, in this process **P-controller is used as level controller** and **PI and PID controllers are used for other two SISO pairing loops** for the better performance.

7.2.2 Closed loop performance

The desired closed loop performance are expressed as some constraints on the time response of the closed loop system. The performance specifications can be analyzed in terms of three important properties of the system dynamics:

- Stability
- Response accuracy : little or zero-tracking error at steady state
- Transient (dynamic) performance (i.e. speed of response):
  - Minimum rise time
  - Minimum settling time
  - Specified maximum overshoot
  - For oscillatory response, a specified maximum decay ratio, to ensure sufficiently fast settling of oscillations.

7.2.3 Controller Tuning Process

Choose the controller type and parameters so that some objective criteria of the closed loop performance are satisfied.

Open-Loop tuning method: Ziegler-Nichols Open-Loop method

- Assumption: The process step rresponse is a 's-shape' curve.

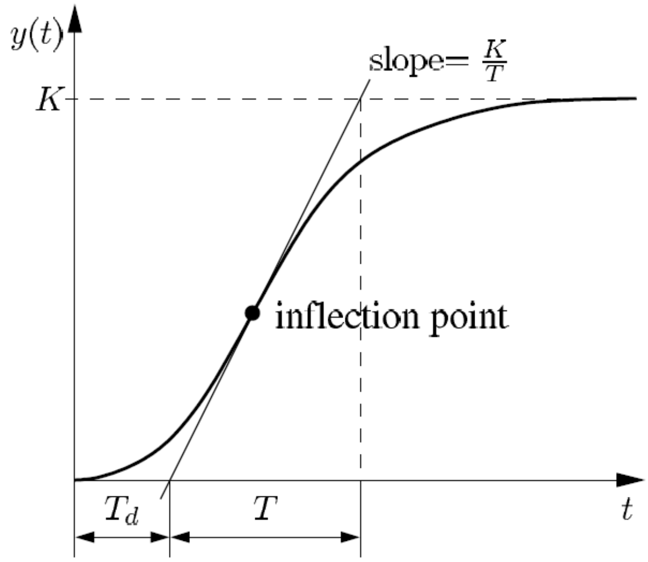

Figure 7.2: Unit step response

- typical response curve of many systems, can be approximated by the step response of the first order system with time delay:

$$G(s) = \frac{K}{Ts+1}e^{-sT_d}$$

- The constants K (gain), T (time constant) and Td (dead time) can be determined from the unit step response. (see figure 7.2)

– Alternatively if the atep response has no s-shape curve, then another approach should be followed which is called two-point method. (see figure 7.3)

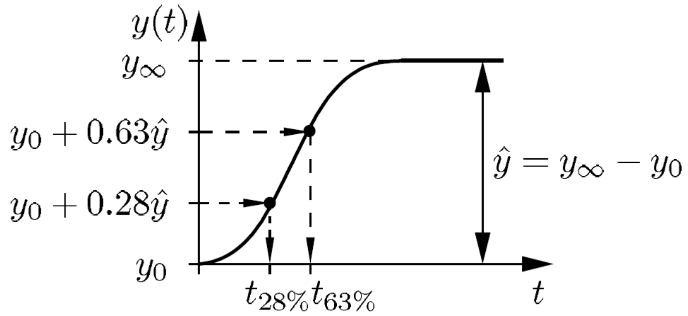

Figure 7.3: Unit step response with two point method

- In this method two values of time is measured to calculate T and T_d.

$$T = \frac{3}{2}(t_{63\%} - t_{28\%})$$

$$T_d = |t_{63\%} - T|$$

$$K = \frac{\hat{y}}{\hat{u}}$$

- Based on the value of K, T and T_d find the proportinal gain, integral time, and derivative time from open-loop tuning rules and implement those values in the controller block.

- Disadvantage of open-loop test is that process variables may drift away from the normal oprating point.

Close-Loop tuning method: Relay Feedback tuning method

- It is an automated tuning procedure that can be implemented on an industrial process control system.

- effective method for determing ultimate gain (K_u) and ultimate period (P_u).

- An on-off(ideal) relay is placed in the feedback loop.

- Control law: $u(t) = \begin{cases} h & \text{for} \quad e(t) > 0 \\ -h & \text{for} \quad e(t) < 0 \end{cases}$
  where, h = relay height

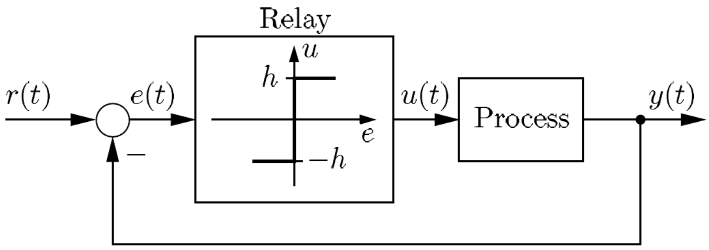

Figure 7.4: Relay feedback system

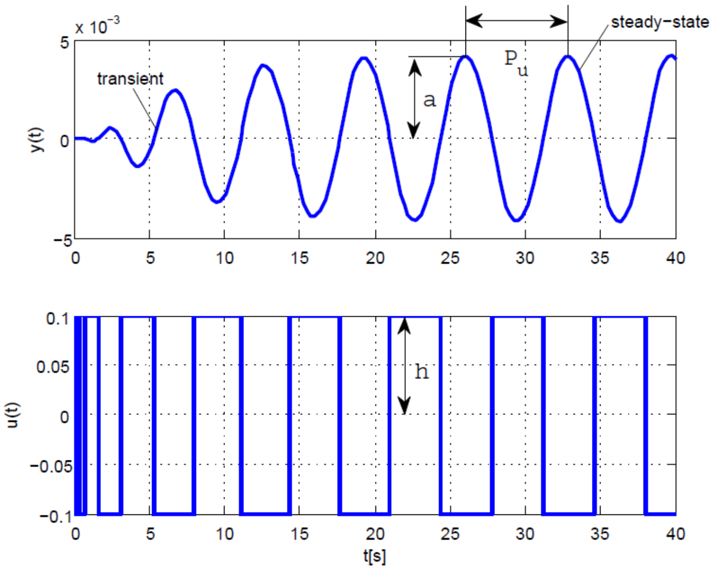

Figure 7.5: Relay feedback response plot

where, $a \rightarrow$ amplitude of oscillation
$P_u \rightarrow$ ultimate period
$K_u = \dfrac{4h}{\pi a} \rightarrow$ ultimate gain

- Design procedure:
  - Keep the feedback control system is at steady-state.
  - Switching procedure (increase and decrease u(t) by h) until sustained oscillation is observed, magnitude h depends on allowable deviations in the process output y(t), typical values are 3% ∼ 10% of y0.
  - Read of P_u and a from the steady-state oscillation and compute K_u.
  - Find the K_p, K_I and K_D based on the Ziegler-Nichols closed loop tuning rules using the values of K_u and P_u

7.2.4 PID controller implementation issues

1. Controller BIAS

- For a real processes, the controller needs an initialization constant or a bias signal u0.

