People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdès

Institute of Electrical and Electronic Engineering

Department of Control and Power Engineering

Project Report Presented in Partial Fulfilment of

the Requirements of the Degree of

'MASTER'

In Control Engineering

Title:

Observer-based Active Fault Tolerant Control of a Three Tank system (DTS-200)

Presented By:

- **TAIB Mohamed Amine**

Supervisor:

Pr. Abdelmalek KOUADRI
Dr. Tewfik YOUSSEF (co-supervisor)

Registration number: ……./2022

Abstract

Multi Input Multi Output (MIMO) systems theory is concerned with complex dynamical systems with multiple input (control) variables u_r that regulate multiple output variables y_m. In general, a MIMO system is a collection of independent or interconnected Single Input Single Output (SISO) subsystems that can be conditionally represented. The system's separate subsystems are referred to as direct (or forward path) channels, and each one establishes a connection between the appropriate scalar input and output.

This project deals with nonlinear multivariable system control; the work will begin with linearizing the model around a suitable equilibrium point and extracting the system's transfer function matrix representation. The RGA will be used to select the optimal pairings, and two observers will be designed to estimate the states and decouple disturbance and noise. The system will be influenced by two types of faults under the scope of Fault Tolerant Control: actuator faults and sensor faults, both abrupt and random faults are estimated. Finally, P and PI controllers are used to design the system's regulation and tracking mode responses in order to improve system performance, particularly settling time.

A simulation under Matlab Simulink will be carried on to simulate and control the system.

Key words: MIMO systems, linearization, observers, fault tolerant control, estimation, regulation, tracking.

Acknowledgment

I would like to express my sincere gratitude to my primary supervisor **Pr. Abdelmalek KOUADRI** for his support, guidance, and for allowing me to duly embrace this project under his auspices despite his busy agenda.

In addition, I would like to acknowledge **Dr. Tewfik YOUSSEF** who has been an ideal co-supervisor and mentor. I am proud of, and grateful for, my time working with them. I am thankful for their confidence and the freedom they gave me throughout the course of this work.

I am deeply grateful to the members of the jury for taking the time to read and analyze this project report. My warmest thanks go to the teachers of IGEE for the devotion they have shown and the knowledge they have passed to me.

Dedication

On this special day of my graduation, I want to thank my family members, one by one, starting by my dearest parents; Thank you father, thank you mother for all the support you have provided me since the first step I took on this hard path, your son could not have done it without you. May God bless you both. Also, my two beloved sisters, you have done so much for me.

I would also thank my friends, who are considered as my second family and a gift from God, namely, my best friend Abderraouf SENHADJI, the long list of friends I had the pleasure of meeting during my years in college: Assia BOUGRID, Hamza BENRABAH, Kenza BENSLIMANE, Mahfod HEMISSI, Meriem BENDAHMAN, Meriem CHAIB, Moncef MAOUCHE, Nesrine CHIKER, Nezha CHOUIDER, Sara ALILAT, Wassim OULDACHE, Youcef BENAZIZA. Thanks for being a part of my journey and making it so special and beautiful.

Contents

List of Figures

List of Tables

List of Abbreviations

AFTCS Active Fault Tolerant Control Systems

DCS Digital Control Signals

DNA Direct Nyquist Array

DRGA Dynamic Relative Gain Array

DTS Dosing Tank Stations

FD Fault Diagnosis

FE Fault Estimation

FR Frequancy Response

FTCS Fault Tolerant Control Systems

HIIA Hankel Intersection Index Array

HSV Hankel Singular Value

I/O Input Output

INA Inverse Nyquist Array

LTI Linear Time Invariant

MIMO Multiple Input Multiple Output

NI Niederlinski Index

NSRGA Nonsquare Relative Gain Array

PFTCS Passive Fault Tolerant Control Systems

PID Proportional-Integral-Derivative

PM Participation Matrix

RGA Relative Gain Array

SISO Single Input Single Output

SVD Singular Value Decomposition

TITO Two Input Two Output

UIO Unknown Input Observer

General introduction

Automation is at the heart of the technology that allows machines to perform tasks with minimal human intervention. The aim is to focus the labor on sophisticated creative challenges; instead of mundane, repetitive tasks that are performed in a better, faster, and more efficient way by machines, thereby making it a revolutionary domain for the development and the better performance of a production cycle.

In industrial production, the use of so-called MIMO systems, which are machines with the ability to perform multiple tasks simultaneously, i.e., a single set of machinery is designed to perform several tasks simultaneously.

The controlling process of MIMO systems often faces a challenging problem due to the cross-coupling effects that occur between the inputs and the outputs; thus, a single input may influence all system outputs. Boksenbom and Hood first mentioned this problem in 1950 [1]. However, coupling was only treated as a complicated design idea and had not been widely explored at that instant. With the rapid development of manufacturing, methodologies aiming at eliminating or decreasing multi-loop interaction issues have received enough attention in the past decades.

In the 1980s, Professor Waller mentioned that one of the subjects of great research activity in chemical process control in the U.S. today is interaction analysis, in which coupling between inputs and outputs in MIMO systems is studied [1]. Such a problem can be solved using decouplers, which allow achieving efficient control by eliminating such cross-coupling effects. If properly designed, decouplers permit deriving control inputs for fully-actuated and over-actuated MIMO systems to attain reference tracking of the individual outputs. Nevertheless, the reference tracking for individual outputs is almost unachievable in under-actuated MIMO systems with decoupling control methods.

In recent decades, the frequent use of MIMO systems led to the emergence of decoupling control algorithms study. The main idea of the decoupling algorithm proposed by Boksenbom and Hood is to make the overall closed-loop transfer function of the controlled MIMO system diagonal. So far, this is still the primary solution to the coupling problem. Some other remarkable contributions have been made based on this idea. For example Mesarovic divided the controlled systems with identical inputs and outputs into two different categories, i.e., P-canonical and V-canonical systems based on system transfer function; correspondingly. Sonquist and Morgan proposed a state space approach of decoupling control in a necessary and sufficient condition of the solvability of square system decoupling problem based on state-space was put forward by Falb and Wolovich. Then, the equivalent condition for transfer function expressed system was obtained by Gilbert [1].

Within the general framework of the multi-loop control of multi-variable systems, considerable attention will be given to the analysis of interactions. In this perspective, it is most likely to seek for system compensation so that:

- Each input affects only one output.

- The disturbance on a single output, with zero inputs, only affects this same output.

This report is divided into the following sections:

Chapter 1: it introduces multivariate systems and their main characteristics, such as representation, frequency response, directions in multivariable systems, and singular value decomposition, as well as the pivotal concept of stability in control theory.

Chapter 2: it covers the important notions on the existing interactions phenomenon in multivariable systems which causes a lot of problems for control. For a good analysis, couplings must be measured using well defined interactions measurement techniques, then the system is decoupled using decoupling algorithms. This choices are essential and decisive for the desired performance. Both approaches are presented, in this chapter, with illustration examples.

Chapter 3: it discusses the issue of component failure and proposes a solution, namely the concept of fault-tolerant control, to reduce the impact of faults. It also addresses the problem of measurements in a system where some parameters are difficult to measure, in which case the estimation approach for states and faults using observers is followed. Correction methods are a key area of analyzing multivariable systems; it is critical to comprehend the specifications. To address this, design compensators are presented with the goal of improving system performance characteristics.

Chapter 4: it presents a simulation of a three-tank hydraulic stand. This chapter is an important part of my research since it attempts to evaluate the practical relevance of the methodologies discussed in the previous chapters. Simulink software is used for the simulation.

Finally, this thesis includes a general conclusion where all the results obtained are summarized and discussed in details,

Chapter 1

Generalities on MIMO systems

1.1 Introduction

In industrial automation, the most used systems are usually multivariate at the strategic variable level. Such systems have specific characteristics and are analyzed differently.

Controlling these systems is a fundamental object in the subject of automatic control, and it necessitates a methodical approach to attain the desired results. In this regard, numerous control techniques have been developed with the goal of minimizing the phenomena of interactions between the system's variables.

This chapter provides an overview on multivariable systems, their properties, and how they can be represented.

1.2 Multivariable systems

1.2.1 Definition

Multiple variable systems are simply defined as systems with more than one input $u = (u_1, u_2, u_3, ..., u_r)$ and/or more than one output $y = (y_1, y_2, y_3, ..., y_m)$; where each output is influenced by more than one input, that is, a change in a single input may cause changes in many outputs, such that:

- u: is the input vector, with dimension $(r \times 1)$.
- y: is the output vector, with dimension $(m \times 1)$.
- r: is the number of inputs.
- m: is the number of outputs.

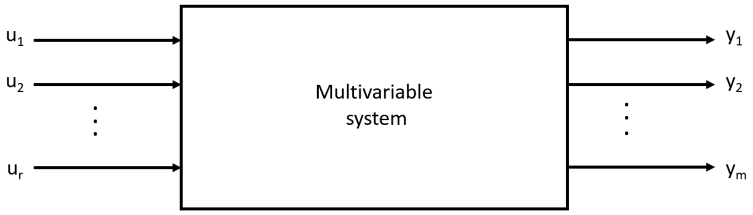


Figure 1.1: Multivariable system.

1.2.2 Representation

Multiple Inputs Multiple Outputs (MIMO) systems can be represented in more than one form:

1.2.2.1 State-variable Form

A state-variable representation of a system consists of a set of n first-order differential equations and an algebraic output equation:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned} \tag{1.1}$$

Such that:

- f: is function matrix.
- g: is function matrix.

The full form of f and g is shown below:

$$f(x,u) = \begin{bmatrix} f_1(x_1,\dots,x_n,u_1,\dots,u_r) \\ f_2(x_1,\dots,x_n,u_1,\dots,u_r) \\ \vdots \\ f_n(x_1,\dots,x_n,u_1,\dots,u_r) \end{bmatrix}$$

$$g(x,u) = \begin{bmatrix} g_1(x_1,\dots,x_n,u_1,\dots,u_r) \\ g_2(x_1,\dots,x_n,u_1,\dots,u_r) \\ \vdots \\ g_m(x_1,\dots,x_n,u_1,\dots,u_r) \end{bmatrix}$$

The expended form for Eq 1.1 is of the form:

$$\begin{cases} \dot{x}_1 = f_1(x_1,\dots,x_n,u_1,\dots,u_r) \\ \dot{x}_2 = f_2(x_1,\dots,x_n,u_1,\dots,u_r) \\ \vdots \\ \dot{x}_n = f_n(x_1,\dots,x_n,u_1,\dots,u_r) \end{cases} \tag{1.2}$$

$$\begin{cases} y_1 = g_1(x_1,\dots,x_n,u_1,\dots,u_r) \\ y_2 = g_2(x_1,\dots,x_n,u_1,\dots,u_r) \\ \vdots \\ y_m = g_m(x_1,\dots,x_n,u_1,\dots,u_r) \end{cases} \tag{1.3}$$

Such that: $x = [x_1,\dots,x_n]^T$ is the state vector. u and y are called the control input and the system output respectively.

1.2.2.2 State-space representation

Dynamic systems can be represented with differential equations where the behavior of a system can be described as a function of its current state and an external input. The state-space representation is simply a repackaging of the high order differential equations into a set of first order differential equations that focus on this relationship, this repackaging makes the system easier to analyze as it focuses on the underlined behavior of the interconnected system as well as how the system is effected by a single or multiple external inputs.

This representation offers a great advantage during simulation since the vector of derivatives is built first and then it is integrated to obtain the states, the state space representation is given by:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t) + D(t)u(t) \end{cases} \tag{1.4}$$

Such that:

- $x(t)$: is the states vector, with dimensions $(n \times 1)$.
- $y(t)$: is the outputs vector, with dimensions $(m \times 1)$.

- $u(t)$: is the control vector, with dimensions $(r \times 1)$.
- $A(t)$: is the states dynamics matrix, with dimensions $(n \times n)$.
- $B(t)$: is the inputs matrix, with dimensions $(n \times r)$.
- $C(t)$: is the output matrix, with dimensions $(m \times n)$.
- $D(t)$: is the direct transmission matrix, with dimensions $(m \times r)$.

The relation between the number of inputs and outputs and the dimensions of the matrices is demonstrated in the following illustration.

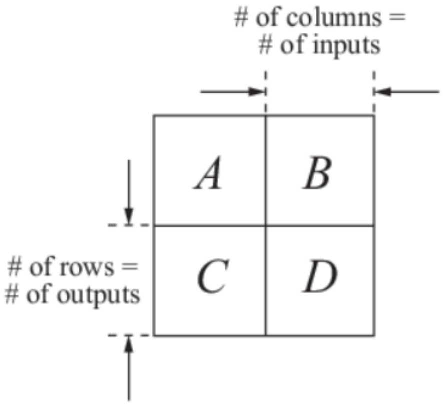


Figure 1.2: Illustration of the relation between the dimensions of matrices and the number of I/O .

For an LIT (Linear Time Invariant) system, the matrices $A(t)$, $B(t)$, $C(t)$, and $D(t)$ become constant and Eq 1.4 can be written as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{1.5}$$

Expending the state equation yields the following form:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1r} \\ b_{21} & \dots & b_{2r} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nr} \end{bmatrix}\begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \tag{1.6}$$

For the output form, the expended representation is as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & a_{mn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} d_{11} & \dots & d_{1r} \\ b_{21} & \dots & d_{2r} \\ \vdots & \ddots & \vdots \\ d_{m1} & \dots & d_{mr} \end{bmatrix}\begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \tag{1.7}$$

The derivatives of the state variables are clearly expressed in terms of the states and the inputs in the matrix-based state equations. The state vector is expressed in this manner as the direct outcome of vector integration. Fig 1.3 depicts the block diagram representation. This generic block diagram shows matrix operations from input to output in terms of the A, B, C, and D matrices, but, does not include individual variable paths.

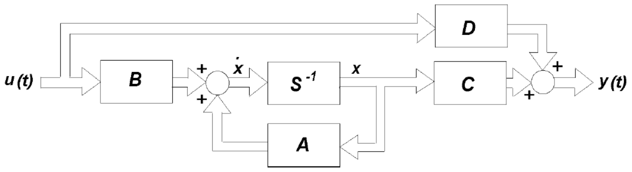


Figure 1.3: Vector block diagram for a linear system described by state-space representation

1.2.2.3 Transfer function matrix representation

It is an approach for representing a process linear model based solely on the relationship between inputs and outputs. To obtain it, for a discrete time system, the Z transfer is applied, while for continuous time equivalent is the Laplace transform, yielding the following formula:

$$Y(s) = G(s)U(s) \tag{1.8}$$

Such that:

- s : is the Laplace constant.
- $Y(s)$: is the Laplace transform of $y(t)$.
- $U(s)$: is the Laplace transform of $u(t)$.
- $G(s)$: is a matrix relating $Y(s)$ and $U(s)$.

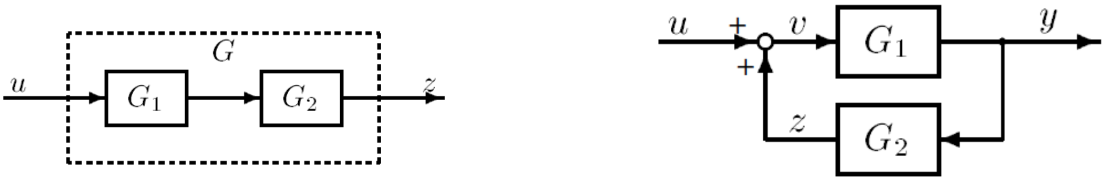
The extended form of Eq 1.8 becomes:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ \vdots \\ y_m(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \dots & g_{1r}(s) \\ g_{21}(s) & g_{22}(s) & \dots & g_{2r}(s) \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1}(s) & g_{m2}(s) & \dots & g_{mr}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_r(s) \end{bmatrix}$$

The relation between a specific output $Y_i(s)$ and a specific input $U_i(s)$ is given by:

$$Y_i(s) = \sum_{j=1}^{r} g_{ij} U_j(s) \tag{1.9}$$

For a system represented with a transfer function matrix, the representation may come in the form of a block diagram, the following two figures present the most common forms of block diagrams:



a Cascade system

b Positive feedback system

Figure 1.4: block diagrams of MIMO systems

- **Cascade rule:** for the cascade (series) interconnection of G_1 and G_2, the overall transfer function matrix is $G = G_2G_1$. The order of the transfer function matrices is reversed from how they appear.
- **Feedback rule:** with reference to the positive feedback system, the value of v is given by $v = (I - L)^{-1}u$ where $L = G_2G_1$ is the transfer function around the loop.
- **Push-though rule:** for matrices of appropriate dimension:

$$G_1(I - G_2G_1)^{-1} = (I - G_1G_2)^{-1}G_1$$

- As a general rule, one should start from the output and write down the blocks as they appear moving towards to the input. When exiting a positive feedback loop the following term is included $(I - L)^{-1}$, for a negative feedback loop the term is $(I + L)^{-1}$; where L is the loop transfer function. Parallel branches should be treated independently, and their contributions added together.

Remark:

For the transfer function matrix representation, all initial conditions are supposed to be equal to zero.

1.2.3 Transition between the representations

1.2.3.1 From state-variable to state space

Nonlinear systems are complicated because of the high dependency of the system variables on each other. Also, the nonlinear characteristic of the system abruptly changes due to some slight changes in valid parameters, thus, making nonlinear systems hard to control. One solution for that is to linearize the nonlinear system to be able to fully analyze its behavior.

- **Equilibrium point:** for a nonlinear system described by

$$\dot{x}(t) = f\big(x(t), u(t)\big) \tag{1.10}$$

A point $x_0 \in \mathbb{R}^n$ is called an equilibrium point if there exist a specific $u_0 \in \mathbb{R}^r$ such that:

$$f(x_0, u_0) = 0_n \tag{1.11}$$

- **Deviation variables:**

In control engineering the main focus is on how things change around an operating point. In solving the equations, the choice of using normal, full-valued, or variables is open, but then the solution will be complicated by the steady-state information. This information is exactly the same at the start and end of the solution, but makes the algebra really complicated. To get around this, deviation variables are normally used. Deviation variables have a value which is equal to the full value variable minus the variables nominal, steady-state, value, i.e.

$$\delta x = x - x_0 = \begin{cases} \delta x_1 & = x_1 - x_{10} \\ \delta x_2 & = x_2 - x_{20} \\ \vdots & \\ \delta x_n & = x_n - x_{n0} \end{cases} \tag{1.12}$$

$$\delta u = u - u_0 = \begin{cases} \delta u_1 & = u_1 - u_{10} \\ \delta u_2 & = u_2 - u_{20} \\ \vdots & \\ \delta u_r & = u_r - u_{r0} \end{cases} \tag{1.13}$$

1.2.3.2 Linearization around an equilibrium point

Due to the difficulty in analyzing nonlinear systems, it is advantageous to look for approximations of nonlinear systems. Under specific conditions, it is possible to replace the nonlinear system with an approximate linear system. One of those cases is when the aim is to study whether small perturbations away from an equilibrium point (of the nonlinear system) grow or decay with time. The reason why it is not wise to linearize around non-equilibrium points is because it does not provide useful information.

