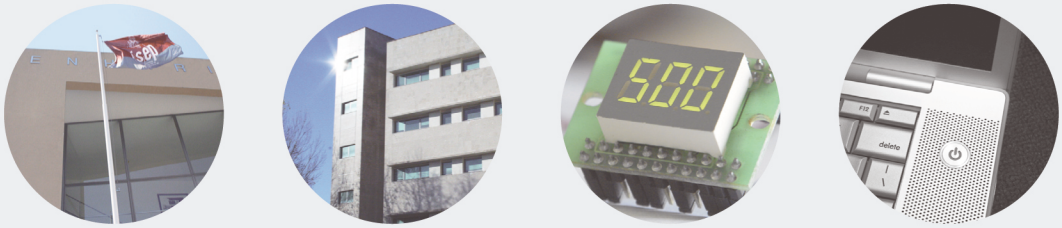
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



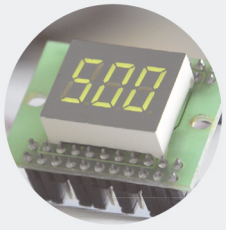
Fractional Control of a Heat Diffusion System

TIAGO FILIPE RIBEIRO SANTOS
Julho de 2025

POLITÉCNICO
DO PORTO

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES





Fractional Control of a Heat Diffusion System

TIAGO FILIPE RIBEIRO SANTOS
Julho de 2025

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Fractional Control of a Heat Diffusion System

Tiago Filipe Ribeiro Santos

Master's Degree in Electrical Engineering

Specialization Area of Autonomous Systems

Department of Electrical Engineering

Supervisor: Prof. Isabel Maria de Sousa de Jesus, PhD

3 de julho de 2025

THIS PAGE
INTENTIONALLY
LEFT BLANK

"Pressure is something you feel when you don't know what the hell you're doing"

Peyton Manning

THIS PAGE
INTENTIONALLY
LEFT BLANK

Abstract

Thermal systems with long delay times and slow dynamic response are common in industrial processes, where precise temperature regulation is necessary to ensure efficiency. Traditional control strategies, specifically **P**roportional **I**ntegral **D**erivative (PID) controllers, struggle with systems that demonstrate significant delays. This dissertation seeks to develop and implement a fractional-order controller for a thermal diffusion system with time delay.

In order to achieve this goal, a comparative study was carried out between the classical and fractional controllers, along with an analysis of the thermal diffusion system in order to find the fractional-order model that best represents its behaviour.

The controllers were tested through simulations, testing both performance and regression metrics, as well as prediction techniques such as the Smith Predictor, anti-windup system and non-linearities. The results demonstrated that the fractional controller outperformed the classical ones, especially in the implemented system with a significant delay, as it showed better robustness and response to the actual system, which was further confirmed after the frequency-domain studies, especially the Bode Diagrams. And, together with the Root Locus, it confirmed the system's stability.

These results showed that the fractional-order controller with Smith Predictor and Anti-windup is a viable solution for thermal systems with high delay values.

Keywords: Fractional Controller, Fractional Calculus, PID Controller, MATLAB, FOMCON Toolbox, Heat Diffusion systems

THIS PAGE
INTENTIONALLY
LEFT BLANK

Acknowledgments

There wouldn't be enough pages to thank everyone who accompanied me on this journey, so I'll just mention them in general terms.

I would firstly like to thank my supervisor, Professor Isabel Jesus, for all her patience, availability and help throughout this dissertation.

I would also like to thank my parents and grandmother, for always being willing to help in any way they could.

I would like to thank all my friends for always being willing to help me in any way they could, if only to remind me that there is more to life than work.

Thank you all!