2. Direcct or reverse action

– Direct and reverse action are related to the sign relationship of the input to the output of a system and refer to the sign of the system gain.

3. Implementation of Integral action (anti-windup)

– All actuators have limitations. For example, a valve cannot be more than fully open or fully closed, a motor has a limited speed. It may happen, that the manipulated variable reaches the actuator limits. (i.e. The actuator saturates) To avoid that, a back calculation method is used.

4. Implementation of Derivative action

– There might always be a measurement noise in reality, and differentiator may cause some error due to that, to avoid that, a low padd filter can be added to the pure differentiator.

5. Proportional action (set-point weighting)

– A step change in the reference signal will cause a step change in the error and that will generate a proportional kick and that can damage the process. To avoid that, set-point weightage can be added to the proportional gain carefully.

6. Digital controller

– In the past, implementations of PID controllers were analog, but now most are digital and directly implemented in computers. So, continuous control laws have to be discretized for practical applications.

Chapter 8

Implementation and Comparison of Designed Controllers

As stated in the previous part-I that, first level in the separator needs to be controlled as it may cause some serious damage to the whole system.

Firstly, level controller (simple proportional controller) is designed and then after two PID controllers for P100-X2 and F200-P2 I/O pair needs to be design using two methods (i.e. open and close loop tuning method)

8.1 Designing of a level controller (F2-L2 pair)

As seen earlier, this F2-L2 process loop is reverse-acting, means the gain has to be negative. ($K < 0$)

So, first the F2-L2 loop is being closed using normal P-controller and started changing the value of gain unitill the stable response is achieved for the value of proportional gain of -16.

Now that level controllr is set, other two controllers needs to be implemented using two different method.

8.2 Open-Loop tuning method

8.2.1 Product composition controller (P100-X2)

To determine the ultimate gain, time constant, and dead time a step-response need to be found.

To find the step-response, a transfer function for SISO process is needed and to find that, this I/O pair is being linearized and then from that , transfer function is being determind and ultimately step-response is found. (see figure 8.1)

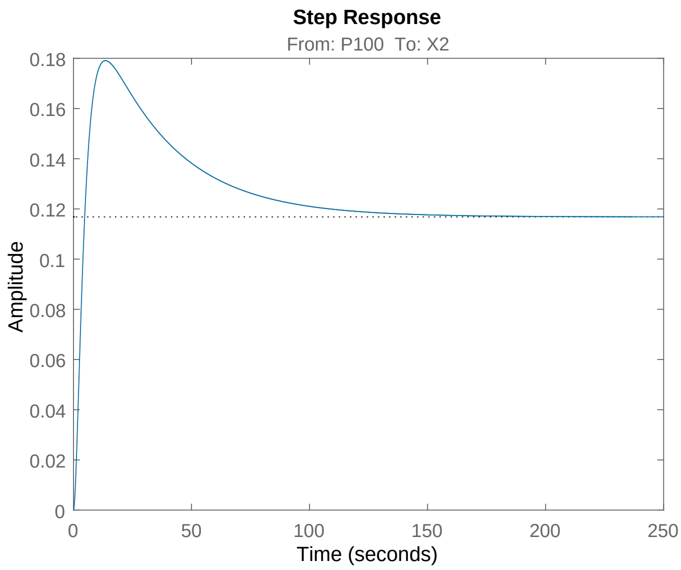

Figure 8.1: Step-response for P100-X2 I/O loop

Since, the step-response has no s-shape, alternate two-point method is being used to find the ultimate gain, time constant, and dead time.

From the value of ultimate gain, time constant, and dead time the values of K_P, K_I, and K_D was calculated based on open-loop tuning rules which are listed in table 8.1.

K	0.1170
T	2.0700
T_d	0.9800
K_P	21.6641
K_I	11.0531
K_D	10.6154

Table 8.1: Open loop tuning parameter for P100-X2 loop

8.2.2 Operating pressure controller (F200-P2)

Since, sequential approach causes system to malfunction, here independent approach is being used for convenience.

So, by keeping only the level control loop closed along with the implmented controller and by breaking the P100-X2 loop, F200-P2 loop is being linearized in order to find the transfer function. which is being used to find the step-response. (see figure 8.2)

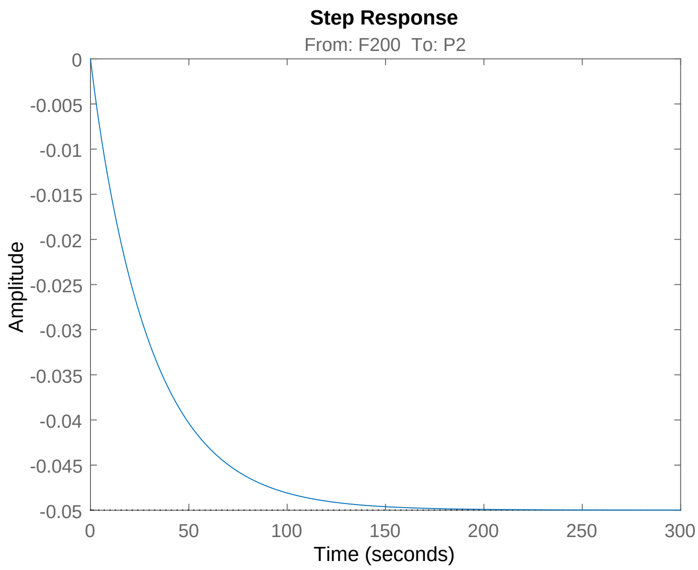

Figure 8.2: Step-response for F200-P2 I/O loop

From the step-response again ultimate gain, time constant, and dead time was determined. Now, using the Ziegler-Nichols open loop tuning rules, K_P,

K_I, and K_D was calculated (see table 8.2)and implemented in the PID block and simulate the system, which ultimately gives the stable ouput.