The process of linearization is a set of mathematical tools designed to represent the system in a state space representation, it can be summarized in the following:

$$\begin{cases} \dot{x} = f(x,u,t) \xrightarrow{linearization} \dot{x} = Ax + Bu \\ y = g(x,u,t) \xrightarrow{linearization} y = Cx + Du \end{cases} \tag{1.14}$$

Such that:

$$A = \frac{\partial f(x,u,t)}{\partial x}\bigg|_{\substack{x = x_0 \\ u = u_0}}$$

$$B = \frac{\partial f(x,u,t)}{\partial u}\bigg|_{\substack{x = x_0 \\ u = u_0}}$$

$$C = \frac{\partial g(x,u,t)}{\partial x}\bigg|_{\substack{x = x_0 \\ u = u_0}}$$

$$D = \frac{\partial g(x,u,t)}{\partial u}\bigg|_{\substack{x=x_0\\ u=u_0}}$$

$$\begin{bmatrix} \delta \dot{x_1} \\ \delta \dot{x_2} \\ \vdots \\ \delta \dot{x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\substack{x=x_0\\ u=u_0}} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_n \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_r} \\ \frac{\partial f_2}{\partial u_1} & \cdots & \frac{\partial f_2}{\partial u_r} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_r} \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \vdots \\ \delta u_r \end{bmatrix} \tag{1.15}$$

$$\begin{bmatrix} \delta y_1 \\ \delta y_2 \\ \vdots \\ \delta y_m \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \cdots & \cdots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}_{\substack{x=x_0\\ u=u_0}} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_n \end{bmatrix} + \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_r} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial u_1} & \cdots & \frac{\partial g_m}{\partial u_r} \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \vdots \\ \delta u_r \end{bmatrix} \tag{1.16}$$

1.2.3.3 From state space to transfer function

To see how the transfer function is obtained, consider the Laplace transfer of Eq 1.5:

$$\begin{aligned} sX(s) - X(0) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \end{aligned}$$

Reordering the first term yields:

$$\begin{aligned} (sI - A)X(s) &= BU(s) + X(0) \\ X(s) &= (sI - A)^{-1}(BU(s) + X(0)) \end{aligned}$$

Replacing the expression of $X(s)$ in the output equation gives:

$$Y(s) = \Big(C\big((sI-A)^{-1}B\big) + D\Big)U(s) + C(sI-A)^{-1}X(0)$$

Which, as can be seen, depends on the initial conditions. setting initial conditions to zero results in the well-known expression.

$$Y(s) = \underbrace{\Big(C\big((sI-A)^{-1}B\big) + D\Big)}_{\text{H(s)}}U(s) \tag{1.17}$$

1.2.3.4 Transfer function to state space

Although the transformation from transfer function to state-space model is not unique, a method is provided for obtaining state variables in the form of phase variables here. When each following state is defined as the derivative of the preceding state variable, the state variables are phase variables.

$$\frac{d^n y(t)}{dt^n} + a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + \ldots + a_1\frac{dy(t)}{dt} + a_0 y(t) = b_0 u(t) \tag{1.18}$$

A convenient way to chose the state variables is to choose the output $y(t)$ and its $n-1$ derivatives as the state variables. They are called phase variables :

$$\begin{aligned} x_1 &= y \\ x_2 &= \frac{dy}{dt} \\ &\vdots \\ x_n &= \frac{d^{n-1}y}{dt^{n-1}} \end{aligned} \tag{1.19}$$

Differentiating both sides of the system Eq 1.19 yields:

$$\dot{x}_k = \frac{d^k y}{dt^k} \quad , \qquad \forall k \in \{0, n\} \tag{1.20}$$

making $\dot{x}_i = \frac{d^i y}{dt^i}$, the system Eq 1.19 can be written also as:

$$\begin{aligned}
x_1 &= y \\
x_2 &= \frac{dy}{dt} = \frac{dx_1}{dt} = \dot{x}_1 \\
x_3 &= \frac{d^2 y}{dt^2} = \frac{dx_2}{dt} = \dot{x}_2 \\
&\vdots \\
x_n &= \frac{d^{n-1} y}{dt^{n-1}} = \frac{dx_{n-1}}{dt} = \dot{x}_{n-1}
\end{aligned} \tag{1.21}$$

Substituting the definitions Eq 1.19 and Eq 1.20 into Eq 1.18 yields:

$$\dot{x}_n + a_{n-1}x_n + \cdots + a_1 x_2 + a_0 x_1 = b_0 u \tag{1.22}$$

The $n-th$ order differential equation Eq 1.18 is equivalent to a system of n first order differential equations obtained from the definitions of the derivatives from Eq 1.21 together with the $\dot{x}_n$ that results from Eq 1.22:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= -a_0 x_1 - a_1 x_2 - \cdots - a_{n-1} x_n + b_0 u
\end{aligned} \tag{1.23}$$

In a matrix-vector form the set of equations in Eq 1.23 become:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b_0 \end{bmatrix} u \tag{1.24}$$

Eq 1.24 is the phase-variable form of the state equation. This form is easily recognized by the pattern of 1's above the main diagonal and 0's for the rest of the state matrix, except for the last row that contains the coefficients of the differential equation written in reverse order. The output equation is:

$$y = x_1$$

Or, in vector form

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + 0 \cdot u \tag{1.25}$$

1.2.4 Controlability and Observability

The controlability gramian, $P \in \mathbb{R}^{n \times n}$, and the obsevability gramian, $Q \in \mathbb{R}^{n \times n}$, which are hermitian non negative definite matrices are obtained, for discrete time case, using the following formulas:

$$\begin{cases} P = \sum_{k=0}^{\infty} A^k BB^T (A^k)^T \\ Q = \sum_{k=0}^{\infty} A^k C^T CA^k \end{cases} \tag{1.26}$$

And do satisfy the Lyapunov equations:

$$\begin{cases} AP + PA^H + BB^H = 0 \\ A^H Q + QA + C^H C = 0 \end{cases} \tag{1.27}$$

For a continuous time analysis, the gramian matrices are:

$$\begin{cases} P = \int_0^{\infty} e^{A\tau} BB^T e^{A^T \tau} d\tau \\ Q = \int_0^{t} e^{A^T \tau} C^T C e^{A\tau} d\tau \end{cases} \tag{1.28}$$

State controlability and state observability can also be examined by considering the matrices

$$W_c \triangleq \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \tag{1.29}$$

$$W_o \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{1.30}$$

The system (A,B) is state controllable if W_c has full rank n. Similarly, the system (A,C) is state observable if W_o has full rank n.

1.3 Characteristics of a multivariable system

1.3.1 Frequency response (FR)

The frequency response is obtained by replacing s by $j\omega$ in the transfer function $G(s)$ (or transfer function matrix for a MIMO system). it can be used to describe system's response to sinusoidal of varying frequencies.

This interpretation has the benefit of being immediately related to the time domain and at each frequency ω the complex number $G(j\omega)$ has a physical meaning. It provides the frequency response to an input sinusoidal of frequency ω.

For a stable linear system described by $y = G(s)u$ interpreting frequency response means applying a sinusoidal signal with frequency $\omega[rad/s]$ and magnitude u_0, such that:

$$u(t) = u_0 sin(\omega t + \alpha)$$

The output signal is also a sinusoidal of the same frequency, but, with different amplitude and phase shift from the input.

$$y(t) = y_0 sin(\omega t + \beta)$$

Using the Laplace transform y_0/u_0 and Φ are obtained as follows:

$$\frac{y_0}{u_0} = |G(j\omega)| \quad , \quad \Phi = \angle G(j\omega)[rad] \triangleq \beta - \alpha \tag{1.31}$$

1.3.1.1 Shortcut method to find FR

The shortcut method consists of the following steps:

- Set $s = j\omega$ in $G(s)$ to obtain $G(j\omega)$.
- Rationalize $G(j\omega)$ to have it in the form of $G(j\omega) = R + jIM$, where R and IM are functions of ω.
- Simplify $G(j\omega)$ by multiplying the numerator and denominator by complex conjugate of the denominator.
- The amplitude ratio and phase angle of $G(s)$ are given by:

$$|G(j\omega)| = \sqrt{R^2 + IM^2} \quad , \quad \Phi = tan^{-1}\left(\frac{IM}{R}\right)$$

1.3.2 Directions in multivariable systems

For a Single Input Single output (SISO) system described by $y = Gu$ [2], the gain is given by:

$$\frac{|y(\omega)|}{|u(\omega)|} = \frac{|G(j\omega)u(\omega)|}{|u(\omega)|} = |G(j\omega)|$$

The gain in this case is dependent on ω, but is independent on the magnitude $|u(\omega)|$. For the case of a MIMO system, due to the nature of input and output (vectors), a measure appropriate for vectors is called for, that is, the vector 2-norm:

$$\begin{aligned} ||u(\omega)||_2 &= \sqrt{\sum_{j=1}^{r} |u_j(\omega)|^2} = \sqrt{u_{10}^2 + u_{20}^2 + \cdots + u_{r0}^2} \\ ||y(\omega)||_2 &= \sqrt{\sum_{j=1}^{m} |y_j(\omega)|^2} = \sqrt{y_{10}^2 + y_{20}^2 + \cdots + y_{m0}^2} \end{aligned} \tag{1.32}$$

The gain of the system transfer function matrix $G(s)$ is

$$\frac{||y(\omega)||_2}{||u(\omega)||_2} = \frac{||G(j\omega)u(\omega)||_2}{||u(\omega)||_2} \tag{1.33}$$

In the MIMO case, the gain depends on frequency ω, but is independent of the norm $||u(\omega)||_2$. it is, however, dependent on the direction of the input vector u.

The MIMO gain given above is known as the induced 2-norm, its maximum and minimum values are computed as the maximum and minimum singular values of G respectively.

$$\begin{aligned} \underset{u\neq 0}{max}\frac{||Gu||_2}{||u||_2} &= \underset{||u||_2=1}{max}\, ||Gu||_2 = \bar{\sigma}(G) \\ \underset{u\neq 0}{min}\frac{||Gu||_2}{||u||_2} &= \underset{||u||_2=1}{min}\, ||Gu||_2 = \underline{\sigma}(G) \end{aligned}$$

Example 1.1:

Considering the five different inputs shown below such that ($||u_i||_2 = 1, i \in \{1,2,3,4,5\}$):

$$u_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad u_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad u_3 = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}, \quad u_4 = \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix}, \quad u_5 = \begin{bmatrix} 0.6 \\ -0.8 \end{bmatrix}$$

The system is a 2 × 2 system described by:

$$G_1 = \begin{bmatrix} 5 & 4 \\ 3 & 2 \end{bmatrix}$$

The inputs u_i lead to the following outputs y_i:

$$y_1 = \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \; y_2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \; y_3 = \begin{bmatrix} 6.36 \\ 3.54 \end{bmatrix}, \; y_4 = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}, \; y_5 = \begin{bmatrix} -0.2 \\ 0.2 \end{bmatrix}$$

The corresponding 2-norms are as follows:

$$y_1 = 5.83, \; y_2 = 4.47, \; y_3 = 7.3, \; y_4 = 1.00, \; y_5 = 0.28$$

Plotting a functional plot of the matrix gain as a function of a parameterized input direction gives:

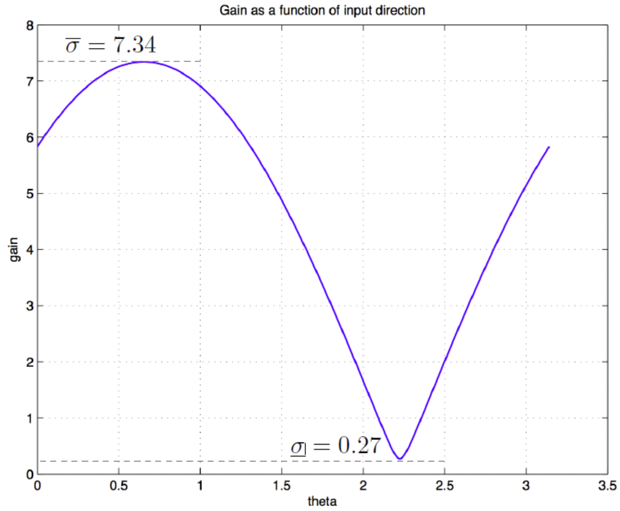


Figure 1.5: The gain of G.

The maximum and minimum gains are achieved at distinct vactor directions.

1.3.3 Singular Value Decomposition (SVD)

Definition 1: A matrix U is said to be unitary if

$$U^H = U^{-1}$$

All the eigenvalues of a unitary matrix have absolute value equal to 1, and all its singular values are equal to 1. Where: U^H denotes the Hermitian form of matrix U, i.e. $u_{ij} = \bar{u}_{ji}$.

Definition 2: Any complex $m \times r$ transfer function matrix G can be factorized into a singular value decomposition [3]. using the following formula :

$$\underset{m\times r}{G} = \underset{m\times m}{U} \; \underset{m\times r}{\Sigma} \; \underset{r\times r}{V^H} \tag{1.34}$$

Such that:

$$\begin{cases} UU^H = I_m \\ VV^H = I_r \end{cases}$$

And $\Sigma \in \mathbb{C}^{m\times r}$ is matrix containing a diagonal matrix Σ_1 of real, non-negative singular values, σ, arranged in a descending order as in:

$$\Sigma = \begin{cases} \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} & for \;\; m > r \\ \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix} & for \;\; m < r \;\; or \;\; \Sigma_1 = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_k \end{bmatrix}, \;\; k = min(m,r) \\ \Sigma_1 & for \;\; m = r \end{cases}$$

The scalars σ_i are called singular values of the matrix $G(j\omega)$, and are the squre roots of the eigenvalues of the matrix G^*G, i.e. $\sigma_i = \sqrt{\lambda_i(G^*G)}$.

Example 1.2:

Consider the following transfer function matrix [4]:

$$G = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \implies G^*G = \begin{bmatrix} 25 & -15 \\ -15 & 25 \end{bmatrix}$$

The eigenvalues of the matrix are $\lambda_1 = 40, \lambda_2 = 10$. Thus, the resulting eigenvectors are:

$$\begin{aligned} eig_1 &= \begin{bmatrix} -1 \\ 1 \end{bmatrix} & \Rightarrow \quad \sigma_1 = \sqrt{(-1)^2 + 1^1} = 1.41421 \\ eig_2 &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} & \Rightarrow \quad \sigma_2 = \sqrt{1^2 + 1^1} = 1.41421 \end{aligned}$$

$$\begin{aligned} v_1 &= \frac{1}{1.41421}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.7071 \\ 0.7071 \end{bmatrix} \\ v_2 &= \frac{1}{1.41421}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix} \end{aligned}$$

Therefore, we have $V = [v_1, v_2]$, and U is calculated using the formula $u_i = \frac{1}{\sigma_i} G \cdot v_i$. The final result is shown below:

$$G = U\Sigma V^H = \begin{bmatrix} -0.44722 & 0.89443 \\ -0.89443 & -0.44722 \end{bmatrix} \begin{bmatrix} 6.32456 & 0 \\ 0 & 3.16228 \end{bmatrix} \begin{bmatrix} -0.70711 & 0.70711 \\ 0.70711 & 0.70711 \end{bmatrix}^H$$

1.4 Stability

Stability is the most important concept in control engineering. A system is said to be stable, if its output is under control. Otherwise, it is said to be unstable. A stable system produces a bounded output for a given bounded input.

For an LTI system represented in state space form where $[A, B, C, D]$ are minimal realization, the system is said to be stable if and only if all eigenvalues of A have a negative real part.

1.5 Conclusion

This chapter introduced the multivariable systems, and the different forms they can be represented in, these forms are state-variable, state-space, and the transfer function matrix representation. In order to have more flexibility when working with these forms, this chapter explained the transition process between them, in this course, it addresses the equilibrium point and the linearization method for nonlinear systems.

Finally, in the framework of system characterization, this chapter tackles the concept of stability, the frequency response, directions in multivariable systems, and the singular value decomposition.

Chapter 2

Analysis of interactions in MIMO systems

2.1 Introduction

As noted previously, MIMO systems encounter the dilemma of coupling, which leads to a number of control issues. Several tools have been introduced to tackle this issue; such techniques aim to analyze the system and more importantly allow the selection of the optimal input/output pairs.

Technique selection is critical and decisive for achieving the desired performance as well as the optimal control configuration to ensure a weak interaction between loops.

This chapter covers a variety of issues that arise when working with multivariable systems, beginning with the coupling phenomenon and the general techniques for dealing with it. Also, it introduces several decoupling algorithms.

2.2 Interactions in multivariable systems

Compared to single-input single-output (SISO) systems, the control design for MIMO systems is more elaborate. One reason for this, as mentioned above, is that different parts of a multivariable system may intersect and cause couplings in the system.

2.2.1 Definition:

Couplings in a multivariable system in close loop can be defined as the effect of one input $R_i(s)$ on all outputs $\{Y_j(s) \quad \forall j \neq i\}$.

Example 2.1:

Consider a shower with separate hot and cold water flow controls. This is a MIMO system with two inputs, hot and cold water flows, which are used to control the two outputs, the flow from the tap and the temperature of the effluent water. Changing one of the inputs will obviously affect both of the outputs. This indicates that the system has significant couplings. In other words, interactions occur when a change in one input affects multiple outputs [5].

Fig. 2.1 shows a schematic representation of the system. G_{ij} denotes the transfer function between input u_j and output y_i. If the selected input/output pairing is $y_1 \rightarrow u_1$ and $y_2 \rightarrow u_2$ then the transfer functions G_{12} and G_{21} represent the cross couplings (channel interactions) in the system.

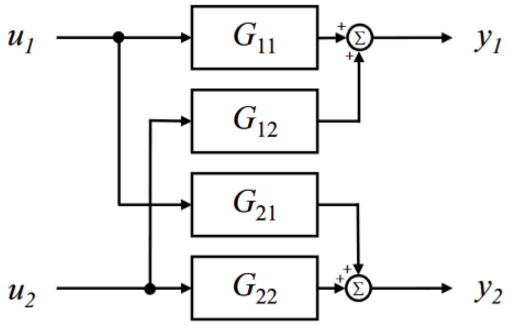


Figure 2.1: Block diagram of TITO system.

2.3 Interaction measures

2.3.1 Relative Gain Array (RGA)

The most commonly used interactions measure is the Relative Gain Array (RGA) developed by Bristol (1966). The RGA analyzes the plant's steady-state properties and suggests an option for the pairing problem in the case of a decoupled (decentralized) control structure, this type of structure will be diagonal, and the RGA also specifies which pairings should be avoided due to potential stability and performance issues [5].

For a non-singular square complex matrix G, the RGA is a square complex matrix defined as:

$$RGA(G) = \Lambda(G) \triangleq G(0) \times (G(0)^{-1})^T \tag{2.1}$$

where × denotes element-by-element multiplication (the Hadamard or Schur product).

Following Bristol [3]. It is possible to demonstrate that the RGA provides a measure of interactions. Assume that u_j and y_i represent a specific input output pair for the multivariable plant $G(s)$, and that the task is to use u_j to control y_i. Bristol contended that there will be two extreme scenarios:

- All other loops open: $u_k = 0 \quad \forall k \neq j$.
- All other loops closed with perfect control: $y_k = 0 \quad \forall k \neq i$.

Perfect control is only possible at steady-state, but it is a good approximation at frequencies within the bandwidth of each loop. We now evaluate our gain $\frac{\partial y_i}{\partial u_j}$ for the two extreme cases:

$$Other\, loops\, open: \qquad \left(\frac{\partial y_i}{\partial u_j}\right)_{u_k=0, k\neq j} = g_{ij} \tag{2.2}$$

$$Other\, loops\, closed: \qquad \left(\frac{\partial y_i}{\partial u_j}\right)_{y_k=0, k\neq i} \triangleq \hat{g}_{ij} \tag{2.3}$$

Here $g_{ij} = [G]_{ij}$ is the ij'th element of G, whereas $\hat{g}_{ij}$ is the inverse of the ji'th element of G^{-1}.

$$\hat{g}_{ij} = \frac{1}{G^{-1}_{ij}} \tag{2.4}$$

To derive Eq.2.4 equation, it is important to denote that:

$$y = Gu \left(\frac{\partial y_i}{\partial u_j}\right)_{u_k=0, k\neq j} = [G]_{ij} \tag{2.5}$$

And interchange the roles of G and G^{-1}, of u and y, and of i and j to get

$$u = G^{-1}y \left(\frac{\partial u_j}{\partial y_i}\right)_{y_k=0, k\neq i} = [G^{-1}]_{ij} \tag{2.6}$$

And Eq.2.4 follows. Bristol argued that the ratio between the gains in Eq.2.2 and Eq.2.3 is a useful measure of interactions, and defined the ij'th "relative gain" as:

$$\lambda_{ij} \triangleq \frac{g_{ij}}{\hat{g}_{ij}} = [G]_{ij}[G^{-1}]_{ij} \tag{2.7}$$

2.3.1.1 Algebraic properties

The RGA has a variety of useful algebraic properties. Out of these properties we mention the following:

- **Property 1:** the division in eq. (2.1) normalizes the RGA in such a way that the numerical sum of each column and row in the RGA equals one, i.e. for a $n \times n$ RGA:

$$\sum_{i=1}^{n} \Lambda_{ij} = \sum_{j=1}^{n} \Lambda_{ij} = 1 \tag{2.8}$$

- **Property 2:** the division in eq. (2.1) insures that the RGA to be scaling independent, i.e:

$$\Lambda(G) = \Lambda(S_1 G S_2) \tag{2.9}$$

  Such that: S_1 and S_2 are diagonal scaling matrices of the same dimension as G.

- **Property 3:** for a system with a 2×2 plant G with nonzero elements only, we have the following:

  a - If the number of positive elements in $G(0)$ is odd then $\Lambda_{ij} \in (0,1)$

  b - If the number of positive elements in $G(0)$ is even then $\Lambda_{ij} \in (-\infty, 0) \cup (1, \infty)$

- **Property 4:** if rows and columns are permuted in the transfer function matrix G then the rows and columns in the RGA are permuted in the same way.

- **Property 5:** if the transfer function matrix, G, is diagonal or triangular, and the rows in the transfer function matrix are permuted to produce nonzero elements along the diagonal in the case of a triangular G, the RGA equals the identity matrix. As a result, the RGA does not differ between diagonal and triangular plants.

2.3.1.2 Pairing recommendation

In the case of a 2×2 system, the following RGA matrix is obtained:

$$\Lambda(G) = \begin{bmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{bmatrix}$$

Depending on the value of λ, five different cases occur (Kinnaert, 1995):

- $\lambda = 1$: this is an ideal case when no interaction between the loops is present. The pairing should be along the diagonal, i.e. $y_1 \rightarrow u_1, y_2 \rightarrow u_2$.
- $\lambda = 0$: this is the same situation as above, except that now the suggested pairing is along the off-diagonal, i.e. $y_1 \rightarrow u_2, y_2 \rightarrow u_1$.
- $0 < \lambda < 1$: now, the gain increases (i.e. $\hat{G}_{ij}$ increases) when the loops are closed, hence, there is interaction. $\lambda = 0.5$ corresponds to the worst interaction.
- $\lambda > 1$: now, the gain decreases when the loops are closed. The interaction gets worse the larger λ is.
- $\lambda < 0$: now, even the sign changes when the loops are closed and this is highly undesirable. The more negative λ, the worse the interaction.