Tiago Santos

THIS PAGE
INTENTIONALLY
LEFT BLANK

Contents

THIS PAGE
INTENTIONALLY
LEFT BLANK

List of Figures

THIS PAGE
INTENTIONALLY
LEFT BLANK

List of Tables

THIS PAGE
INTENTIONALLY
LEFT BLANK

List of Abbreviations

CRONE	*Commande **R**obuste d'**O**rdre **N**on **E**ntier*
FC	**F**ractional **C**alculus
FOMCON	**F**ractional-**O**rder **M**odeling and **C**ontrol
FOIPDT	**F**ractional-**O**rder **I**ntgrating **P**lus **D**ead **T**ime
FOPDT	**F**irst-**O**rder **P**lus **D**ead **T**ime
FOPID	**F**ractional-**O**rder **P**roportional **I**ntegral **D**erivative
FOTF	**F**ractional-**O**rder **T**ransfer **F**unctions
GUIs	**G**raphical **U**ser **I**nterfaces
IAE	**I**ntegral **A**bsolute **E**rror
IPDT	**I**ntegrating **P**lus **D**ead **T**ime
ISE	**I**ntegral **S**quare **E**rror
ITAE	**I**ntegral **T**ime **A**bsolute **E**rror
ITSE	**I**ntegral **T**ime **S**quare **E**rror
MAE	**M**ean **A**bsolute **E**rror
MAPE	**M**ean **A**bsolute **P**ercentual **E**rror
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput
MSE	**M**ean **S**quared **E**rror
ODE	**O**rdinary **D**ifferential **E**quation
PID	**P**roportional **I**ntegral **D**erivative
RMSE	**R**oot **M**ean **S**quared **E**rror
SISO	**S**ingle **I**nput **S**ingle **O**utput
SQP	**S**equential **Q**uadratic **P**rogramming

Chapter 1

Introduction

1.1 Contextualization and Motivation

Heat diffusion systems are fundamental to a wide range of industrial and engineering processes, including thermal management in electronic devices. The ability to control the temperature in these systems is crucial to guarantee energy efficiency [1]. However, the control of thermal processes is often difficult due to their intrinsic characteristics, such as slow dynamics, distributed parameters and, principally, time delays.

Time delays in thermal systems result from physical transport phenomena, such as, the latency of sensors and actuators or communication delays in distributed control systems. These delays can worsen system performance and, in some cases, destabilise conventional controllers [2, 3]. Consequently, there is a growing demand for control strategies that are capable of maintaining performance and deal with great delays.

Although PID controllers are vastly utilized because of their simplicity and well-known, they are notably limited when applied to system with complex dynamics or delays [4]. To face this, fractional-order control has recently emerged as a powerful alternative. By lengthening the classical controller to allow non-integer integrals and derivatives, fractional controllers offer more flexibility and better adaptability to systems with memory dependencies, outperforming classical controllers, specifically in systems with significant delays [5–7].

Even with such outstanding results, fractional-order controllers still face challenges. Implementation in real-time systems, the complexity of tuning five parameters instead of three limit their practical usage. In addition, there is still a need for more comparative studies that explore the behaviour of fractional and classical controllers in realistic conditions.

This dissertation was motivated by the need to better understand and evaluate the application of fractional order control strategies in heat diffusion systems. Thus, by exploring system behaviour, comparing the control with classical and evaluating performance under

delay conditions, this research aims to contribute valuable knowledge of more effective solutions for heat diffusion systems.

1.2 Objectives

The primary objective of this dissertation is to design a fractional-order controller for a heat diffusion system. To achieve this goal, a preliminary evaluation was conducted to determine the most effective implementation approach. The following research objectives were established:

- Investigation of heat diffusion system dynamics;
- Study of fractional calculus;
- Study of classical and fractional-order controllers;
- Analysis of control algorithms for time-delay compensation in heat diffusion systems;
- Performance comparison between fractional and classical controllers in time-delay systems;

1.3 Dissertation Structure

Chapter 2 presents the current state of the art. This Chapter gives a short introduction to calculus and fractional controllers. It then introduces classical controllers, along with the importance of parameter tuning and the Ziegler-Nichols method. Finally, a brief description of thermal diffusion systems is given, followed by the first law of thermodynamics which introduces the model of a thermal system.