K	-0.1000
T	54.9000
T_d	0.2000
K_P	-1.195
K_I	-0.971
K_D	-0.367

Table 8.2: Open loop tuning parameter for F200-P2 loop

Now, the full simultion model for open-loop tuning method is shown in figure 8.3.

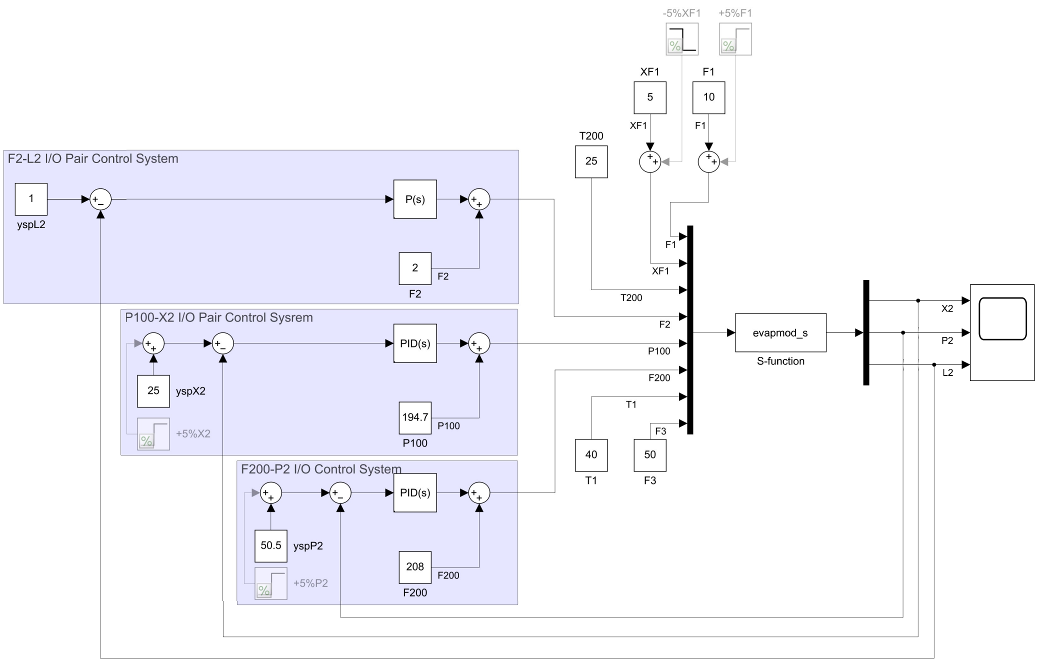

Figure 8.3: Simulation model with open loop tuning method

8.3 Close-Loop tuning method

In the close loop relay feedback method one relay block is added and the relay feedback tuning method procedure is being followed as stated in chapter 3.

8.3.1 Product composition controller (P100-X2)

By keeping the level control loop is closed with one relay block insted of PID-controller and set value of h as +10 to -10 (because of positive error) which results in oscillatory output, from that output amplitude of oscillation and ultimate period was determined and ultimate period was calculated.

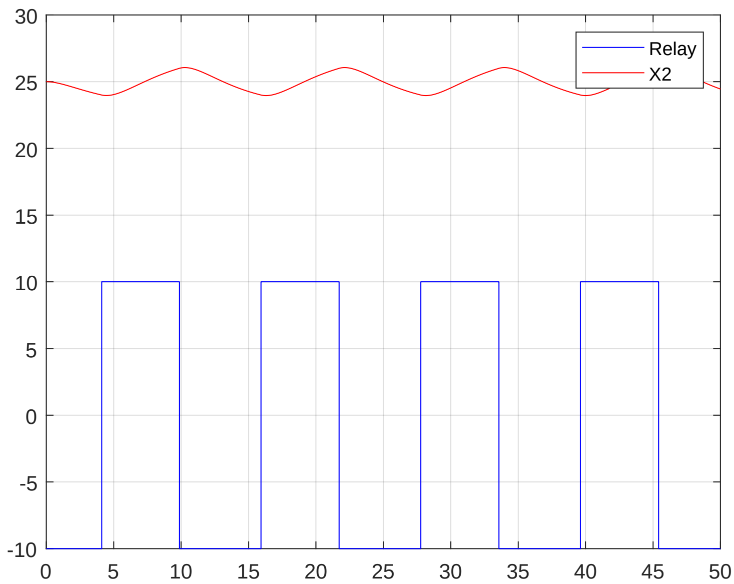

Figure 8.4: Relay response of P100-X2 loop

h	10
a	1
P_u	11.95
K_u	6.366
K_P	7.6394
K_I	1.2786
K_D	11.4114

Table 8.3: Close loop tuning parameter for P100-X2 loop

From the determined value of Ku and Pu, tuning parameters are being found using Zieglar-Nichols close loop tuning rules. and then implement those gains in PID block, which results the stable response.

8.3.2 Operating pressure controller (F200-P2)

Since, sequential design approach is being followed, other two loops are kept as it is and now simillar procedure is being followed for F200-X2 loop.

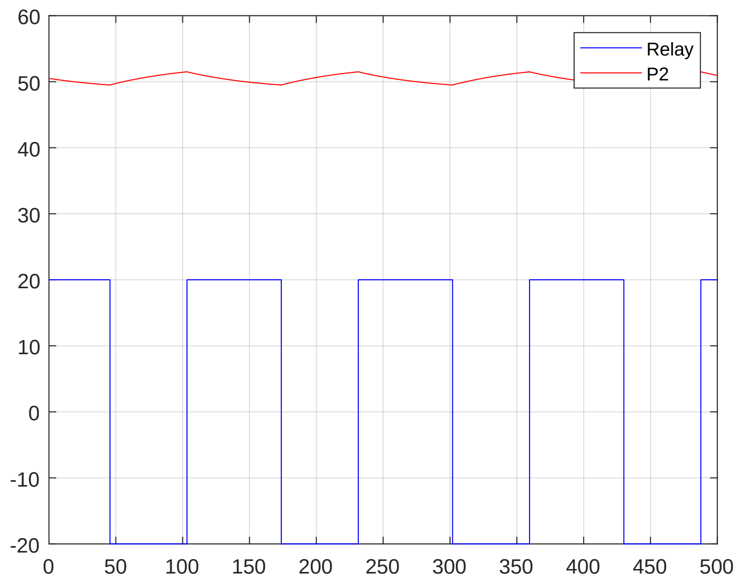

Figure 8.5: Relay response of F200-P2 loop

h	-20
a	1
P_u	128.23
K_u	-25.4648
K_P	-15.2789
K_I	-0.2383
K_D	-244.9313

Table 8.4: Close loop tuning parameter for F200-P2 loop

From the determined value of Ku and Pu, tuning parameters are being found using Zieglar-Nichols close loop tuning rules. and then implement those gains in PID block, which results the stable response.