Example2.2: let us consider a 2×2 system with plant model.

$$y_1 = g_{11}(s)u_1 + g_{12}(s)u_2 \tag{2.10}$$

$$y_2 = g_{21}(s)u_1 + g_{22}(s)u_2 \tag{2.11}$$

Suppose that the aim is to control y_1 using u_1. Take the case where the other loop is open, that is u_2 is constant or $\rightarrow u_2 = 0$. Then:

$$u_2 = 0: \; y_1 = g_{11}(s)u_1$$

Consider the case where the other loop is closed perfectly, i.e. $y_2 = 0$. Because of interactions, when changing u_1, u_2 will also change. Setting $y_2 = 0$ in eq. (2.11), for example, yields:

$$u_2 = -\frac{g_{21}(s)}{g_{22}(s)}u_1$$

Substituting it into eq. (2.10) gives:

$$y_2 = 0 : y_1 = \left(g_{11} - \frac{g_{21}}{g_{22}}g_{12}\right)u_1 = \hat{g}_{11}(s)u_1$$

Closing the other loop results in gain changes from $g_{11}(s)$ to $\hat{g}_{11}(s)$, and the associated RGA element becomes

$$\lambda_{11}(s) = \frac{g_{11}(s)}{\hat{g}_{11}(s)} = \frac{1}{1 - \frac{g_{12}(s)g_{21}(s)}{g_{11}(s)g_{22}(s)}}$$

Intuitively, for decentralized control, the preferable pairing variables u_j and y_i so that λ_{ij} is close to 1 at all frequencies, as it means that closing the other loops has no effect on the gain from u_j to y_i. To be more specific, the following pairing rule suggest:

- **Pairing rule 1:** the input-output pair with corresponding RGA element close to one.
- **Pairing rule 2:** Niederlinski Index have to be positive.
- **Pairing rule 3:** larger RGA elements are not appropriate for input-output pairing.
- **Pairing rule 4:** RGA elements corresponding to the input-output pair must be positive.

Example 2.3: RGA for the wood and binary distillation column whose transfer function matrix is given by:

$$G(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}$$

The steady-gain matrix is obtained from the transfer function matrix by setting $s = 0$, giving:

$$G(0) = \begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix}$$

The inverse and transpose of inverse matrices are as follows:

$$G^{-1}(0) = \begin{bmatrix} 0.157 & -0.153 \\ 0.053 & -0.104 \end{bmatrix} \quad , \quad (G^{-1}(0))^T = \begin{bmatrix} 0.157 & 0.053 \\ -0.153 & -0.104 \end{bmatrix}$$

Applying es. (2.11) yields the RGA of the system:

$$\Lambda = \begin{bmatrix} 2.0 & -1.0 \\ -1.0 & 2.0 \end{bmatrix}$$

The RGA strongly advocates that the input-output pairs should be $u_1 \rightarrow y_1$ and $u_2 \rightarrow y_2$.

2.3.1.3 The Niederlinski Index (NI)

The NI is best used with the RGA to check if the recommended pairs are realizable in terms of stability. It is defined as follows:

$$NI = \frac{|G(0)|}{\prod_{i=1}^{n} g_{ii}} \tag{2.12}$$

When all control loops are closed, a negative value for NI indicates that the system will be integrally unstable for all possible controller parameter values.

Given the transfer function, the RGA is used to obtain a tentative loop pairing, then the NI is used to determine the stability of the closed loop system using the recommended RGA pairing, and finally simulation runs are used to verify if the recommended pairings are suitably stable.

2.3.1.4 The zeta ratio for a 2×2 system

Considering a TITO system with a steady-state gain matrix given by:

$$K = G(0) = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

Therefore, the RGA is:

$$\Lambda = \frac{1}{|K|} \begin{bmatrix} k_{11}k_{22} & -k_{12}k_{21} \\ -k_{12}k_{21} & k_{11}k_{22} \end{bmatrix}$$

While the NI is given by:

$$NI = \frac{k_{11}k_{22} - k_{21}k_{12}}{k_{11}k_{22}}$$

Setting:

$$\zeta = \frac{k_{21}k_{12}}{k_{11}k_{22}}$$

Rewriting the RGA and NI in terms of ζ yields:

$$\Lambda = \frac{1}{1-\zeta} \begin{bmatrix} 1 & -\zeta \\ -\zeta & 1 \end{bmatrix} \qquad \textit{and} \qquad NI = 1 - \zeta$$

Both RGA and NI can thus be said to be functions of ζ.

As a result, the unique ratio - the zeta ratio - can fully characterize a 2×2 system, and the smaller the value of ζ, the more perfect the diagonal pairing. However, there are many more ratios to consider for higher order systems.

2.3.2 Dynamic Relative Gain Array (DRGA)

Bristol (1966) calculated the RGA using only the plant steady-state gain, G(0). This is most likely due to the fact that in the process industry, the steady state measure is frequently far easier to obtain than the dynamic counterpart G(jω) [5].

However, a dynamic extension of the RGA was later proposed.

$$DRGA(G) = \Lambda(G(j\omega)) \triangleq (G(j\omega)) \times (G(j\omega)^{-1})^T \tag{2.13}$$

When analyzing a system, it is recommended to use this dynamic RGA and investigate the behavior of $\Lambda(G)$ in the relevant frequency range. As Skogestad and Postlethwaite (1996) pointed out, it is often enough to require $\Lambda(G(j\omega))$ to be close to the identity matrix at the crossover frequency to avoid instability. A pairing that results in negative RGA elements, on the other hand, should be avoided for any frequency of interest [5].

2.3.3 Nonsquare Relative Gain Array (NSRGA)

Definition: nonsquare plants are defined as multivariable plants with unequal number of inputs and outputs, based on this definition, it is possible to distinguish two types: more inputs than outputs or more outputs than inputs [6].

2.3.3.1 Control configuration selection of Nonsquare Multivariable plans

Nonsquare plant analysis and control theory is not as sophisticated as one may think, despite the fact that they are often encountered in many engineering disciplines. The squaring down procedure is the primary control approach for nonsquare plants. To obtain a square plant, the requisite number of outputs or inputs are added or eliminated from the transfer function matrix. The nonsquare plants can then be controlled using well-established control paradigms. However, each of these methods has its own set of issues. Adding unneeded outputs and inputs increases expenses and maintenance concerns; eliminating manipulated inputs reduces degrees of freedom for attaining desired responses; deleting outputs results in less reliable measurable information about plant performance.

Consider the linear multivariable plant described by the following transfer function matrix model.

$$Y(s) = G(s)U(s) \tag{2.14}$$

Such that:

- $G(s)$ is a $m \times r$ matrix
- $Y(s)$ is $m \times 1$ output vector.
- $U(s)$ is the $r \times 1$ input vector.

for this case, the following condition is considered: $m > r$.

It is not possible to derive the RGA definition. This is readily apparent from the plant's lack of functional controllability. As a result, the concept of perfect control is adjusted to accommodate this circumstance.

$$RGA^N(G) = \Lambda^N(G) = G(0) \times (G(0)^+)^T \tag{2.15}$$

Such that the (+) indicates the Moore-Penrose pseudo inverse.

Example 2.4: consider the steady state of a side stream distillation column with the following transfer function matrix (Chang and Yu 1990) [6].

$$G(0) = \begin{bmatrix} -9.811 & 0.374 & -11.3 \\ 5.984 & -1.986 & 5.24 \\ 2.38 & 0.0204 & -0.33 \\ -11.67 & -0.176 & 4.48 \end{bmatrix}$$

Its pseudo inverse is as follows:

$$G(0)^+ = \begin{bmatrix} -0.0245 & 0.0010 & 0.0117 & -0.0622 \\ -0.2748 & -0.5509 & 0.0205 & -0.0473 \\ -0.0762 & -0.0191 & -0.0060 & 0.0530 \end{bmatrix}$$

Applying eq. (2.15) gives the NSRGA:

$$\Lambda^N = \begin{bmatrix} 0.2406 & -0.1028 & 0.8606 \\ 0.0061 & 1.0940 & -0.1001 \\ 0.0277 & 0.0004 & 0.0020 \\ 0.7256 & 0.0083 & 0.2375 \end{bmatrix}$$

2.3.4 Gramian based interaction measures

Gramians are matrices that describe the controllability and observability features of a particular stable system. They may be calculated for both continuous and discrete time systems. For the sake of clarity, only the continuous time scenario is considered in this study.

Let's consider the system described by the following state-space representation:

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

Such that:

- $A \in \mathbb{R}^{n \times n}$.
- $B \in \mathbb{R}^{n \times r}$.
- $C \in \mathbb{R}^{m \times n}$.
- $D = 0_{m \times r}$.

2.3.4.1 Hankel singular value (HSV)

In order to extract valuable information from the matrices P and Q, the product PQ is calculated, the eigenvalues, $\lambda_i(i = 1,2,...,n)$, also known as the HSV are non negative and used to build an interaction measure.

Elementary system

Considering the previous MIMO system described by (A, B, C, 0), it is possible have a set of SISO systems where each one have a single input $u_i(i \in 1,2,...,r)$, and a single output $y_j(j \in 1,2,...,m)$, each SISO system is described with $(A, b_i, c_j^T, 0)$.

The gramian P_i and Q_i therefore satisfy:

$$\begin{cases} AP_i + P_iA^T + b_ib_i^T = 0 \\ A^TQ_j + Q_jA + c_jc_j^T = 0 \end{cases} \tag{2.16}$$

Such that:

- b_i is the i^{th} column of matrix B.
- c_j is the j^{th} column of matrix C^T.

Therefore, obtaining the pair (P_i, Q_i) and the HSV associated with it gives a description of the pair (u_i, y_i)'s ability to control and to observe the system state.

It is possible, therefore, to prove that:

$$P = \sum_{i=1}^{r} P_i \qquad Q = \sum_{j=1}^{m} Q_j$$

From the gramian decomposition it can be shown that the product PQ for the MIMO system is given by:

$$PQ = \left(\sum_{i=1}^{r} P_i\right)\left(\sum_{j=1}^{m} Q_j\right) = \sum_{i,j=1}^{r,m} P_iQ_j$$

2.3.4.2 The Hankel interaction Index Array (HIIA)

The Hankel norm [5], for a stable MIMO system represented by $(A, B, C, 0)$, considering a subsystem (i, j) given by $(A, b_{*j}, c_{i*}, 0)$, calculating the Hankel norm for each fundamental subsystem and arranged in a matrix as follows :

$$|\tilde{\Sigma}_H|_{ij} = ||G_{ij}||_H \tag{2.17}$$

A normalized version is called the Hankel interaction Index Array (HIIA) given by:

$$|\Sigma_H|_{ij} = \frac{||G_{ij}||_H}{\Sigma_{kl}||G_{ij}||_H} \tag{2.18}$$

The aim is to sum the corresponding elements in Σ_H and then find the simplest control structure that gives a sum as close as possible to one.

2.3.4.3 Participation matrix PM

In order to make profit of the previous analysis in terms of quantifying and comparing, it is required to use the trace of the product P_iQ_i which is a convenient basis to measure the interactions and the ability of different controller structures [6].

This measure can be organized in a martix $\Phi = [\phi_{ij}] \in \mathbb{R}^{n\times n}$ called the participation matrix, defined as:

$$\Phi_{ij} = \frac{trace[P_iQ_j]}{trace[PQ]} \leq 1 \tag{2.19}$$

Such that the trace of P_jQ_i is equal to the sum of squared HSVs of the subsystem with input u_i and y_j. It can be shown that the trace of PQ is equal to the sum of all trace of P_jQ_i.

2.3.4.4 The H_2 norm

The system H_2 norm [5]. For a stable and strictly proper system, i.e $(D = 0)$, with transfer function $G(s)$ is given by:

$$||G(s)||_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} tr\big(G^H(j\omega)G(j\omega)\big)d\omega} \tag{2.20}$$

For a system described by the state-space set of matrices $(A,B,C,0)$ with controllability Gramian P, observability Gramian Q, if the system is stable, continuous-time, and strictly proper, then the H_2 norm can be obtained as follows:

$$||G||_2 = \sqrt{tr\big(B^T QB\big)} = \sqrt{tr\big(CPC^T\big)} \tag{2.21}$$

2.3.4.5 the Σ_2 interaction measure

Suggested by Birk and Medvedev in 2003 [5], it is similar to the HIIA and obtained using the following formula:

$$|\Sigma_2|_{ij} = \frac{||G_{ij}||_2}{\Sigma_{kl}||G_{kl}||_2} \tag{2.22}$$

This measure is normalized in the same manner as the HIIA and the PM, it aims to find a controller structure that corresponds to a sum of the elements in Σ_2 as close as possible to one.

Example 2.4: (A quadruple-tank process)

The system is described by the following set of matrices:

$$A = \begin{bmatrix} -0.0159 & 0 & 0.1590 & 0 \\ 0 & -0.0159 & 0 & 0.02651 \\ 0 & 0 & -0.1590 & 0 \\ 0 & 0 & 0 & -0.02651 \end{bmatrix} \quad , \quad B = \begin{bmatrix} 0.05459 & 0 \\ 0 & 0.07279 \\ 0 & 0.01820 \\ 0.03639 & 0 \end{bmatrix}$$

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

The steady state transfer function is:

$$G(0) = \begin{bmatrix} 3.4326 & 1.1442 \\ 2.2884 & 4.5768 \end{bmatrix}$$

The RGA matrix for the system, denoted Λ, is:

$$\Lambda(G(0)) = \begin{bmatrix} 1.2 & -0.2 \\ -0.2 & 1.2 \end{bmatrix} \tag{2.23}$$

The gramian-based interaction matrices are:

$$\Sigma_H = \begin{bmatrix} 0.2866 & 0.1029 \\ 0.2285 & 0.3821 \end{bmatrix} \tag{2.24}$$

$$\Phi = \begin{bmatrix} 0.2809 & 0.0364 \\ 0.1834 & 0.4994 \end{bmatrix} \tag{2.25}$$

$$\Sigma_2 = \begin{bmatrix} 0.3146 & 0.1000 \\ 0.1658 & 0.4195 \end{bmatrix} \tag{2.26}$$

In order to have direct element by element comparison for the sake of simplifying the analysis, it is required to calculate the squar of the PM matrix then normalize it which results in:

$$\bar{\Phi} = \begin{bmatrix} 0.2809 & 0.0364 \\ 0.1834 & 0.4994 \end{bmatrix} \tag{2.27}$$

All gramian-based interaction matrices along with the RGA matrix suggest that the optimal input/output pairing is $y_1 \rightarrow u_1, y_2 \rightarrow y_2$.

2.3.5 Direct Nyquist Array (DNA)

The DNA is a graphical method with the following principle:

- Draw the Nyquist graph of every diagonal element $G_{ij}(s)$ of the transfer function matrix $G(s)$ for ω varying from 0 to $+\infty$.
- Superpose every graph with the Geshgorin circles obtained by varying ω from 0 to $+\infty$.
- The coordinations of a circle's center are the real and imaginary parts of $G_{ij}(s)$, and the vector $R_{ii}(s)$ of the circle is the sum of the modules of elements of the i^{th} column except the one of element $G_{ij}(s)$ considering the vector $R_{ii}(s)$ which is given by the following formula:

$$R_{ii}(s) = \sum_{j=1, j\neq i}^{m} |G_{ji}(s)| \tag{2.28}$$

2.3.6 Inverse Nyquist Array (INA)

Rosenbrock developed the Inverse Nyquist Array (INA), a frequency domain method. Its goal is to design compensators for multivariable systems so that they can efficiently be decoupled, thereby being treated using univariable procedures. It is only applicable to square, controllable systems in their original form since it necessitates the calculation of the inverse transfer function. It is proposed to extend it to non-square systems with a given feedback structure that includes an inner loop. The graphical interpretation as well as the new form of the basic operations are presented [3].

This method is based on the inverse characteristics $\hat{Q}(s)$ and $\hat{H}(s)$ resulting from the following simple relation.

$$\hat{H} = \hat{Q} + F \tag{2.29}$$

Such that:

– $\hat{H}(s)$: the inverse of closed-loop system transfer function matrix $H(s)$.

– $\hat{Q}(s)$: the inverse of open-loop system forward transfer function matrix $Q(s)$.

– F: the feedback gain matrix.

2.3.6.1 Principle of INA

As stated before, the INA uses the inverse of the transfer function matrix of the system, its principle is as follows:

- Calculate $\hat{G}(s)$ which is the inverse of the matrix $G(s)$, $\hat{G}(s) = G^{-1}(s)$.
- Construct the nyquist of all diagonal elements $\hat{G}_{ii}(s)$ of the matrix $\hat{G}$ for ω varying from 0 to $+\infty$.
- obtain the Gershgorin circles for ω varying from 0 to $+\infty$ and superpose them.
- The coordinations of a circle's center are the real and imaginary parts of $\hat{G}_{ii}(s)$, and the vector $\hat{R}_{ii}(s)$ of the circle is the sum of the modules of elements of the $i^t h$ column except the one of element $\hat{G}_{ii}(s)$ considering the vactor $\hat{R}_{ii}(s)$ which is given by the following formula:

$$\hat{R}_{ii}(s) = \sum_{j=1, j\neq i}^{m} |\hat{G}_{ji}(s)| \tag{2.30}$$

2.3.6.2 Advantages of using INA

- The properties of the relation ($\hat{H} = \hat{Q} + F$), such that the hat denotes the inverse of the matrix.
- The inverse transfer function tends to be more diagonally dominate than the direct one.
- The feedback gains of all but loop j become indefinitely large, the transfer function between input j and output j, h_j, approaches $(\hat{q}_{jj})^{-1}$, such that $\hat{q}_{jj}$ is the jj element of matrix $\hat{Q}$.

2.3.6.3 DNA and INA interpretation

The two methods presented allow interaction analysis, these interactions occur between the loops of the configuration and are defined by the elements of the diagonal of the transfer matrix. This configuration presents weak interactions, if the Geshgorin circles of each element of the diagonal of the system $G(s)$ or $\hat{G}(s)$ according to the considered analysis method, do not encircle the origin of the complex plane in the working frequency band of the system.

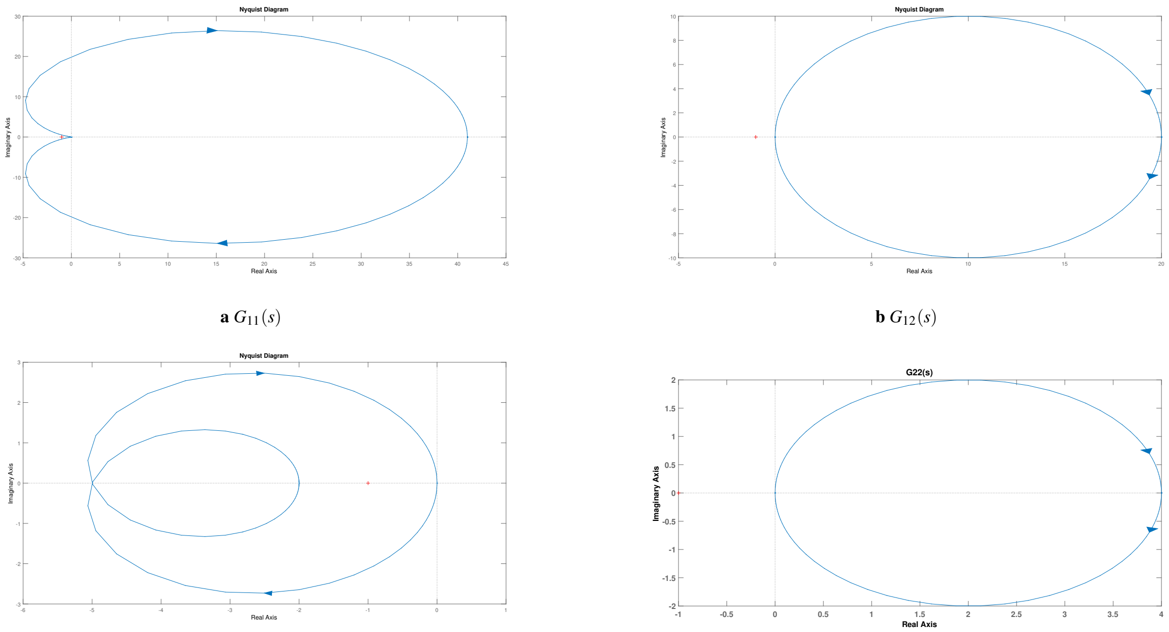
Example 2.5:

Since frequency domain analysis is not the aim of this work, only the DNA is obtained for this example is the scope of explanation.

Let us consider the following transfer function matrix representing a multivariable system:

$$\begin{bmatrix} Y_1 \\ \\ Y_2 \end{bmatrix} = \begin{bmatrix} \frac{s+41}{s^2+2s+1} & \frac{-10s-2}{s^2+2s+1} \\ \\ \frac{20}{s+1} & \frac{4}{s+1} \end{bmatrix} \begin{bmatrix} U_1 \\ \\ U_2 \end{bmatrix} \tag{2.31}$$

Using the **DNA** to analyze the interactions



a $G_{11}(s)$

b $G_{12}(s)$

c $G_{21}(s)$

d $G_{22}(s)$

Figure 2.2: DNA

2.4 General purpose decoupling algorithms

The construction of an adequate decoupler or controller will be the next crucial step after a proper input-output pairing has been selected in a MIMO system. A centralized MIMO controller with a series of decouplers or a series of SISO decentralized controllers can both be used to implement an effective MIMO control strategy.