Chapter 3 details the controller development process for the heat diffusion system. The chapter initially describes the toolbox implemented. Subsequently, it presents testing of all control algorithms to identify the optimal solution for the specific system requirements. The chapter concludes with an analytical discussion of the experimental results.

Lastly, Chapter 4 consist of the conclusions and possible ideas for future work.

Chapter 2

State of Art

2.1 Introduction

This chapter presents the main concepts related to the control of heat systems using fractional controllers. Firstly, in section 2.2, the fundamentals of **F**ractional **C**alculus (FC) and their usage in dynamic system models are discussed. The section 2.3 presents the fractional controllers, promoting their advantages when compared to traditional methods. Relatively to section 2.4, a revision is made about classic controllers, followed by section 2.5 explains the importance of parameter tuning. The section 2.6 introduces the Ziegler-Nichols tuning, a special method for parameter tuning that follows certain rules. Finally, the sections 2.7 and section 2.8 not only present the features of heat systems and their value to the industry, but also their relation to the advanced controller approach.

2.2 Fractional Calculus

Fractional calculus is a generalization of integration and differentiation to non-integer orders, that can be expressed by the fundamental operator ${}_aD_t^r$, where a and t define the operation limits [8]. The continuous integro-differential operator can be written as

$$ {}_aD_t^r = \begin{cases} \frac{d^r}{dt^r} & \Re e(r) > 0, \\ 1 & \Re e(r) = 0, \\ \int_a^t (d\tau)^- r & \Re e(r) < 0, \end{cases} \tag{2.1} $$

where $r \in \Re e$, but it can also be a complex number. Different from traditional calculus, responsible for dealing with integer-order differentiation and integration, fractional calculus extends these concepts to arbitrary orders, providing a flexible mathematical framework. It also unifies integer-order differentiation and multiple integrations while capturing both short and long term memory effects. While the short term corresponds to a distribution of time constants, the long term reflects on the absence of a specific timescale, allowing fractional calculus to be highly usable in modeling complex dynamical systems [9–11].

2.2.1 The Origin of Fractional Calculus

Over the past several decades, many researchers have applied fractional calculus in several areas of engineering and science, such as control systems [9, 10]. However, fractional calculus has been acknowledged since the improvement of the regular calculus, having its first mention in a letter from L'Hopital to Leibniz about the notation used in his publication for the n-th derivative of a function;

$$\frac{D^n f(x)}{Dx^n} \tag{2.2}$$

where L'Hopital asks *"What would be the result if $n=\frac{1}{2}$?"*, to which Leibniz answered *"an apparent paradox from which one day useful consequences will be drawn"* [9, 10, 12]. However, the applications and mathematical foundations of fractional calculus are not paradoxical. While the physical interpretation may be challenging to grasp, the definitions are just as rigorous as those for integer order calculus. In 1812, Laplace introduced the concept of a fractional derivative of arbitrary order, which years later appeared in Lacroix's writings in 1819. The Laplace work began as a mathematical exercise, extending the concept of differentiation from integer orders to fractional ones. Starting with $y = x^m$, where m is a positive integer, Lacroix derived the n-th derivative with ease [11–13]:

$$\frac{d^n y}{dx^n} = \frac{m!}{(m-n)!} x^{m-n}, m \geq n \tag{2.3}$$

Applying the Legengre's symbol for the generalized factorial (the complete Gamma function [14]), the function will be:

$$\frac{d^n y}{dx^n} = \frac{\Gamma(m+1)}{\Gamma(m-n+1)} x^{(m-n)} \tag{2.4}$$

Assuming $y = x$ and $n = \frac{1}{2}$, we have:

$$\frac{d^{\frac{1}{2}} y}{dx^{\frac{1}{2}}} = \frac{2\sqrt{x}}{\sqrt{\pi}} \tag{2.5}$$

