Order N°……..../FHC/UMBB/2024

People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M’Hamed BOUGARA-Boumerdes

Faculty of Hydrocarbons and Chemistry

Final Year Thesis in Order to Obtain the Degree of

MASTER

In Industrial Processes Automation

Presented by

MAZOUZI Abderrahmane

Branche: Hydrocarbons
Major: Industrial Processes Automation: Automatic Control

Theme

Multivariable control of Shell Heavy Oil Fractionator Column

Jury Members:

KESRAOUI	Mohamed	PR	UMBB	President
MANDIL	Chafiaa	MAB	UMBB	Examinator
KHELASSI	Abdelmadjid	MCA	UMBB	Supervisor

Academic Year: 2023/2024

People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M’Hamed BOUGARA-Boumerdes

Faculty of Hydrocarbons and Chemistry

Department: Automation of Industrial Processes and Electrification
Branche: Hydrocarbons
Major: Industrial Processes Automation: Automatic Control

Final Year Thesis in Order to Obtain the Degree of

MASTER

In Industrial Processes Automation

Theme

Multivariable control of Shell Heavy Oil Fractionator Column

Presented by:
MAZOUZI Abderrahmane

Supervisor Favorable Opinion
Full Name: KHELASSI Abdelmadjid
Signature:

Juries President Favorable Opinion
Full Name: KESRAOUI Mohamed
Signature:

Stamp and Signature

ACKNOWLEDGEMENT

First and foremost, praises, thanks, and gratefulness are to Allah for granting me the strength, will, and courage to accomplish this humble work.

I would like to express my deep and sincere gratitude to my project Supervisor Dr. KHELASSI Abdelmadjid, it was a great privilege and honor to work under his supervision.

I would also like to extend my sincere gratitude to the jury members for accepting to evaluate this work and for their valuable feedback and critiques, as well as to the Department of Industrial Automation and Electrification staff and teachers.

Finally, my deep thanks to all my friends who have supported me to complete this work.

Thank you all.

DEDICATION

I dedicate this work to my Parents

To family, friends, and classmates

TABLE OF CONTENT

LIST OF FIGURES

Abstract

A Shell Heavy Oil Fractionator (SHOF) is a type of distillation column widely used in oil refineries. It is designed for the separation of heavy crude oil into various components based on differences in their boiling points. This project aims to control a Shell Heavy Oil Fractionator (SHOF), which is characterized by high nonlinear dynamic behavior and strong loop interactions. The Relative Gain Array (RGA) was used to determine the best loop pairing configuration for the system, and the Dynamic Relative Magnitude Array (DRMA) was used to determine the level of interaction. A multivariable control system is designed to control the fractionator using a Proportional-Integral (PI) controller, which is tuned using the Biggest Log Modulus Tuning (BLT) method.

General introduction

General introduction

The control of multivariable systems has become very important in automatic control because most chemical processes, especially refining and petrochemical processes, are multivariable. These systems, characterized by multiple inputs and outputs, require sophisticated control strategies to manage the interactions between variables and maintain optimal performance. The Shell Heavy Oil Fractionator (SHOF) is a prime example of such a system.

The SHOF is crucial for separating heavy oil fractions, contributing significantly to product quality and operational efficiency. However, the interactions within its multivariable framework present substantial challenges for the control. These interactions can lead to performance degradation, increased energy consumption, and higher operational costs if not properly managed.

The primary objective of this work is to explore and address the multivariable control challenges associated with the SHOF. By analyzing the interactions within the system and applying advanced control strategies, we aim to enhance the stability, efficiency, and overall performance of the fractionator.

This work is divided into four chapters:

In the first chapter, we provide a comprehensive study of the Shell Heavy Oil Fractionator (SHOF). This includes an introduction to the fundamentals of the separation process and a detailed presentation of the mathematical model represented as a transfer function matrix.

In the second chapter, we offer an overview of multivariable systems and multivariable control. We explored the interaction phenomenon, discussing various methods for analyzing these interactions and their impact on control strategies.

In the third chapter, we present the Proportional-Integral-Derivative (PID) controller, exploring its actions. We then discuss both single-variable and multivariable control methods. For single-variable control, we examine the Ziegler-Nichols and Evans methods. For multivariable control, we explore the Biggest Log Module Tuning method (BLT).

In the fourth chapter, we explored into the control process of the Shell Heavy Oil Fractionator (SHOF). We begin by selecting the best loop pairing configuration, and then we analyze the system interactions in the closed-loop. Finally, we apply a multivariable control method the BLT to the process. The simulations are conducted in MATLAB, and the results are thoroughly discussed.

Chapter I:

System’s description and model

Chapter I: System's description and model

I.1 Introduction:

A Shell Heavy Oil Fractionator Column is widely used equipment in oil refineries. It is designed for the separation of crude oil into various components based on differences in their boiling points. This chapter aims to provide a general overview of the shell heavy oil fractionator columns and present the Mathematical model that will be used in this work.

I.2 Shell Heavy Oil Fractionator Column [6]:

Shell Heavy Oil Fractionator Columns are widely used equipment in the petroleum industry to separate mixtures of two or more components based on differences in their boiling points and volatility. This separation is achieved by applying and removing heat, which turns the mixtures into products of desired purity. When a liquid mixture of two volatile materials is heated, the vapor that comes off will have a higher concentration of the more volatile material (lower boiling point) than the liquid from which it evolved. On the other hand, the less volatile (higher boiling point) material tends to condense in a larger proportion than the more volatile material when a vapor is cooled.

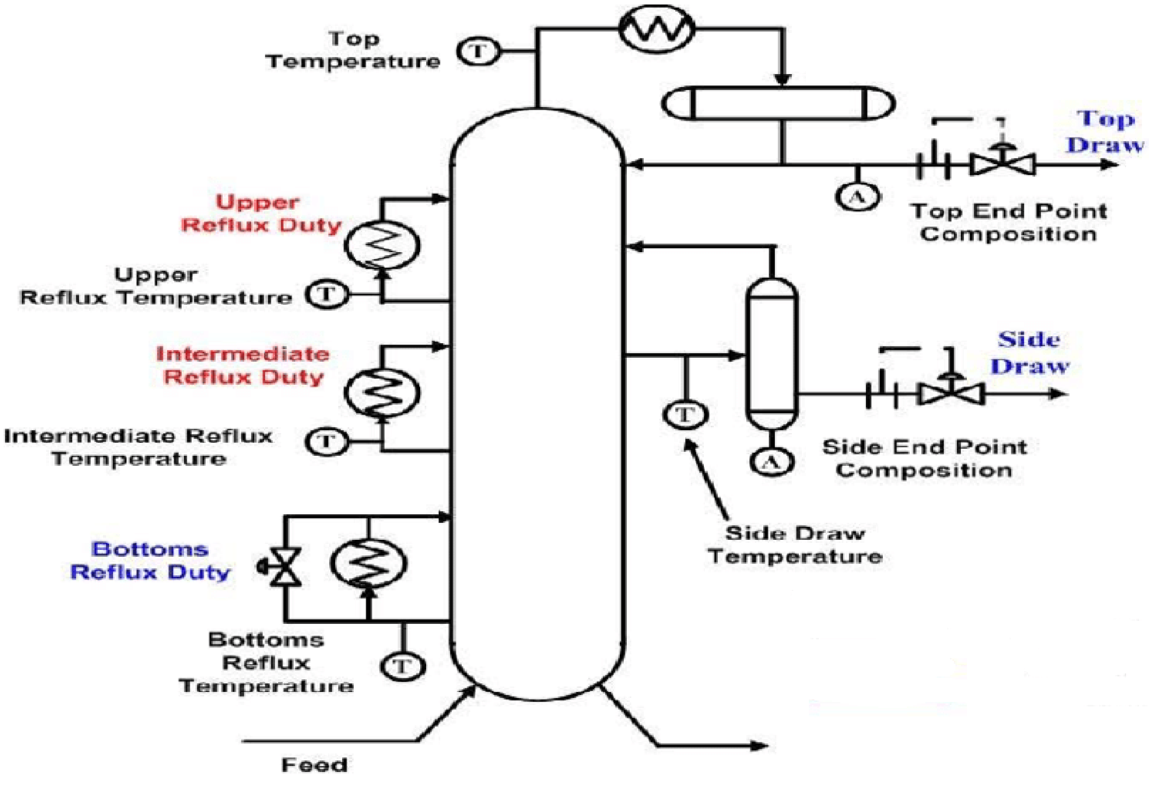


Figure I.1 : Shell’s Heavy oil fractionator (Shead et al., 2007)

I.3 Process Description:

First the heavy oil is heated to about 600 degrees . To get evaporated, then the vaporized heavy oil enters the (SHOF) through the bottom as the vapor rises it becomes cooler.

At the head of the column, tiny hydrocarbon molecules with low boiling points escape as gases. Larger hydrocarbon molecules, on the other hand, condense into liquids and accumulate at different levels due to their greater boiling temperatures.

The condensed liquid fractions are gathered by trays or plates inside the column and then sent through pipes outside the column.

The separated fractions, such as gasoline, diesel, kerosene, and bitumen, are either stored in tanks or sent to other areas for further chemical processing.

I.4 Mathematical Model:

I.4.1 Mathematical model representation:

A physical system is described by a mathematical model. In multivariable (MIMO) systems, we can represent it in three different ways:

- State-space representation
- Representation by differential equations
- Transfer function matrix representation

I.4.1.1 State-space representation:

The state-space model of a continuous linear system is given by:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{I.1}$$

where:

$\dot{x}(t)$: vector of state variables
$u(t)$: vector of input (control) variables
$y(t)$: vector of output variables
A : the state matrix
B : the input matrix
C : the output matrix
D : the input/output matrix

I.4.1.2 Representation by differential equations:

A multivariable linear time-invariant system (LTI) of order n can be described by a system of linear differential equations with constant coefficients. After applying the laws of physics, many multivariable systems can be described by differential and algebraic equations of the form:

$$F(x,u)=\begin{bmatrix}F_1(x_1,\dots,x_n,u_1,\dots,u_m)\\F_2(x_1,\dots,x_n,u_1,\dots,u_m)\\\vdots\\F_n(x_1,\dots,x_n,u_1,\dots,u_m)\end{bmatrix} \tag{I.2}$$

The expended form for Equation:

$$\begin{cases}\dot{x_1}=F_1(x_1,\dots,x_n,u_1,\dots,u_m)\\\dot{x_2}=F_2(x_1,\dots,x_n,u_1,\dots,u_m)\\\qquad\vdots\\\dot{x_n}=F_n(x_1,\dots,x_n,u_1,\dots,u_m)\end{cases} \tag{I.3}$$

I.4.1.3 Transfer function matrix representation:

The transfer function matrix representation is a way to describe the input-output relationship of a multi-input multi-output (MIMO) system in the frequency domain. In the context of a linear time-invariant (LTI) system, the relationship between the input vector $u(s)$and the output vector $y(s)$ in the frequency domain is given by:

$$y(s)=G(s)u(s) \tag{I.4}$$

Where, for an m-input, n-output system $G(s)$ is a $m\ x\ n$ transfer function matrix with shown as:

$$G(s)=\begin{bmatrix}g_{11}(s)&g_{12}(s)&\cdots&g_{1m}(s)\\g_{21}(s)&g_{22}(s)&\cdots&g_{2m}(s)\\\vdots&\vdots&\ddots&\vdots\\g_{n1}(s)&g_{n2}(s)&\dots&g_{nm}(s)\end{bmatrix} \tag{I.5}$$

And each typical $g_{ij}(s)$ transfer function element will be represented as:

$$g_{ij}(s) = \frac{Y_i(s)}{X_j(s)} = \frac{K\, e^{-Ds}}{as+1} \qquad (I.6)$$

where:

$Y_i(s)$: Laplace transform of the output
$X_j(s)$: Laplace transform of the input
K : the steady-state gain of the model
D : the model dead time
a : model time constant

I.4.1.4 Transition from state space to transfer function matrix representation:

The state space representation:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \\ X(0) = x_0 \end{cases} \qquad (I.7)$$

Due to the linearity of the Laplace transform, it is possible to apply it to the above equations.

$$\begin{cases} S\, X(S) = A\, X(S) + B\, U(S) \\ y(S) = C\, X(S) + D\, U(S) \end{cases} \qquad (I.8)$$

We get :

$$(SI - A)X(s) \;=\; B\, U(s) \qquad (I.9)$$

With

I: Identity matrix of dimension $n \times n$

After rearranging the terms, we will have:

$$\begin{cases} X(S) = (SI - A)^{-1}B\, U(S) \\ y(S) = C\, (SI - A)^{-1}B\, U(S) + D\, U(S) \end{cases} \qquad (I.10)$$

Since the transfer function matrix representation assumes that the initial conditions are zero; then:

$$T(s) = C\ (SI - A)^{-1}B + D \qquad \text{(I.11)}$$

I.4.2 Mathematical model of the (SHOF) [6]:

SHOF process is a multivariable process seven measured outputs, three manipulated inputs and two input disturbances as shown in Table I.1.

SHOF model is a 5 × 7 matrix of the first order plus dead time step responses as presented in Table I.1.

inputs outputs	TD(u1)	SD(u1)	BRD(u3)	URD(u4)	IRD(u5)
TEP(y1)	$\frac{4.05e^{-27s}}{(50s + 1)}$	$\frac{1.77e^{-28s}}{(60s + 1)}$	$\frac{5.88e^{-27s}}{(50s + 1)}$	$\frac{1.20e^{-27s}}{(45s + 1)}$	$\frac{1.44e^{-27s}}{(40s + 1)}$
SEP (y2)	$\frac{5.39e^{-18s}}{50s + 1}$	$\frac{5.72e^{-14s}}{(60s + 1)}$	$\frac{6.90e^{-15s}}{(40s + 1)}$	$\frac{1.52e^{-15s}}{(25s + 1)}$	$\frac{1.83e^{-15s}}{(20s + 1)}$
TT(y3)	$\frac{3.66e^{-2s}}{(9s + 1)}$	$\frac{1.65e^{-20s}}{(30s + 1)}$	$\frac{5.53e^{-2s}}{(40s + 1)}$	$\frac{1.16}{(11s + 1)}$	$\frac{1.27}{(6s + 1)}$
URT (y4)	$\frac{5.92e^{-11s}}{(12s + 1)}$	$\frac{2.54e^{-12s}}{(27s + 1)}$	$\frac{8.10e^{-2s}}{(40s + 1)}$	$\frac{1.73}{(5s + 1)}$	$\frac{1.79}{(19s + 1)}$
SDT (y5)	$\frac{4.13e^{-5s}}{(8s + 1)}$	$\frac{2.38e^{-7s}}{(19s + 1)}$	$\frac{6.23e^{-2s}}{(10s + 1)}$	$\frac{1.31}{(2s + 1)}$	$\frac{1.26}{(22s + 1)}$
IRT (y6)	$\frac{4.06e^{-8s}}{(13s + 1)}$	$\frac{4.18e^{-4s}}{(33s + 1)}$	$\frac{6.53e^{-1s}}{(9s + 1)}$	$\frac{1.19}{(19s + 1)}$	$\frac{1.17}{(24s + 1)}$
BRT (y7)	$\frac{4.38e^{-20s}}{(33s + 1)}$	$\frac{4.42e^{-22s}}{(44s + 1)}$	$\frac{7.2}{(19s + 1)}$	$\frac{1.14}{(27s + 1)}$	$\frac{1.26}{(32s + 1)}$

Table I.1 : Mathematical Model of the SHOF.

With:

Inputs:

Y1= Top End point Composition (TEP).

Y2= Side End Point Composition (SEP).

Y3= Top temperature (TT).

Y4= Upper reflux temperature (URT).

Y5= Side draw temperature (SDT).

Y6= Intermediate Reflux Temperature (IRT).

Y7= Bottom Reflux Temperature (BRT)

Outputs:

U1= Top Draw (TD).

U2= Side Draw (SD).

U3= Bottom Reflux Duty (BRD).

U4= Upper reflux duty (URD).

U5= Intermediate reflux duty (IRD).

In this work we are going to focus only on three manipulated variables (u1, u2 and u3) that have direct influence on the Top End point Composition (y1), Side End Point Composition (y2) and the Bottom Reflux Temperature (y7).

We eliminated the upper reflux duty loop and the intermediate reflux duty loop because they have fast dynamic compared to other loops so they are not affected by our controllers.

After removing the upper and intermediate reflux duty loops we will get this transfer function model:

$$\begin{pmatrix} Y1 \\ Y2 \\ Y3 \end{pmatrix} = \begin{pmatrix} \frac{4.05\mathrm{e}^{-27\mathrm{s}}}{(50\mathrm{s}\ +\ 1)} & \frac{1.77\mathrm{e}^{-28\mathrm{s}}}{(60\mathrm{s}\ +\ 1)} & \frac{5.88\mathrm{e}^{-27\mathrm{s}}}{(50\mathrm{s}\ +\ 1)} \\ \frac{5.39\mathrm{e}^{-18\mathrm{s}}}{50\mathrm{s}\ +\ 1} & \frac{5.72\mathrm{e}^{-14\mathrm{s}}}{(60\mathrm{s}\ +\ 1)} & \frac{6.90\mathrm{e}^{-15\mathrm{s}}}{(40\mathrm{s}\ +\ 1)} \\ \frac{4.38\mathrm{e}^{-20\mathrm{s}}}{(33\mathrm{s}\ +\ 1)} & \frac{4.42\mathrm{e}^{-22\mathrm{s}}}{(44\mathrm{s}\ +\ 1)} & \frac{7.2}{(19\mathrm{s}\ +\ 1)} \end{pmatrix} \begin{pmatrix} U1 \\ U2 \\ U3 \end{pmatrix}$$

With:

Y1= Top End point Composition (TEP).

Y2= Side End Point Composition (SEP).

Y3= Bottom Reflux Temperature (BRT).

U1= Top Draw (TD).

U2= Side Draw (SD).

U3= Bottom Reflux Duty (BRD).