Now, the full simultion model for close-loop tuning method is shown in figure 8.6.

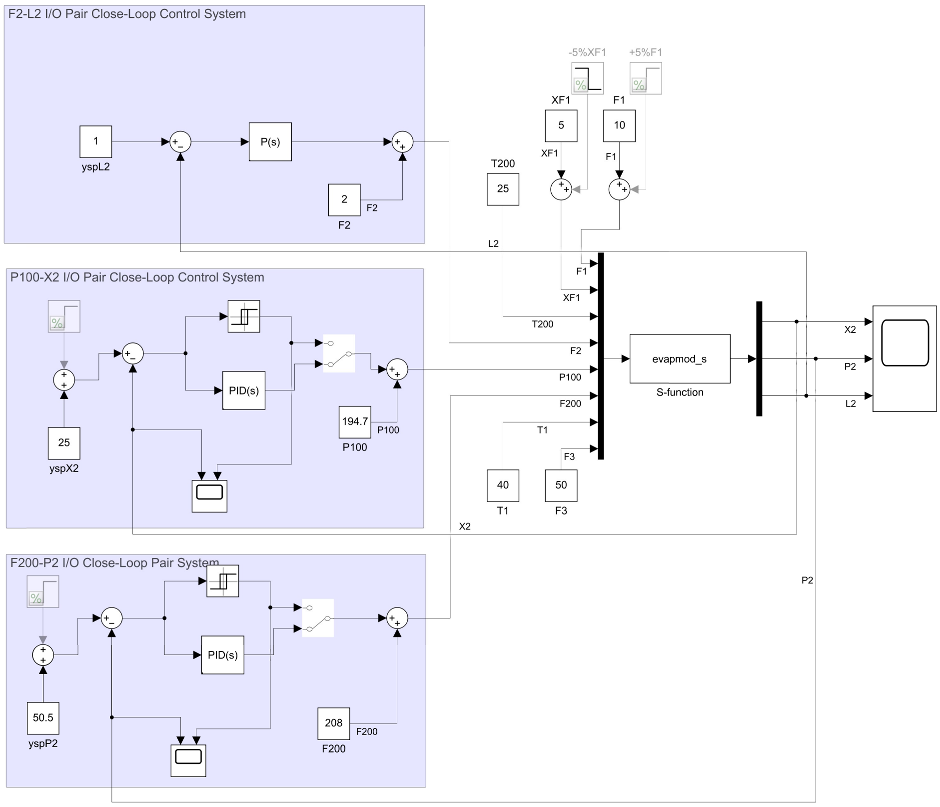

Figure 8.6: Simulation model with close loop tuning method

8.4 Performance of Designed Controllers

When a controller is designed for any system, it is always necessary to check the performance of the controller by giving some disturbance in the system and see how effective a controler is in stabilizing the system again. For well tuned controller it is nearly impossible to check the performance of a controller just by looking at the output plots.

So, there are some other methods available to check the performance of a controller which are listeed below.

1. **IAE(Integral of Absolute Error)**

– In this method absolute of all the errors (because some errors can be positive while others can be negative) are integrated and displayed as a single value, which represents the performance of a controller.

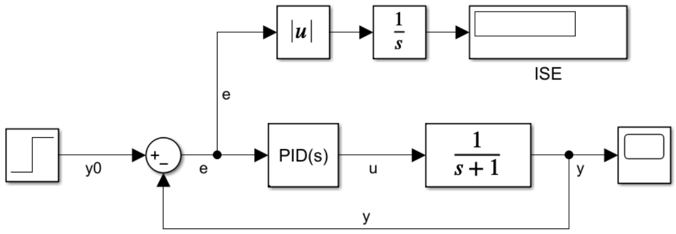

Figure 8.7: IAE(Integral of Absolute Error)

2. **ISE(Integral of Square of Error)**

– In this method square of all the errors (because some errors can be very small, by doing the square it signifies the error) are integrated and displayed as a single value, which represents the performance of a controller.

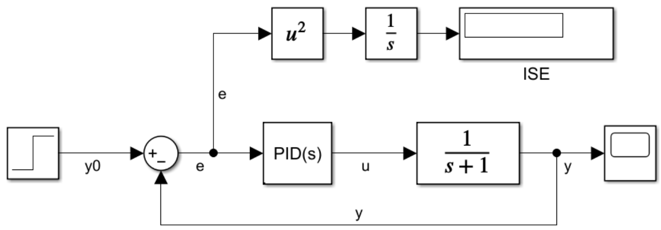

Figure 8.8: ISE(Integral of Square of Error)

8.4.1 Disturbance rejection scenarios

1. +10% step change in feed flowrate (F1) to be applied at t = 10.