2.4.1 Decentralized controller

Let $G(s)$ be an $n \times n$ transfer function describing a MIMO system such that:

$$G(s) = \{g_{ij}(s), \quad i = 1, \ldots, n, \quad j = 1, \ldots, n\} \tag{2.32}$$

The resulting decentralized controller is the diagonal matrix $C(s)$:

$$C(s) = \begin{bmatrix} C_{11}(s) & 0 & \dots & 0 \\ 0 & C_{22}(s) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_{nn}(s) \end{bmatrix} \tag{2.33}$$

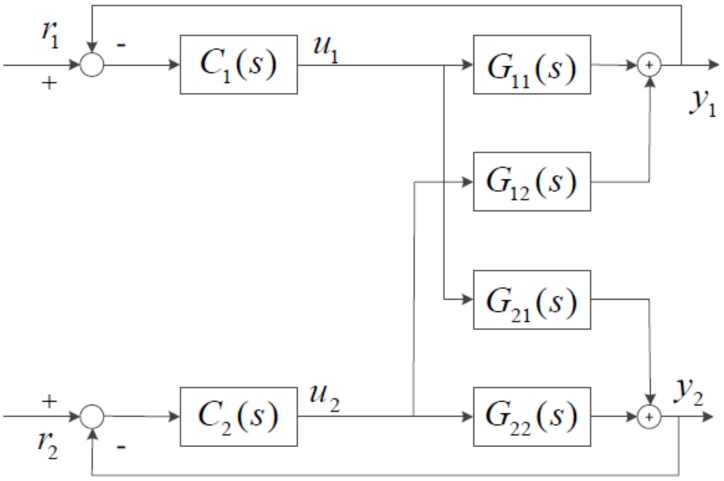


Figure 2.3: Simplest decentralized control structure of TITO system.

2.4.2 Static decoupling

If the available information is limited, it is advisable to design a static decoupler. For a MIMO system with the transfer function shown in eq. (2.34). Let the parameters k_{ij} $\{i = 1,\dots,n \quad j = 1,\dots,n\}$ be steady state gains of $g_{ij}(s)$. The steady state gain for matrix $G(0)$ is obtained as follows:

$$G(0) = \{k_{ij}, \quad i = 1,\dots,n \quad j = 1,\dots,n\} \tag{2.34}$$

The static decoupler is derived as:

$$D = G^{-1}(0) \tag{2.35}$$

Static decouplers are recommended for various industrial processes because they require less information from the controlled system and can minimize the danger of influence caused by model uncertainties. In the meanwhile, a static decoupler may be easier to create than a dynamic one [7].

In a closed loop, however, a static decoupler may not provide adequate decoupling performance. It may also have an adverse effect on the high frequency responsiveness of various MIMO processes. In systems with integral elements, the static decoupler is recommended. This is because, when frequency rises, the magnitudes of non-diagonal terms fall quicker than those of diagonal terms.

2.4.3 Dynamic decoupling

When compared to static decoupling, a MIMO system with a dynamic decoupler consistently outperforms static decoupling at the cost of acquiring an accurate process model. Ideal decoupling, simplified decoupling, and inverted decoupling are three types of dynamic decoupling algorithms that have been intensively investigated and utilized in industrial processes. Each of these three decouplers has its own set of characteristics and drawbacks [7].

For the sake of simplicity, a TITO system will serve as an example in this part.

- **Ideal decoupling**

For a TITO system defined by the following:

- $C(s)$: controller matrix.
- $D(s)$: decoupler matrix.
- $G(s)$: controlled plant.
- r_i: set-point signals.
- u_i: control signals.
- y_i: output signals.

Such that:

$$C(s) = \begin{bmatrix} C_1(s) & 0 \\ 0 & C_2(s) \end{bmatrix} \tag{2.36}$$

$$D(s) = \begin{bmatrix} D_{11}(s) & D_{12}(s) \\ D_{21}(s) & D_{22}(s) \end{bmatrix} \tag{2.37}$$

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \tag{2.38}$$

If the controlled system $G(s)$ is ideally decoupled, the matrix representing the decoupled system, defined as $M(s) = G(s)D(s)$, should be diagonal. the matrix $D(s)$ is therefore obtain as:

$$\begin{aligned} D(s) &= G^{-1}(s)M(s) \\ &= \frac{1}{G_{11}(s)G_{22}(s) - G_{12}(s)G_{21}(s)} \times \begin{pmatrix} G_{22}(s)M_{11}(s) & -G_{12}(s)M_{22}(s) \\ -G_{21}(s)M_{11}(s) & G_{11}(s)M_{22}(s) \end{pmatrix} \end{aligned} \tag{2.39}$$

The aim of ideal decoupling is to set $M_{11}(s) = G_{11}(s)$ and $M_{22}(s) = G_{22}(s)$, this results in a product matrix of the form:

$$M(s) = \begin{bmatrix} M_{11} & 0 \\ 0 & M_{22}(s) \end{bmatrix} \tag{2.40}$$

This shows a full decoupled system, therefore, controller $C_1(s)$ and $C_2(s)$ can be designed in the same manner under ideal decoupling. The controller does not need to be redesigned even if different loops are set in different modes.

While ideal decoupling provides obvious operational advantages, the complicated presentation of D(s), which includes sums of transfer functions, is frequently a challenge.

Furthermore, the problem of limited applicability, as well as the sensitivity to model flaws and system dimensions of ideal decoupling, should not be overlooked. As a result, optimal decoupling is rarely applied in real-world applications.

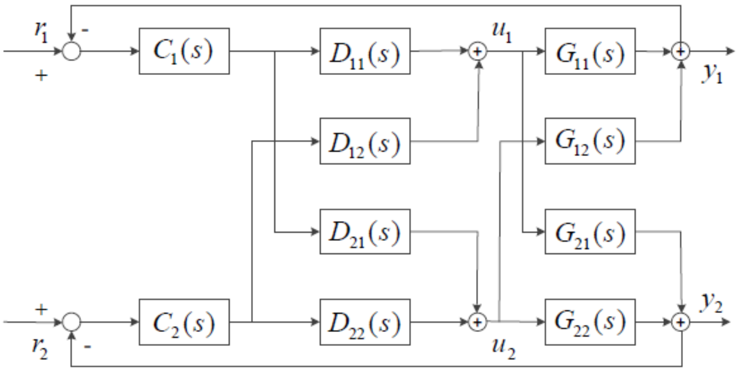


Figure 2.4: Ideal decoupling structure.

- **Simplified decoupling**

It is the most used in literature [7]. The decoupler matrix has the form:

$$D(s) = \begin{bmatrix} 1 & -G_{12}(s)/G_{11}(s) \\ -G_{21}(s)/G_{22}(s) & 1 \end{bmatrix} \tag{2.41}$$

For simplified decoupling, only two decouplers are generated, there exist three alternative configurations of simplified decoupling, two elements in different columns of matrix $D(s)$ are set to 1. These three alternatives configurations are obtained as:

$$D(s) = \begin{bmatrix} -G_{22}(s)/G_{21}(s) & 1 \\ 1 & -G_{11}(s)/G_{12}(s) \end{bmatrix} \tag{2.42}$$

$$D(s) = \begin{bmatrix} -G_{22}(s)/G_{21}(s) & -G_{12}(s)/G_{11}(s) \\ 1 & 1 \end{bmatrix} \tag{2.43}$$

$$D(s) = \begin{bmatrix} 1 & 1 \\ -G_{21}(s)/G_{22}(s) & -G_{11}(s)/G_{12}(s) \end{bmatrix} \tag{2.44}$$

In practice, the simplified decoupling technique is simple to implement. However, because the decoupler expression still contains some summation elements, the controller tuning process may be problematic.

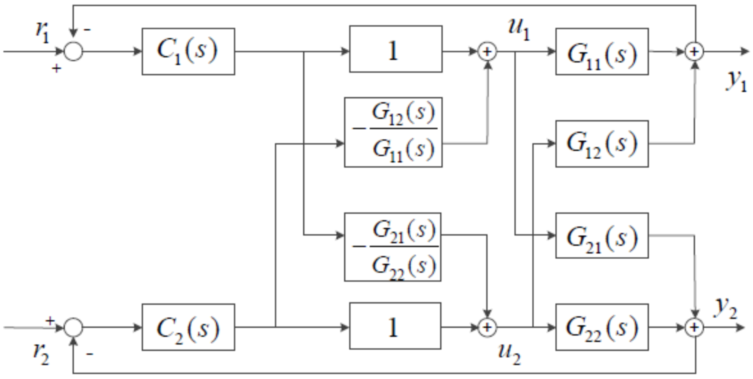


Figure 2.5: Simplified decoupling structure.

- **Inverting decoupling**

Inverted decoupling is another extensively used decoupling algorithm that can derive the same decoupled process model as an ideal decoupler without using a difficult $D(s)$ expression [7].

$$D_{11}(s) = D_{22}(s) = 1 \tag{2.45}$$

This results in:

$$\begin{aligned} D_{12}(s) &= -\frac{G_{12}(s)}{G_{11}(s)} \\ D_{21}(s) &= -\frac{G_{21}(s)}{G_{22}(s)} \end{aligned} \tag{2.46}$$

The decoupled transfer function of inverted decoupling is the same as that of ideal decoupling, and it is implemented in the same way as simple decoupling. Therefore, it should be able to provide both ideal and simplified decoupling. This results in three advantages:

- When inverted decoupling is used, the decoupled system behaves as if there is no interaction between control loops and the alternate controllers are in manual mode; each decoupled loop can be kept from acting as a secondary to other control loops.
- It can be implemented in DCS as a feed-forward input.
- When the system mode is changed, the initialization and bumpless issues will not emerge.

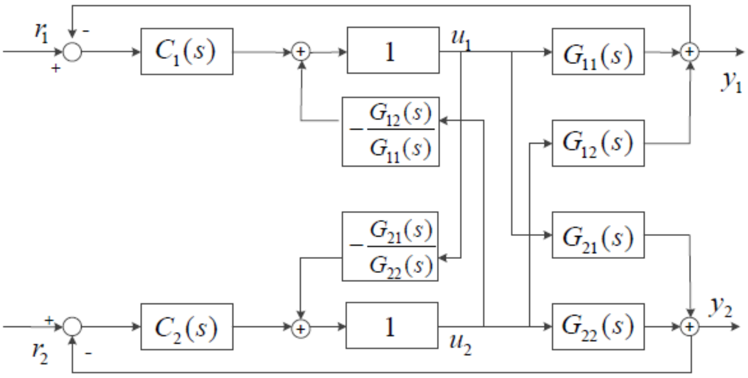


Figure 2.6: Inverted decoupling structure.

2.5 Conclusion

This chapter examines and analyzes the interactions that occur between inputs and outputs in multivariable systems.

It also provides an overview of the techniques used to examine this phenomenon, such techniques aim to analyze the degree of interaction, and suggest the best input-output pairs suitable for the application taking into consideration stability. First, the most used decoupling method is the RGA approach to control configuration selection for steady-state and a dynamic extension known as DRGA, as well as the NSRGA, a nonsquare RGA approach. In addition, the critical Niederlinski tool and definitions for integrity analysis were presented. Moreover, there are the Gramian-based interaction methods, which include the HIIA, the PM the H_2 norm, and the Σ_2 methods, an example is presented to compare the suggestions of each method. Also, the direct Nyquist array and the inverse Nyquist array, are two powerful multivariable approaches that involve stability analysis.

Finally, this chapter provides a comprehensive review of the existing cross-loop interactions analysis and decoupling control methods. Special purpose decoupling algorithms are introduced with their properties, advantages, and application domains.

Chapter 3

Fault Tolerant Control

3.1 Introduction

One of the problems that face the design and analysis of MIMO systems is the component failures, to overcome that, the concept of fault-tolerant control is found for mitigating the effects of system component failures.

Due to system design limitations or sensors expenses, some system parameters are hard/impossible to measure, to overcome this issue, observers are designed to estimate these parameters with a high degree of accuracy, higher than the sensed signals in some cases, and reduce the costs related to hardware redundancy of sensors.

Correction methods are a big part of analyzing multivariable systems, it is fundamental to understand that the specifications stipulating the closed loop performances will be translated by the constraints on the frequency response of the open loop corrected system.

Designing compensators often rises a trade-off issue between stability and rapidity, that is, increasing system response speed often has stability risks, while aiming for a stable system results in a slow response system that does not satisfy the design requirements.

This chapter addresses three important concepts in control engineering, the fault-tolerant control, the states and faults estimation using observers, and the feedback control for controller design.

3.2 Fault tolerant control

3.2.1 Faults

A fault is defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable behavior. The fault is a state that may lead to a malfunction or a failure in the system. There are many types of faults [8].

3.2.2 Fault types

Physical location classification yields the following fault types: actuator fault, sensor fault, plant component fault.

- **Actuator faults:** they vary from complete loss of control to loss of partial control effectiveness, such faults affect severely the system performance since the actuator is considered as the entrance of the system [9].
- **Sensor faults:** they mostly include incorrect readings due to malfunction in the sensor circuit, fortunately, increasing sensor reliability can be achieved using parallel hardware redundancy [9].
- **Plant component faults:** they are caused by physical parameter variations in the system, and result in changes in the dynamical relationship between system variable.

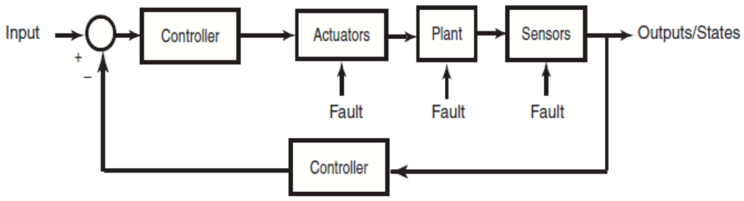
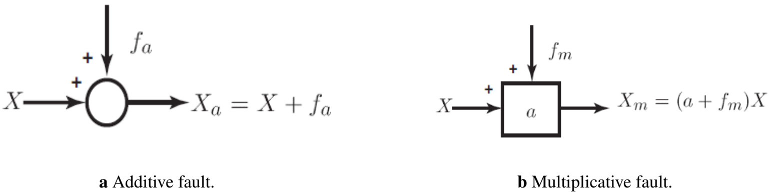


Figure 3.1: Location of potential faults in a control system.

Classifying faults based on their induced effects on the system performance yields two types: additive faults and multiplicative faults.

- **Additive faults:** they affect the mean value of the system output signal only.
- **Multiplicative faults:** they affect variance and correlations of the system as well as its spectral characteristics and dynamics.



a Additive fault.

b Multiplicative fault.

Figure 3.2: Different faults induced changes.

A severity based classification of faults results in three main types of faults: drift-like faults, abrupt faults, and intermittent faults.

- **Drift-like (incipient) fault:** it refers to the situation where a signal slowly deviates (linearly) from the actual value. Until the effect on the process is corrected, it remains constant. Sensor/actuator inaccuracy or partial failure are the sources of these faults.
- **Abrupt (stepwise) fault:** It is a sudden and considerable change in parameter values, it is more rapid than the standard dynamic process. The fault's cause and/or effects will continue until they are corrected.
- **Intermittent fault:** it is a malfunction that occurs at irregular intervals. This type of faults, which is common in most systems, can be caused by a number of factors, including improper electrical wire connections to sensors, actuators, and other components. Intermittent faults are more likely to occur as the system becomes more complex. The detection of intermittent faults is difficult for most detection algorithms due to their inconsistent nature.

3.2.3 Faults description:

For the sake of simplicity, only additive faults are considered, for both actuator faults and sensor faults, the mathematical description can be obtained as follows:

- **Actuator faults:** In the presence of an actuator fault, the linear system can be given by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + F_a f_a(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

Such that: $F_a = B$, and f_a is a matrix containing the magnitudes of the actuators faults. For the nonlinear system, the actuator faults are injected as follows:

$$\begin{aligned} &\dot{x}(t) = f\big(x(t)\big) + \sum_{j=1}^{n} \big(g_j(x(t))u_j(t)\big) + \sum_{j=1}^{n} \big(F_{a,j}(x(t))f_{a,j}(t)\big) \\ &y_i(t) = g_i\big(x(t)\big) \end{aligned}$$

Such that: $F_{a,j}(x(t))$ corresponds to the j^{th} column of matrix $G\big(x(t)\big)$, and $f_{a,j}(t)$ corresponds to the magnitude of the fault affecting the j^{th} actuator.

- **Sensor faults:** In a similar way, considering f_s as an unknown input illustrating the presence of a sensor fault, the linear faulty system will be represented by

$$\begin{aligned}\dot{x}(t) &= Ax(t)+Bu(t)\\ y(t) &= Cx(t)+Du(t)+F_s f_s(t)\end{aligned}$$

Where F_s can be considered as an identity matrix of proper dimensions. The affine nonlinear systems can be defined in continuous-time through an additive component such as:

$$\begin{aligned}\dot{x}(t) &= f\big(x(t)\big)+\sum_{j=1}^{n}\big(g_j(x(t))u_j(t)\big)\\ y_i(t) &= g_i\big(x(t)\big)+\sum_{j=1}^{m}\big(F_{s,j}(x(t))f_{s,j}(t)\big)\end{aligned}$$

3.2.4 Fault Diagnosis (FD)

It refers to the task of inferring the concurrence of faults in a process and finding their root causes using: quantitative models, qualitative models, and historical data. since diagnosis from raw data is often difficult, quantitative and qualitative model-based FDI techniques are combined to supervise the process and to ensure appropriate reliability and safety in the industry [10].

3.2.5 Fault Tolerant Control Systems (FTCS)

A FTCS is a control system that can accommodate system component faults and can maintain stability and an acceptable degree of performance not only when the system is fault free but also when there are component malfunctions. FTCS prevents faults in a subsystem from developing into failures at the system level [9].

3.2.5.1 Classification of FTCS

There are two approaches when designing an FTCS: Passive FTCS and Active FTCS. the best approach is determined based on the redundancy being utilized in the system, the ability to determine the faults that a system may undergo during the design phase, and the behavior of fault-induced changes.

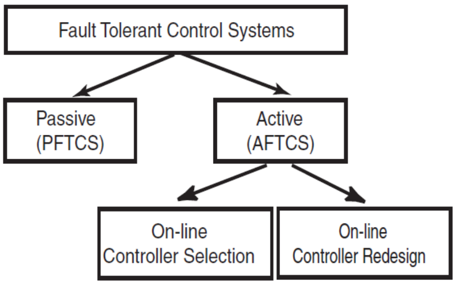

Fault Tolerant Control Systems
Passive (PFTCS)
Active (AFTCS)
On-line Controller Selection
On-line Controller Redesign

Figure 3.3: Classification of fault tolerant control systems.

- **PFTCS:** for this approach, the system can tolerate a limited number of faults that are known prior to the design of the controller, thus, it has a limited fault tolerance capability [9].
- **AFTCS:** it is designed using the available resources and employs both physical and analytical system redundancy to accommodate unanticipated faults, using FDI algorithms to identify the fault-induced changes, it can compensate for faults by either selecting a pre-computed control law or by synthesizing a new control law on-line in real-time [9].

3.3 states and faults estimation

- **State estimation:** control of spacecraft and other systems requires reliable real-time estimates of system state. Unfortunately, the complete state is not always observable. State Estimation takes all the telemetry seen so far and uses it to determine the underlying behavior of the system at any point in time. It includes fault detection and isolation and continuous system parameter estimation [11].
- **Fault estimation (FE):** it provides information on the fault such as the location, size, and duration. Thus, it is especially useful for incipient faults and slow drifts, which are very difficult to detect. Also, fault estimation is vital in FTCS which improve the system's performance.

3.3.1 Observers:

Observers are algorithms that combine sensed signals with other knowledge of the control system to produce observed signals. The principle of an observer is that by combining a measured feedback signal with knowledge of the control system components, the behavior of the plant can be known with greater precision than by using the feedback signal alone.

In other cases, observers can reduce system costs by augmenting the performance of a low-cost sensor so that the two together can provide performance equivalent to a higher-cost sensor.

In the extreme case, observers can eliminate a sensor altogether, reducing sensor cost and the associated wiring. Phase lag and attenuation can be caused by the physical construction of the sensor or by sensor filters, which are often introduced to attenuate noise. The key detriment of phase lag is the reduction of loop stability.

3.3.1.1 Unknown Input Observer (UIO):

In order to design an UIO [12], the following procedure is applied. For the case of linear dynamic systems in which system uncertainty can be modeled as an additive unknown disturbance term in the dynamic equation:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + E_d d(t) \\ y(t) = Cx(t) \end{cases} \tag{3.1}$$

The disturbance distribution matrix E_d must be full column rank, in case it is not, matrix decomposition to the following form is performed:

$$E_d d(t) = E_1 E_2 d(t)$$

Where E_1 is a full column rank matrix and $E_2 d(t)$ can be considered as the new unknown input or disturbance acting on the system.

Second, the following existence condition must be checked, (C, A_1) must be a detectable pair, where:

$$A_1 = A - E_d[(CE_d)^T CE_d]^{-1}(CE_d)^T CA$$

For an observer to be called Unknown Input Observer for the system described by eq. (3.1), the state estimation error vector denoted as $e(t)$ and defined as:

$$e(t) = x(t) - \hat{x}(t) \tag{3.2}$$

Must approach zero asymptotically regardless of the process of unknown inputs, $d(t)$, in the system. Furthermore, the structure for the full order UIO is given by the dynamic system:

$$\begin{aligned} \dot{z}(t) &= F_{obs} z(t) + TBu(t) + Ky(t) \\ \hat{x}(t) &= z(t) + Hy(t) \end{aligned} \tag{3.3}$$

Where $\hat{x}$ is the state estimate, z is the state of full-order dynamic observer, and F_{obs}, T, K, H are matrices to be designed for the purpose of achieving the unknown input decoupling. In order to achieve this decoupling $\dot{e}(t)$ is expended to be:

$$\begin{aligned} \dot{e}(t) = {} & (A - HCA - K_1C)e(t) \\ & + [F_{obs} - (A - HCA - K_1C)]z(t) \\ & + [K_2 - (A - HCA - K_1C]y(t) \\ & + [T - (I - HC)]Bu(t) \\ & + (HC - I)E_d d(t) \end{aligned} \tag{3.4}$$

Such that: $K = K_1 + K_2$, and K_1 is calculated using pole placement technique with the following Matlab command:

K_1 = place(A',C',P)'.