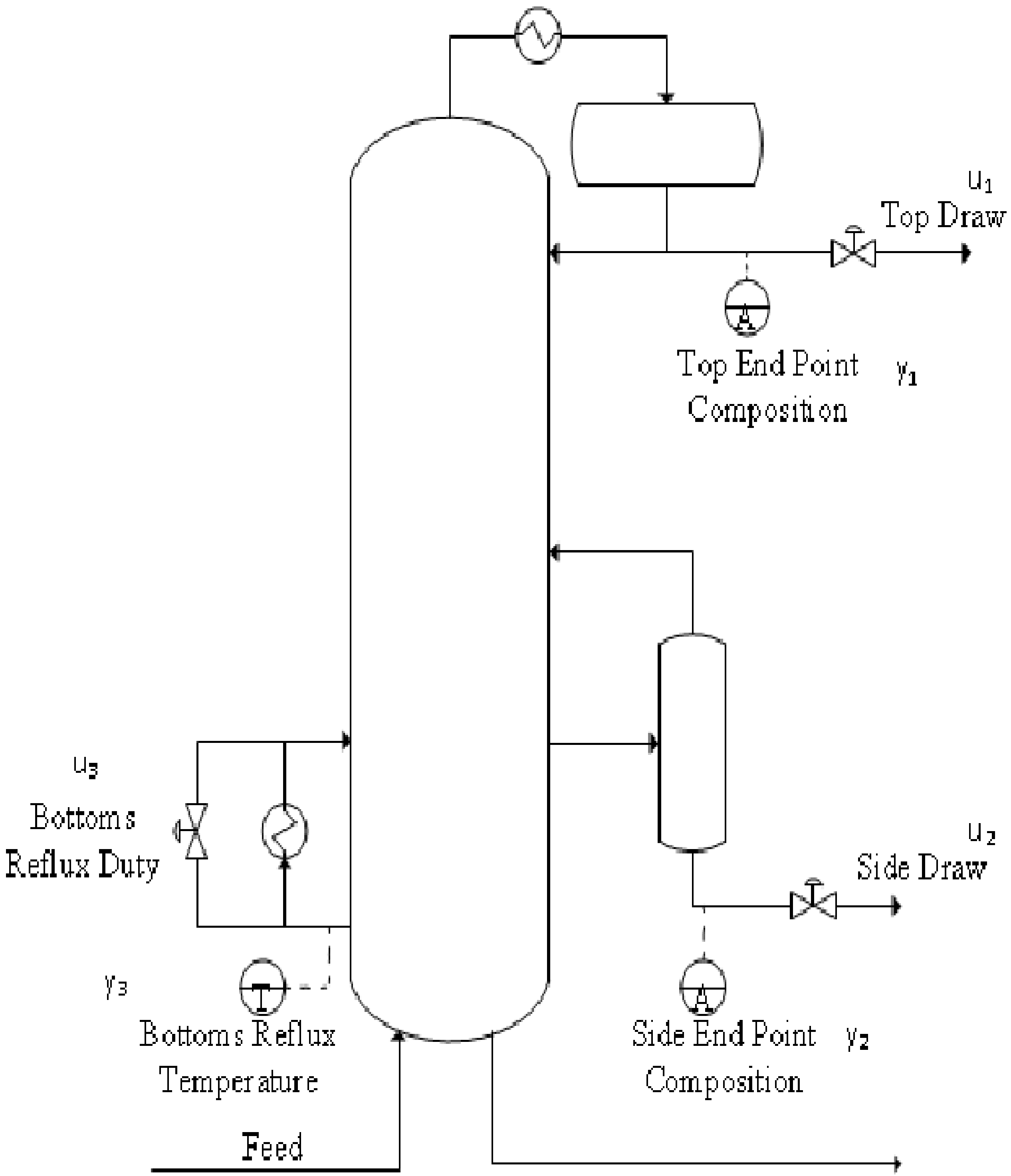

u1
Top Draw
Top End Point Composition
y1
u3
Bottoms Reflux Duty
u2
Side Draw
y3
Bottoms Reflux Temperature
Side End Point Composition
y2
Feed

Figure I.2 : Shell's Heavy oil fractionator with the three outputs.

I.5 Conclusion:

In this chapter, we have explored the principle of Shell Heavy Oil Fractionator (SHOF) process and the Mathematical Model of multivariable systems. We began by the definition of the SHOF then we have explained the process of separation of heavy crude oil. Additionally, we discussed various representations of multivariable systems and presented the multivariable model in the form of a Transfer function matrix, which simplifies the process of analyzing the system and facilitates the implementation of advanced control strategies that we will see in the chapters.

Chapter II:

Multivariable Systems and Interactions Analysis

Chapter II: Multivariable systems and interactions

II.1 Introduction:

In this chapter, we will explore multivariable systems, which occur in nearly all petroleum industry processes. We will address one of the major challenges in multivariable control: the interaction between variables.

II.2 Multivariable Systems:

A system is called multivariable or MIMO (Multiple Input Multiple Output) if it has multiple inputs $u = [u_1, u_2, \dots, u_p]$ and/or multiple outputs $y = [y_1, y_2, \dots, y_q]$, in MIMO systems a manipulated variable affects more than one output, and an output is influenced by more than one input.

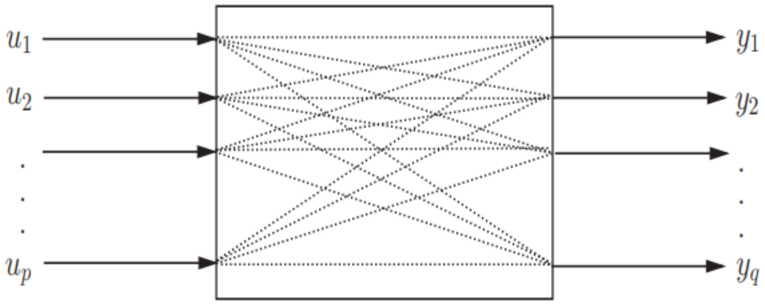


Figure II.1 : Multivariable system.

- u: is the input vector
- y: is the output vector

II.3 Multivariable Control:

Designing a control system suitable for an industrial multivariable process presents a number of challenges. Among these, the issue of interactions between input-output variables of the system is the primary cause for the difficulty in controlling and operating the multiloop system, loop by loop. A change in one input variable results in changes in several output variables, making it difficult to maintain the performance of each loop. Furthermore, the performance of one control loop can be significantly affected by the settings of a controller in other loops.

In the context of multivariable system control, considerable attention has been given to the concept of interaction analysis. In this perspective, the goal is often to compensate the system so that:

- Each input affects only one output.
- Disturbances on one output affect only that particular output.

II.3.1 Multiloop Control [1]:

Multiloop control techniques typically achieve acceptable performance levels in most cases. The synthesis of a multiloop control system is carried out in two steps:

Step 1: Determining the control configuration by selecting input-output pairs (each input should be looped with a specific output, introducing a well-designed controller).

Step 2: Choosing the control law and determining the controller settings for each loop to ensure desired performance.

In the first step, the selection of an appropriate control configuration—where interactions between resulting control loops are minimal—is guided by the use of an interaction analysis method, which also evaluates the level of interaction.

For a multivariable system with two inputs and two outputs, as illustrated in figure II.2:

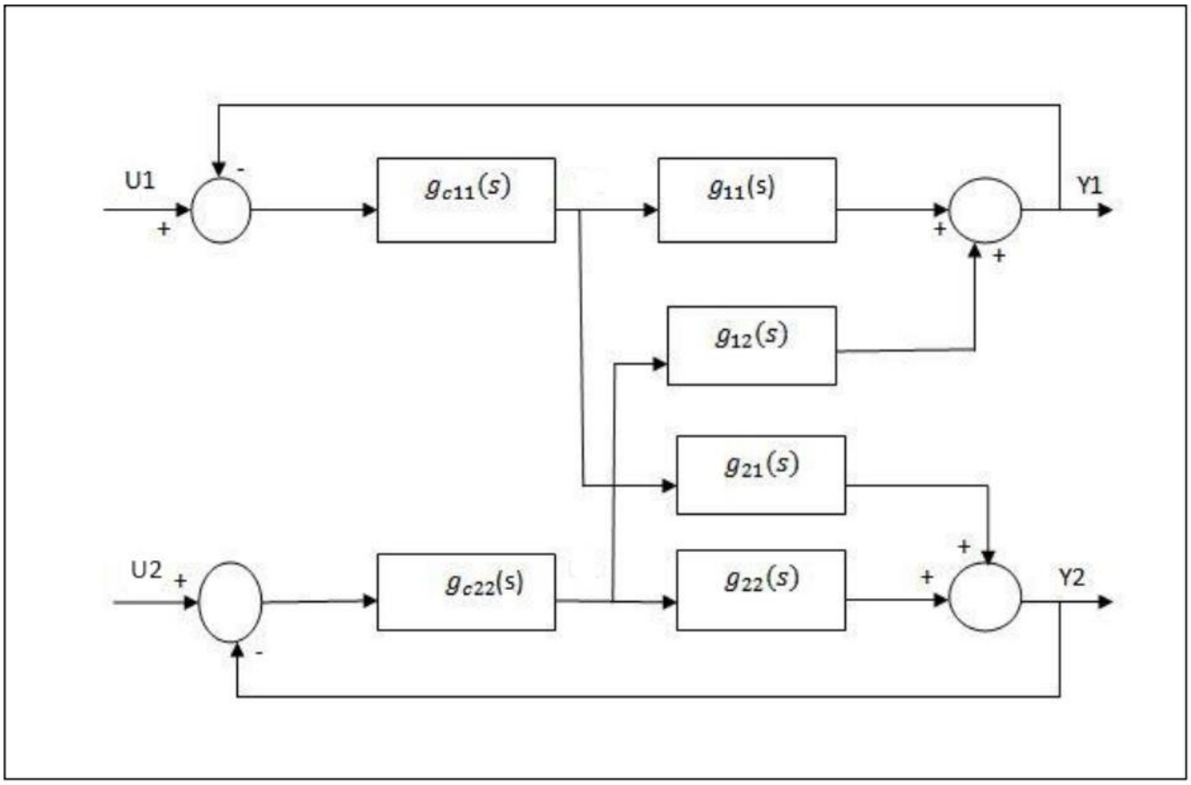


Figure II.2 : 2×2 Multivariable system.

Two control configurations are possible:

- $u1$ controls $y1$ and $u2$ controls $y2$, $[u_1 - y_1]$; $[u_2 - y_2]$;.
- $u1$ controls y2 and $u2$ controls $y1$, $[u_1 - y_2]$; $[u_2 - y_1]$;.

In the general case, for a system with m inputs and m outputs, there are $m!$ possible control configurations.

For the multiloop control of a system, the most important step is choosing the best control configuration (input-output pairs). This decision is made by analyzing the interactions present in the system. The preference is for a configuration where the level of interactions between different control loops is very low, while ensuring the stability of each loop and the overall system (closed-loop system).

II.3.2 Advantages of multiloop control:

Multiloop control offers certain advantages than centralized multivariable. Among these advantages are:

- Simplicity and speed of implementation on industrial equipment.
- The ability to keep certain outputs constant while modifying others.
- Prevention of disturbance propagation affecting one output throughout the system.
- After choosing the correct configuration of input/output pairs, one of the loops can be opened without causing instability.
- The ease of specifying different performance criteria for each variable to be controlled (for each loop).

II.4 Interactions in multivariable systems [1]:

We can determine if a multivariable system is interactive, if a control action $u_k(s)$ in Nth-order loop (resulting from a disturbance $z_k(s)$ or a set point change $c_k(s)$ causes a control action $u_l(s)$ $(l \neq k)$in one or more loops, with the aim of keeping the output variables $y_l(s)$ $(l \neq k)$assigned to these loops at their set points, as it is explained in the next point.

II.4.1 Explanation of the Interaction Phenomenon [1]:

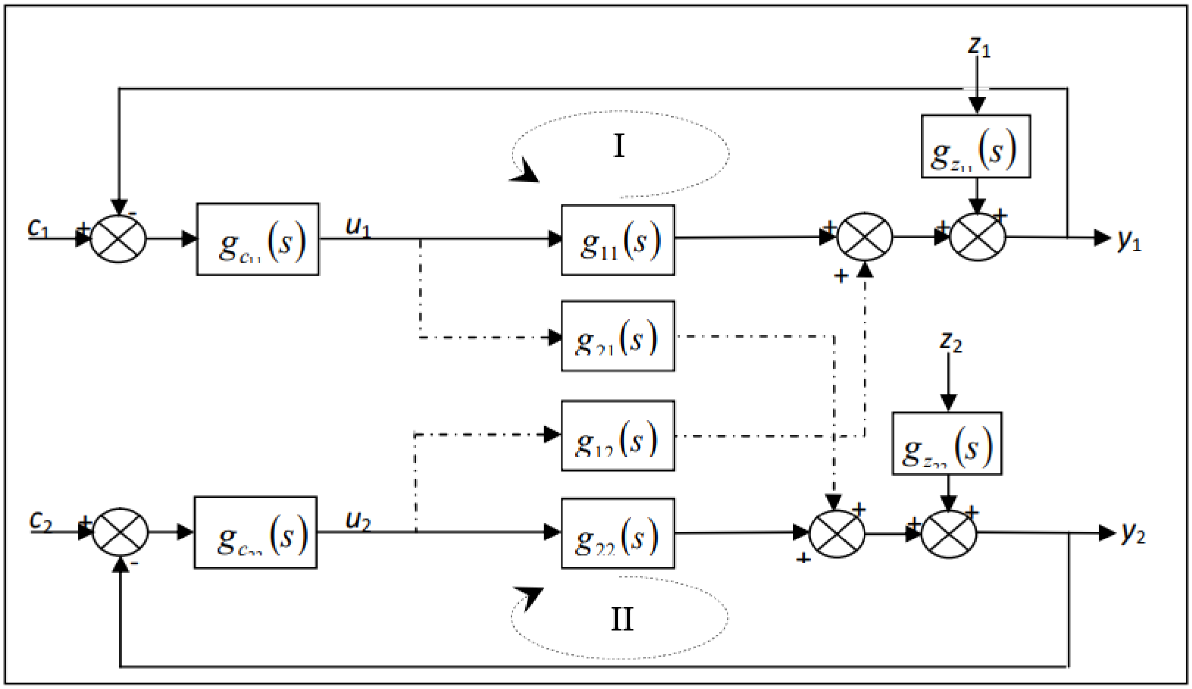


Figure II.3 : multiloop control

When disturbance z1 affects output y1, the error will deviate this output from its set point value c1. That will cause the controller g_{C11}(S) to generate an action u1 to eliminate this deviation (solid line). However, the generated control action u1 will also affects output y2 through the transmittance g_{21}(S) (dashed line), causing output y2 to deviate from its set point value c2. This necessitates the controller g_{C22}(S) to generate an action u2 to maintain output y2 at the desired position c2. The corrective action of the regulator g_{C22}(S) in the second loop (II) (the control action u2) also affects output y1 through the transmittance g_{12}(S). Therefore, maintaining outputs y1 and y2 at their desired positions, despite the disturbance z1 that must be canceled by the regulator g_{C11}(S), is a challenging task.

II.4.2 Methods for interactions analyzing:

II.4.2.1 Direct interaction analysis methods:

- Interaction Quotient method**(IQ)**
- Relative gain array **(RGA)**
- Dynamic Relative Gain Array **(DRGA)**.
- Nyquist-based methods **(DNA, INA).**

II.4.2.1.1 Interaction Quotient method (IQ) [2]:

The first works on interaction analysis date back to the studies by Rijnsdorp in 1965, Rijnsdorp proposed the Interaction Quotient (IQ) method, which provides information on the level of interactions between the loops of a candidate configuration by calculating the IQ for each possible configuration to identify the best one.

For a 2x2 system, the interaction quotient is defined by:

$$K(s) = \frac{G_{12}(s)*G_{21}(s)}{G_{11}(s)*G_{22}(s)} \tag{II.1}$$

The analysis of interactions relies on the calculation of the static value of $K(s)$:

$$K(s) = \lim_{s\to 0} \mathrm{K(S)} \tag{II.2}$$

Interpretation of (IQ):

- If $K(s) > 1$ interactions cause instability problems.
- If $-1 \le K(s) \le \frac{1}{3}$ weak interactions between system variables.
- If $K(s) \ge -1$ and $\frac{1}{3} \le K(s) \le 1$ strong interactions between system variables.

Note:

The IQ does not directly determine the best control configuration but provides information on the level of interaction between the loops of a candidate configuration. To determine the best configuration, one must calculate the IQ for each possible configuration in order to identify the optimal one.

II.4.2.1.2 Relative gain array (RGA) [3]:

The Relative Gain Array (RGA) is a method developed by Bristol in 1966, allows for the loop paring of a control system with a low level of interaction. The calculation of the RGA is based on the system's steady-state gain matrix. Each element of the RGA is determined by the following expression:

$$\lambda_{ij} = \frac{\left(\frac{\partial \mathrm{y}_i}{\partial \mathrm{u}_j}\right)_{u_k=0,k\neq j}}{\left(\frac{\partial \mathrm{y}_i}{\partial \mathrm{u}_j}\right)_{y_k=0,k\neq i}} \qquad \text{(II.3)}$$

The numerator represents the open-loop static gain between u_j and y_i , and the denominator is the static gain between u_j and y_i when the other outputs are controlled by perfect controllers. The relative gain λ_{ij} indicates whether the gain of an open-loop [u_j -y_i] changes when all other loops are closed.

Calculation of the RGA:

The Relative Gain Array is calculated directly using the steady-state gain matrix K_s as follows:

$$\mathrm{RGA} = K_s.*[\,K_s^{-1}\,]^T, \qquad \text{(II.4)}$$

with :

$$\mathrm{RGA} = [\,\lambda_{ij} : i,j = 1, \ldots., m], \qquad \text{(II.5)}$$

$$k_s = [\,K_{s_{ij}} : i,j = 1, \ldots\ldots, m], \qquad \text{(II.6)}$$

where :

.* : is the Hadamard product

K_s : is the steady-state gain matrix.