Type of tuning method	IAE	ISE
Open Loop Tuning	14.54	1.006
Close Loop Tuning	807	1841

Table 8.5: IAE and ISE for disturbance scenario-1

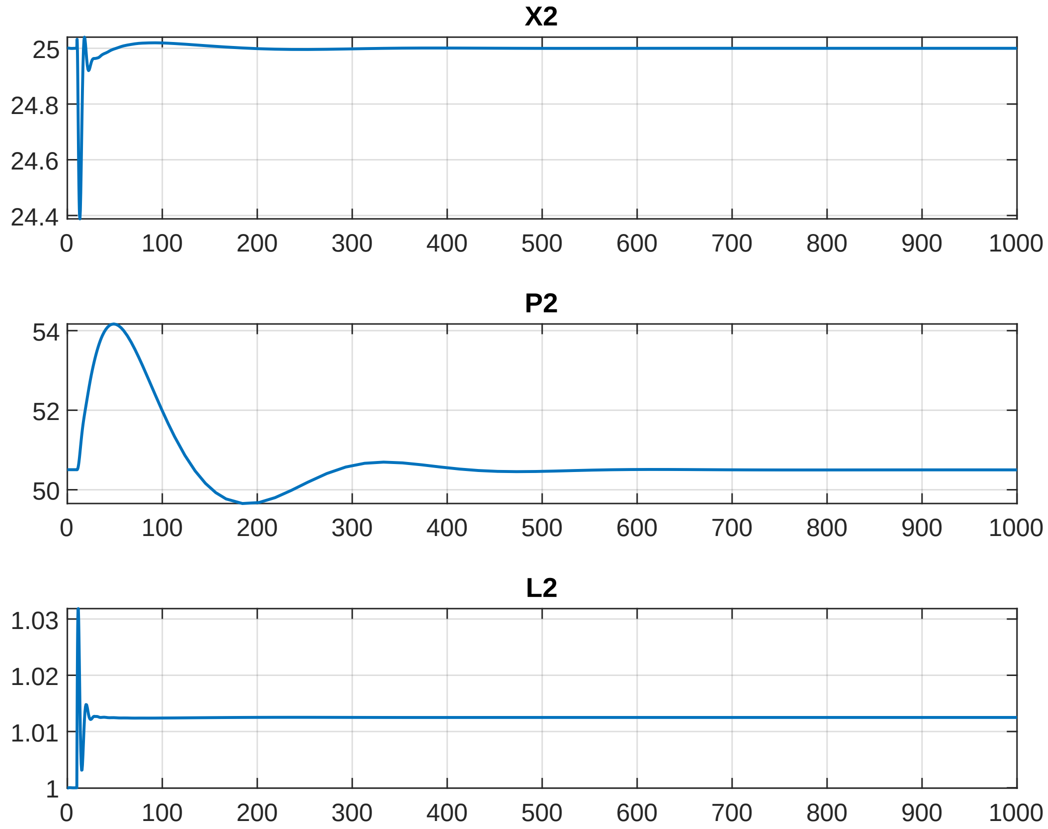

Figure 8.9: Disturbance rejection scenarios-1 for open loop tuning method

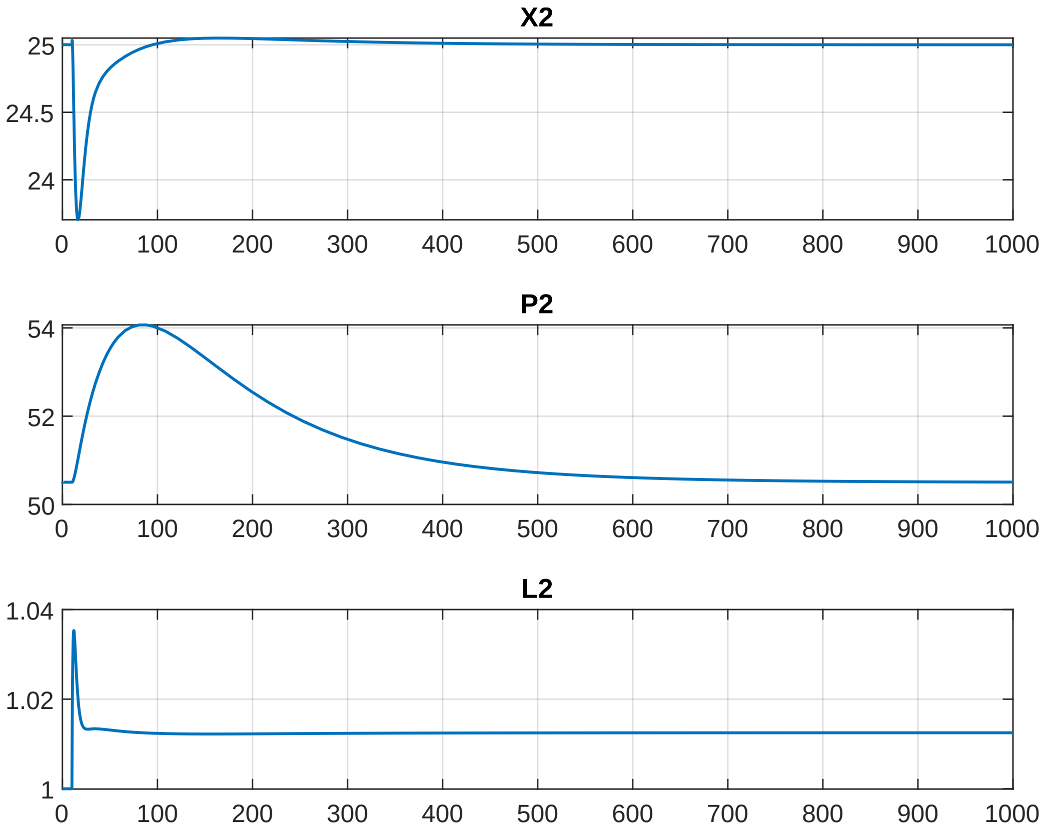

Figure 8.10: Disturbance rejection scenarios-1 for close loop tuning method

2. +5% step change in feed flowrate (F1) to be applied at t = 10 and −5% step change in feed composition (XF1) to be applied at t = 300 in the same simulation run.

Type of tuning method	IAE	ISE
Open Loop Tuning	3.55	0.2619
Close Loop Tuning	394	379