The objective is make the estimation error a function of $e(t)$, which means:

$$\dot{e}(t) = Fe(t) \tag{3.5}$$

In order to satisfy Eq 3.5, the following must hold.

$$0 = (HC - I)E_d \tag{3.6}$$

$$T = I - HC \tag{3.7}$$

$$F_{obs} = A - HCA - K_1C \tag{3.8}$$

$$K_2 = F_{obs}H \tag{3.9}$$

If all the eigenvalues in F are stable, the $e(t)$ will approach zero asymptotically. notice that the estimation error is not a function of E_d or d. Therefore, it approaches zero independently of the disturbance terms, achieving the desired decoupling of the state estimate from the unknown disturbance inputs.

The disturbance estimation can be obtained by the following relation:

$$\hat{d} = (CE)^{+}[\dot{\hat{y}} - CA\hat{x} - CBu]$$

Such that $(CE)^{+}$ denotes the Moore-Penrose Pseudo inverse of the matrix CE.

The following figure shows a block diagram of the UIO.

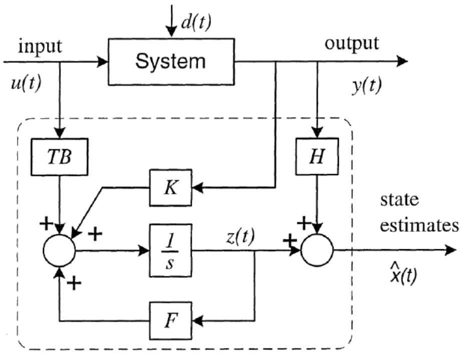


Figure 3.4: Unknown Input Observer (UIO).

3.3.1.2 Luenberger Observer

Named after David Gilbert Luenberger, a Professor in Management Science and Engineering at Stanford University, who first introduced these methods for constructing state observers in his doctoral dissertation at Caltech. Considering a system modeled by the n-dimensional, r-input, m-output LTI system.

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

The Luenberger observer will employ a copy of the system with one notable difference.

$$\dot{x}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)]$$
$$\hat{y}(t) = C\hat{x}(t) + Du(t)$$

The term $L[y(t) - \hat{y}(t)]$ injects the error between measurements and model prediction, scaled by a user-selectable "observer gain" vector $L \in \mathbb{R}^{n\times m}$. This concept is logically named "output error injection", and is the key feature of state estimation design.

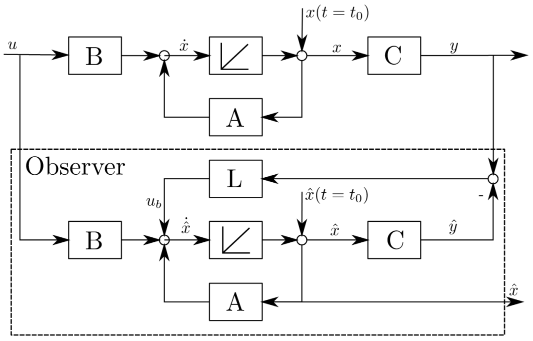


Figure 3.5: Block diagram of Luenberger observer.

To understand this state estimation algorithm, the error dynamics, $e(t)$, is considered to be $e(t) = x(t) - \hat{x}(t)$ which evolves according to

$$\begin{aligned}\dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\ &= Ax(t) + Bu(t) - A\hat{x}(t) - Bu(t) - L[y(t) - \hat{y}(t)] \\ &= A(x(t) - \hat{x}(t)) - L[Cx(t) + Du(t) - C\hat{x}(t) - Du(t)] \\ &= A(x(t) - \hat{x}(t)) - LC(x(t) - \hat{x}) \\ &= (A - LC)e(t)\end{aligned} \tag{3.10}$$

The aim is to select L such that the eigenvalues of $(A - LC)$ have negative real parts, that way, the estimation error system is asymptotically stable. selecting L appropriately enables assigning the eigenvalues (i.e. speed) of the error system. Mathematically, $e(t) \to 0$ or $\hat{x}(t) \to x(t)$ as $t \to \infty$. the speed of convergence is characterized by the eigenvalues of $(L - AC)$

Placement rule: as a general rule-of-thumb, the observer eigenvalues should be placed from 2 to 10 times faster than the slowest stable eigenvalue of the system itself. if λ_A is the slowest stable eigenvalue of the system, we have:

$$-\infty < 10 \cdot Re[\lambda_A] \leq Re[\lambda_{A-LC,i}] \leq 2 \cdot Re[\lambda_A] < 0 \qquad \forall i$$

Where: $\lambda_{A-LC,i}$ are the eigenvalues of $A - LC$, this rule ensures fast convergence of the estimations, without risking destabilizing the system. Given that the eigenvalues of $(A - LC)$ can be assigned arbitrarily for (A, C) observable, one might suggest setting the eigenvalues at -10^{99}, the reason why the choice of placing eigenvalues relatively at negative infinity is not desirable is the sensor noise. to demonstrate more, suppose the following dynamic system:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) + n(t)\end{aligned}$$

Where $n(t)$ is the measurement noise that exists within any real-world sensor. Extracting the error dynamics for the system yields:

$$\dot{e}(t) = (A - LC)e(t) - Ln(t)$$

A choice of eigenvalues close to negative infinity will render a very large observer gain L. This will result in an amplified sensor noise. To summarize, there must be a balance in the trade-off between speed and robustness to noise in order to have optimal results.

3.4 Proportional Integral Derivative (PID) Controllers

Definition:

A PID controller is an instrument used in industrial control applications to regulate temperature, flow, pressure, speed, and other process variables. PID controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controllers.

PID controllers have the purpose of forcing the feedback to match a certain setpoint. For example, a thermostat forces the heating and cooling unit to turn on or off based on a set temperature. PID controllers are best used in systems with relatively small mass and these which react quickly to changes in energy added to the process. The controller is expected to compensate automatically for loud changes due to frequent changes in setpoint.

Definition: Rapidity:

The character of rapidity can be perceived in two ways, either directly by observing a time or frequency response, or indirectly through the concept of dynamic error (a rapid system enables the pursuit of an input that rapidly varies and hence a low dynamic error) [13].

Inserting a controller can be done in different configurations, for this case, direct insertion in cascade with the the plant is considered as shown in Fig 3.6.

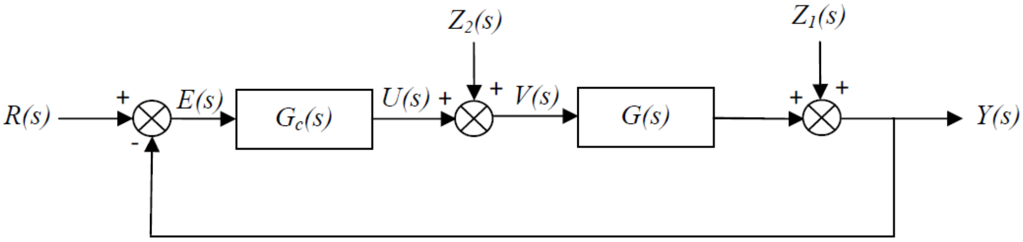


Figure 3.6: Close loop configuration.

Such that:

- $R(s)$: reference signal.
- $E(s)$: error signal.
- $V(s)$: control signal for the plant.
- $U(s)$: control signal for the controller.
- $Z_1(s)$: output disturbance.
- $Z_2(s)$: input disturbance.
- $G_c(s)$: transfer function of the regulator.
- $G(s)$: transfer function of the plant.

The error is defined as the difference between the reference signal and the output signal. i.e. $E(s) = R(s) - Y(s)$. and the open loop transfer function becomes:

$$L(s) = G_c(s)G(s)$$

The controller is added to maintain an error as close as possible to zero, also, fix the output to a constant value regardless of the disturbances (regulation), and bring the output to a predetermined value (tracking mode).

3.5 Controller types

PID controllers relate the error to the actuating signal either in a proportional (P), integral (I), or derivative (D) manner. PID controllers can also relate the error to the actuating signal using a combination of these controls.

3.5.1 Proportional controller (P):

One type of action used in PID controllers is proportional control [14]. Proportional control is a form of feedback control. It is the simplest form of continuous control that can be used in a closed-looped system.

A P-only control minimizes the fluctuation in the process variable, but it does not always bring the system to the desired set point. It provides a faster response than most other controllers, initially allowing the P-only controller to respond a few seconds faster. However, as the system becomes more complex (i.e. more complex algorithm) the response time difference could accumulate, allowing the P-controller to possibly respond even a few minutes faster. Although the P-only controller does offer the advantage of faster response time, it produces deviation from the set point. This deviation is known as the offset, and it is usually not desired in a process.

P-control linearly correlates the controller output (actuating signal) to the error (difference between measured signal and set point). This P-control behavior is mathematically illustrated in Eq 3.11.

$$u(t) = K_c e(t) + b \tag{3.11}$$

Such that:

- $u(t)$: is the controller output.
- K_c : is the controller gain.
- $e(t)$: is the error.
- b : is the bias.

The following figure shows the bode plot of a P controller, where $K_c = 1$ and $b = 0$.

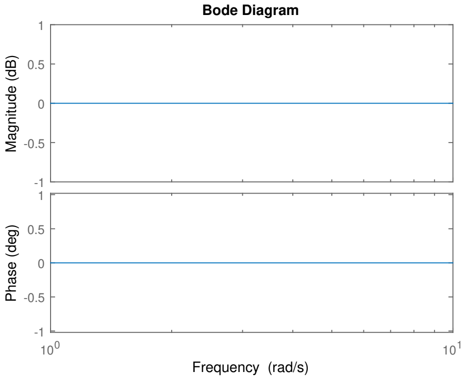


Figure 3.7: Bode plot of a P controller.

3.5.2 Integral (I) Control

Another type of action used in PID controllers is integral control [14]. Integral control is the second form of feedback control. It is often used because it can remove any deviations that may exist. Thus, the system returns to both a steady state and its original setting. A negative error will cause the signal to the system to decrease, while a positive error will cause the signal to increase. However, I-only controllers are much slower in their response time than P-only controllers because they are dependent on more parameters. If it is essential to have no offset in the system, then an I-only controller should be used, but it will require a slower response time. This slower response time can be reduced by combining I-only control with another form, such as P or PD control.

I-only controls are often used when measured variables need to remain within a very narrow range and require fine-tuning control. I-control affect the system by responding to accumulated past errors. The philosophy behind the integral control is that deviations will be affected in proportion to the cumulative sum of their magnitude.

I-control correlates the controller output to the integral of the error. The integral of the error is taken with respect to time. It is the total error associated over a specified amount of time. This I-control behavior is mathematically illustrated in Eq 3.12.

$$u(t) = \frac{1}{T_i}\int e(t)dt + u(t_0) \tag{3.12}$$

Such that:

- $u(t)$: is the controller output.
- T_i : is the integral time.
- $e(t)$: is the error.
- dt : is the differential change in time.

- $u(t_0)$: is the controller output before integration.

The following figure shows the bode plot of a I controller.

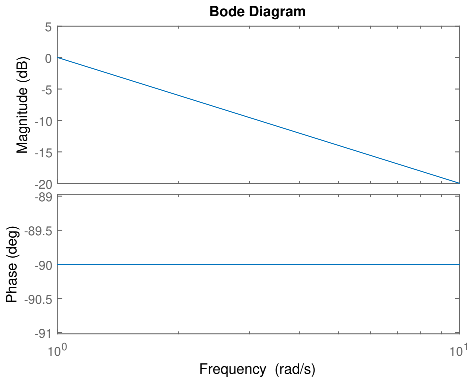


Figure 3.8: Bode plot of a I controller.

3.5.3 Derivative (D) Control

Another type of action used in PID controllers is the derivative control [14]. Unlike P-only and I-only controls, D-control is a form of feed forward control. D-control anticipates the process conditions by analyzing the change in error. It functions to minimize the change of error, thus keeping the system at a consistent setting.

The primary benefit of D controllers is to resist change in the system, the most important of these being oscillations. The control output is calculated based on the rate of change of the error with time. The larger the rate of the change in error, the more pronounced the controller response will be. Unlike proportional and integral controllers, derivative controllers do not guide the system to a steady state. Because of this property, D controllers must be coupled with P, I, or PI controllers to properly control the system.

D-control correlates the controller output to the derivative of the error. The derivative of the error is taken with respect to time. It is the change in error associated with the change in time. This D-control behavior is mathematically illustrated in Eq 3.13.

$$u(t) = T_d \frac{de}{dt} \tag{3.13}$$

Such that:

- $u(t)$: is the controller output.
- T_d : is the derivative time constant
- de : is the differential change in error.
- dt : is the differential change in time.

The following figure shows the bode plot of a D controller.

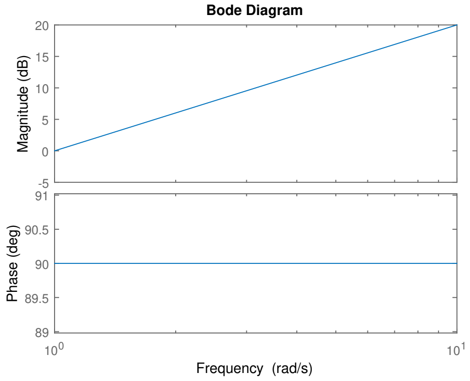


Figure 3.9: Bode plot of a D controller.

3.5.4 Proportional Integral (PI) control:

One combination is the PI-control [14], which lacks the D-control of the PID system. PI control is a form of feedback control. It provides a faster response time than I-only control due to the addition of the proportional action. PI control stops the system from fluctuating, and it is also able to return the system to its set point. Although the response time for PI-control is faster than the I-only control, it is still up to 50% slower than the P-only control. Therefore, in order to increase response time, PI control is often combined with D-only control.

PI-control correlates the controller output to the error and the integral of the error. This PI-control behavior is mathematically illustrated in Eq 3.14.

$$u(t) = K_c\left(e(t) + \frac{1}{T_i}\int e(t)dt\right) + u(t_0) \tag{3.14}$$

Such that:

- $u(t)$: is the controller output.
- K_c : is the controller gain.
- $e(t)$: is the error.
- T_i : is the integral time.
- dt : is the differential change in time.
- $u(t_0)$: is the initial value of the controller.

The following figure shows the bode plot of a PI controller.

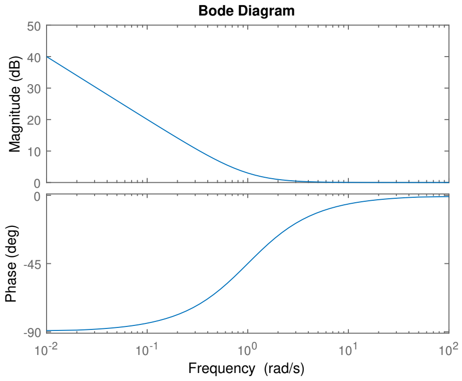


Figure 3.10: Bode plot of a PI controller.

3.5.5 Proportional-derivative (PD) control:

Another combination of controls is the PD-control [14], which lacks the I-control of the PID system. PD-control is a combination of feedforward and feedback control because it operates on both the current process conditions and predicted process conditions. In PD-control, the control output is a linear combination of the error signal and its derivative. PD-control contains the proportional control's damping of the fluctuation and the derivative control's prediction of process error.

As mentioned, PD-control correlates the controller output to the error and the derivative of the error. This PD-control behavior is mathematically illustrated in Eq 3.15.

$$u(t) = K_c\left(e(t) + T_d\frac{de}{dt}\right) + u(t_0) \tag{3.15}$$

Such that:

- $u(t)$: controller output.
- K_c : the proportional gain.
- $e(t)$: the error.
- T_d : is the derivative time constant
- de : is the differential change in error.
- dt : is the differential change in time.
- $u(t_0)$: the initial of controller.

The equation indicates that the PD-controller operates like a simplified PID-controller with a zero integral term. Alternatively, the PD-controller can also be seen as a combination of the P-only and D-only control equations. In this control, the purpose of the D-only control is to predict the error in order to increase the stability of the closed loop system.

P-D control is not commonly used because of the lack of the integral term. Without the integral term, the error in steady state operation is not minimized.

The following figure shows the bode plot of a PD controller.

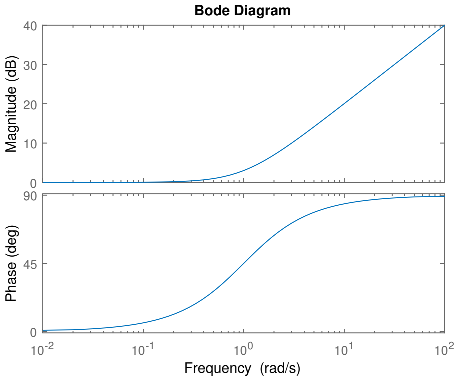


Figure 3.11: Bode plot of a PD controller.

3.5.6 Proportional-Integral-Derivative (PID) Control

Proportional-integral-derivative control is a combination of all three types of control methods. PID-control is most commonly used because it combines the advantages of each type of control. This includes a quicker response time because of the P-only control, along with the decreased/zero offset from the combined derivative and integral controllers. This offset was removed by additionally using the I-control. The addition of D-control greatly increases the controller's response when used in combination because it predicts disturbances to the system by measuring the change in error. On the contrary, as mentioned previously, when used individually, it has a slower response time compared to the quicker P-only control [14].

Although the PID controller seems to be the most adequate controller, it is also the most expensive controller. Therefore, it is not used unless the process requires the accuracy and stability provided by the PID controller.

PID-control correlates the controller output to the error, integral of the error, and derivative of the error. This PID-control behavior is mathematically illustrated in Eq 3.16.

$$u(t) = K_c\left(e(t) + T_d\frac{de}{dt} + \frac{1}{T_i}\int e(t)dt\right) + u(t_0) \tag{3.16}$$

Such that:

- $u(t)$: is the controller output.
- K_c : is the controller gain.
- $e(t)$: is the error.
- T_d : is the derivative time constant.
- de : is the differential change in error.
- dt : is the differential change in time.
- Ti : is the integral time.
- $u(t_0)$: is the initial value of controller.

As shown in the above equation, PID control is the combination of all three types of control. In this equation, the gain is multiplied with the integral and derivative terms, along with the proportional term, because in PID combination control, the gain affects the I and D actions as well. Because of the use of derivative control, PID control cannot be used in processes where there is a lot of noise, since the noise would interfere with the predictive, feedforward aspect. However, PID control is used when the process requires no offset and a fast response time.

The following figure shows the bode plot of a PID controller.

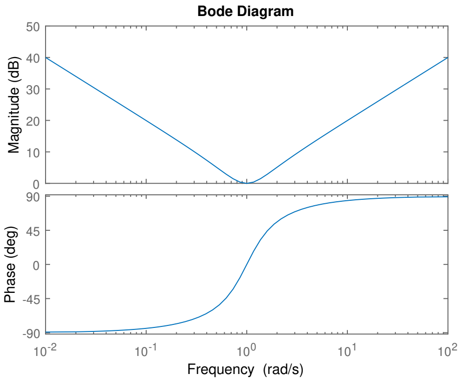


Figure 3.12: Bode plot of a PID controller.

Remarque: For the graphs shown above, the values of each constant is 1.

3.5.7 Summary tables:

The following tables show a summary of the advantages and disadvantages of each type of controls P,I, and D.

A guide for the typical uses of the various controllers.

Table 3.1: Advantages and disadvantages of controls

	P	I	D
Advantages	Reduce steady-state error, thus make the system more stable, making the slow response of the overdamped system faster	Known as reset controllers, they can set the controlled variable back the set point.	Improve the transient response of the system
Disadvantages	Cause some offsets, increase the maximum overshot	Make the system unstable due to slow respond towards the produced error	Slow response time

Table 3.2: Estimate and uses of controls

Controller	Estimates	When to use
P	Present	Systems with slow response, systems tolerant to offset
I	Back	Not often used alone, as it is slow
D	Forward	Not used alone because of noise sensitivity, but, they minimize the transient errors like overshoot and oscillations
PI	Present and back	Reduce both the rise time and the steady state errors
PD	Present and forward	Reduce the transients like rise time, overshoot, and oscillations
PID	All of the time	Often used, most robust

3.5.8 General Guidelines for Designing a PID Controller

When designing a PID controller for a given system, general guidelines to obtain the desired response are as follows:

- Obtain the transient response of the closed-loop transfer function and determine what needs to be improved.
- Insert the proportional controller, Design the value of 'K_c' through Routh-Hurwitz or suitable software.
- Add an integral part to reduce steady-state error.
- Add the derivative part to increase damping (damping should be between 0.6-0.9). The derivative part will reduce overshoots and transient time.

Remark: it is worth mentioning that the above steps of tuning parameters (designing a control system) are general guidelines. There are no fixed steps for designing controllers.

3.6 Conclusion

This chapter addressed the concept of fault tolerant control providing a description of fault types based on location and induced effects on the system performance, and how to inject faults in both nonlinear and linear systems.

It also goes over the estimation approach for both states and faults, in the process, it introduces observers in control systems, with the UIO and Luenberger observer taken as case studies.

Finally, it discusses feedback control, for which PID controllers are described, with emphasis on the advantages, disadvantages, and typical applications of the various types of PID controllers.