It is interesting to note that Lacroix's result, which is characteristic of the classical formalists of the time, aligns with the result obtained from the Riemann-Liouville definition of fractional derivatives. Lacroix's expression is also known as Euler's formula, that gives a foundational link between the beginning of fractional calculus and modern theoretical frameworks. By seeking to use this formula to evaluate the fractional derivative of $f(t) = \exp(t)$, the function can be showed as power series [12, 13]

$$f(t) = e^t = \sum_{k=0}^{\infty} \frac{t^k}{k!} \tag{2.6}$$

Applying this term to Euler's expression (as mentioned above), the following result is obtained:

$$\frac{d^v}{dt^v} e^t = \sum_{k=0}^{\infty} \frac{t^{k-v}}{\Gamma(k-v+1)} \tag{2.7}$$

For $v > 0$ and $v \in \Re$, the fractional derivative of the exponential function does not result in another exponential function. Instead, the expression takes the form of a series, which is classified as one of the Higher Transcendental Functions. In this context, the function is represented as the Miller-Ross function, denoted by $E_t(-v, 1)$ [12, 13].

2.2.2 Functions Applied in Fractional Calculus

Fractional calculus extends classical differentiation and integration to non-integer orders, relying heavily on specialized mathematical functions to bridge the gap between theory and application. Among these, three functions play crucial roles [15, 16], present in the next subsections.

2.2.2.1 Gamma

The Gamma function, denoted as $\Gamma(z)$, is the most fundamental of all special functions. Being one of the foundational tools of fractional calculus, Euler's Gamma function serves a critical purpose: the generalization of the factorial $n!$ to non-integer values, thereby extending discrete operations into the continuous domain. This generalization is indispensable for defining fractional derivatives and integrals, bridging the gap between classical calculus and its fractional part [12, 15–18]. It can be defined by the following integral:

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt \tag{2.8}$$

where, by considering z a real number, it will converge in the right half of the complex plane $\Re e(z) > 0$, implying that gamma function is continuously defined as positive real values of z. With that being said, among the fundamental properties of the Gamma function, the following holds particular significance:

$$\Gamma(z+1) = z\Gamma(z) \text{ for } z \in \mathbb{C} \notin (0, -1, -2, -3, ...) \tag{2.9}$$

By using the property mentioned before, we obtain values for $z = 1, 2, 3, ...$:

$$\begin{aligned} \Gamma(2) &= 1.\Gamma(1) = 1! \\ \Gamma(3) &= 2.\Gamma(2) = 2! \\ \Gamma(4) &= 3.\Gamma(3) = 3! \end{aligned} \tag{2.10}$$

Thus, the Gamma function serves as a generalization of the factorial, extending its domain to non-integer and even complex arguments.

2.2.2.2 Beta

The Beta function is defined usually through its Laplace convolution:

$$B(p, q) = \int_0^1 \tau^{p-1} (1-\tau)^{q-1} dt \tag{2.11}$$

or by its Euler representation, known as the *Euler integral of the first kind*:

$$B(p,q) = \int_0^1 u^{p-1}(1-u)^{q-1}du, \;\; \mathbb{R}(p,q) > 0. \tag{2.12}$$

The Beta function is a complex-valued function of two complex variables, whose analytic properties derive from its relationship with the Gamma function, since the Gamma function is referred to as the *Euler integral of the second kind* [12, 14, 15]:

$$B(p,q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} \tag{2.13}$$

2.2.2.3 Mittag-Leffler

The Mittag-Leffler function is a special function that generalizes the exponential function and plays a crucial role in fractional calculus and related work, and can be applied to express several physical processes. The classical Mittag-Leffler function, also known as the one-parametric Mittag-Leffler function, can be defined as [12, 14, 15, 18, 19]:

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \;\; \alpha > 0, \; z \in \mathbb{C} \tag{2.14}$$

This function can be reduced to the exponential function when $\alpha = 1$:

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k+1)} \tag{2.15}$$

$$= \sum_{k=0}^{\infty} \frac{z^k}{k!} \tag{2.16}$$

$$= e^z \tag{2.17}$$

The same equation, for an $\alpha = 2$, it relates to hyperbolic function:

$$E_2(z) = \cosh(\sqrt{z}) \tag{2.18}$$

Aside from the Equation 2.14, there is also the two-parametric Mittag-Leffler function that extends the classic form and is defined as:

$$E_\alpha, \beta(z) = \sum_{k=0}^{\infty} = \frac{z^k}{\Gamma(\alpha k + \beta)}, \;\; \alpha, \beta > 0, \; z \in \mathbb{C} \tag{2.19}$$

where if $\beta = 1$:

$$E_{\alpha,1}(z) = E_\alpha(z) \tag{2.20}$$

In case of specific values, the Equation 2.19 connects to elementary functions such as:

$$E_{1,2}(z) = \frac{e^z - 1}{z}, \quad E_{2,2}(z) = \frac{\sinh(\sqrt{z})}{\sqrt{z}} \tag{2.21}$$

The Mittag-Leffler function serves as a fundamental generalization of the exponential function, bridging the gap between classical exponential behaviour and complex fractional-order dynamics. By carefully choosing the parameters α and β, it can also represent other functions, including trigonometric, hyperbolic, and error functions. This versatility makes it a cornerstone in the study of fractional systems, enabling precise modelling of phenomena with memory effects, anomalous diffusion, and non-local interactions [12, 14, 15, 18, 19].

2.2.3 Definitions of Fractional Derivatives and Integrals

Fractional calculus encompasses a variety of definitions and studies, ranging from n-fold. Some of the definitions are straightly an extension of the conventional integer order calculus, since these operators are characterized by irrational continuous transfer functions in the Laplace domain or infinite-dimensional discrete transfer functions in the Z-domain. Evaluation challenges frequently arise during simulations. The more often than not implemented definitions can be summarized in the following subsections [9, 10, 20]:

2.2.3.1 Fractional Order Cauchy Integral Formula

This equation drawn-out from integer order calculus:

$$D^{\alpha}f(t) = \frac{\Gamma(\alpha+1)}{j2\pi}\int_{C}\frac{f(\tau)}{(\tau-t)^{\alpha+1}}d\tau \tag{2.22}$$

where, C, represents the closed path that enclose the poles of the function$f(t)$.The integrals and derivatives of sinusoidal and cosine functions can be expressed as:

$$\frac{d^k}{dt^k}[\sin at] = a^k \sin\left(at + \frac{k\pi}{2}\right),\ \frac{d^k}{dt^k}[\cos at] = a^k \cos\left(at + \frac{k\pi}{2}\right) \tag{2.23}$$

It can also be demonstrated using Cauchy's formula that, even when k is not an integer, the above formula remains valid [10, 12].

2.2.3.2 Grünwald-Letnikov Definition

This definition unifies the fractional order differentiation and integral in a way that can be expressed as

$${}_aD_t^{\alpha}f(t) = \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{j=0}^{[\frac{(t-a)}{h}]}(-1)^j\begin{pmatrix}\alpha\\ j\end{pmatrix}f(t-jh) \tag{2.24}$$

$$\begin{pmatrix}\alpha\\ j\end{pmatrix} = \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} \tag{2.25}$$

with $\begin{pmatrix}\alpha\\ j\end{pmatrix}$ representing the binomial coefficients. The subscripts to the left and right side of D denote the lower and upper bounds of the integral, respectively. The value of α

can either be positive or negative, corresponding to differentiation and integration, with α being a non-integer. This definition is particularly advantageous for obtaining numerical solutions to fractional differential equations [8–10, 20, 21].