Table 8.6: IAE and ISE for disturbance scenario-2

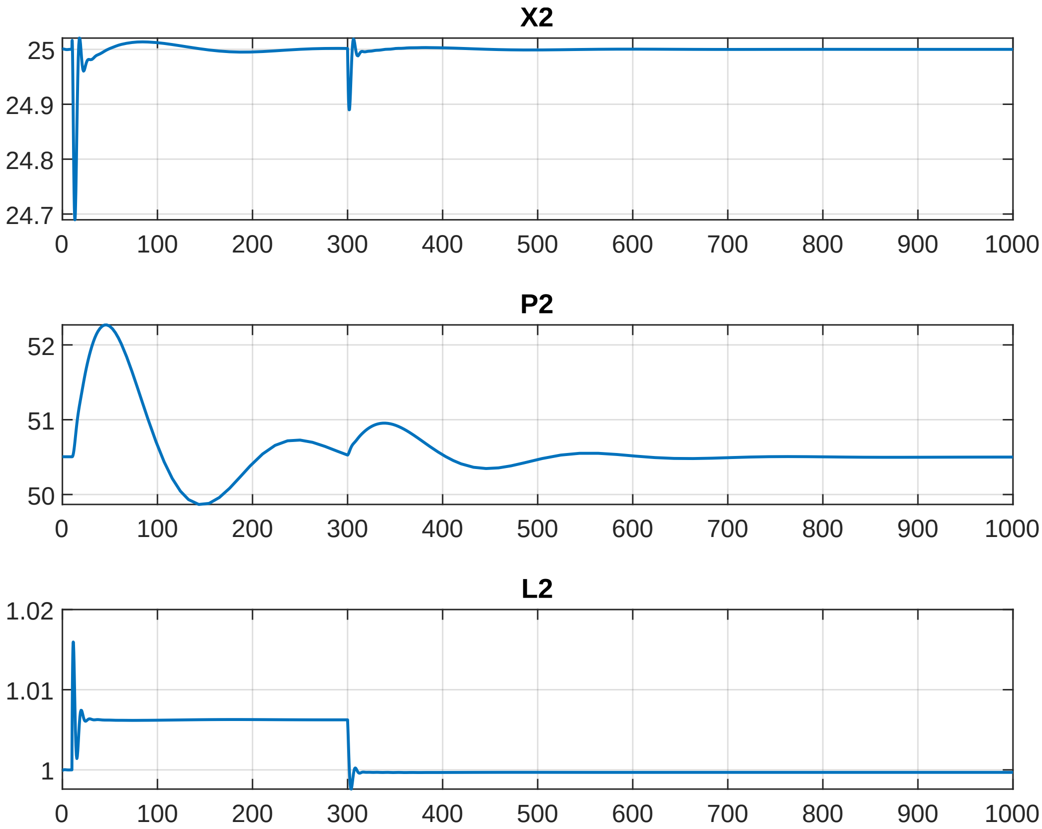

Figure 8.11: Disturbance rejection scenarios-2 for open loop tuning method

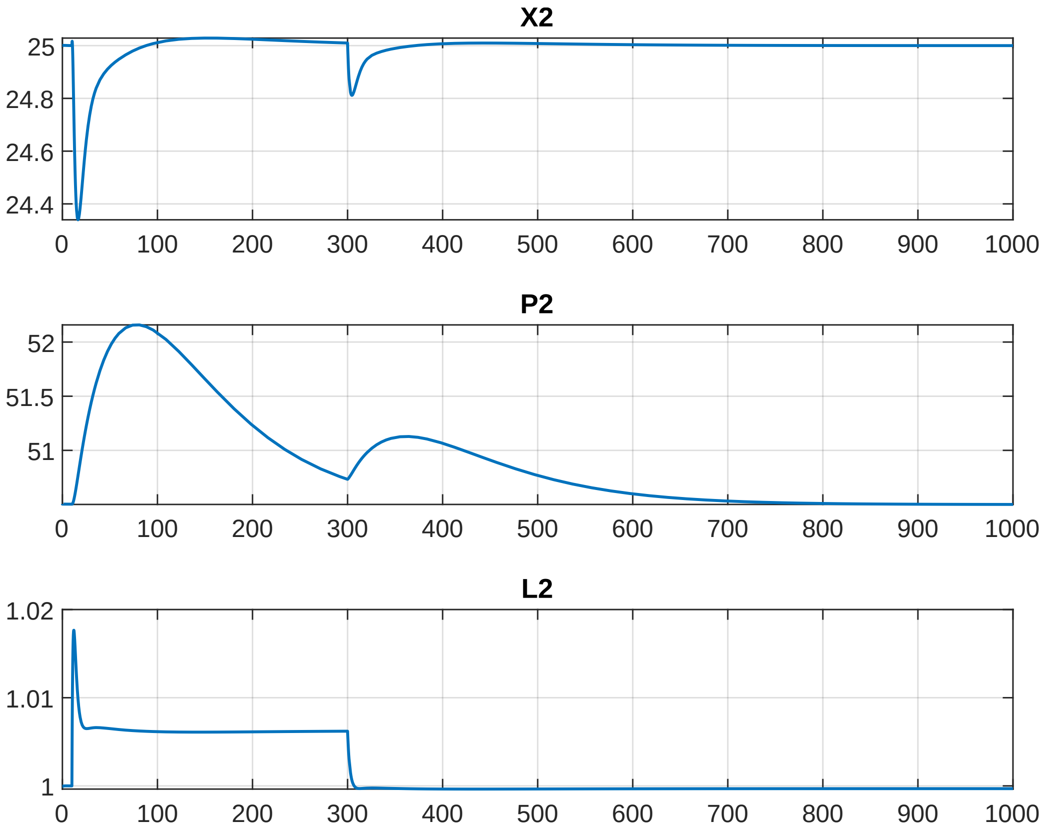

Figure 8.12: Disturbance rejection scenarios-2 for close loop tuning method

8.4.2 Set-point tracking scenarios

1. +10% step change in set-point X2 to be applied at t = 10.

Type of tuning method	IAE	ISE
Open Loop Tuning	1.77e+04	3.256e+05
Close Loop Tuning	119.3	61.65