Chapter 4

Simulation and results

4.1 Introduction

This chapter is devoted to an application of the methodologies presented in the previous chapters on a multivariate nonlinear model of a hydraulic process experimental, called DTS-200 station shown in Fig 4.1.

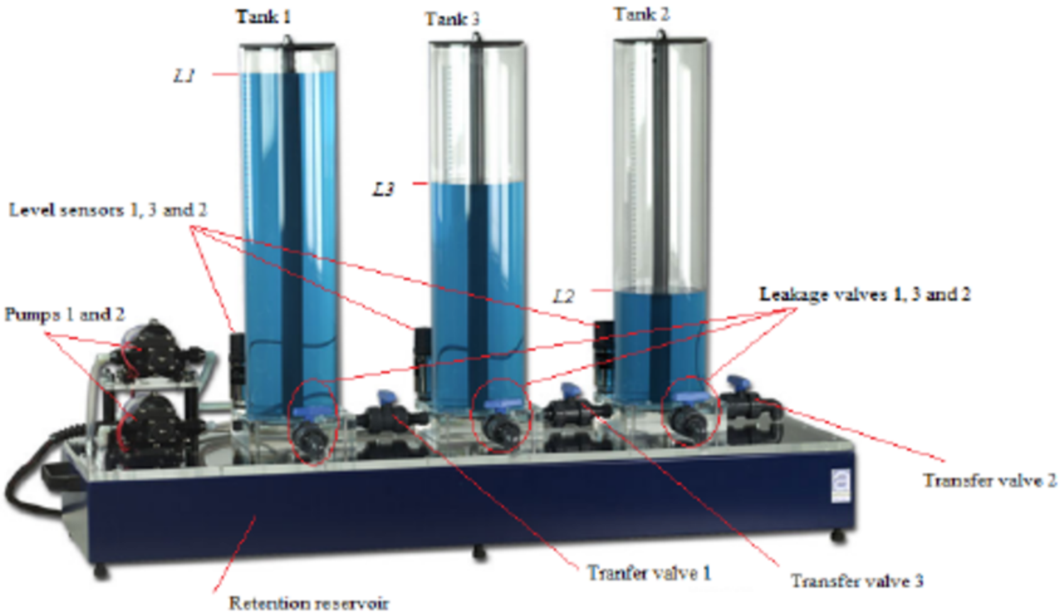


Figure 4.1: Station DTS-200

System description:

The system consists of three interconnected cylindrical tanks, two pumps, five valves, pipes, a water reservoir in the bottom denoted subscript T_0, measurement of liquid levels and other elements. The pumps pump water from the bottom reservoir to the top of the left and right tanks. A simplified scheme of the system is shown in Fig 4.2. The pump P_1 controls the inflow to tank T_1 while the pump P_2 controls the liquid inflow to tank T_2. There is no pump connected to the middle tank T_3. The characteristic of the flow between tank T_1 and tank T_3 can be affected by valve V_1, flow between tanks T_3 and T_2 can be affected by valve V_2 and the outflow of tank T_2 can be affected by valve V_3. The system also provides the capability of simulating leakage from individual tanks by opening the valves V_{l1} and V_{l2}.

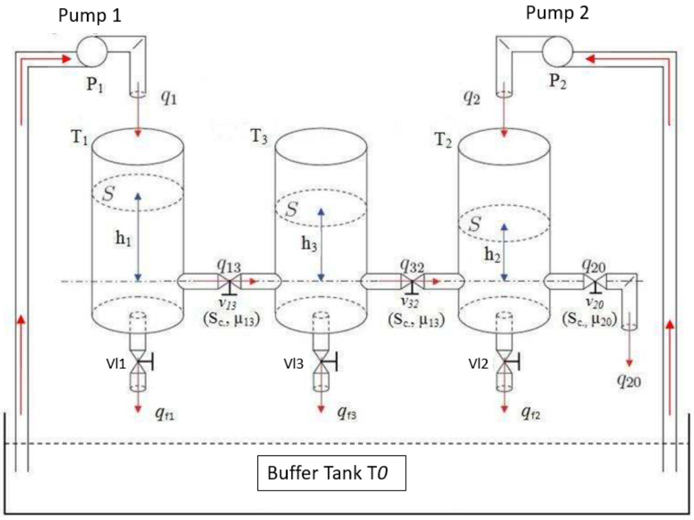


Figure 4.2: Block diagram of the hydraulic system

With:

- T_0 : buffer tank.
- T_i : tank i.
- P_i : feeding pump of tank i.
- h_i : water level in tank $i[m]$.
- V_{li} : leakage valve between tank i and buffer tank.
- V_{ij} : communication valve between tank i and tank j.
- μ_{ij} : viscosity coefficient of valve of communication between Tank i and Tank j.
- q_i : rate of flow of pump i.
- q_{li} : leakage flow of valve V_{li}.
- q_{ij} : circulating flow of the valve of communication V_{ij}.
- S : cylindrical tank cross section $[m^2]$.
- S_s : cross section of communication valves and leakage valves $[m^2]$.

4.2 Mathematical model of the system

As part of our application, the configuration of the schematized process in Fig 4.3 is chosen. The latter allows to work with a device made up of three tanks, the flow of water out of the cylindrical tanks into the buffer tank will be provided by the outlet located at the end of tank T_2, which requires the complete opening of the communication valves V_{13}, V_{32} and V_{20}, and the complete closing of leakage valves V_{l1} and V_{l2}. The three tanks are cylinders of revolution, of section $S = 0.0154\ m^2$.

According to the fundamental law of conservation of matter, the variation in the volume of water stored by a unit of time by a tank is given by the difference between the flow entering the tank and the outgoing flow of the latter, i.e.:

$$\begin{bmatrix} \text{Water accumulation} \\ \text{in Tank } i \end{bmatrix} = \begin{bmatrix} \text{The flow of water} \\ \text{entering Tank } i \end{bmatrix} - \begin{bmatrix} \text{The flow of water} \\ \text{leaving Tank } i \end{bmatrix}$$

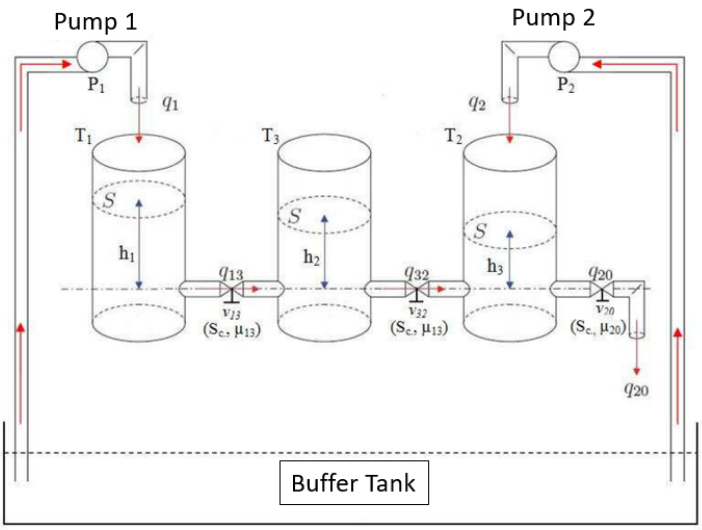


Figure 4.3: Synoptic diagram of the experimental configuration chosen for our application

- The mass balance in tank 1 gives the following relation:

$$S \cdot \frac{dh_1}{dt} = q_1 - q_{13} \tag{4.1}$$

- The mass balance in tank 2 gives the following relation:

$$S \cdot \frac{dh_2}{dt} = q_2 + q_{32} - q_{20} \tag{4.2}$$

- The mass balance in tank 3 gives the following relation:

$$S \cdot \frac{dh_3}{dt} = q_{13} - q_{32} \tag{4.3}$$

The following state representation results in a model analysis of the system represented by three nonlinear differential equations of prime order:

$$\sum NL = \begin{cases} S \cdot \frac{dh_1}{dt} = q_1 - q_{13} \\ S \cdot \frac{dh_2}{dt} = q_2 + q_{32} - q_{20} \\ S \cdot \frac{dh_3}{dt} = q_{13} - q_{32} \end{cases} \tag{4.4}$$

Such that: the q_{ij} parameters represent the flow of liquid from the i^{th} tank to the j^{th} $\{i, j \in (1,2,3), \ \forall i \neq j\}$, and are calculated using Torricelli's law

The general formula is given by:

$$q_{ij} = \mu_{ij}.S_n.sign\big(h_i(t) - h_j(t)\big).\sqrt{2g|h_i(t) - h_j(t)|} \tag{4.5}$$

Where:

- μ_{ij}: is the outflow coefficient.
- $sign\big(h_i(t) - h_j(t)\big)$: is the sign of the argument $h_i(t) - h_j(t)$.
- g: is the acceleration as a result of gravity.

For the output flow, it is represented as follows:

$$q_{20}(t) = \mu_{20}.S_n.\sqrt{2.g|h_2(t)|} \tag{4.6}$$

For this study the only case considered is when the level in the three tanks respects the following inequality $h_1 > h_3 > h_2$. The resulting equations to calculate the partial flows are as follows:

$$\begin{cases} q_{13} = \mu_{13}.S_n.sign\big(h_1(t) - h_3(t)\big).\sqrt{|h_1 - h_3|} \\ q_{32} = \mu_{32}.S_n.sign\big(h_3(t) - h_2(t)\big).\sqrt{|h_3 - h_2|} \\ q_{20} = \mu_{20}.S_n.\sqrt{|h_2|} \end{cases} \tag{4.7}$$

Replacing Eq 4.7 in Eq 4.4 yields the following mathematical model:

$$\begin{cases} \dot{h_1} = \frac{1}{S}[q_1 - C_{13}.\sqrt{h_1 - h_3}] \\ \dot{h_2} = \frac{1}{S}[q_2 + C_{32}.\sqrt{h_3 - h_2} - C_{20}.\sqrt{h_{20}}] \\ \dot{h_3} = \frac{1}{S}[C_{13}.\sqrt{h_1 - h_3} - C_{32}.\sqrt{h_3 - h_2}] \end{cases} \tag{4.8}$$

The equivalent state space representation can be obtained easily by making:

$$\begin{cases} h_i \to x_i, & \forall(i \in \{1,2,3\}) \\ q_i \to u_j, & \forall(j \in \{1,2\}) \end{cases}$$

$$\begin{cases} \dot{x_1} = \frac{1}{S}[u_1 - C_{13}.\sqrt{x_1 - x_3})] \\ \dot{x_2} = \frac{1}{S}[u_2 + C_{32}.\sqrt{x_3 - x_2} - (C_{20}.\sqrt{x_{20}}] \\ \dot{x_3} = \frac{1}{S}[C_{13}.\sqrt{x_1 - x_3} - (C_{32}.\sqrt{x_3 - x_2})] \end{cases} \tag{4.9}$$

$$\begin{cases} y_1 = x_1 \\ y_2 = x_2 \end{cases}$$

Such that: the state vectors is $x = [x_1,x_2,x_3]^T = [h_1,h_2,h_3]^T$ whereas the control vector $u = [u_1,u_2]^T = [q_1,q_2]^T$, the output vector is represented by $y_1(t)$ and $y_2(t)$.

The parameters C_{13}, C_{32} and C_{20} are constants, they are determined experimentally.

Variable	Symbol	Value	Unit
Cross section of the Tank	S	1.54^{-4}	m^2
Cross section of the pipes	S_n	0.5×10^{-4}	m^2
Coefficients	C_{13}	1.01×10^{-4}	
	C_{32}	0.99×10^{-4}	
	C_{20}	1.32×10^{-4}	
Maximal flow	$q_{1max} = q_{2max}$	10^{-4}	m^3/s
Maximal level	$h_{imax}(i = 1,2,3)$	0.60	m

Table 4.1: Parameters of the system.

Simulating the system using Matlab Simulink software gives the following results:

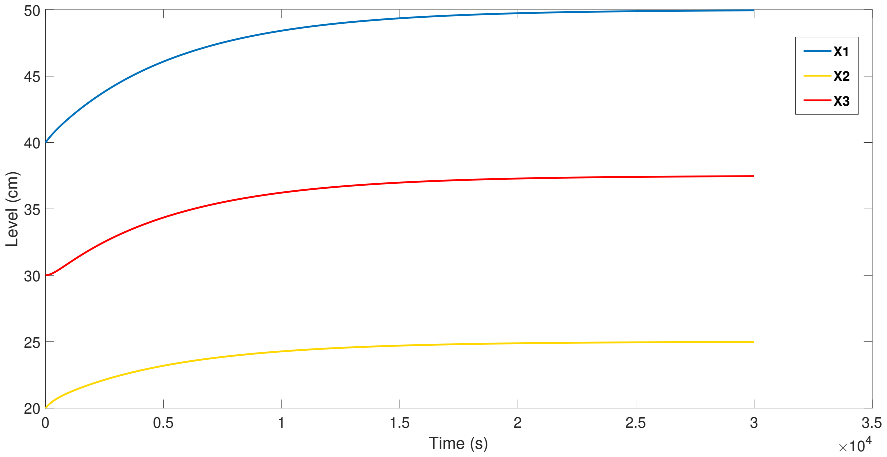


Figure 4.4: The behavior of the nonlinear system.

Discussion:

As the figure shows, the system has a first order behavior, starting from the initial levels set previously and evolving exponentially towards the the final values, the system passes by a transient state and finally set into a stead-state.

4.3 Linearization around an equilibrium point

The linearization process follows the steps explained in chapter 1.

First, a proper choice of the equilibrium point, and the control required to obtain it are demonstrated in the following table:

Equilibrium point	i=1	i=2	i=3
h_{i0} (m)	0.40	0.30	0.20
$q_{i0}(m^3/s) \times 10^{-5}$	3.2200	2.7897	

Table 4.2: Equilibrium point value.

The deviation variables for the systems are:

$$\delta x = x - x_0 \begin{cases} \delta x_1 = x_1 - x_{10} \\ \delta x_2 = x_2 - x_{20} \\ \delta x_3 = x_3 - x_{30} \end{cases} \tag{4.10}$$

$$\delta x = u - u_0 \begin{cases} \delta u_1 = u_1 - u_{10} \\ \delta u_2 = u_2 - u_{20} \end{cases} \tag{4.11}$$

In order to do a Jacobian linearization, the states and output must be written in the form:

$$\dot{x} = f(x,u)$$
$$y = g(x,u)$$

The state $\dot{x}$ is expended to three states, white the output y is expended to two outputs results in:

$$\begin{cases} \dot{x_1} = \frac{1}{S}[u_1 - C_{13}.\sqrt{x_1 - x_3})] = f_1(x,u,t) \\ \dot{x_2} = \frac{1}{S}[u_2 + C_{32}.\sqrt{x_3 - x_2} - (C_{20}.\sqrt{x_{20}}] = f_2(x,u,t) \\ \dot{x_3} = \frac{1}{S}[C_{13}.\sqrt{x_1 - x_3} - (C_{32}.\sqrt{x_3 - x_2})] = f_3(x,u,t) \end{cases} \tag{4.12}$$

$$\begin{cases} y_1 = x_1 = g_1(x,u,t) \\ y_2 = x_2 = g_2(x,u,t) \end{cases}$$

$$\begin{bmatrix} \delta\dot{x_1} \\ \delta\dot{x_2} \\ \delta\dot{x_3} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{\substack{x=x_0 \\ u=u_0}} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} \tag{4.13}$$

$$\begin{bmatrix} \delta y_1 \\ \delta y_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} \end{bmatrix}_{\substack{x=x_0 \\ u=u_0}} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \end{bmatrix} + \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} \tag{4.14}$$

$$A = \begin{bmatrix} -\frac{C_{13}}{2S\sqrt{x_{10}-x_{30}}} & 0 & \frac{C_{13}}{2S\sqrt{x_{10}-x_{30}}} \\ 0 & -\frac{C_{32}}{2S\sqrt{x_{30}-x_{20}}} - \frac{C_{20}}{2S\sqrt{x_{20}}} & -\frac{C_{32}}{2S\sqrt{x_{30}-x_{20}}} \\ \frac{C_{13}}{2S\sqrt{x_{10}-x_{30}}} & \frac{C_{32}}{2S\sqrt{x_{30}-x_{20}}} & -\frac{C_{13}}{2S\sqrt{x_{10}-x_{30}}} - \frac{C_{32}}{2S\sqrt{x_{30}-x_{20}}} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{S} & 0 \\ 0 & \frac{1}{S} \\ 0 & 0 \end{bmatrix} \quad , \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad , \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- **Numerical Application**

Replacing with:

$x = [x_{10}, x_{20}, x_{30}]^T = [0.4, 0.2, 0.3]^T$ and $u = [u_1, u_2]^T = [3.2200, 2.7897]^T \times 10^{-5}$

$$A = \begin{bmatrix} -0.0105 & 0 & 0.0105 \\ 0 & -0.0199 & 0.0102 \\ 0.0105 & 0.0102 & -0.0207 \end{bmatrix} \quad , \quad B = \begin{bmatrix} 0.0065 & 0 \\ 0 & 0.0065 \\ 0 & 0 \end{bmatrix}$$

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

Representing the state space in Simulink, such that it is connected to same input as the nonlinear system and the same initial conditions are added to compares the outcomes:

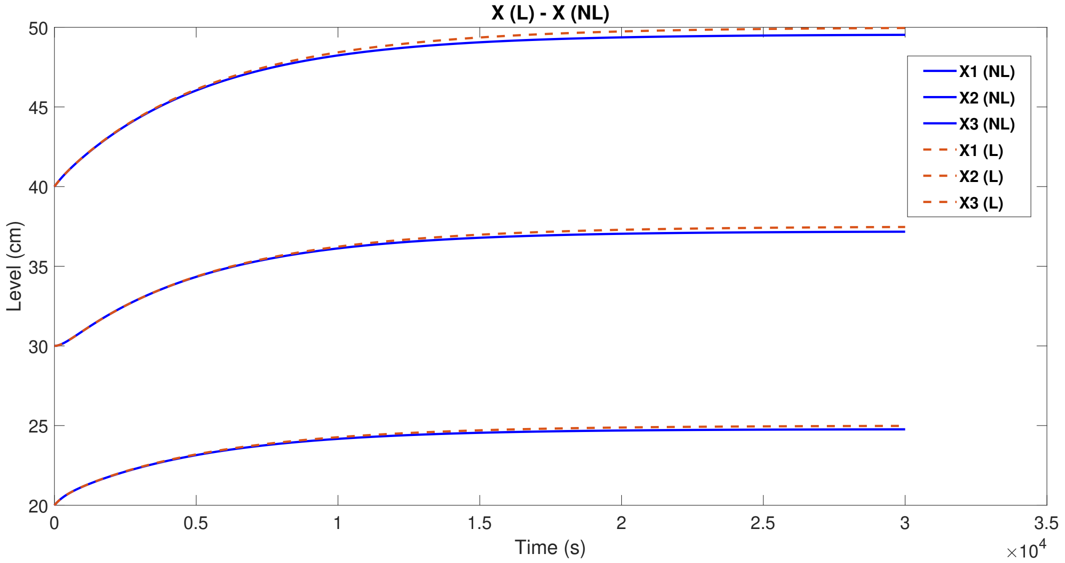


Figure 4.5: Linear system behavior with leaks

To test the accuracy of the linear system, a second operating point is chosen, such that it deviates from the original one by 20%, thus, the new equilibrium point is $Xe = [0.5, 0.25, 0.375]$. Calculating the error between the linear system and the nonlinear system for such deviation yields the following values:

$$e_{hi} = \begin{cases} 4.31\% & for \;\; i = 1 \\ 2.86\% & for \;\; i = 2 \\ 4.92\% & for \;\; i = 3 \end{cases}$$

The error resulting from the deviation between the nonlinear model and the linear model is:

$$e_{hi} < 5\%, \quad for \;\; i = \{1, 2, 3\}$$

The error between the nonlinear and the linear system is acceptable, therefore, it is safe to say that linear approximation represents, accurately, the behavior of the nonlinear system around the chosen equilibrium point.

4.3.1 Transfer function matrix representation

The transfer function matrix is obtained using the following formula:

$$G(s) = \frac{h(s)}{q(s)} = C(sI_n - A)^{-1}B + D \tag{4.15}$$

Replacing the values of A,B,C, and D yields:

$$G(s) = \begin{bmatrix} \frac{64.94s^2+2.633s+0.01989}{s^3+0.05101s^2+0.000621s+1.031e-06} & \frac{0.006943}{s^3+0.05101s^2+0.000621s+1.031e-06} \\ \\ \frac{0.006943}{s^3+0.05101s^2+0.000621s+1.031e-06} & \frac{64.94s^2+2.022s+0.006943}{s^3+0.05101s^2+0.000621s+1.031e-06} \end{bmatrix}$$

4.3.2 Study of stability

To check for stability, it is mandatory first to verify is controlable and observable, the controlability and observability matrices are:

$$W_c = \begin{bmatrix} 64.9351 & 0 & -0.6789 & 0 & 0.0142 & 0.0069 \\ 0 & 64.9351 & 0 & -1.2902 & 0.0069 & 0.0324 \\ 0 & 0 & 0.6789 & 0.6641 & -0.0211 & -0.0269 \end{bmatrix}$$

$$W_o = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 1.0000 & 0 \\ -0.0105 & 0 & 0.0105 \\ 0 & -0.0199 & 0.0102 \\ 0.0002 & 0.0001 & -0.0003 \\ 0.0001 & 0.0005 & -0.0004 \end{bmatrix}$$

Using Matlab to check, $rank(W_c) = rank(W_o) = 3$, thus, both the controlability and observability matrices are full rank, the system is, therefore, controlable and observable, this means that the poles of the system are the eigenvalues of matrix A.