The elements K_s of are defined by the following expression:

$$K_{s_{ij}} = \lim_{s\to 0} \mathrm{g}_{ij}(\mathrm{S}). \qquad \text{(II.7)}$$

Properties of the RGA:

The algebraic sum of the elements of the RGA along any row i or column j is equal to 1.

$$\sum_{j=1,i=cst}^{m} \lambda_{ij} = 1$$

$$\sum_{i=1,j=cst}^{m} \lambda_{ij} = 1$$

Interpretation of the RGA:

- If the elements on the diagonal of the RGA (λ_{ij}: $i = j$) are close to 1, the level of interaction in the system is very low. However, if they are less than or greater than 1, the interactions are strong.
- For $\lambda_{ij} = 1$, the response for the input-output pair $[uj - yi]$ will be the same whether all other loops are open or closed, meaning the other loops have no influence on the $[uj - yi]$ loop.
- If λ_{ij} is negative, the response of the corresponding loop can change direction (inverse response system) when the other loops are closed. Additionally, the loop itself may become unstable, or the overall system may become unstable if the considered loop is opened, hence the corresponding pair should not be chosen in the control configuration.
- The selection of the control configuration should focus on pairs with a relative gain λ_{ij} close to 1.

Note:

The RGA method has its limitations, because:

- We assume the controllers used in the RGA are perfect.
- The RGA is only applicable to systems that operate around zero frequency.
- It only considers static cases, while the degree of interaction depends on the actual dynamics of the system.

II.4.2.1.3 Dynamic relative gain array (DRGA) [8]:

The dynamic relative gain array (DRGA) is extension of the Relative Gain Array (RGA) proposed by Witcher and McAvoy (1977) in order to study these processes dynamically; this dynamic extension of the RGA is represented by Equation.

$$\text{DRGA(s)} = \text{G (s).* [G (s)}^{-1}]^{T} \tag{II.8}$$

$$\text{With: DRGA}= [\, \lambda_{ij} : i, j = 1, \ldots\ldots, m], \tag{II.9}$$

The examination of the bode plots of the elements λ_{ij} allows for the analysis of the interactions between the system loops and the determination of the best control configuration.

Interpretation of the DRGA:

The interactions in the system are weak if the magnitude of each element λ_{ij} (diagonal elements) is close to 1 and the magnitudes of the other elements (off-diagonal elements) are close to zero within the system's working frequency.

Note:

The limitation of both the DRGA methods is the assumption that the controllers of the other loops are perfect. In other words, it is assumed that the output y2 remains constant at all times even if the control input u1 is modified.

II.4.2.1.4 Direct Nyquist Array (DNA) [5]:

The direct Nyquist matrix is a graphical method with the following principle:

• Construct the Nyquist plot for each element $G_{ij}(s)$of the transfer diagonal $G(s)$ for ω varying from 0 to +∞.

•Superimpose each plot with the Gershgorin circles obtained by varying ω from 0 to +∞.

•The coordinates of the center of a circle are the real and imaginary parts of $G_{ij}(s)$. The radius $R_{ij}(s)$ of the circle is the sum of the magnitudes of the elements of the column except for the magnitude of the considered element $G_{ij}(s)$. The radius $R_{ij}(s)$ is given by the following formula:

$$R_{ij}(s) = \sum_{j=1, j\neq i}^{m} |G_{ij}(s)| \tag{II.10}$$

II.4.2.1.5 Inverse Nyquist Array (INA) [5]:

INA is a method to find the stability of a process which is based on frequency response approximation that was developed by Rosenbrock in 1969. This stability method is worked on MIMO processes where there is interaction in every loop. Then he also introduced Gershgorin ring which can be applied to the INA so that it can diagnose how big the interaction effect is on influencing the stability.

Unlike the direct nyquist array (DNA), the Inverse nyquist array (INA) uses the inverse of the system transfer matrix $G(s)$, designated as $\hat{G}(s)$. The principle of INA is as follows:

- Calculate $\widehat{G}(s)$, which the inverse is of $G(s)$.
- Construct the nyquist plot for each element $\hat{G}(s)$, of the transfer matrix diagonal $\hat{G}(s)$, for ω varying from 0 to $+\infty$.
- Superimpose each plot with the Gershgorin circles obtained for ω varying from 0 to $+\infty$.

The coordinates of the center of a circle are the real and imaginary parts of $\widehat{G}(s)$. The radius $R_{ij}(s)$ of the circle is the sum of the magnitudes of the elements of the iii-th column except for the magnitude of the considered element$G(s)$. The radius $R_{ij}(s)$ is given by the following formula:

$$\hat{R}_{ij}(s) = \sum_{j=1,j\neq i}^{m} \left|\hat{G}_{ij}(s)\right| \qquad \text{(II.11)}$$

Interpretation of DNA and INA:

Both methods presented allow for analyzing the interactions between the loops of the configuration defined by the elements of the diagonal of the transfer matrix. This configuration exhibits weak interactions if the Gershgorin circles of each element of the diagonal of the system $G(s)$ or $\hat{G}(s)$do not overlap significantly.

II.4.2.2 dynamic interaction analysis Methods :

- Dynamic relative magnitude array (DRMA)
- Generalized relative dynamic gain (GRDG)

II.4.2.2.1 Dynamic relative magnitude array (DRMA) [1]:

This method does not provide a control configuration, but it allows for analyzing the interactions present in a closed-loop system (multi-loop control). The principle of DRMA involves first choosing a control configuration using one of the direct interaction analysis methods, then determining the controller for each loop. After synthesizing the multi-loop control system, DRMA is calculated to analyze the interactions in the resulting closed-loop system.

The calculation of the DRMA elements for a 3 by 3 system is given by:

$$\mathrm{DRMA} = \begin{bmatrix} \dfrac{\left(\frac{y_1(S)}{u_1(S)}\right)_{ooo}}{\left(\frac{y_1(S)}{u_1(S)}\right)_{occ}} & \dfrac{\left(\frac{y_1(S)}{u_2(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \dfrac{\left(\frac{y_1(S)}{u_3(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} \\ \dfrac{\left(\frac{y_2(S)}{u_1(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \dfrac{\left(\frac{y_2(S)}{u_2(S)}\right)_{ooo}}{\left(\frac{y_2(S)}{u_2(S)}\right)_{coc}} & \dfrac{\left(\frac{y_2(S)}{u_3(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} \\ \dfrac{\left(\frac{y_3(S)}{u_1(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \dfrac{\left(\frac{y_3(S)}{u_2(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \dfrac{\left(\frac{y_3(S)}{u_3(S)}\right)_{ooo}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{cco}} \end{bmatrix} \qquad \text{(II.12)}$$

Where:

ooo : all the loops are open.

occ : the first loop is open , the second t and the third loop are closed.

ccc : all the loops are close.

coc : the first and the third loop are close ,the second is open.

cco : the first and the second loop are closed , the third is open

Once the DRMA is determined, the Bode diagram (magnitude plot) of each element $\mathrm{X}_{ij}(\mathrm{S})$ of the matrix is plotted. The resulting Bode diagrams allow for the analysis of interactions between the loops of the chosen configuration.

Interpretation of the dynamic relative magnitude array (DRMA):

If the diagonal elements of the DRMA are close to 1 in the useful frequency range, the interactions in the closed-loop system will be insignificant.

The off-diagonal elements allow for examining the effect of each loop on another. If the element $X_{ij}(S)$ (i$\neq j$) is large (in the working frequency range), then the control u_j strongly affects y_i. On the other hand, if element $X_{ij}(S)$ (i$\neq j$) is close to zero, the output y_i is weakly affected by u_j.

Notes:

To analyze interactions using the DRMA, we can choose for example the control configuration with the RGA. This approach is only valid for systems operating around zero frequency; therefore we use the DRMA, to verifier the adequacy of the configuration derived from the application of the RGA

If the analysis results obtained using the RGA are not convincing, then the control configuration is reconsidered (m! configurations are possible), we analyze again to determine the best configuration.

The off-diagonal elements of the DRMA provide information on the effect of one loop on another (spread of a disturbance), mainly due to the influence of one loop's regulator settings on the others, as the DRMA does not assume that the regulators are perfect.

II.4.2.2.2 Generalized relative dynamic gain (GRDG) [4]:

The generalized relative dynamic gain (GRDG) is a method interactions analyzing created by Huang, Ahsima in 1994 as an amelioration of the RDGA. Unlike the RDGA, GDRG does not hypothesis that the closed-loop controllers are ideal. The method's idea is to analyze interactions after selecting a configuration and synthesizing controllers. Applying the method allows evaluating the level of interaction between loops in the considered configuration.

$$\text{GDRG} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \end{bmatrix} \qquad \text{(II.13)}$$

$$\lambda_{ij}(\mathrm{S}) = \frac{\left(\frac{\partial \mathrm{y}_i(\mathrm{S})}{\partial \mathrm{u}_j(\mathrm{S})}\right)_{open\ loop}}{\left(\frac{\partial \mathrm{y}_i(\mathrm{S})}{\partial \mathrm{u}_j(\mathrm{S})}\right)_{closed\ loop}} \tag{II.14}$$

With:

$$\mathrm{GRDG} = [\lambda_{ij} : i, j = 1, \ldots\ldots, m], \tag{II.15}$$

The Generalized Relative Dynamic Gain (GRDG) indicates the change in the open-loop gain between u_j and y_j , due to interactions and the effects of the settings of the controllers in the other loops.

II.5 Conclusion:

In this chapter, we have explored the principle of multivariable systems and one of the major problems in multivariable control the interaction. We began by defining multivariable systems and then we explained more details about the multivariable control. We also examined the phenomenon of interaction, which characterizes the majority of multivariable processes. Additionally, we discussed various interaction analysis methods, including direct analysis methods for determining the best loop pairing configuration and dynamic analysis methods for the determination of the level of interaction. Interaction analysis is crucial in multivariable control as it helps in the determination of the appropriate control strategy, whether single-variable or multivariable methods. In Chapter 3, we will discuss this topic in more detail.

Chapter III:

Control methods

Chapter III: Control methods

III.1 Introduction:

To improve the precision, stability, and response speed of a system, it is essential to incorporate a Proportional Integral Derivative (PID) controller into the control loop. This chapter aims to present the fundamental concepts of the PID controller and to explain key control methods, such as the Ziegler-Nichols method and the EVANS method for single-variable systems, as well as the Biggest log Module Tuning (BLT) method for multivariable systems.

III.2 Proportional-Integral-Derivative PID controller:

Classical PID controllers are one of the widely used controllers today, having been widely employed in industries since the 1940s. The PID controller can be found in many application areas: petroleum processing, chemical industries, robotics and many more. The PID controller is the combination of three actions: Proportional, Integral, and Derivative. The structure of this controller depends on the three coefficients. The presence of the integral action gives of a statical precision to the system, and the presence of the derivative action improves stability. Its equation is given by:

$$u(t) = K_p * (e(t) + \frac{1}{T_i} * \int_{-\infty}^{t} e(\tau) * d_\tau + T_d * \frac{de}{dt}) \qquad \text{(III.1)}$$

III.2.1 Study of PID actions [1]:

In this paragraph, we are interested in the study of the actions of the PID module composed of the three basic actions. We will highlight the effect produced by each action in a control loop, its advantages, as well as its limitations.

III.2.1.1 Proportional action analysis (P):

The proportional action is the basic action of the controller, by increasing the Proportional gain it reduces the steady-state error but also decreases the stability margins of the closed-loop system. Additionally, the proportional action quickly corrects any deviation in the controlled variable. It minimizes control error and make the system faster.

In the case of a proportional controller, the corrected control law is proportional to the error $e(t)$:

$$u(t) = K_P e(t) + u_0 \tag{III.2}$$

Where:

- $e(t)$ is the error signal
- K_P is the proportional gain
- u_0 is the initial value of u

The transfer function of the controller is therefore:

$$C(s) = \frac{U(s)}{e(s)} = K_P \tag{III.3}$$

III.2.1.2 Integral action analysis (I):

Integral action is the complements of proportional action. Its primary role is to eliminate steady-state error. However, it unfortunately introduces a phase lag and can make the system unstable. In the case of an integral controller, the control signal $u(t)$ is equal to the integral of the error $e(t)$:

$$u(t) = \frac{1}{Ti} \int e(t)\, dt \tag{III. 4}$$

The coefficient $\frac{1}{Ti}$ acts like a gain, and its value affects the transient behavior of the closed loop.

In industry, the integral action is used whenever perfect accuracy is needed for technological reasons – for example, the regulation of pressure or temperature. Integral action is never used alone but is combined with proportional action.

III.2.1.3 Derivative action analysis (D):

Derivative action is used in industry for adjusting slow variables such as temperature. It is not recommended for adjusting noisy or highly dynamic variables like pressure, as deriving noise can amplify. These noises are also added to the error signal. The derivative controller not only derives the useful error signal but also amplifies the noise signal.

In the case of a derivative controller, the control signal $u(t)$ is equal to the derivative of the error $e(t)$:

$$u(t) = T_d \frac{de(t)}{dt} \tag{III.5}$$

Where:

- T_d is the derivative time constant .

III.3 Single-variable control methods:

Two well-known methods are cited in this project:

- Root Locus Method (Evans Method)
- Ziegler-Nichols Method

III.3.1 Root Locus Method (Evans Method) [1]:

Let's consider as the standard form for constructing this root locus, the open-loop transfer function given by:

$$GH(s) = \frac{K(s+z_1)(s+z_1)\ldots\ldots\ldots(s+z_n)}{(s+\tau_1)(s+\tau_2)\ldots\ldots\ldots(s+\tau_n)} \tag{III.6}$$

The root locus construction is simplified by applying the following rules:

1. The locus starts from the poles of the open-loop transfer function (with the controller gain $Kc = 0$) and ends at the zeros of the open-loop transfer function or at infinity ($Kc = \infty$).
2. Complex poles or zeros appear as complex conjugates, thus, the plot is always symmetric about the real axis.
3. The number of loci approaching infinity as Kc increases is equal to the number of poles minus the number of zeros.

4. The centroid is calculated by the formula:

$$G_0 = \frac{\sum_{K=1}^{N_P} R_e(P_k) - \sum_{K=1}^{N_Z} R_e(Z_k)}{N_P - N_Z} \qquad \text{(III.7)}$$

N: number of poles

Z: number of zeros

5. Number of asymptotes = $N_P - N_Z$
6. The angles between the asymptotes and the real axis are given by:

$$\gamma = \frac{(1+2n)\pi}{N_P - N_Z}, n = 1,2, \dots \dots \dots \qquad \text{(III.8)}$$

7. The departure angle of the root locus of a complex pole is given by:

$$\theta_D = 180^{\circ} + argGH' \qquad \text{(III.9)}$$

$argGH'$ is the phase of $GH(s)$ calculated at the complex pole without considering the contribution of this particular pole.

8. Branching points these are the points where the locus leaves or joins the real axis. This corresponds to values of K such that the closed-loop system has double poles. To find these points, we apply the formula:

$$\frac{dK}{dS} = 0 \qquad \text{(III.10)}$$

9. Intersection with the imaginary axis, if the locus intersects with the imaginary axis, it means that for certain values of K, the closed-loop transfer function has purely imaginary poles. To find these points, we use techniques that examine the characteristic equation for critical stability, such as the use of the Routh-Hurwitz matrix."

III.3.2 Ziegler-Nichols Method:

The Ziegler-Nichols method is a widely used technique for tuning PID controllers. This method was introduced by John G. Ziegler and Nathaniel B. Nichols in their 1942.
The Ziegler-Nichlos (ZN) method consists of first finding the ultimate gain K_u , the value of gain at which the loop is at the limit of stability with only a proportional controller. The period of the resulting oscillation is called the ultimate period P_u. The ZN settings are then calculated from K_u and P_u by the formulas given in table:

Type of controller	K_c	$\frac{1}{T_i}$	T_d
P	$\frac{K_u}{2}$		
PI	$\frac{K_u}{2.2}$	$\frac{P_u}{1.2}$	
PID	$\frac{K_u}{1.7}$	$\frac{P_u}{2}$	$\frac{P_u}{8}$

Table III.1 : Ziegler-Nichols controller settings.

This method can only be used if the studied system tolerates overshoots. Therefore, there are many cases where the settings in Table (III.1) are not crucial because they are too oscillatory. This is why other settings (Table III.2), which lead to a highly damped response, are used. One can choose either a response to a set point change with slight overshoot or a response without overshoot.

Type of controller	K_c	$\frac{1}{T_i}$	T_d
Original	$0.6K_u$	$\frac{P_u}{2}$	$\frac{P_u}{8}$
Slight overshoot	$0.33K_u$	$\frac{P_u}{2}$	$\frac{P_u}{3}$
No overshoot	$0.2K_u$	$\frac{P_u}{2}$	$\frac{P_u}{3}$

Table III.2 : Modified Ziegler-Nichols settings.