2.2.3.3 Riemann-Liouville Definition

This definition not only bridges the ideas of Liouville and Riemann but also provides a robust mathematical framework for fractional calculus. Its integro-differential structure makes it literally suitable for theoretical analysis, while its connection to the Grünwald-Letnikov definition highlights the interplay between continuous and discrete formulations. Regardless of the challenges associated with the manipulation of Grünwald-Letnikov derivatives, the Riemann-Liouville definition remains a cornerstone in the field, allowing advancements in both pure and applied mathematics. Furthermore, its versatility extends to practical applications, such as modeling complex systems in physics [3, 9, 14, 20, 22]. The derivative definition can be expressed as:

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt}\right)^n \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \; with \; (n-1) \leq \alpha < n \tag{2.26} $$

and the integral [14, 21]:

$$ I_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau \tag{2.27} $$

2.2.3.4 M. Caputo Fractional-Order Derivative Definition

Certain application problems, such as those encountered in viscoelasticity theory and solid mechanics, it is crucial the formulation of initial conditions for accurate modeling. Nevertheless, the Riemann-Liouville definition derivative introduces initial conditions that involve limit values of fractional derivatives. While these problems can be addressed mathematically, the resulting solutions are often impractical because the initial conditions lack clear physical interpretation [8, 9, 14, 22]. The Caputo definition proposed can be expressed as:

$$ {}_aD_t^\alpha = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^n(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \tag{2.28} $$

Compared to the other definitions, the primary advantage of Caputo definition lies in its treatment of initial conditions for differential equations. Unlike the Riemann-Liouville definition, Caputo allows initial conditions to take the same form as in the integer-order case, which ensures they have a clear physical interpretation. However, it is crucial to note that this definition requires the function f to be at least n times differentiable.

In contrast, the Riemann-Liouville definition provides a more generic framework for fractional derivatives, encompassing a broader class of functions. While the Riemann-Liouville derivative serves as a general notation for fractional differentiation, the Caputo

derivative aligns more closely with classical differentiation, making it particularly useful for practical applications where physical interpretation is essential.

2.2.4 Properties of Fractional Derivatives and Integrals

2.2.4.1 Linearity

Equivalently to integer-order differentiation, the fractional differentiation also maintains the linearity operation. Its preservation constitutes a vital bridge between both integer and fractional calculus [23]. At its core, linearity asserts that for any admissible fractional derivative and arbitrary constants, the operation commutes with linear combinations:

$$D^p(\lambda f(t) + \mu g(t)) = \lambda D^p f(t) + \mu D^p g(t) \tag{2.29}$$

This behaviour arises primarily from the structure of fractional operators. In the subsection 2.2.3.2 derivative, linearity manifests through the distributivity of limits and finite differences, and in the subsection 2.2.3.3, linearity inherits from the integral's inherent additivity and homogeneity [24].

The applications of this property are far-reaching. Analytically, its solutions to fractional differential equations must obey superposition, enabling decomposition into simpler subproblems. Computationally, it justifies the term-wise manipulation of fractional operators in numerical schemes, ensuring stability in discretization methods. Most importantly, linearity underpins the compatibility of fractional calculus with Laplace transform methods [16]. Related to the subsection 2.2.3.4 derivative, with the Laplace transform retains its familiar algebraic form:

$$\mathcal{L}\{D^p y(t)\} = s^p Y(s) - \sum_{k=0}^{n-1} s^{p-k-1} y^{(k)}(0^+) \tag{2.30}$$

so that enabling the solution of fractional order systems through frequency domain techniques. The implication of linearity in fractional calculus generalizes classical calculus, preserving its operational framework even with the introduction of non-local dynamics and memory effects [25].