Stability is an important concept is control engineering. For a system to be stable, all eigenvalues of the matrix A have to be real strictly negative numbers.

$$det(\lambda I_n - A) = \begin{cases} \lambda_1 = -0.0333 \\ \lambda_2 = -0.0158 \\ \lambda_3 = -0.0020 \end{cases} \tag{4.16}$$

All the eigenvalues of the matrix A are real and have negative signs, therefore, the system is stable.

4.4 Interaction analysis using the RGA method

The RGA aims for finding the best pairing that corresponds to good controller performance. The steady-state gain that is used in pairing analysis purpose is obtained from closed and open loop simulation of the process.

There exist two possible configurations:

1. The first possible configuration : [q_1 - h_1][q_2 - h_2]
2. The second possible configuration : [q_1 - h_2][q_2 - h_1]

Applying the RGA process explained previously in chapter 2 gives:

- **The static gain matrix:**

$$K_s = \lim_{s\to 0} G(s) = \begin{bmatrix} 1.9292 & 0.6734 \\ 0.6734 & 0.6734 \end{bmatrix}$$

- **The inverse of the static gain matrix K_s:**

$$K_s^{-1} = \begin{bmatrix} 0.7963 & -0.7963 \\ -0.7963 & 2.2813 \end{bmatrix}$$

- **The transpose of the matrix $(K_s)^{-1}$:**

$$(K_s^{-1})^T = \begin{bmatrix} 0.7963 & -0.7963 \\ -0.7963 & 2.2813 \end{bmatrix}$$

- **The relative gain matrix matrix:**

$$RGA = K_s \times [K_s^{-1}]^T = \begin{bmatrix} 1.5363 & -0.5363 \\ -0.5363 & 1.5363 \end{bmatrix}$$

Knowing that each row corresponds to an output and each column corresponds to an input, and the fact that choosing negative gain results in system instability, scanning the rows for columns containing the closest element to one such that it is positive results in the following optimal configuration:

$$\begin{cases} q_1 \longrightarrow h_1 \\ q_2 \longrightarrow h_2 \end{cases}$$

4.5 Algebraic multi-variable control (decoupled)

4.5.1 Calculating the decoupler

For this case, the ideal decoupling is chosen. The open loop transfer function matrix has the following form:

$$G(s)D(s) = T(s) = \begin{bmatrix} G_{11}(s) & 0 \\ 0 & G_{22}(s) \end{bmatrix} \tag{4.17}$$

To obtain the decoupling matrix $D(s)$ $G^{-1}(s)$ and $M(s)$ are placed in Eq 4.18, thus:

$$\begin{aligned} D(s) &= G^{-1}(s)T(s) \\ &= \frac{1}{G_{11}(s)G_{22}(s) - G_{12}(s)G_{21}(s)} \begin{bmatrix} G_{22}(s)G_{11}(s) & -G_{12}(s)G_{22}(s) \\ -G_{21}(s)G_{11}(s) & G_{11}(s)G_{22}(s) \end{bmatrix} \end{aligned} \tag{4.18}$$

The numerical application yields transfer functions of the 10^{10}. For a better analysis, the zero-pole cancellation using the 'minreal' function in Matlab is used to eliminate uncontrollable or unobservable states in state-space models, or cancel pole-zero pairs in transfer functions or zero-pole-gain models. The resulting transfer function matrix has minimal order and the same response characteristics as the original model, it allows the user to choose a certain degree of tolerance in which if a straightforward search through the zeros and poles results in a pair within that tolerance, the pair is eliminated, for this study an acceptable degree of tolerance is 0.00001. This reduction technique yields the following result:

$$D_{11}(s) = \frac{s^7+0.1273s^6+0.006333s^5+0.0001556s^4+1.974e-06s^3+1.247e-08s^2+3.515e-11s+3.376e-14}{s^7+0.1273s^6+0.006333s^5+0.0001556s^4+1.963e-06s^3+1.188e-08s^2+2.805e-11s+2.198e-14}$$

$$D_{12}(s) = -\frac{0.0001069s^5+1.028e-05s^4+3.455e-07s^3+4.779e-09s^2+2.531e-11s+3.376e-14}{s^7+0.1273s^6+0.006333s^5+0.0001556s^4+1.963e-06s^3+1.188e-08s^2+2.805e-11s+2.198e-14}$$

$$D_{21}(s) = -\frac{0.0001069s^5+8.835e-06s^4+2.492e-07s^3+2.765e-09s^2+1.052e-11s+1.176e-14}{s^7+0.1273s^6+0.006333s^5+0.0001556s^4+1.963e-06s^3+1.188e-08s^2+2.805e-11s+2.198e-14}$$

$$D_{22}(s) = \frac{s^7+0.1273s^6+0.006333s^5+0.0001556s^4+1.974e-06s^3+1.247e-08s^2+3.515e-11s+3.376e-14}{s^7+0.1273s^6+0.006333s^5+0.0001556s^4+1.963e-06s^3+1.188e-08s^2+2.805e-11s+2.198e-14}$$

4.5.2 Calculating the RGA matrix

To test the effectiveness of the ideal decoupling, the RGA method is used again for the new decoupled system. the following result is obtained:

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

Clearly, the test of the RGA shows that the new system does not include any coupling, thus, it is safe to say that the system is ideally decoupled.

The transfer function matrix of the decoupled system is of form:

$$M(s) = \begin{bmatrix} G_{11}(s) & 0 \\ 0 & G_{22}(s) \end{bmatrix} \tag{4.19}$$

4.6 Actuator disturbance and sensor noise

4.6.1 Actuator disturbance

Based on the fact that disturbance is a signal that represents unwanted inputs which affect the output thereby increase the error, disturbance in the system can be modeled as an addition mass that flows out of the tanks

$$\begin{cases} \theta_{T_1}\sqrt{2gh_1} \\ \theta_{T_2}\sqrt{2gh_2} \\ \theta_{T_3}\sqrt{2gh_3} \end{cases}$$

Since the value of h_3 is immeasurable for this case study, only leaks occurring in tanks T_1 and T_2 are considered.

In Simulink, it is possible to control the time of occurrence of such disturbances in order to fully observe their effect on the behaviour of the system, the first disturbance occurs during the time interval $[t_{d11} = 5000; t_{d12} = 10000]$, while the second disturbance occurs in the following interval $[t_{d21} = 15000; t_{d22} = 20000]$, also, the constants θ for both tanks are set to a constant value:

$$\theta_{T_1} = \theta_{T_2} = 50\%$$

For the nonlinear system, the disturbance is injected as a negative additional terms in the dynamic equations describing the behavior of the system:

$$\begin{cases} \dot{h_1} = \frac{1}{S}[q_1 - C_{13}.\sqrt{h_1 - h_3})] - \theta_{T_1}\sqrt{2gh_1} \\ \dot{h_2} = \frac{1}{S}[q_2 + C_{32}.\sqrt{h_3 - h_2} - (C_{20}.\sqrt{h_{20}}] - \theta_{T_2}\sqrt{2gh_2} \\ \dot{h_3} = \frac{1}{S}[C_{13}.\sqrt{h_1 - h_3} - (C_{32}.\sqrt{h_3 - h_2})] \end{cases} \tag{4.20}$$

For the linear system, another matrix, E_d, is introduced, this allows entering the faults correctly to the system. Since they effect the system by reducing the level in each tank, the matrix E_d is entered with a minus sign, the state space equations become

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) - E_d d(t) \\ y(t) &= Cx(t) \end{aligned} \tag{4.21}$$

Such that:

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

Injecting the faults as previously explained gives the following graphs:

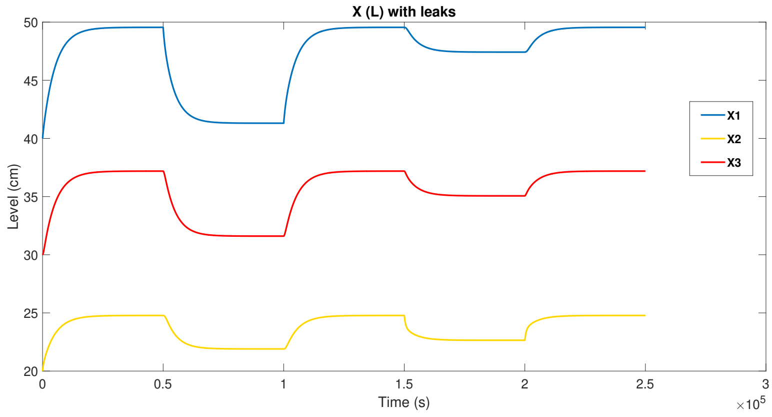


Figure 4.6: Linear system behavior with leaks

There is an immediate disturbance in the system once the leaks are entered. Due to the flow of the liquid in a specific direction ($T_1 \dashrightarrow T_3 \dashrightarrow T_2$), the leak in tank T_1 has more significant effect on the overall behaviour of the system compared to that in tank T_2. Moreover, the system returns back the original state once the leaks are over.

4.6.2 Sensor noise

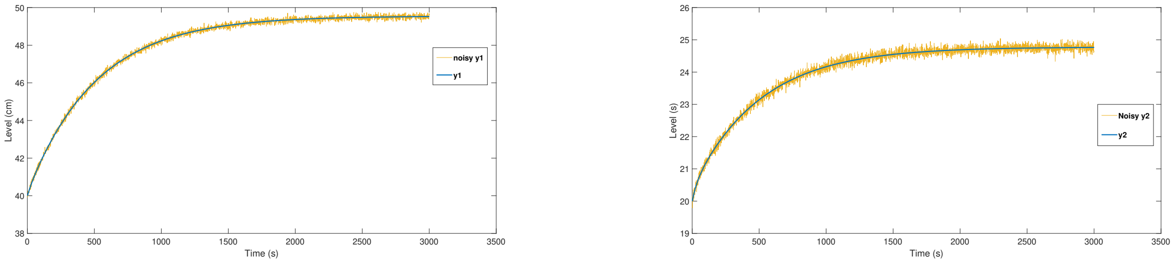
Sensor noise can be injected as a random variations of sensor output unrelated to variations in sensor input. For this part white noise is added to the output signal, the system is then described as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) - E_d d(t) \\ y(t) &= Cx(t) + W_n \end{aligned} \tag{4.22}$$

Such that: W_n is a white noise signal characterized by:

$$\begin{cases} Mean(W_n) = 0 \\ Var(W_n) = 1 \end{cases}$$

Fig 4.7 illustrates the outputs along with the noisy output:



a Sensor 1 noise.

b Sensor 2 noise.

Figure 4.7: Sensors noise.

4.7 State and faults estimation

4.7.1 Observer design

The aim of this part is to estimate with a high degree of accuracy the states of the system, due to the level of liquid in tank (T_3) being immeasurable. Also, designing an observer has a great advantages in decoupling the disturbances.

Let the system be defined by the following state space representation

$$\begin{cases} \dot{x(t)} = Ax(t) + Bu(t) - E_d d(t) \\ y(t) = Cx(t) \end{cases} \tag{4.23}$$

First, to check weather the the disturbance distribution matrix E_d is full column rank. Using MATLAB yields: rank(E_d)=rank($C^* E_d$)=2.

To design an unknown input observer for the system described by Eq 4.23, let the error; $e(t)$, be defined as the difference between the states $x(t)$ and the estimations $\hat{x}(t)$. i.e:

$$e(t) = x(t) - \hat{x}(t) \tag{4.24}$$

This error must approach zero asymptotically regardless of the process of unknown inputs, $d(t)$, in the system. Furthermore, the structure for the full order UIO is given by the dynamic system:

$$\begin{aligned} \dot{z}(t) &= F_{obs} z(t) + TBu(t) + Ky(t) \\ \hat{x}(t) &= z(t) + Hy(t) \end{aligned} \tag{4.25}$$

Such that:

- $\hat{x}$: is the state estimate.
- z: is the state of full-order dynamic observer.

- F_{obs}, T, K, H: are matrices to be designed for achieving the unknown input decoupling.

In order to achieve this decoupling we expend $\dot{e}(t)$ to obtain:

$$\begin{aligned}\dot{e}(t) = &(A - HCA - K_1C)e(t)\\ &+ [F_{obs} - (A - HCA - K_1C)]z(t)\\ &+ [K_2 - (A - HCA - K_1C]y(t)\\ &+ [T - (I - HC)]Bu(t)\\ &- (HC - I)E_d d(t)\end{aligned} \tag{4.26}$$

As previously explained,tThe following equations must hold:

$$0 = (HC - I)E_d \tag{4.27}$$

$$T = I - HC \tag{4.28}$$

$$F_{obs} = A - HCA - K_1C \tag{4.29}$$

$$K_2 = F_{obs}H \tag{4.30}$$

To check the second existence condition which is (C,A_1) is a detectable pair, where:

$$A_1 = A - E_d[(CE_d)^T CE_d]^{-1}(CE_d)^T CA$$

This gives:

$$\begin{aligned}H &= E_d[(CE_d)^T CE_d]^{-1}(CE_d)^T\\ T &= I - HC\\ A_1 &= TA\end{aligned}$$

$$H = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0 & 0 \end{bmatrix} \quad , \quad T = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 0.5 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad , \quad A_1 = \begin{bmatrix} -0.0052 & 0.0099 & 0.0001 \\ 0.0052 & -0.0099 & -0.0001 \\ 0.0105 & 0.0102 & -0.0207 \end{bmatrix}$$

Next, the observability condition is checked, using Matlab: $rank(obsv(A_1,C) = 3$.

Now, applying pole placement in order to place the observer poles at $[-1, -0.03 + 0.003j, -0.03 - 0.003j]$, and get the matrix K_1.

$$K_1 = \begin{bmatrix} 0.6864 & 0.4741 \\ 0.4385 & 0.3226 \\ 0.4968 & 0.3603 \end{bmatrix}$$

Finally, the observer F_{obs} and K matrices are computed:

$$\begin{aligned}F_{obs} &= A_1 - K_1C\\ K &= K_1 + K_2 = K_1 + F_{obs}H\end{aligned}$$

$$F_{obs} = \begin{bmatrix} -0.6916 & -0.4642 & 0.0001 \\ -0.4333 & -0.3325 & -0.0001 \\ -0.4863 & -0.3500 & -0.0207 \end{bmatrix} \quad , \quad K = \begin{bmatrix} 0.1085 & -0.1038 \\ 0.0556 & -0.0603 \\ 0.0786 & -0.0579 \end{bmatrix}$$

Fig 4.8 illustrates the estimation of the states via a UIO.

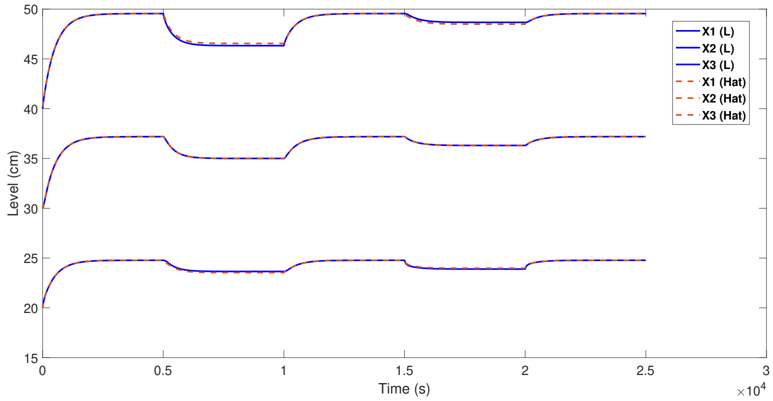
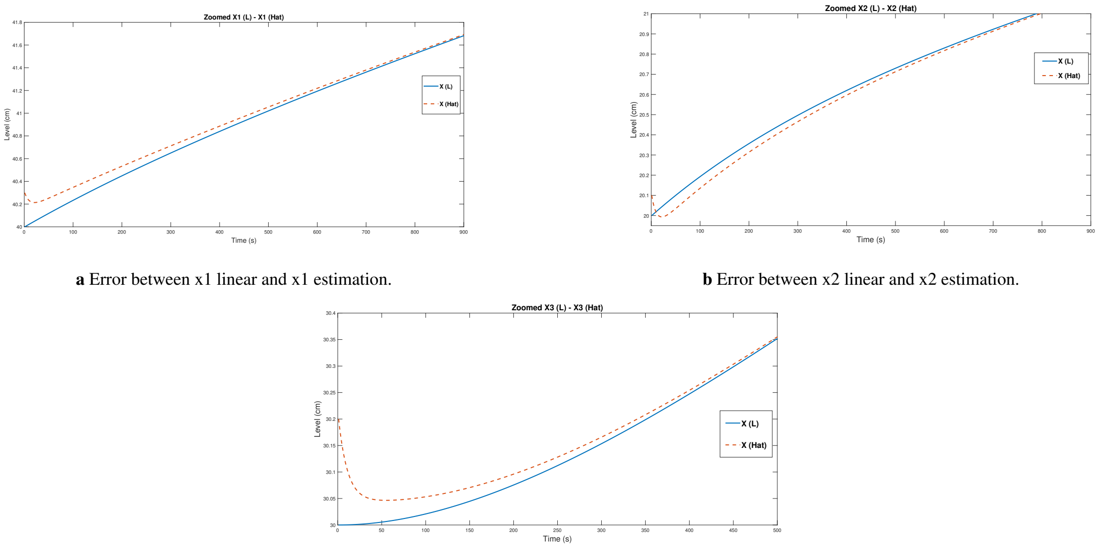


Figure 4.8: Linear system behavior with leaks

In order to better analyze the results obtained, it is required to choose different initial conditions for the observer than those of the linear system, this difference results in a slice error that rapidly fades away as the simulation progresses.

The following figures demonstrates clearly the error occurring at an early stage of the simulation, then, soon the states of the linear system and the estimation of he observer are a total match.



a Error between x1 linear and x1 estimation.

b Error between x2 linear and x2 estimation.

c Error between x3 linear and x3 estimation.

Figure 4.9: The error in the states between the linear system and the UIO.

4.7.2 Modeling faults

For this case, two type of faults are considered: actuator faults, and sensor faults. Different scenarios may be tested for different combinations of faults:

1. **Case 1: 2 actuator faults:** denoted as f_{a1} and f_{a2}; which are pumps faults. The state space representation of a system with actuator faults is as follows:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) - E_d d + F_a f_a \\ y(t) &= Cx(t)\end{aligned}$$

   The system can be converted into an augmented form, the new state-space form becomes:

$$\begin{aligned}\dot{x_a} &= A_a x_a + B_a u - D_a d \\ y_a &= C_a x\end{aligned} \tag{4.31}$$

Such that:

$$x_a = \begin{bmatrix} x \\ f_a \end{bmatrix}, \quad f_a = \begin{bmatrix} f_{a1} \\ f_{a2} \end{bmatrix}, \quad F_a = B, \quad A_a = \begin{bmatrix} A & F_a \\ 0_{2\times3} & 0_{2\times2} \end{bmatrix}, \quad B_a = \begin{bmatrix} B \\ 0_{2\times2} \end{bmatrix}, \quad D_a = \begin{bmatrix} -E_d \\ 0_{2\times2} \end{bmatrix}, \quad C_a = \begin{bmatrix} C & 0_{2\times2} \end{bmatrix}$$

For the faults, the magnitude is chosen be 10% of the maximum actuator value $10^{-4}\ m^3/s$, thus, $f_a = \begin{bmatrix} 10\times10^{-5} \\ 10\times10^{-5} \end{bmatrix}$

2. **Case 2: 2 sensor faults:** there exist two sensors to measure the level of liquid in tanks T_1 and T_2, thus, it is possible to represent the faults as f_{s1} and f_{s2}. The following augmented system results:

$$\begin{aligned} \dot{x_a} &= A_a x_a + B_a u + D_a d \\ y_a &= C_a x \end{aligned} \tag{4.32}$$

Such that:

$$x_a = \begin{bmatrix} x \\ f_s \end{bmatrix}, \quad f_s = \begin{bmatrix} f_{s1} \\ f_{s2} \end{bmatrix}, \quad A_a = \begin{bmatrix} A & 0_{3\times2} \\ 0_{2\times3} & 0_{2\times2} \end{bmatrix}, \quad B_a = \begin{bmatrix} B \\ 0_{2\times2} \end{bmatrix}, \quad D_a = \begin{bmatrix} E_d \\ 0_{2\times2} \end{bmatrix}, \quad C_a = \begin{bmatrix} C & F_s \end{bmatrix}, \qquad F_s = I_{2\times2}$$

For the faults we chose the values to be as follows:

$$f_s = \begin{bmatrix} 0,02 \\ 0.01 \end{bmatrix}$$

3. **Case 3: faults in actuator 1 and sensor 1:** the new system becomes as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu - E_d d + b_1 f_{a1} \\ y &= Cx + F_{s1} f_{s1} \end{aligned} \tag{4.33}$$

Writing the system in augmented form yields:

$$\begin{aligned} \dot{x_a} &= A_a x_a + B_a u - D_a d \\ y_a &= C_a x \end{aligned} \tag{4.34}$$

Such that:

$$x_a = \begin{bmatrix} x \\ f \end{bmatrix}, \quad f = \begin{bmatrix} f_{a1} \\ f_{s1} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_2 \end{bmatrix}, \quad F_{s1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A_a = \begin{bmatrix} A & b_1 & 0_{3\times1} \\ 0_{2\times3} & 0_{2\times1} & 0_{2\times1} \end{bmatrix}$$

$$B_a = \begin{bmatrix} B \\ 0_{1\times2} \\ 0_{1\times2} \end{bmatrix}, \quad D_a = \begin{bmatrix} E_d \\ 0_{2\times2} \end{bmatrix}, \quad C_a = \begin{bmatrix} C & 0_{2\times1} & F_{s1} \end{bmatrix}$$

4. **Case 4: faults in actuator 2 and sensor 2:** the new augmented system becomes as follows:

$$\begin{aligned} \dot{x_a} &= A_a x_a + B_a u - D_a d \\ y_a &= C_a x \end{aligned} \tag{4.35}$$

Such that:

$$x_a = \begin{bmatrix} x \\ f \end{bmatrix}, \quad f = \begin{bmatrix} f_{a2} \\ f_{s2} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_2 \end{bmatrix}, \quad F_{s2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad A_a = \begin{bmatrix} A & 0_{3\times1} & b_2 \\ 0_{2\times3} & 0_{2\times1} & 0_{2\times1} \end{bmatrix}$$

$$B_a = \begin{bmatrix} B \\ 0_{1\times2} \\ 0_{1\times2} \end{bmatrix}, \quad D_a = \begin{bmatrix} E_d \\ 0_{2\times2} \end{bmatrix}, \quad C_a = \begin{bmatrix} C & 0_{2\times1} & F_{s2} \end{bmatrix}$$

For the estimation of faults, the Luenberger observer is used, it is described by the following mathematical approach:

$$\begin{cases} \dot{\hat{x}}_a = A_a \hat{x}_a + B_a u - D_a d + L(y_a - \hat{y}_a) \\ \hat{y}_a = C_a \hat{x}_a \end{cases} \tag{4.36}$$

In order to obtain the matrix L, let the error, e_a, be the difference between x and $\hat{x}$. The error dynamics, $\hat{e}(t)$, is given by the following formula:

$$e_a = x_a - \hat{x}_a = (A_a - L_a C_a) e_a \tag{4.37}$$

Using the place function in Matlab to set the poles position as follows:

$$p_i(i = 1,2,3,4,5) = \{-0.03 + 0.03j, -0.03 - 0.03j, -0.4, -0.4, -0.5\}$$

The estimation of faults is illustrated in the following graphs, in order to successfully evaluate the accuracy of the estimation, the standard deviation is used. Such that:

$$\begin{cases} |e| = |x - \hat{x}| \\ \sigma = |e|^2 \end{cases} \tag{4.38}$$

- **Actuator faults:**

  The first actuator fault is injected at time $t = 5000s$ for a period of time of $T_{fa1} = 5000s$, the fault and its estimation are illustrated in Fig 4.10.