Table 8.7: IAE and ISE for disturbance scenario-3

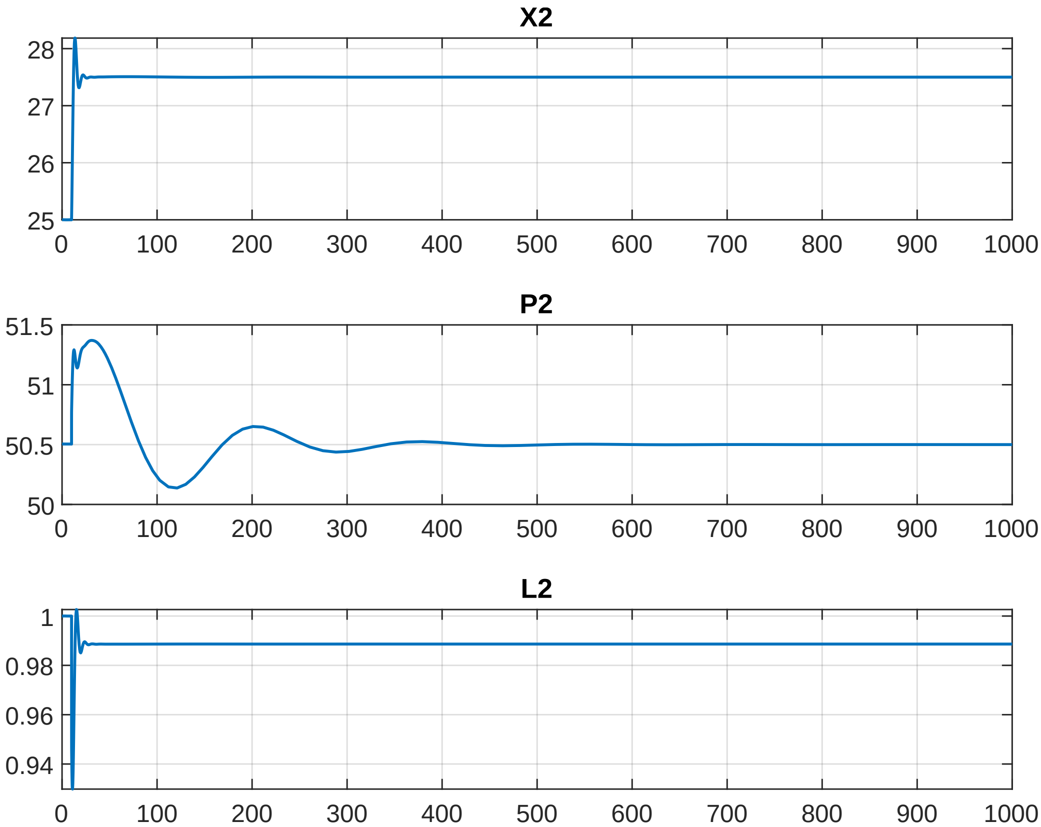

Figure 8.13: Disturbance rejection scenarios-3 for open loop tuning method

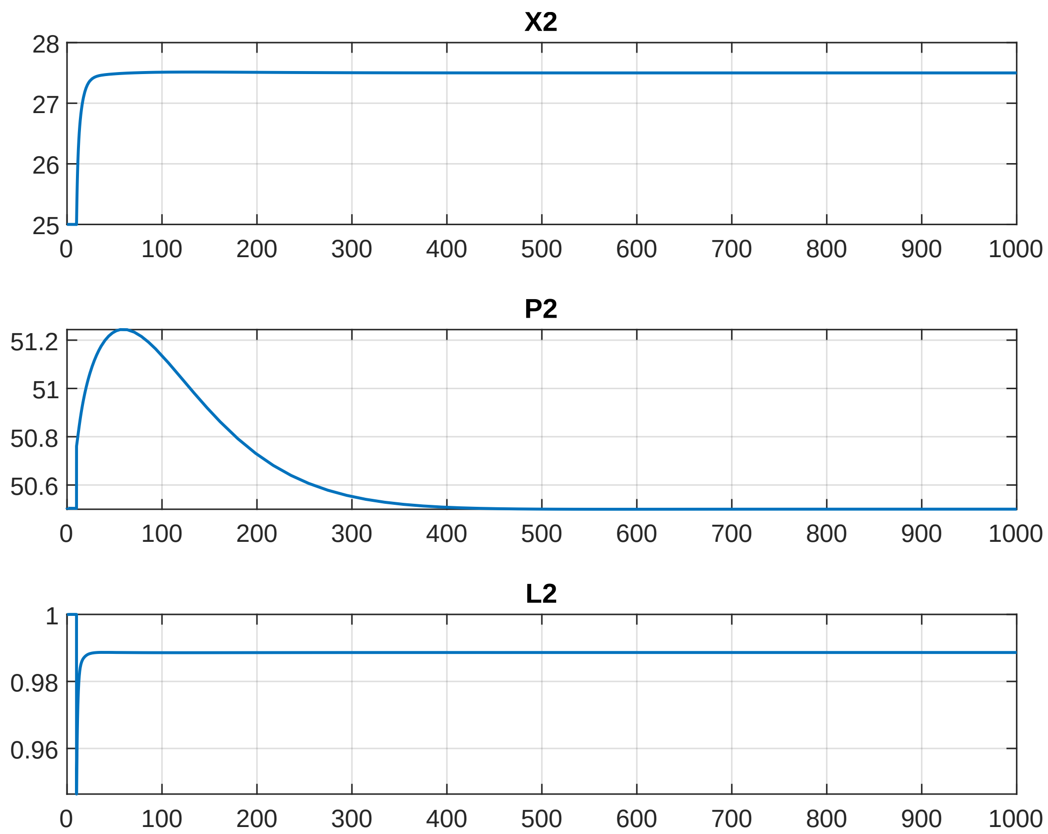

Figure 8.14: Disturbance rejection scenarios-3 for close loop tuning method

2. +5% step change in set-point X2 to be applied at t = 10 and +5% step change in set-point P2 to be applied at t = 300 in the same simulation run.

Type of tuning method	IAE	ISE
Open Loop Tuning	2.036e+04	4.338e+05
Close Loop Tuning	1.731e+04	4.315e+05