III.4 Multivariable control methods:

There are many classical multivariable control method like:

- Biggest log-modulus Tuning method (BLT)
- Internal Module Control (IMC)
- inverse Nyquist Array (INA)

III.4.1 Biggest log-modulus tuning method (BLT) [7]:

Biggest Log-Modulus Tuning (BLT) is a method proposed by William L. Luyben in 1986 [7] for the tuning of controllers in multivariable systems. It is an extension of the Ziegler-Nichols tuning method, specifically designed for systems where the interactions are strong. It allows to determinate the best setting of the controller's to ensure the stability of the system. The design of PI controllers using the BLT method involves the following steps:

Step 1:

Calculating the settings of the PI controller according to the Ziegler-Nichols method for each individual loop starting with determining the ultimate frequency ω_u, which is the frequency corresponding to phase -π, and the ultimate gainK_u, which is the inverse of the real part of $G_{ii}(s)$ at the ultimate frequency. Then, the Ziegler-Nichols parameters are calculated using the following formulas:

$$K_{ZN} = \frac{K_u}{2.2} \tag{III.11}$$

$$T_{ZN} = \frac{\pi}{0.6\omega_u} \tag{III.11}$$

In the following, it will be given the steps for getting the Ziegler-Nichols parameters:

To calculate the ultimate frequency ω_u, we use the phase margin formula:

$$arg(G(j\omega)) = -\pi \tag{III.13}$$

For:

$$G_{11}(s) = \frac{4.05\mathrm{e}^{-27s}}{(50s + 1)}$$

We have:

$$G_{11}(j\omega_u) = \frac{4.05\mathrm{e}^{-27j\omega_u}}{(50j\omega_u + 1)}$$

Calculating the phase:

$$\varphi_1 = \arg\big(G_{11}(j\omega_u)\big) = -\pi$$

$$\varphi_1 = \arg(4.05) + \arg(\mathrm{e}^{-27\mathrm{j}\omega_u}) - \arg\,(50j\omega_u\ +\ 1) = -\pi \qquad \text{(III.14)}$$

$$\varphi_1 = -27\omega_u - \arctan\,(50\omega_u\,) = -\pi$$

Numerically solving the equation, the value of ultimate frequency ω_u:

$\omega_u = \ 0.06867 rad/s$

The calculate of the ultimate gainK_u:

$$K_u\|G_{11}(j\omega_u)\| = 1 \qquad \text{(III.15)}$$

$$K_u\frac{4.05}{\sqrt{(50\omega_u\,)^2+1}} = 1$$

We replace the value of the ultimate frequency ω_u in the equation:

$$\frac{\sqrt{(50\omega_u\,)^2+1}}{4.05} = 1$$

$$K_u = \ 0.883$$

Calculating the controller settings using the Ziegler-Nichols:

$$K_{ZN}\ = \ \frac{K_u}{2.2} = \frac{0.883}{2.2} = 0.883$$

$$T_{ZN} = \frac{\pi}{0.6\omega_u} = 76.248$$

Step 2:

The detuning factor F is assumed ($2 < F < 5$). F should always be bigger than 1. We compute the gains of controllers K_{BLT} by dividing the gains of Ziegler-Nichols K_{ZN} by the factor F.

$$K_{BLT} = \frac{K_{ZN}}{F} \tag{III.16}$$

Then all controllers reset times T_{BLT} are calculated by multiplying the Ziegler-Nichols reset times T_{ZN} by the same factor F.

$$T_{BLT} = T_{ZN}F \tag{III.17}$$

Step 3:

Using the guessed value of F and the resulting controller settings, a multivariable Nyquist plot of the scalar function is made:

$$W\ (jw)\ = -1 + Det\ [I + G\ (jw)G_c(s)] \tag{III.18}$$

With:

$G\ (s)$: transfer function matrices of the system.

$G_c(s)$: diagonal transfer function matrices of the controllers.

I: identity matrix.

$$G_c(s) = \begin{bmatrix} g_{c1}(s) & 0 & \dots & 0 \\ 0 & g_{c2}(s) & \dots & 0 \\ \dots & \dots & \ddots & 0 \\ 0 & 0 & \dots & g_{cn}(s) \end{bmatrix} \tag{III.19}$$

With:

$$G_{ci}(s) = K_P\left(1 + \frac{1}{T_i s}\right) \tag{III.20}$$

We represent function $W\ (jw)$ by Nyquist plot, the closer it is to the point (-1, 0), the closer we are to instability. By analogy with SISO systems, we define the closed-loop multivariable modulus.

$$L_{cm} = 20log\left|\frac{w}{1+w}\right| \qquad \text{(III.21)}$$

Step 4:

The F factor is varied until L_{cm}^{max} is equal to 2N, where N is the order of the system. For N = 1, the SISO case, we get the familiar +2 dB maximum closed loop log modulus. For a $2\ x\ 2$ system, at 4 dB value of L_{cm}^{max} is used; for a 3 x 3, +6 dB; and so forth.

III.5 conclusion:

In this chapter, we provided an overview of the PID controller and some key control methods. We began by defining the PID controller and explaining its various actions that influence the dynamic behavior of a system in a closed-loop. Additionally, we discussed key control methods for single-variable systems, including the Ziegler-Nichols and Evans methods. For multivariable systems, we introduced the Biggest Log Module tuning method. Using interaction analysis methods, we can determine the appropriate control method to implement, which will be explored in the final chapter.

Chapter IV:

Simulation and results

Chapter IV: Simulation and results

IV.1 Introduction:

In this chapter, we will explore various methods to optimize loop pairing and control in multivariable systems. First, we will use the Direct Interaction Analysis Method (RGA) to determine the best loop pairing configuration. Next, we will apply the Dynamic relative magnitude array (DRMA) to study the interactions between the loops. Finally, we will implement the Biggest Log Module Tuning Method (BLT) as a multivariable control method.

IV.2 The RGA matrix:

To calculate the RGA matrix we are going to use the equations (II.4) and (II.7):

$$\mathrm{K} = \lim_{s\to 0} \mathrm{g}(S) = \begin{pmatrix} 4.05 & 1.77 & 5.88 \\ 5.39 & 5.72 & 6.90 \\ 4.38 & 4.42 & 7.2 \end{pmatrix}$$

$$RGA \;= K.* \, [K^{-1}\,]^{T} = \begin{pmatrix} 2.0757 & -0.7289 & -0.3468 \\ 3.4242 & 0.9343 & -3.3585 \\ -4.4999 & 0.7946 & 4.7053 \end{pmatrix}$$

The best loop paring configuration will be as follows:

- the first output by the first input $[u_1 - y_1]$
- the second output by the second input $[u_2 - y_2]$
- the third output by the third input $[u_3 - y_3]$

IV.3 The DRMA matrix:

IV.3.1 The design of the controller for each loop:

Using the Evans method, we can determine the PI controller settings for each loop with a damping factor of $\xi = 0.5$.

The PI controller for the Top End point Composition loop will be :

$$gc1 = \frac{0.3582s + 0.0062}{s}$$

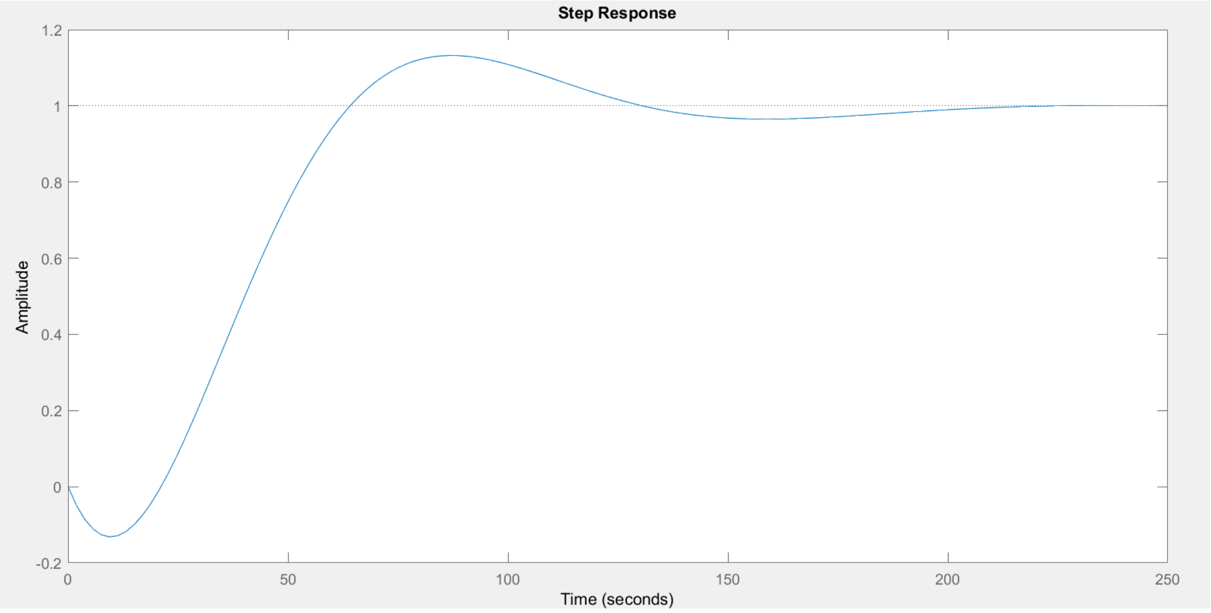


Figure IV.3 : The step response of the first loop

The PI controller for the Side End Point Composition loop is:

$$gc2 = \frac{0.5529s + 0.0115}{s}$$

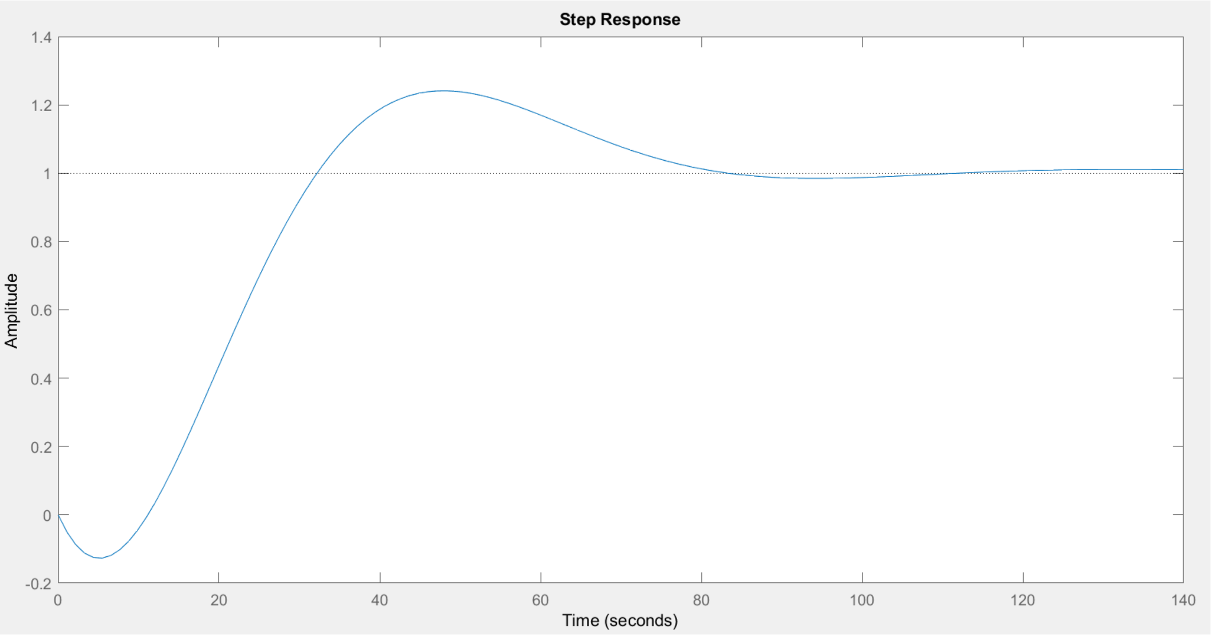


Figure IV.4 : The step response of the second loop

The PI controller for the Bottom Reflux Temperature loop will be:

$$gc3 = \frac{0.1536s + 0.0243}{s}$$

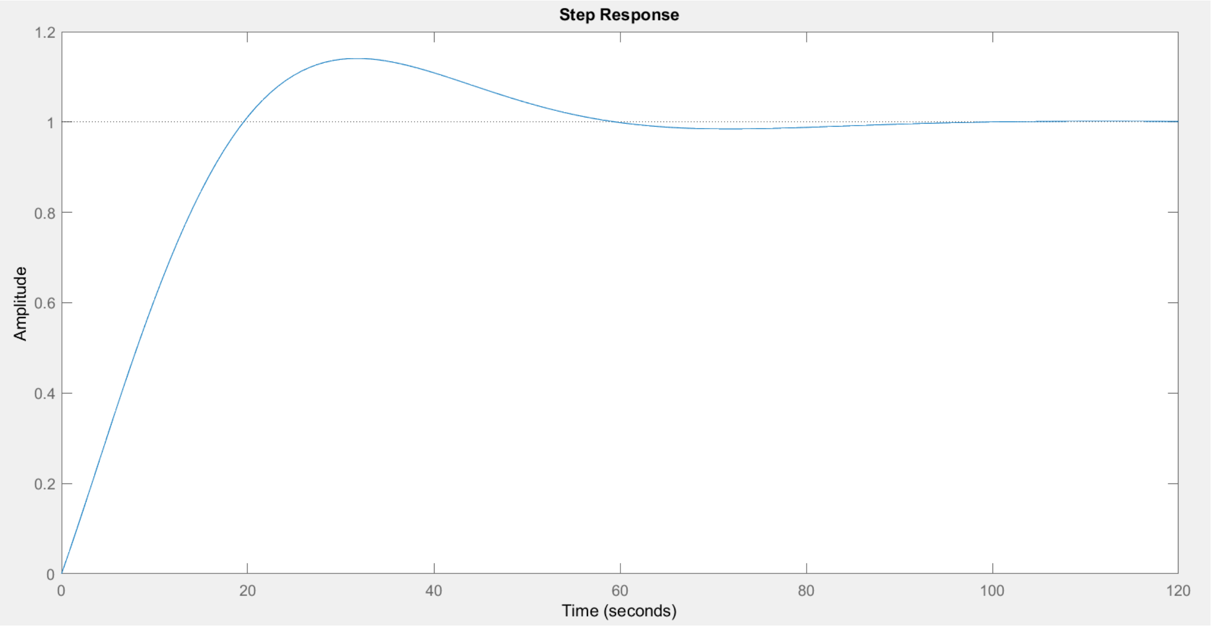


Figure IV.5 : The step response of the third loop

IV.3.2 Operating frequency of each loop:

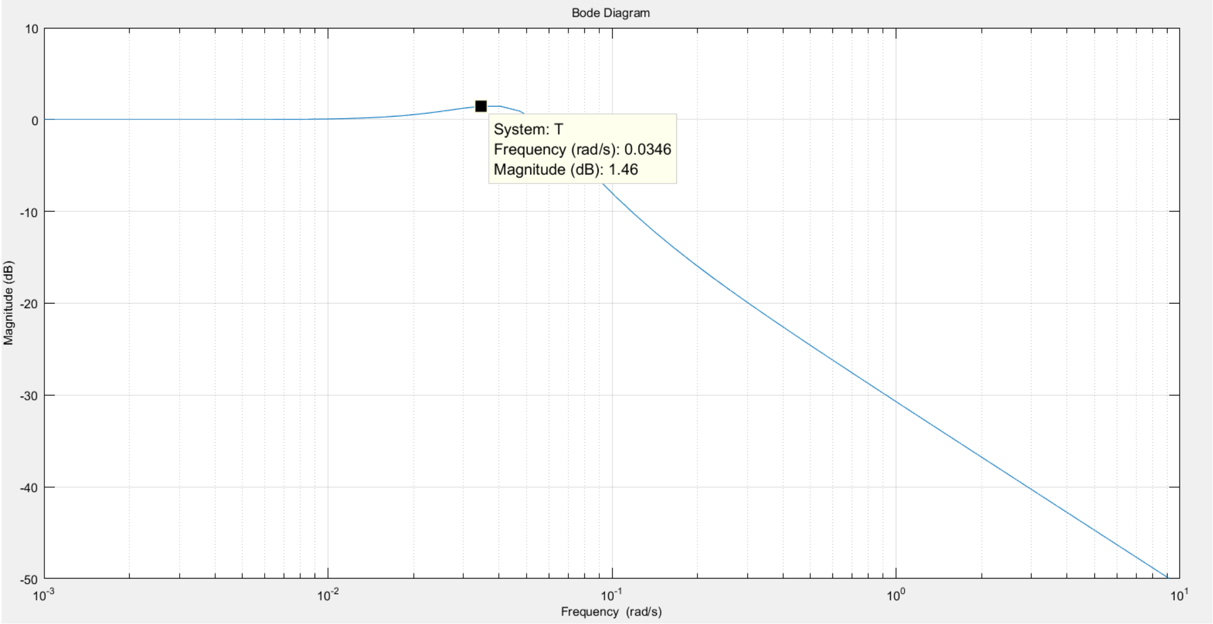


Figure IV.6 : Bode diagram of the first loop

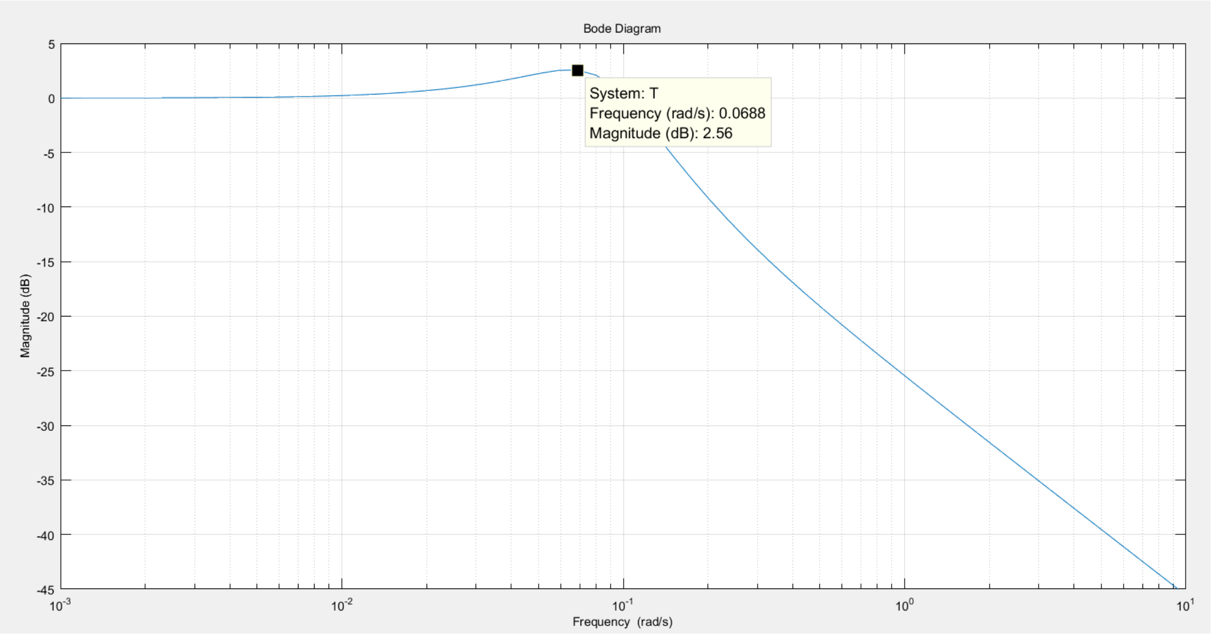


Figure IV.7 : Bode diagram of the second loop

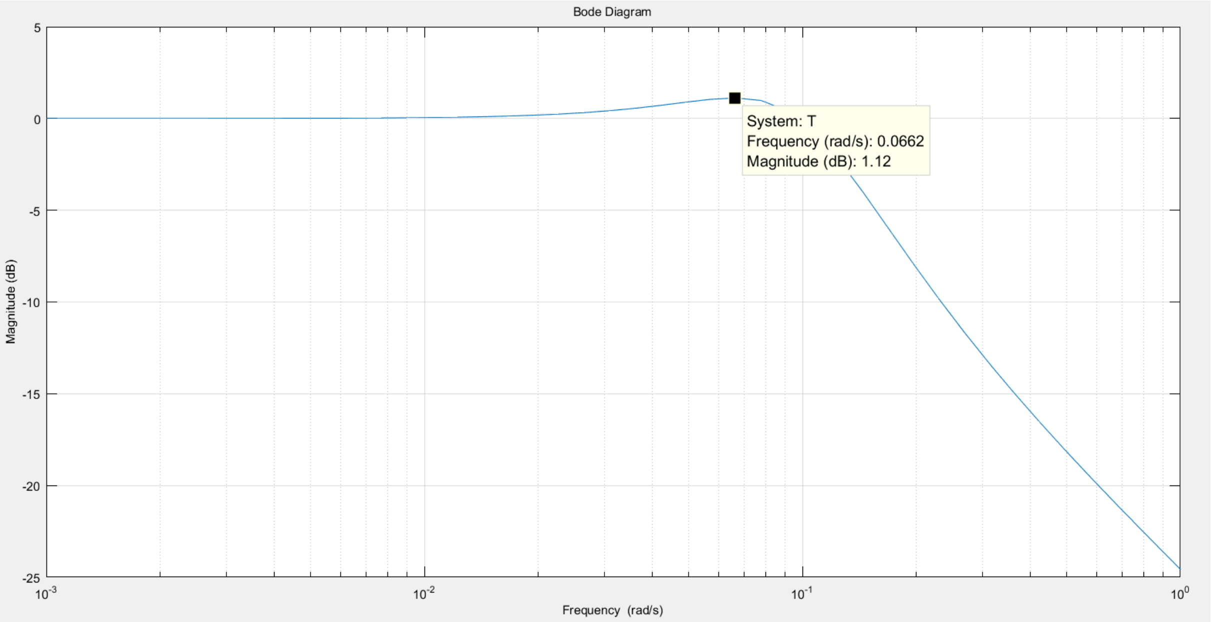


Figure IV.8 : Bode diagram of the third loop

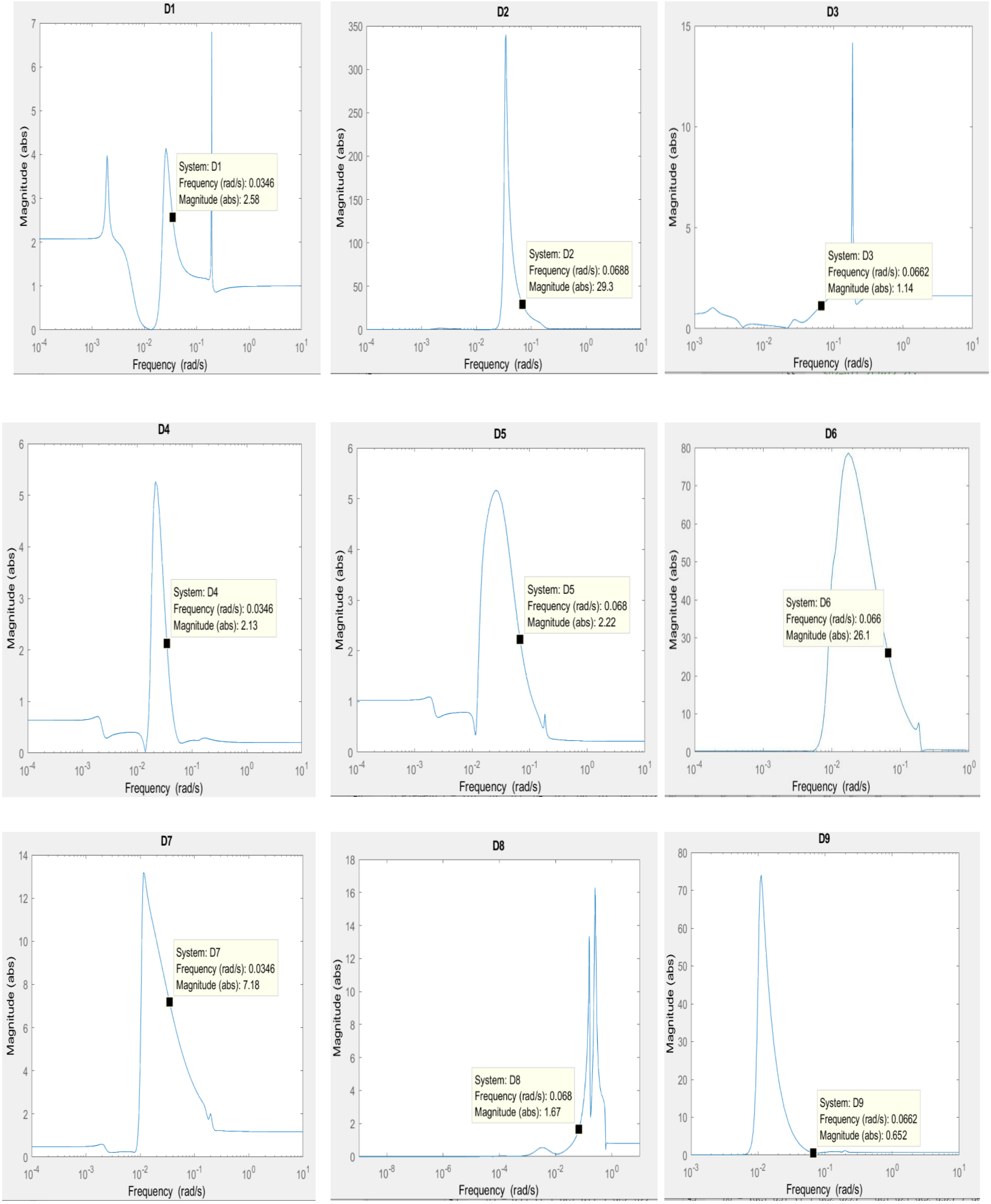
The pic frequency of each loop are:

ω_{o1}=0.0346 rad/s

ω_{o2}=0.0688 rad/s

ω_{o3}=0.0662 rad/s

IV.3.3 Bode diagram of each element of the DRMA matrix:



IV.3.4 DRMA matrix results:

After replacing each value of operating frequencies in there correspondent bode diagram we can determine the DRMA matrix using the equation (II.12):

$$\text{DRMA} = \begin{bmatrix} \frac{\left(\frac{y_1(S)}{u_1(S)}\right)_{ooo}}{\left(\frac{y_1(S)}{u_1(S)}\right)_{occ}} & \frac{\left(\frac{y_1(S)}{u_2(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \frac{\left(\frac{y_1(S)}{u_3(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} \\ \frac{\left(\frac{y_2(S)}{u_1(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \frac{\left(\frac{y_2(S)}{u_2(S)}\right)_{ooo}}{\left(\frac{y_2(S)}{u_2(S)}\right)_{coc}} & \frac{\left(\frac{y_2(S)}{u_3(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} \\ \frac{\left(\frac{y_3(S)}{u_1(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \frac{\left(\frac{y_3(S)}{u_2(S)}\right)_{ccc}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{ccc}} & \frac{\left(\frac{y_3(S)}{u_3(S)}\right)_{ooo}}{\left(\frac{y_3(S)}{u_3(S)}\right)_{cco}} \end{bmatrix}$$

$$DRMA = \begin{pmatrix} 2.58 & 29.3 & 1.14 \\ 2.13 & 2.22 & 26.1 \\ 7.13 & 1.67 & 0.652 \end{pmatrix}$$

Analysis of the Diagonal Elements of the DRMA Matrix:

The diagonal elements (λ_{11} , λ_{22} and λ_{33}) of the DRMA are different from 1.

- The first loop, λ_{11}= 2.58 at the frequency ω_{o1}=0.0356rad/s, indicating that the behavior of this loop when the second and third loops are open is different from when they are closed.
- The second loop, λ_{22}= 2.22 at the resonance frequency ω_{o2}=0.0688rad/s, indicating that the behavior of this loop when the first and third loops are open is different from when they are closed.
- The third loop, λ_{33}= 0.652 at the resonance frequency ω_{o2}=0.0662rad/s, indicating that the behavior of this loop when the first and third loops are open is different from when they are closed.
- We conclude that the system is highly interactive.

Analysis of Off-Diagonal Elements of the DRMA Matrix:

Element λ_{12} = 29.3 and λ_{21} = 2.13:

- The element λ_{12} is large, indicating that input u_2 strongly affects output y_1.
- The element λ_{21} is relatively smaller, suggesting that control u_1 has a weaker effect on output y_2.
- Since λ_{12} is much larger than λ_{21}, the effect of u_2 on y_1, is significantly greater than the effect of u_1 on y_2.

Element λ_{13} = 1.14 and λ_{31} = 7.13:

- The element $\lambda_{13}$is relatively small, indicating that control u_3 has a weak effect on output y_1.
- The element λ_{31}) is larger, suggesting that control u_1 strongly affects output y_3.
- Since λ_{31} is much larger than λ_{13}, the effect of u_1 on y_3 is greater than the effect of u_3 on y_1.

Element λ_{23} = 26.1 and λ_{32} = 1.67:

- The element λ_{23} is large, indicating that control u_3 strongly affects output y_2.
- The element $\lambda_{32}$is relatively small, suggesting that control u_2 has a weaker effect on output y_3.
- Since $\lambda_{32}$is much larger than λ_{32}, the effect of u_3 on y_2 is significantly greater than the effect of $u_2$on y_3.

In conclusion, since all the off-diagonal elements of the DRMA matrix are greater than zero, we can say that the system is highly interactive and all the loops affect each other.

IV.4 The design of the controller for each loop using BLT:

The new PI controller value for the Top End point Composition loop using the BLT method will be:

$$gc1 = \frac{0.32s + \ 0.0035}{s}$$

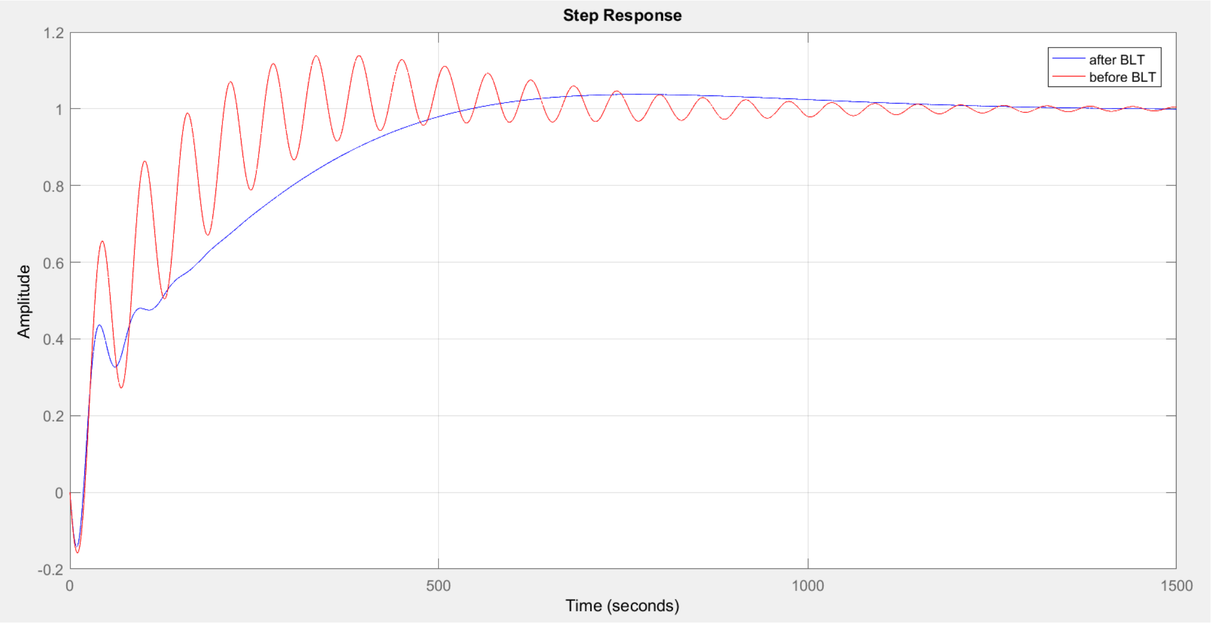


Figure IV.9 : The step response of the first loop (before and after BLT)

The new PI controller value for The Side End Point Composition loop using BLT method will be:

$$gc2 = \frac{0.41s + 0.0052}{s}$$

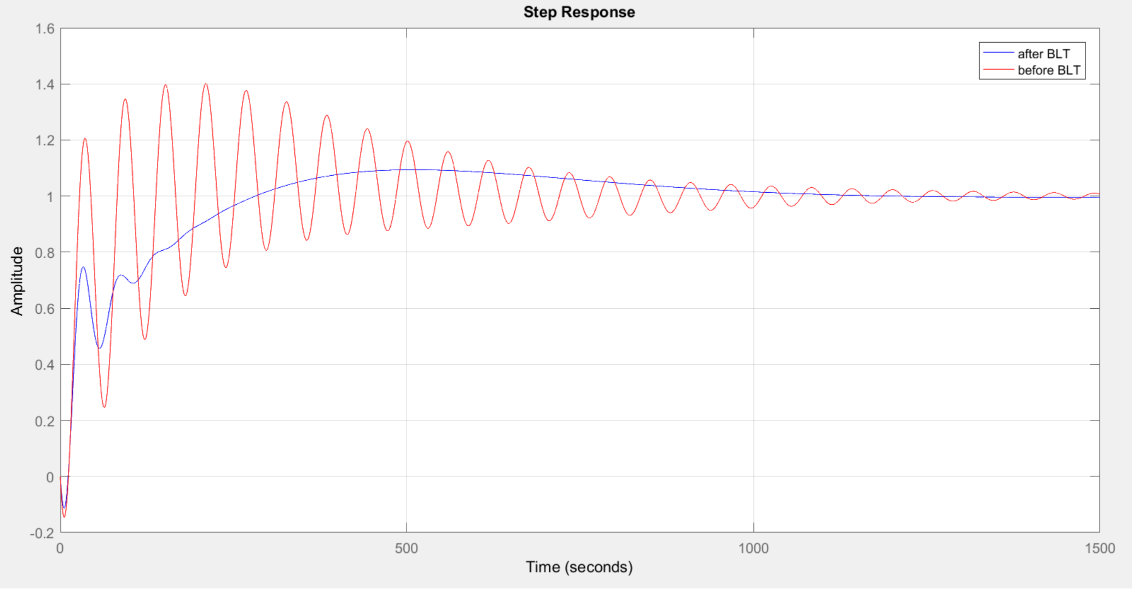


Figure IV.10 : The step response of the second loop (before and after BLT)

The new PI controller value for The Bottom Reflux Temperature loop using the BLT method will be:

$$gc3 = \frac{0.53s + 0.073}{s}$$

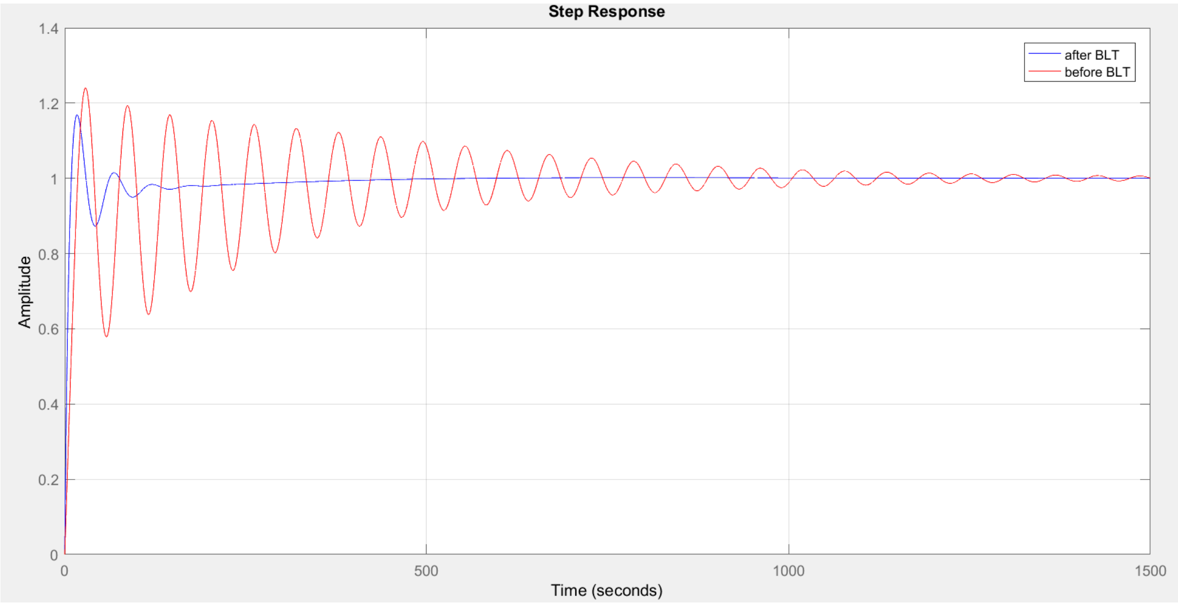


Figure IV.11 : The step response of the third loop (before and after BLT)

Results Discussion:

Based on the simulation results of the responses of each loop when the other loops are closed before and after BLT, we observe that the BLT method provides responses without any oscillations caused by the propagation of disturbances from one loop to the others.