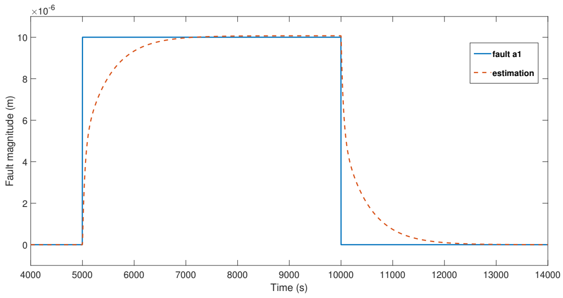


Figure 4.10: Actuator 1 fault and its estimation via Luenberger observer.

Now the error square graph as a function of time is:

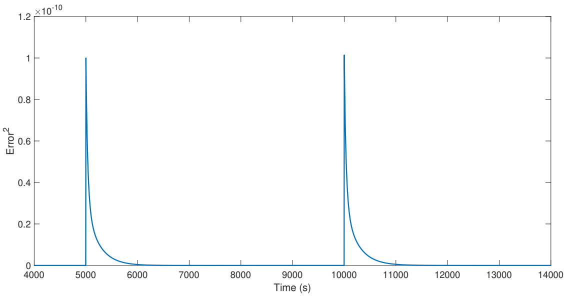


Figure 4.11: Standard deviation of actuator 1 fault.

The second actuator fault is injected at time $t = 2000s$ for a period of time of $T_{fa2} = 10000s$, the fault and its estimation are illustrated in Fig 4.12.

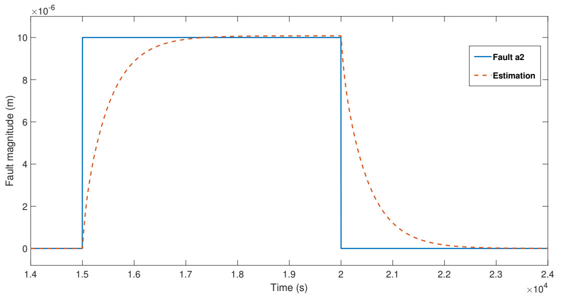


Figure 4.12: Actuator 2 fault and its estimation via Luenberger observer.

Now the standard deviation graph as a function of time is:

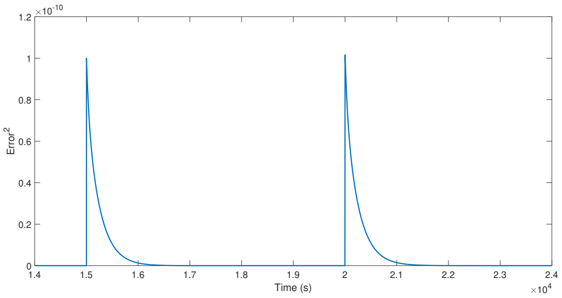


Figure 4.13: Standard deviation of actuator 2 fault.

- **Sensor faults:**

  The first sensor fault is injected at time $t = 2000s$ for a period of time of $T_{fs1} = 10000s$, the fault and its estimation are illustrated in Fig 4.14.

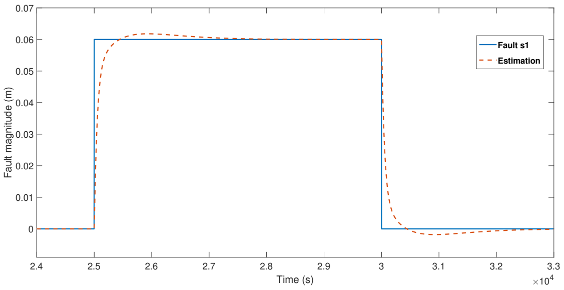


Figure 4.14: Sensor 1 fault and its estimation via Luenberger observer.

Now the standard deviation graph as a function of time is:

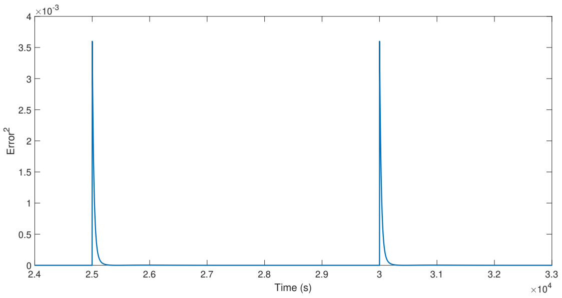


Figure 4.15: Standard deviation of sensor 1 fault.

The second sensor fault is injected at time $t = 3000s$ for a period of time of $T_{fs2} = 10000s$, the fault and its estimation are illustrated in Fig 4.16.

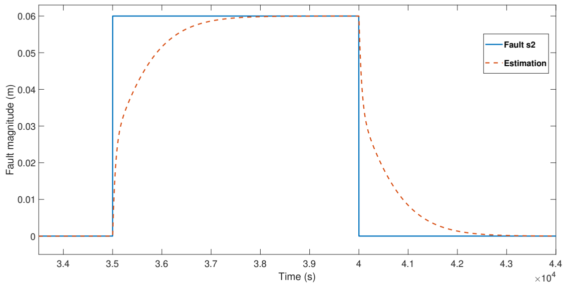


Figure 4.16: Sensor 2 fault and its estimation via Luenberger observer.

Now the standard deviation graph as a function of time is:

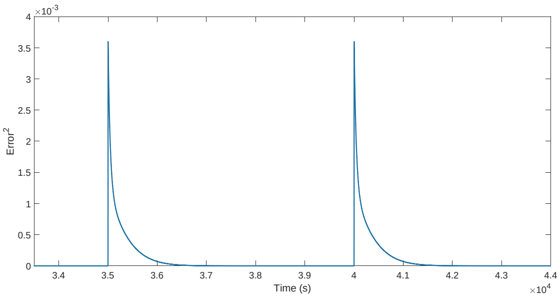


Figure 4.17: Standard deviation of Sensor 2 fault.

Remarque: In order to have a better visualization, the faults are simulated separately. The previously mention scenarios are tested, however, successfully.

The graphs, as well as the standard deviation, clearly show that the Luenberger observer was able to estimate simultaneously actuator and sensor faults, the observer is designed for a linear model. Small size examples have illustrated the efficiency of the proposed approach for constant faults.

Random faults:

For the following part, let's consider a random sensor fault generated by the random number generator in Simulink. The fault is injected in sensor 2 at time $t = 1000s$ for a period of $T_r = 14000s$.

The estimation using the Luenberger observer gave the following graphs:

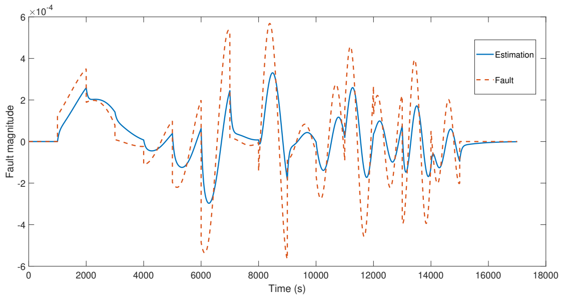


Figure 4.18: Random sensor fault estimation via Luenberger observer.

The standard deviation graphs of the previous estimation is:

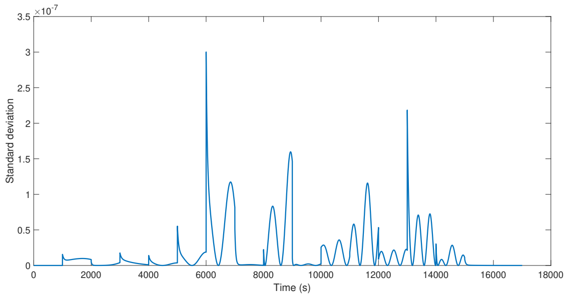


Figure 4.19: Standard deviation for random sensor fault.

Remark: the error graph for all faults is zoomed for a better visualization, the magnitude of the spics is relatively small and can be neglected.

4.8 Regulation and tracking responses

4.8.1 Regulation mode response

Designing a dynamic feedback controller for a given control system such that the output of the resulting closed-loop system tracks (i.e. converges to) a predefined reference signal is an important feedback synthesis problem. This problem is known as the servo problem.

Parameter	Value
h_d (m)	[0.40, 0.20, 0.30]
IC (m)	[-0.1000, -0.0500, -0.0750]

Table 4.3: Regulation mode parameters.

Using the pole placement approach, the poles are set as follows: $[-0.05, -0.02+0.01j, -0.02-0.01j]$, yields the following gain for the controller:

$$K = 10^{-3} \times \begin{bmatrix} 0.3676 & -0.2362 & 0.2366 \\ -0.2362 & 0.2329 & 0.2315 \end{bmatrix}$$

Fig 4.20 illustrates the simulink representation of the regulation mode.

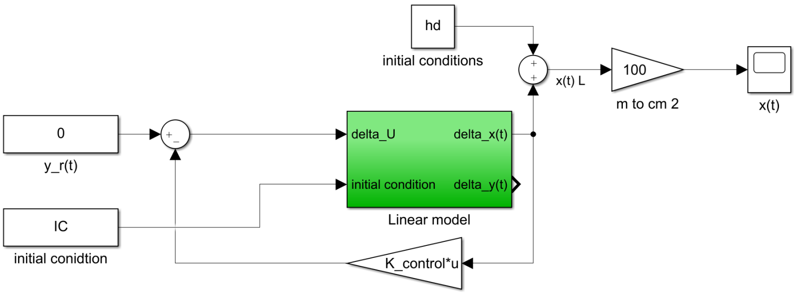


Figure 4.20: Simulink representation of regulation mode.

The regulated system must, as is well known, follow the evolution of the setpoint with a certain level of precision and more or less quickly. In order to address this issue, we will monitor the system's responses to setpoint steps, the obtained results are as follow:

Clearly, the regulation mode yields good results in terms of following the evolution of a setpoint.

4.8.2 Tracking mode response

For the tracking mode, a PI controller is used. The controller is also simulated using a prespecified tracking reference signal. Mathematically, the previous statement can be expressed in in an equation form as follows:

$$\lim_{x \to -\infty} [y(t) - r(t)] = 0 \tag{4.39}$$

By augmenting the state space system as shown in the following equation, integral action will be imparted onto the loop

$$A_{aug} = \begin{bmatrix} A & 0_{3\times2} \\ -C & 0_{2\times2} \end{bmatrix} \quad , \quad B_{aug} = \begin{bmatrix} B \\ 0_{2\times2} \end{bmatrix}$$

The A matrix must remain square, so from inspection it can be seen that the effect of this augmentation is the addition of a number of poles at the origin equal to the number of outputs.

The following tables shows the different parameter required for the racking mode.

Parameter	Value
h_d (m)	[0.5000, 0.2500, 0.3750]
IC (m)	[0.40, 0.20, 0.30]

Table 4.4: Tracking mode parameters.

Using the pole placement approach, the poles are set as follows: $[-0.05, -0.025, -0.04, -0.06, -0.06]$, this yields the following gain for the controller:

$$K_p = \begin{bmatrix} 0.0015 & 0.0001 & -0.0000 \\ 0.0001 & 0.0013 & -0.0000 \end{bmatrix} \quad , \quad K_i = 10^{-4} \times \begin{bmatrix} 0.4722 & 0.0961 \\ 0.0918 & 0.4558 \end{bmatrix}$$

Fig 4.21 Simulink representation of the tracking mode

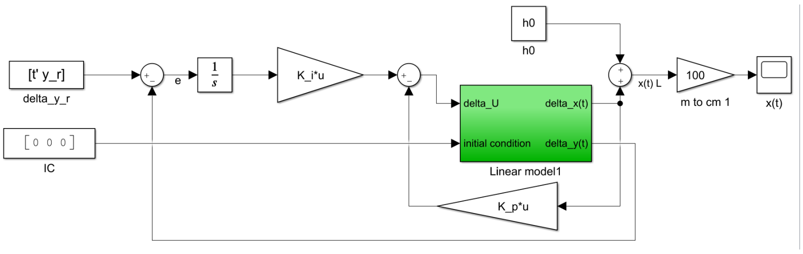


Figure 4.21: Simulink representation of tracking mode.

Figure 4.22 shows the simulated tracking response of the controller.

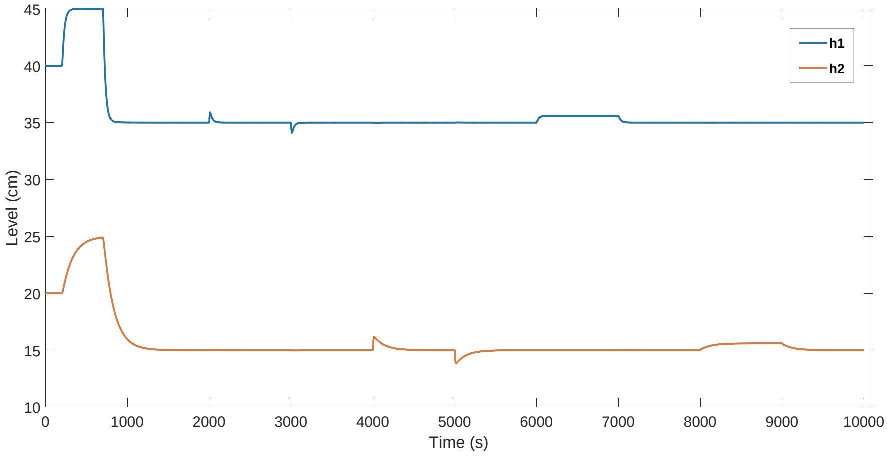


Figure 4.22: System behavior in tracking mode.

Clearly, the tracking mode provides good tracking for the system. mainly reducing the step time for tank T_1 from 2300s to 150s, for tank T_2, the setting time is reduced from 3000s to 600. Consequently, the setting time of tank T_3 is reduced from 3500s to 500s.

The first deviation in the level of tank T_1 represents the first actuator fault, as can be seen, the fault has a minimum effect on tank T_1 and no noticeable effect on tank T_2. Similarly, the first deviation in the level of tank T_2 has no effect on T_1. The same remarks are noticed for when injecting sensor faults.

It is safe to say the FTC was successfully applied to minimize the effect of the injected faults on the system, and the system is ideally decoupled where the faults in tank T_1 have no effect on tank T_2.

4.9 Conclusion

In this chapter, the methodology of the multiloop control of multivariable systems was highlighted, for a hydraulic experimental station with three tanks called DTS-200, the methods developed in the earlier chapters were used.

First, on the basis of a priori model knowledge, the process was described before using non-linear differential equations to model its dynamics.

The characteristics of the station, including stability, and interaction analysis, were then the subject of a study. By using the latter, it was possible to select the best control configuration while still ensuring minimal interaction and asymptotic stability.

Rejecting disturbance and noise is a crucial process in the analysis of a control system, for that disturbance in the form of tank leakage and noise in the form of sensor false readings were injected into the system, and successfully decoupled using well designed observers, these laters have another key rule in the analysis, which estimating both states and faults, for this case abrupt actuator and sensor faults are injected in various scenarios to text the functionality of the observers, moreover, a random sensor fault is injected, and well estimated with a small error percentage.

Finally, the regulation mode and tracking mode responses are obtained, from the simulation, it is clear that the regulated system responds more quickly and without overshoot, also, it is clear that the system becomes less sensitive to the disturbance and faults.

General Conclusion

The work presented in this thesis falls under the category of linear control of multivariable systems. It primarily focuses on the multi-loop control technique. The work's main goal is to propose a methodological approach for synthesis of a multi-loop control system for a multivariable system with at least one suitable control configuration.

First, the project begins with providing generalities on nonlinear multivariable systems and their characteristics.

Next, it covers the coupling phenomena in multivariable systems and how it effects the systems performance, in the same scope, it offers a solution for the problem in the form of interactions measurement techniques and decoupling algorithms

Following that, the key concepts of fault tolerance control and estimation for both states and faults, as well as the feedback control technique for controller design are presented.

Finally, in order to test the previously discussed techniques, chapter 4 covers an application for level control in a hydraulic system with three tanks. The simulation allowed us to illustrate the steps of the distributed control, namely the choice of the better control configuration by exploiting the RGA method, and the synthesis of monovariable correctors using the dominant pole compensation method. The simulation carried out have demonstrated the advantages of the multi-loop control, since tracking and disturbance rejection have been ensured.

In light of the findings, it is possible to conclude that multi-loop control adapts well to the control of a multivariable system when a weakly interactive command configuration is used.

The findings of this thesis are very encouraging and provide interesting avenues for future research. Furthermore, it is important to take into consideration that the proposed technique for synthesis of a multi-loop control system was created for systems with at least one weakly interactive command configuration.

This thesis' methodology showed that the proposed technique for the synthesis of a multi-loop control system has been developed for systems that have at least one command configuration weakly interactive. The methodology developed should be able to be extended to the case of systems that do not present a control configuration. Adequate, as well as for non-square multivariate systems.

References

[1] A. S. Boksenbom and R. F. Hood, "General algebraic method applied to control analysis of complex engine types," 1950.

[2] "Introduction to multivariable control." (accessed on June 15, 2022), [Online]. Available: http://mocha-java.uccs.edu/ECE5580/index.html.

[3] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. 2005.

[4] "Singular value decomposition example." (accessed on June 1,2022), [Online]. Available: https://atozmath.com/.

[5] B. Halvarsson, "Interaction analysis in multivariable control systems: Applications to bioreactors for nitrogen removal," Ph.D. dissertation, Acta Universitatis Upsaliensis, 2010.

[6] A. Khaki-Sedigh and B. Moaveni, *Control Configuration Selection for Multivariable Plants*. 2009.

[7] L. Liu, S. Tian, D. Xue, T. Zhang, Y. Chen, and S. Zhang, "A review of industrial mimo decoupling control," *International Journal of Control, Automation and Systems*, vol. 17, Apr. 2019. DOI: 10.1007/s12555-018-0367-4.

[8] "Faults." (accessed on June 10,2022), [Online]. Available: https://tc.ifac-control.org/6/4/terminology/terminology-in-the-area-of-fault-management.

[9] M. Mahmoud, J. Jiang, and Y. Zhang, *Active Fault Tolerant Control Systems: Stochastic Analysis and Synthesis*. 2003.

[10] H. Noura, D. Theilliol, J. Ponsart, and A. Chamseddine, *Fault-tolerant Control Systems: Design and Practical Applications*. 2009.

[11] "State estimation." (accessed on June 6, 2022), [Online]. Available: https://www.nasa.gov/centers/ames/research/technology-onepagers/state-estimation.html.

[12] S. Nazari, *The unknown input observer and its advantages with examples*, 2015.

[13] P. de Larminat, *Analysis and Control of Linear Systems*. 2007.

[14] P. Woolf, *Chemical Process Dynamics and Controls*, ser. Open textbook library. University of Michigan Engineering Controls Group, 2009. [Online]. Available: https://books.google.dz/books?id=Op87vwEACAAJ.