Table 8.8: IAE and ISE for disturbance scenario-4

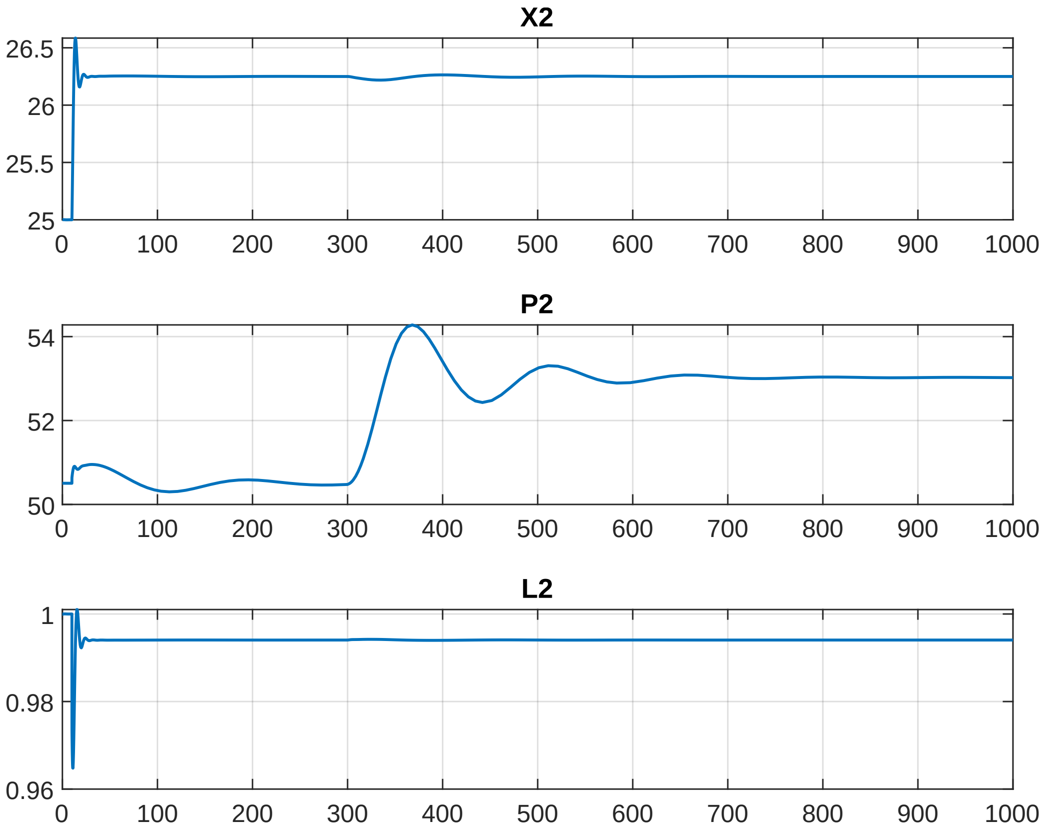

Figure 8.15: Disturbance rejection scenarios-4 for open loop tuning method

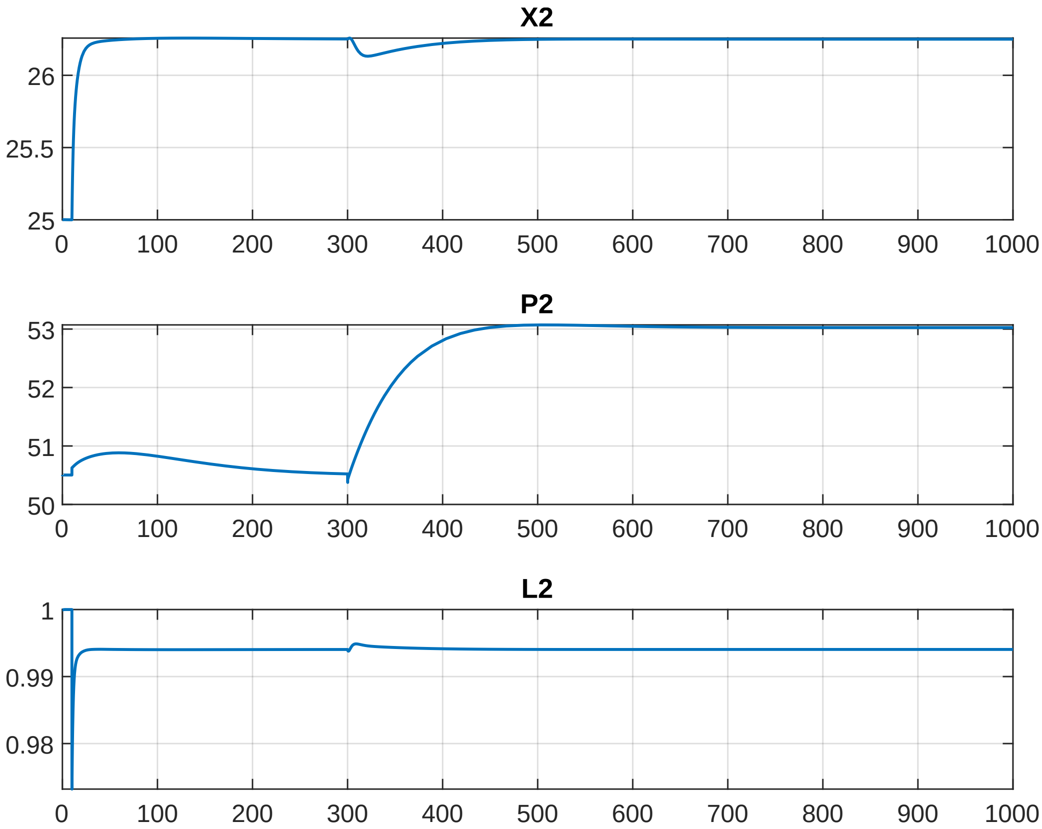

Figure 8.16: Disturbance rejection scenarios-4 for close loop tuning method

Chapter 9

Discussion and conclusions

After perfoming this extensive work certain conclusion need to be made based on the obtained result which are listed below:

- Firstly, designed level controller works perfectly in almost every scenario.
- Product comosition controller works best in both cases, but in case of close loop it takes more time to reach the set-point. but at the end it reaches the set-point.
- Pressure controller needs some further work, because it takes more time to reach the set-point as well as it malfunction when the step change was given in the set point. It needs to be diagnosed and make suitable changes.
- All in all this long process takes time but eventually gives the better results that can be noticed by looking the IAE and ISE values of the disturbance scenarios.
- In further work, this controller performence will be incresed by means of necessary changes.

Bibliography

[1] Newell-lee evaporator system for system identification,. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/68641-newell-lee-evaporator-system-for-system-identification

[2] Robust multivariable control system. [Online]. Available: http://courses.ece.ubc.ca/530/project.pdf

[3] P. Lee, R. Newell, and I. Cameron, *Process Control and Management*. Springer, 1998. [Online]. Available: https://books.google.de/books?id=CM5TAAAAMAAJ

[4] I. Disli-Kienle, "Process control-lecture notes," *Control Loop Interaction*, vol. 2021, no. Lecture 7, pp. 15–26, May 2021.

[5] What is s-function? [Online]. Available: https://www.mathworks.com/help/simulink/sfg/what-is-an-s-function.html#:~:text=You%20can%20use%20S-functions%20for%20a%20variety%20of,the%20inverted%20pendulum%20example%2C%20Inverted%20Pendulum%20with%20Animation%29