In Conclusion the step response graphs of the Shell heavy oil fractionator (SHOF) before and after applying the Biggest Log Tuning (BLT) method show significant improvements in system performance. Before BLT, all responses exhibit pronounced oscillations that persist for a long duration before stabilizing. After BLT, the responses are much smoother with minimal oscillations, leading to quicker stabilization at the reference point. This indicates that the BLT method effectively mitigates the interaction effects and improves the overall performance of the control system, reducing oscillations and allowing faster stabilization, thereby minimizing the propagation of disturbances across the control loops.

IV.4.1 The effect of loops between each other before and after BLT:

To show the effect of BLT on the interactions present in the system, we show the response of the excitation of the 1st loop on the 2nd and the excitation of the 2nd loop on the 1st loop, the response of the excitation of the 1st loop on the 3rd and the excitation of the 3rd loop on the 1st loop, and the response of the excitation of the 2nd loop on the 3rd and the excitation of the 3rd loop on the 2nd loop before and after BLT.

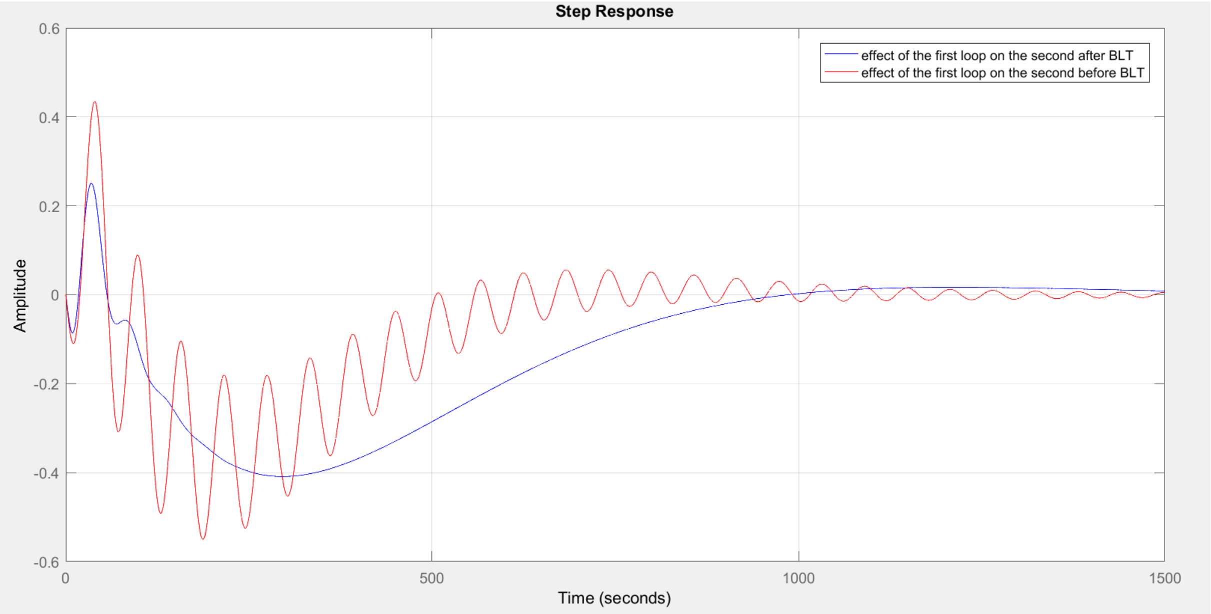


Figure IV.12 : The effect of the first loop on the second loop, before and after applying the BLT.

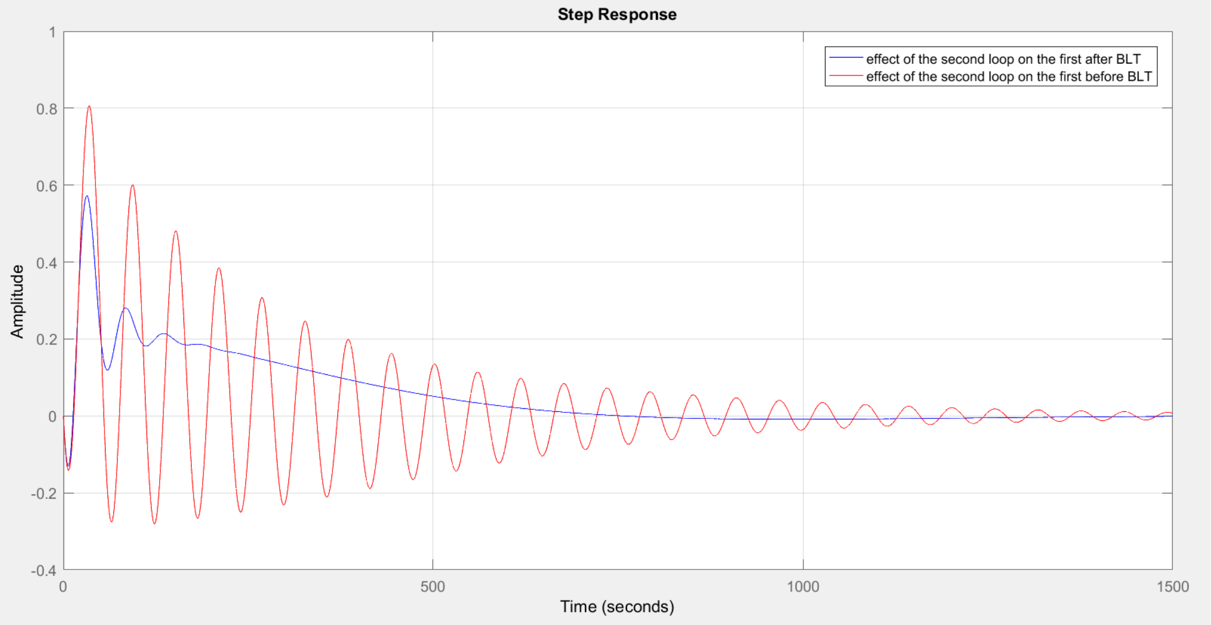


Figure IV.13 : The effect of the second loop on the first loop, before and after applying the BLT.

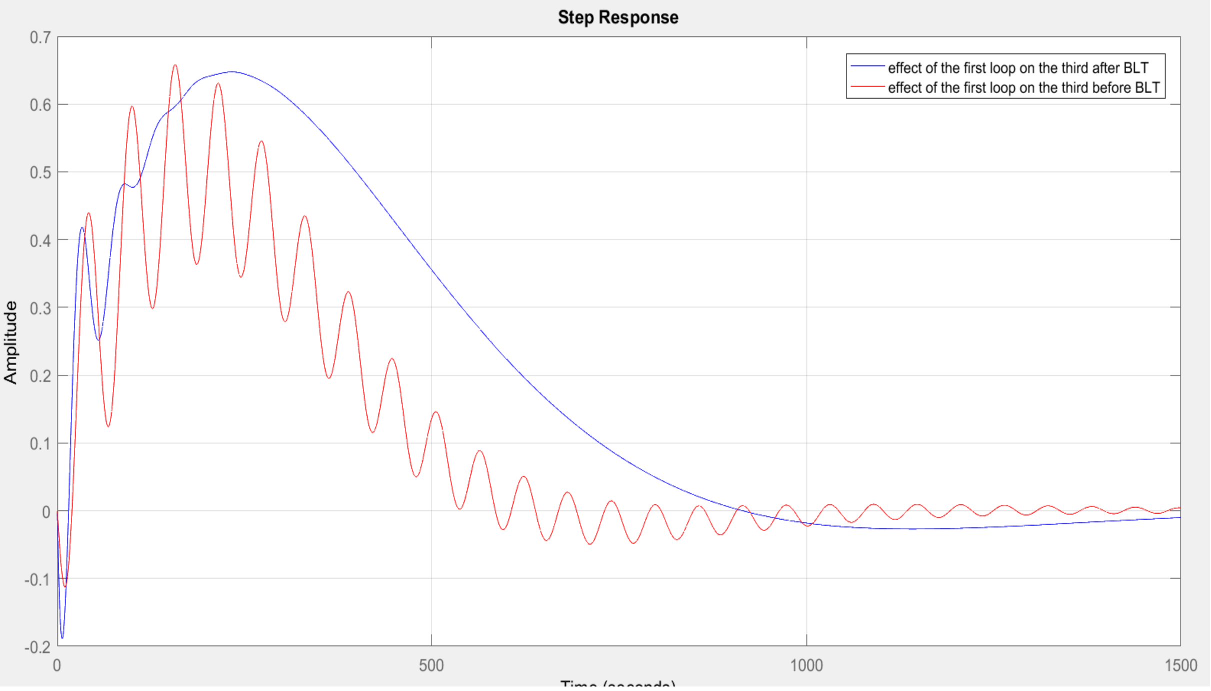


Figure IV.14 : The effect of the first loop on the third loop, before and after applying the BLT.

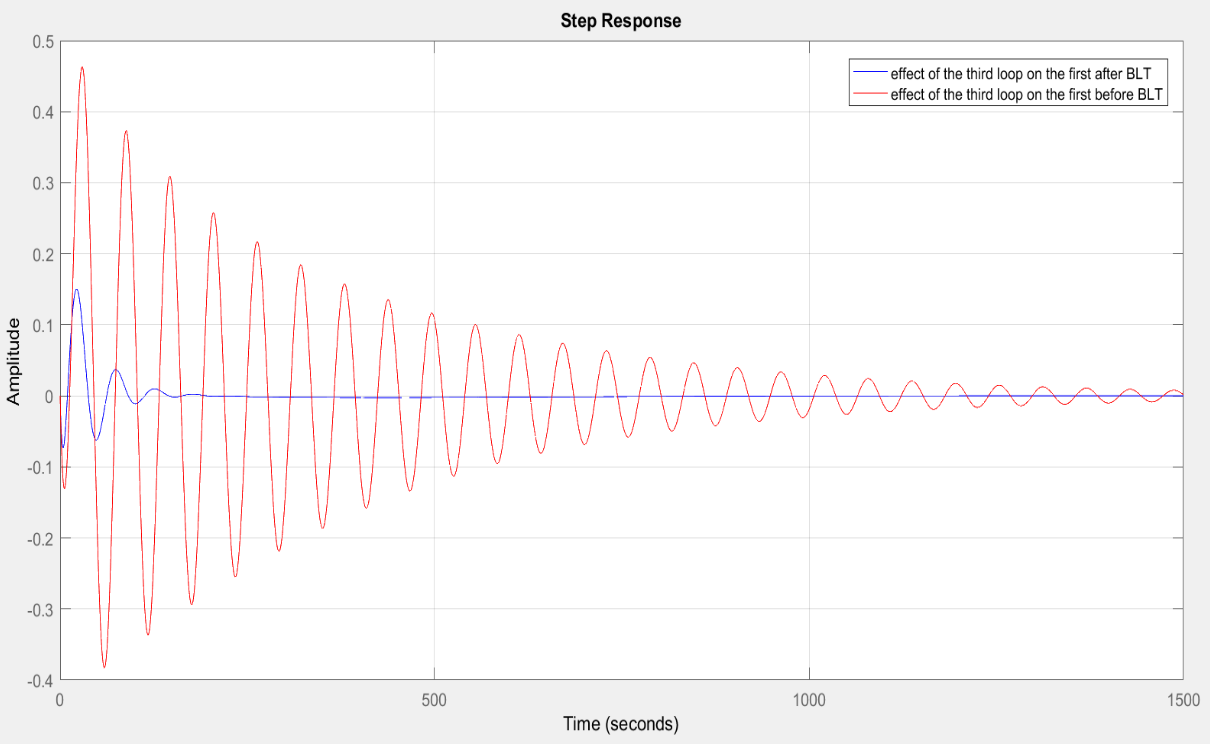


Figure IV.15 : The effect of the third loop on the first loop, before and after applying the BLT.

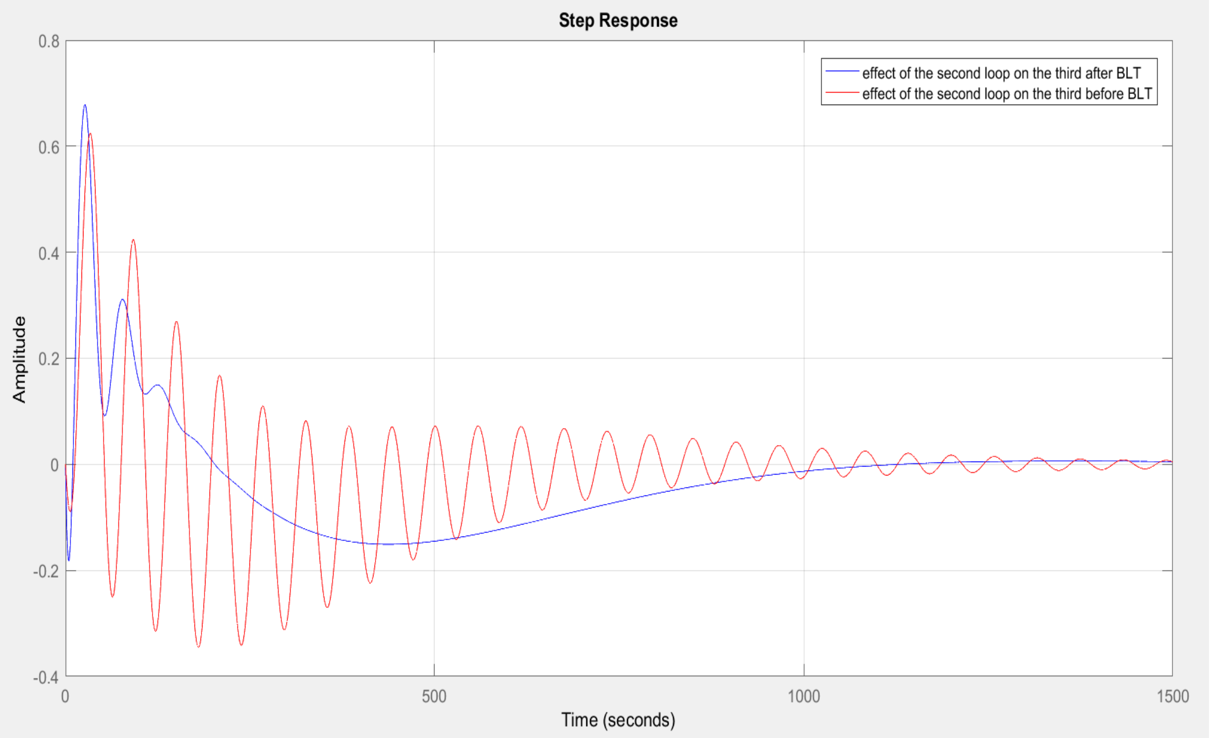


Figure IV.16 : The effect of the second loop on the third loop, before and after applying the BLT.

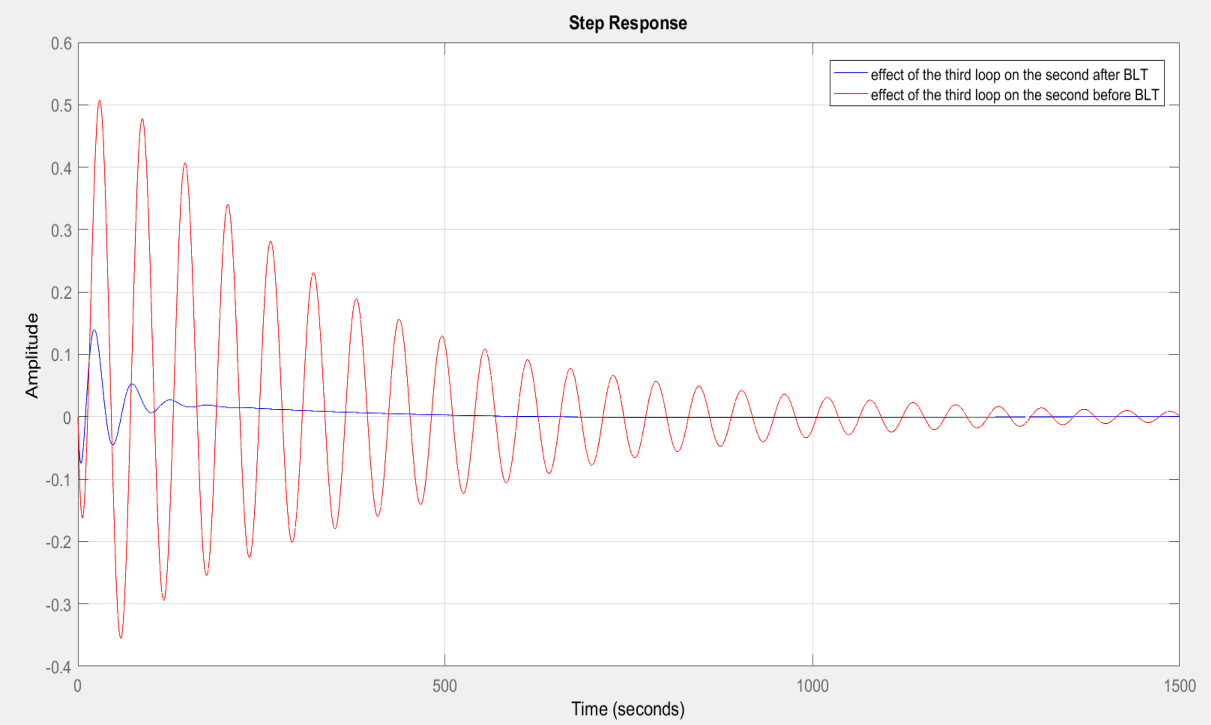


Figure IV.17 : The effect of the third loop on the second loop, before and after applying the BLT.

Results Discussion:

We can see from the figures that the results of the effect of each loop on the others, before and after applying the BLT, clearly demonstrate that the BLT has significantly weakened the interactions present in the system.

IV. Conclusion:

In this chapter, we provided an overview of the simulation and results of the control of the SHOF. We began by calculating the he Direct Interaction Analysis Method (RGA) to determine the best loop pairing configuration, Next, we applied the Dynamic relative magnitude array (DRMA) to study the interactions between the loops. Finally, we implemented the Biggest Log Tuning Method (BLT) as a multivariable control method for the control of the SHOF, which effectively weakened the interactions within the system. This is demonstrated by the simulation results, which show the effect of each loop on the others before and after applying the BLT.

General conclusion

General conclusion

In this work, we have explored the multivariable control of the Shell Heavy Oil Fractionator (SHOF), a crucial component in the petroleum refining process. Our work focused on understanding the interactions between the loops of SHOF in order to develop an effective control strategy, and enhancing the overall performance and stability of the system.

We began with a comprehensive study of the SHOF, detailing the fundamentals of the separation process and presenting a mathematical model in the form of a transfer matrix. Then we examined the multivariable system and multivariable control. We provided an in-depth analysis of the interaction phenomenon, discussing its implications for system performance and control strategy design. Various methods for analyzing interactions were conducted, the Relative Gain Array (RGA) as a direct analysis method and the Dynamic Relative Magnitude Array (DRMA) as a dynamic analysis method. With the RGA method, we determined the best loop paring configuration between the process inputs and outputs, while with the (DRMA), we determined the actual level of interaction between the loops of this SHOF.

The application of the DRMA method provided results on the interaction levels between the three control loops of the SHOF, leading to the use of a multivariable control method, the Biggest Log Tuning (BLT) method.

In conclusion, our study has highlighted that the understanding of interactions in multivariable control systems facilitate the task of choosing the control method for this multivariable system. By comprehensively analyzing the interactions we were able to determine the optimal control strategy. This approach ensures that the chosen control method effectively addresses the unique challenges posed by the interactions within the system, enhancing the stability, efficiency, and overall performance of the fractionator.

References

[1] A.KHELASSI, *« Cours sur l'Automatisation des Systèmes Industriels »*. UMBB-Boumerdes, 2022.

[2] N. Jensen and D. G. Fisher, *« Interaction Analysis in Multivariable Control Systems »*., June 1986.

[3] A. OGUNNAIKE and W. HARMON RAY, *« process dynamics, modeling, and control »*. New York, 1994.

[4] H. HUANG, Ohshima, and Hashimoto, *« Dynamic interaction and multi-loop system design »*., 1994.

[5] J. M. MACIEJOWSKI, *« Multivariable feedback design »*. UK, 1989.

[6] Shead, L.R.E., Anastassakis, C.G. Rossiter, and J.A., *« Steady-state Operability of Multivariable Non-square Systems: Application to Model Predictive Control (MPC) of the Shell Heavy Oil Fractionator (SHOF) »*., 2007.

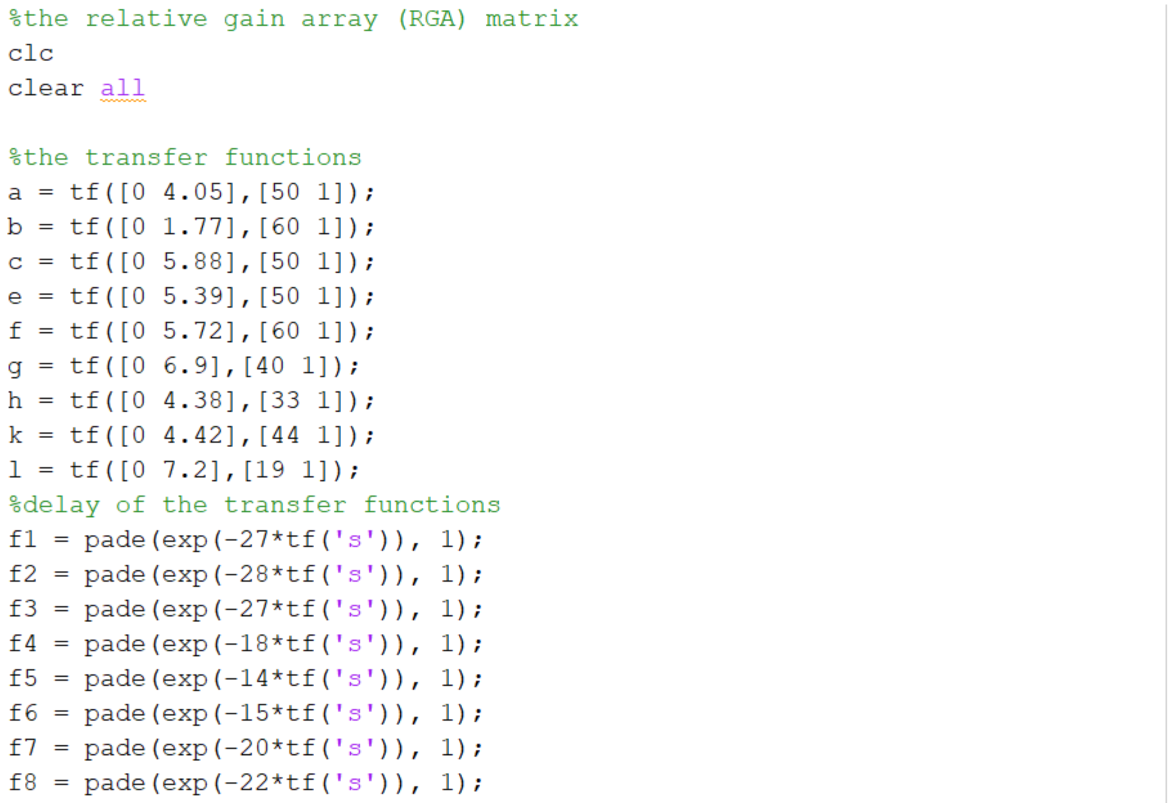
[7] L. LUYBEN WILLIAM and WILLIAM L. LUYBEN, *« PROCESS MODELING, SIMULATION AND CONTROL FOR CHEMICAL ENGINEERS SECOND EDITION »*., 1996.

[8] Witcher, M. F., and MacAvoy T. J, *« Interacting Control Systems : Steady State and Dynamic Measurement of Interaction »*, 16th ed., 1977.

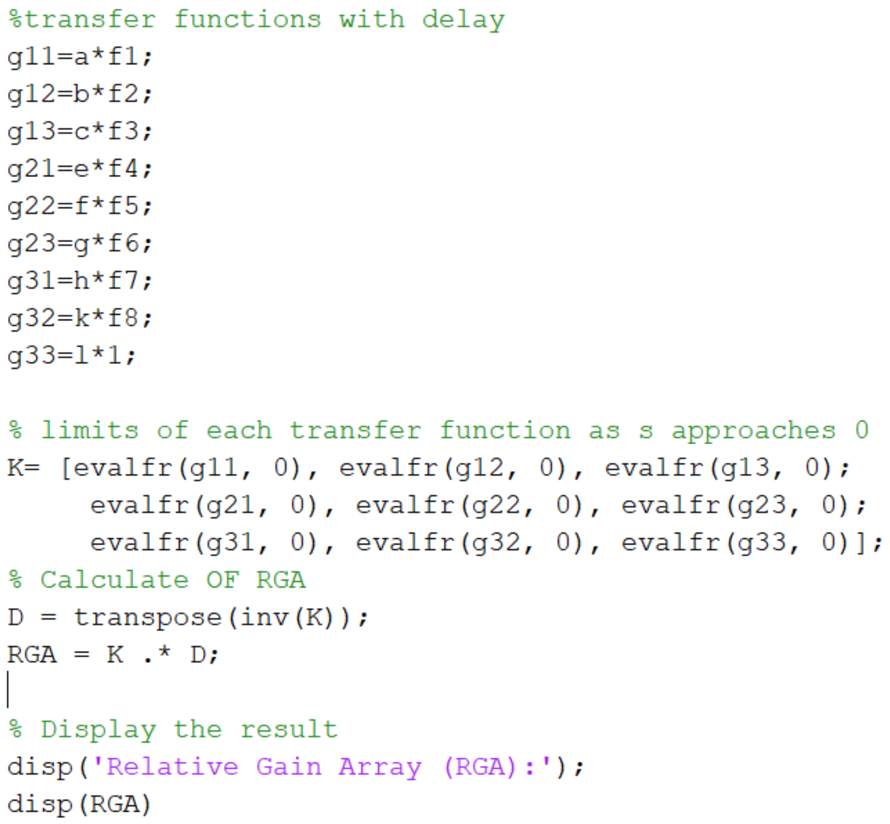
Appendix

Appendix A

The MATLAB Program of the Relative gain array (RGA):



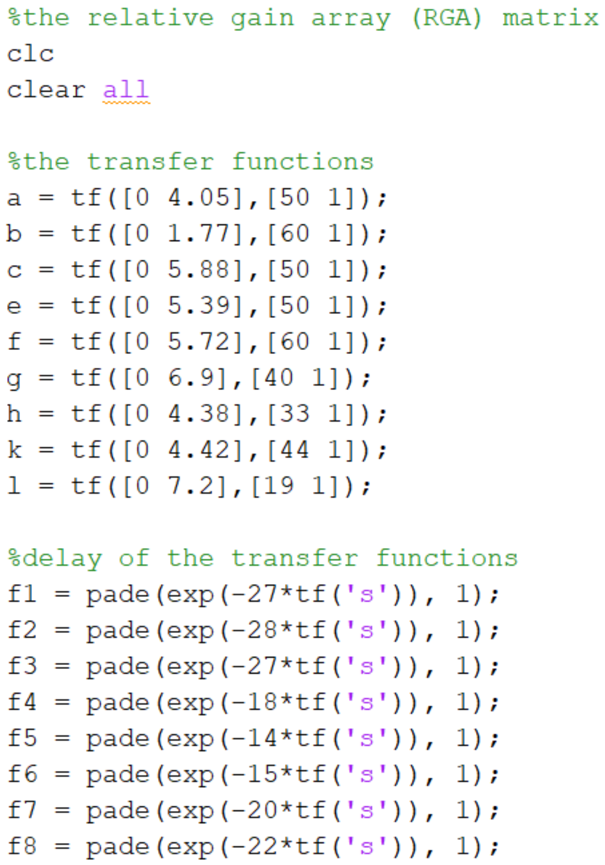
```
%the relative gain array (RGA) matrix
clc
clear all

%the transfer functions
a = tf([0 4.05],[50 1]);
b = tf([0 1.77],[60 1]);
c = tf([0 5.88],[50 1]);
e = tf([0 5.39],[50 1]);
f = tf([0 5.72],[60 1]);
g = tf([0 6.9],[40 1]);
h = tf([0 4.38],[33 1]);
k = tf([0 4.42],[44 1]);
l = tf([0 7.2],[19 1]);
%delay of the transfer functions
f1 = pade(exp(-27*tf('s')), 1);
f2 = pade(exp(-28*tf('s')), 1);
f3 = pade(exp(-27*tf('s')), 1);
f4 = pade(exp(-18*tf('s')), 1);
f5 = pade(exp(-14*tf('s')), 1);
f6 = pade(exp(-15*tf('s')), 1);
f7 = pade(exp(-20*tf('s')), 1);
f8 = pade(exp(-22*tf('s')), 1);
```



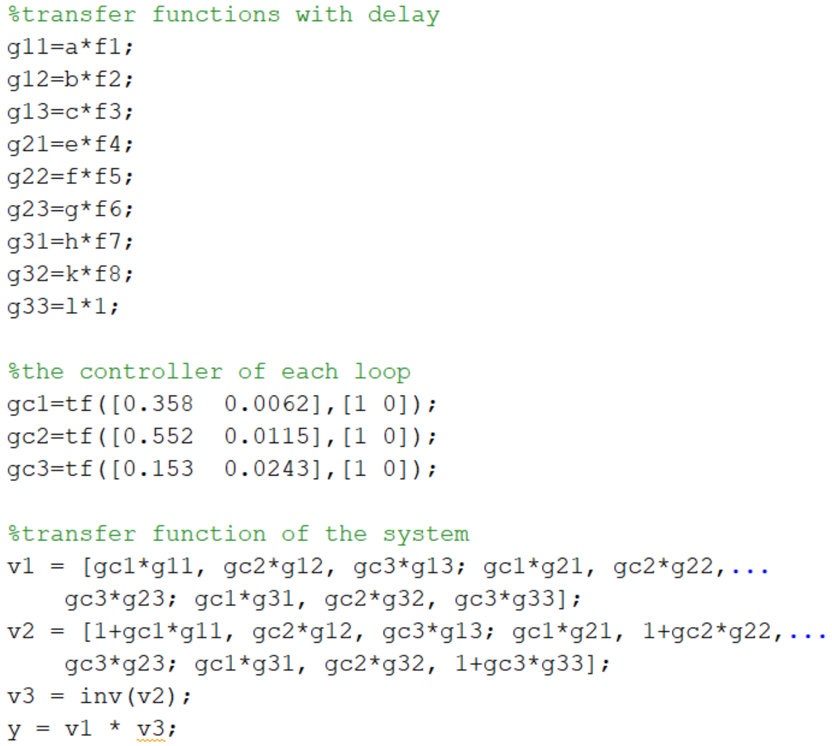
```
%transfer functions with delay
g11=a*f1;
g12=b*f2;
g13=c*f3;
g21=e*f4;
g22=f*f5;
g23=g*f6;
g31=h*f7;
g32=k*f8;
g33=l*1;

% limits of each transfer function as s approaches 0
K= [evalfr(g11, 0), evalfr(g12, 0), evalfr(g13, 0);
     evalfr(g21, 0), evalfr(g22, 0), evalfr(g23, 0);
     evalfr(g31, 0), evalfr(g32, 0), evalfr(g33, 0)];
% Calculate OF RGA
D = transpose(inv(K));
RGA = K .* D;

% Display the result
disp('Relative Gain Array (RGA):');
disp(RGA)
```

Appendix B

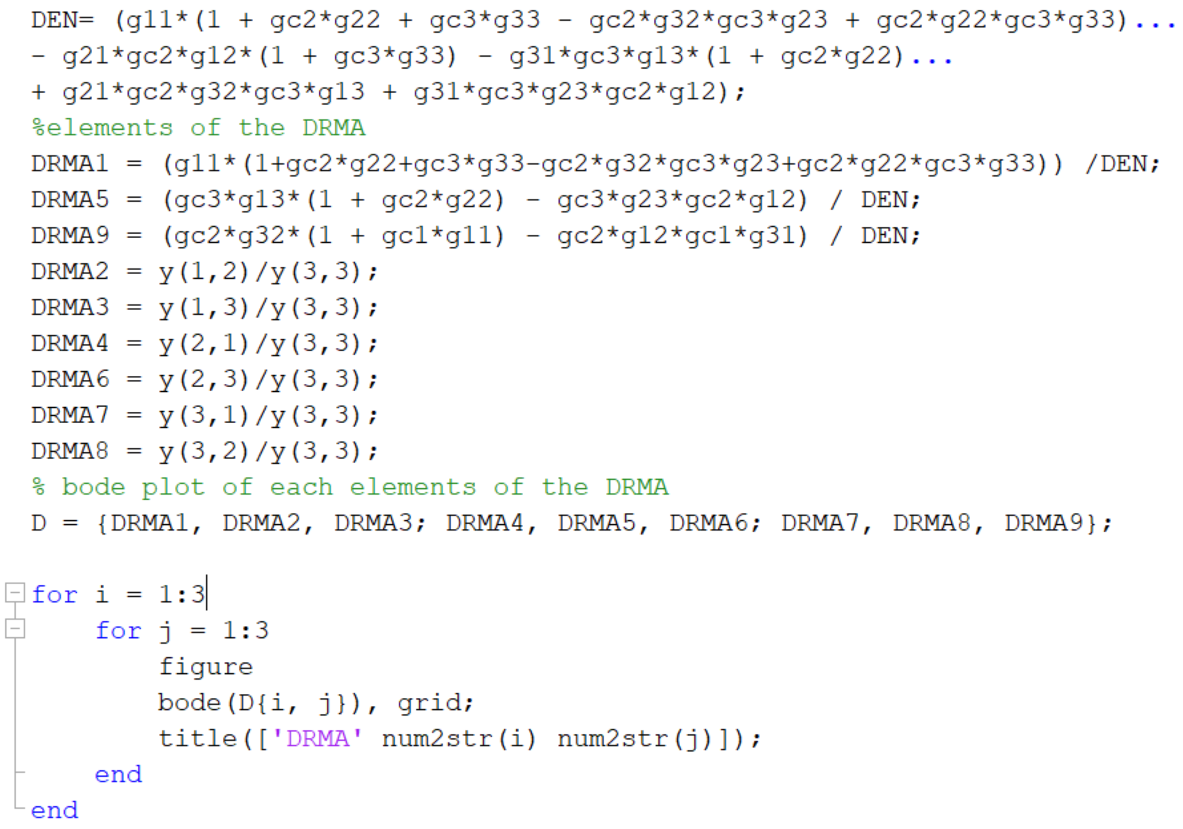
The MATLAB Program of the Dynamic relative magnitude array (DRMA):



```
%the relative gain array (RGA) matrix
clc
clear all

%the transfer functions
a = tf([0 4.05],[50 1]);
b = tf([0 1.77],[60 1]);
c = tf([0 5.88],[50 1]);
e = tf([0 5.39],[50 1]);
f = tf([0 5.72],[60 1]);
g = tf([0 6.9],[40 1]);
h = tf([0 4.38],[33 1]);
k = tf([0 4.42],[44 1]);
l = tf([0 7.2],[19 1]);

%delay of the transfer functions
f1 = pade(exp(-27*tf('s')), 1);
f2 = pade(exp(-28*tf('s')), 1);
f3 = pade(exp(-27*tf('s')), 1);
f4 = pade(exp(-18*tf('s')), 1);
f5 = pade(exp(-14*tf('s')), 1);
f6 = pade(exp(-15*tf('s')), 1);
f7 = pade(exp(-20*tf('s')), 1);
f8 = pade(exp(-22*tf('s')), 1);
```



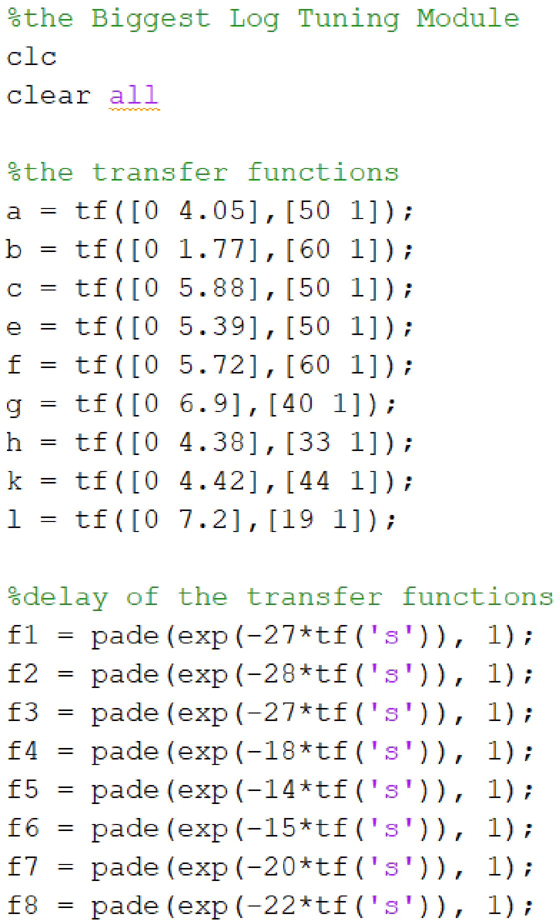
```
%transfer functions with delay
g11=a*f1;
g12=b*f2;
g13=c*f3;
g21=e*f4;
g22=f*f5;
g23=g*f6;
g31=h*f7;
g32=k*f8;
g33=l*1;

%the controller of each loop
gc1=tf([0.358  0.0062],[1 0]);
gc2=tf([0.552  0.0115],[1 0]);
gc3=tf([0.153  0.0243],[1 0]);

%transfer function of the system
v1 = [gc1*g11, gc2*g12, gc3*g13; gc1*g21, gc2*g22,...
    gc3*g23; gc1*g31, gc2*g32, gc3*g33];
v2 = [1+gc1*g11, gc2*g12, gc3*g13; gc1*g21, 1+gc2*g22,...
    gc3*g23; gc1*g31, gc2*g32, 1+gc3*g33];
v3 = inv(v2);
y = v1 * v3;
```



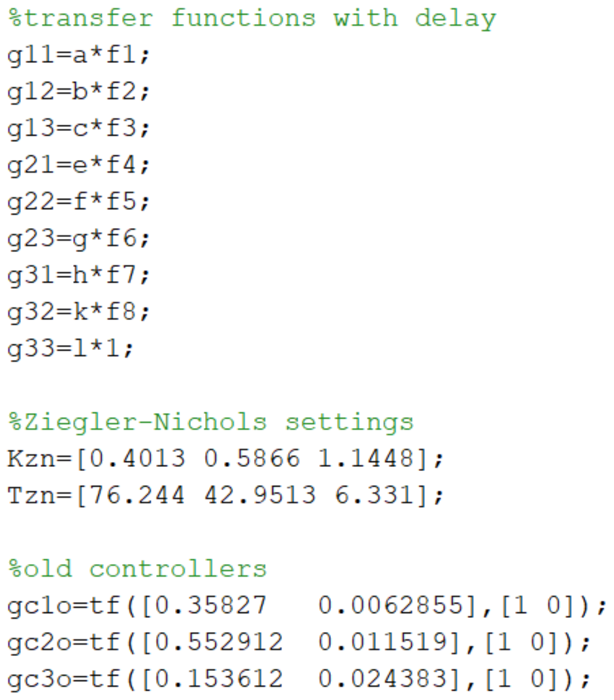
```
DEN= (g11*(1 + gc2*g22 + gc3*g33 - gc2*g32*gc3*g23 + gc2*g22*gc3*g33)...
- g21*gc2*g12*(1 + gc3*g33) - g31*gc3*g13*(1 + gc2*g22)...
+ g21*gc2*g32*gc3*g13 + g31*gc3*g23*gc2*g12);
%elements of the DRMA
DRMA1 = (g11*(1+gc2*g22+gc3*g33-gc2*g32*gc3*g23+gc2*g22*gc3*g33)) /DEN;
DRMA5 = (gc3*g13*(1 + gc2*g22) - gc3*g23*gc2*g12) / DEN;
DRMA9 = (gc2*g32*(1 + gc1*g11) - gc2*g12*gc1*g31) / DEN;
DRMA2 = y(1,2)/y(3,3);
DRMA3 = y(1,3)/y(3,3);
DRMA4 = y(2,1)/y(3,3);
DRMA6 = y(2,3)/y(3,3);
DRMA7 = y(3,1)/y(3,3);
DRMA8 = y(3,2)/y(3,3);
% bode plot of each elements of the DRMA
D = {DRMA1, DRMA2, DRMA3; DRMA4, DRMA5, DRMA6; DRMA7, DRMA8, DRMA9};

for i = 1:3
    for j = 1:3
        figure
        bode(D{i, j}), grid;
        title(['DRMA' num2str(i) num2str(j)]);
    end
end
```

Appendix C

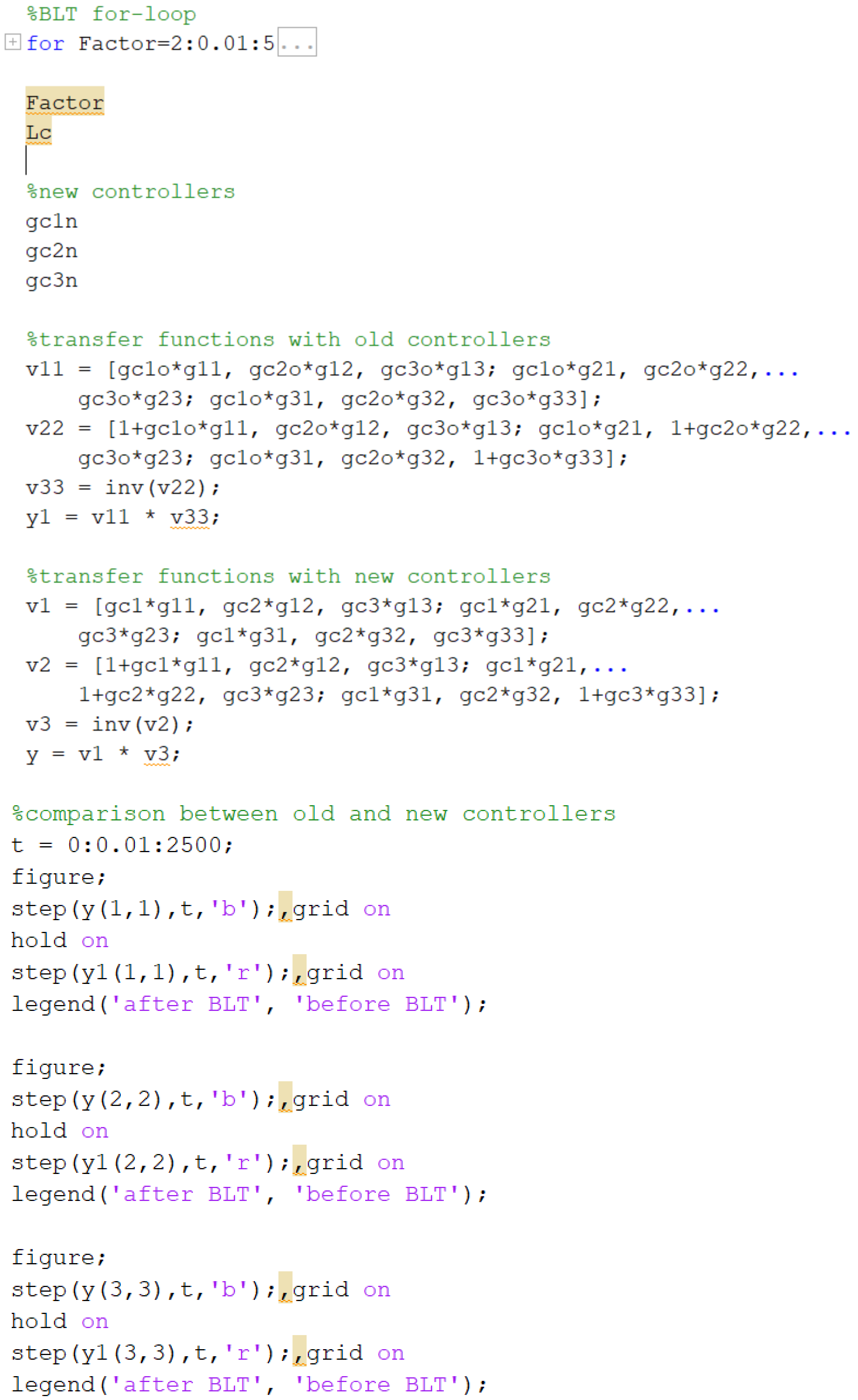
The MATLAB Program of the Biggest log-modulus tuning method (BLT):



```
%the Biggest Log Tuning Module
clc
clear all

%the transfer functions
a = tf([0 4.05],[50 1]);
b = tf([0 1.77],[60 1]);
c = tf([0 5.88],[50 1]);
e = tf([0 5.39],[50 1]);
f = tf([0 5.72],[60 1]);
g = tf([0 6.9],[40 1]);
h = tf([0 4.38],[33 1]);
k = tf([0 4.42],[44 1]);
l = tf([0 7.2],[19 1]);

%delay of the transfer functions
f1 = pade(exp(-27*tf('s')), 1);
f2 = pade(exp(-28*tf('s')), 1);
f3 = pade(exp(-27*tf('s')), 1);
f4 = pade(exp(-18*tf('s')), 1);
f5 = pade(exp(-14*tf('s')), 1);
f6 = pade(exp(-15*tf('s')), 1);
f7 = pade(exp(-20*tf('s')), 1);
f8 = pade(exp(-22*tf('s')), 1);
```



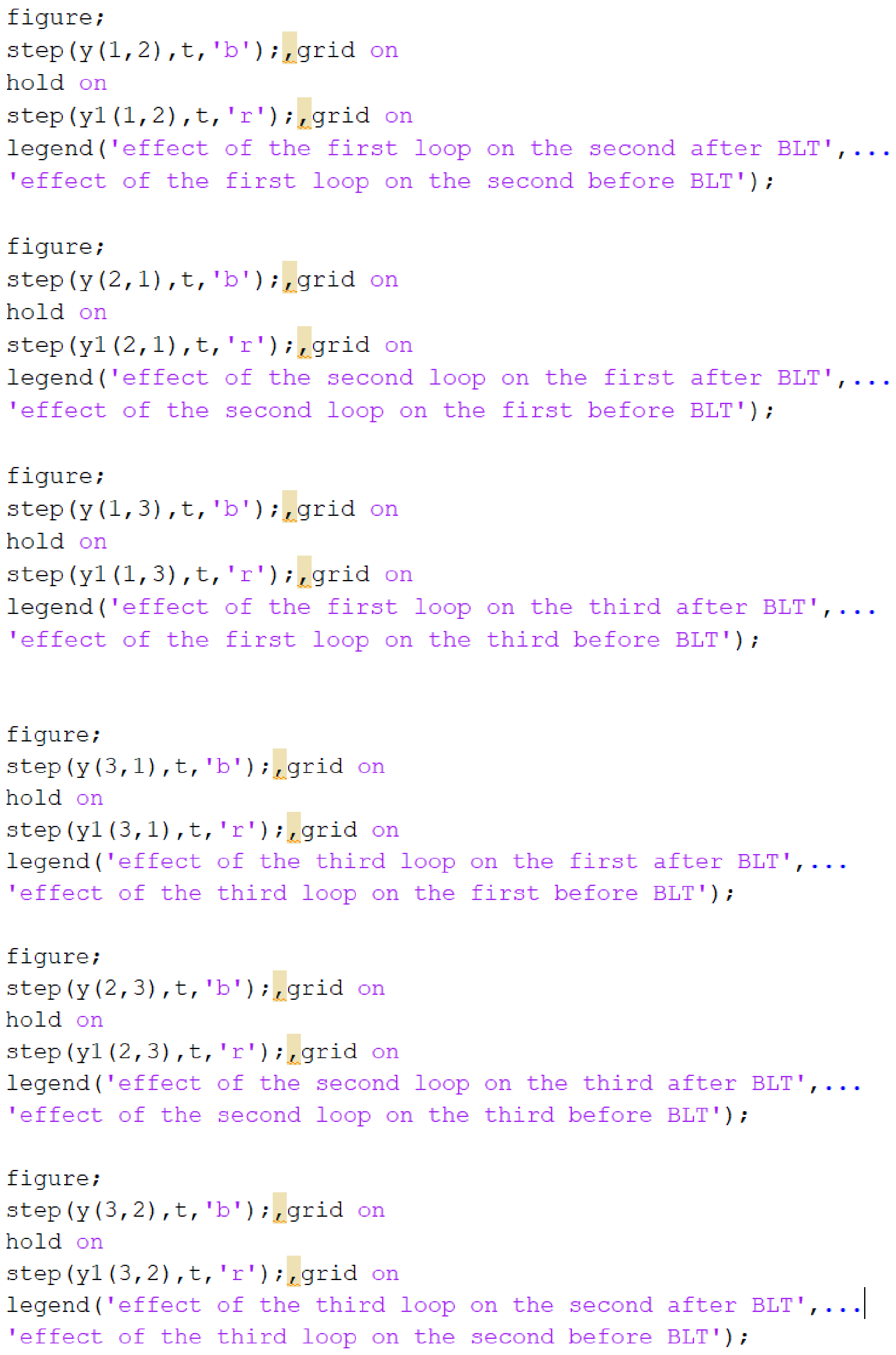
```
%transfer functions with delay
g11=a*f1;
g12=b*f2;
g13=c*f3;
g21=e*f4;
g22=f*f5;
g23=g*f6;
g31=h*f7;
g32=k*f8;
g33=l*1;

%Ziegler-Nichols settings
Kzn=[0.4013 0.5866 1.1448];
Tzn=[76.244 42.9513 6.331];

%old controllers
gc1o=tf([0.35827   0.0062855],[1 0]);
gc2o=tf([0.552912  0.011519],[1 0]);
gc3o=tf([0.153612  0.024383],[1 0]);
```



```
%BLT for-loop
for Factor=2:0.01:5...

Factor
Lc

%new controllers
gc1n
gc2n
gc3n

%transfer functions with old controllers
v11 = [gc1o*g11, gc2o*g12, gc3o*g13; gc1o*g21, gc2o*g22,...
    gc3o*g23; gc1o*g31, gc2o*g32, gc3o*g33];
v22 = [1+gc1o*g11, gc2o*g12, gc3o*g13; gc1o*g21, 1+gc2o*g22,...
    gc3o*g23; gc1o*g31, gc2o*g32, 1+gc3o*g33];
v33 = inv(v22);
y1 = v11 * v33;

%transfer functions with new controllers
v1 = [gc1*g11, gc2*g12, gc3*g13; gc1*g21, gc2*g22,...
    gc3*g23; gc1*g31, gc2*g32, gc3*g33];
v2 = [1+gc1*g11, gc2*g12, gc3*g13; gc1*g21,...
    1+gc2*g22, gc3*g23; gc1*g31, gc2*g32, 1+gc3*g33];
v3 = inv(v2);
y = v1 * v3;

%comparison between old and new controllers
t = 0:0.01:2500;
figure;
step(y(1,1),t,'b');,grid on
hold on
step(y1(1,1),t,'r');,grid on
legend('after BLT', 'before BLT');

figure;
step(y(2,2),t,'b');,grid on
hold on
step(y1(2,2),t,'r');,grid on
legend('after BLT', 'before BLT');

figure;
step(y(3,3),t,'b');,grid on
hold on
step(y1(3,3),t,'r');,grid on
legend('after BLT', 'before BLT');
```

Appendix D

The MATLAB Program of the effect between loops:



```
figure;
step(y(1,2),t,'b');,grid on
hold on
step(y1(1,2),t,'r');,grid on
legend('effect of the first loop on the second after BLT',...
'effect of the first loop on the second before BLT');

figure;
step(y(2,1),t,'b');,grid on
hold on
step(y1(2,1),t,'r');,grid on
legend('effect of the second loop on the first after BLT',...
'effect of the second loop on the first before BLT');

figure;
step(y(1,3),t,'b');,grid on
hold on
step(y1(1,3),t,'r');,grid on
legend('effect of the first loop on the third after BLT',...
'effect of the first loop on the third before BLT');


figure;
step(y(3,1),t,'b');,grid on
hold on
step(y1(3,1),t,'r');,grid on
legend('effect of the third loop on the first after BLT',...
'effect of the third loop on the first before BLT');

figure;
step(y(2,3),t,'b');,grid on
hold on
step(y1(2,3),t,'r');,grid on
legend('effect of the second loop on the third after BLT',...
'effect of the second loop on the third before BLT');

figure;
step(y(3,2),t,'b');,grid on
hold on
step(y1(3,2),t,'r');,grid on
legend('effect of the third loop on the second after BLT',...
'effect of the third loop on the second before BLT');
```