2.2.4.2 Leibniz Rule

The Leibniz rule for fractional derivatives generalizes the product rule of classical ones, but with significant structural differences. The rule is written as [16]: "*If $f(\tau)$ is continuous in $[a,t]$ and $\varphi(t)$ has $n+1$ continuous derivatives in $[a,t]$, then the fractional derivative of the product $\varphi(t)f(t)$ is given by:*"

$${}_aD_t^p(\varphi(t)f(t)) = \sum_{k=0}^{n} \binom{p}{k} \varphi^{(k)}(t)\, {}_aD_t^{p-k} f(t) - R_n^p(t) \tag{2.31}$$

which, in the case of Riemann-Liouville derivatives of order p, where $k-1 \leq p < k$, the rule takes the form [14, 16]:

$$ {}_aD_t^p(\varphi(t)f(t)) = \sum_{k=0}^{\infty} \binom{p}{k} \varphi^{(k)}(t) {}_aD_t^{p-k} f(t) \tag{2.32} $$

where, the binominal coefficients $\binom{p}{k}$ extend to non-integer orders like the Equation 2.25. This infinite series mirrors the non-local nature of fractional operators, which means, each term captures interactions between the history of φ and f. With the introduction of Caputo derivatives, the rule simplifies when $g(t)$ is analytic, but retains its computational complexity due to the slow convergence of the series [13, 16, 24].

2.2.4.3 Semigroup Property

The Semigroup Property affirms that for Riemann-Liouville fractional integrals of orders $\alpha, \beta > 0$, following composition holds:

$$ \mathcal{I}^{\alpha}\mathcal{I}^{\beta} = \mathcal{I}^{\alpha+\beta} f(t) \tag{2.33} $$

which is a generalization of the integer-order case. This property ensures that constant integration of order α followed by β is equal to a single integration of order $\alpha+\beta$. Nonetheless, in the case of fractional derivatives, the property is nuanced. While Riemann-Liouville derivatives satisfy:

$$ {}_a\mathcal{D}_a^{\alpha}\mathcal{D}^{\beta} =_a \mathcal{D}^{\alpha+\beta} \tag{2.34} $$

but only in certain conditions, when β is an integer or when function vanishing conditions at the lower limit are met. Aside from the Riemann-Liouville, it fails due to the non-locality of fractional operators, such as the Caputo derivative that exhibits similar constraints, requiring homogeneous initial conditions for the property to hold [16, 24].

2.2.4.4 Non-locality

The feature that defines fractional derivatives, and their most profound departure from classical calculus, is their non-local nature. Unlike integer-order, which depends only on a function's behaviour in an infinitesimal neighbourhood of a point, fractional derivatives integrate the entire history of the function up to that point. Thus, this arises from the integral kernel in definitions like the Riemann-Liouville operator (Equation 2.26), mathematically:

$$ D^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^t \frac{f^n(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau \tag{2.35} $$

here, the term $(t-\tau)^{-\alpha-1+n}$ ensures that all past values of $f(\tau)$ contribute to the derivative at time t, embedding a memory effect. This property mirrors real-world phenomena where

systems "remember" their past states, as for example the viscoelastic materials whose stress depends on the strain history [23]. These implications are both theoretical and practical. On one hand, non-locality enables fractional calculus to model complex systems with hereditary dynamics, from biological tissues to anomalous diffusion processes [16]. On the other hand, it introduces computational disputes: simulating fractional systems requires storing and processing the entire history of variables, unlike classical **O**rdinary **D**ifferential **E**quation (ODE) where only initial conditions are suffice [26].

Critically, non-locality also disrupts intuitive properties of derivatives. Thus, the semi-group property $D^{\alpha}D^{\beta} = D^{\alpha+\beta}$ fails if the boundary conditions are not carefully prescribed, highlighting the nuanced balance fractional operators strike between generality and tractability [25].

2.2.5 Laplace Transform

The adaptation of the Laplace transform to fractional calculus hinges on the choice of fractional derivatives and integrals. In case of Caputo derivative, the transform follows a natural generalization of the integer-order case. Explicitly, the Laplace transform of this derivative, ${}^{C}D^{\alpha}f(t)$ (where $n-1 < \Re e(\alpha) \leq n$) is expressed as:

$$\mathcal{L}{}^{C}D^{\alpha}f(t) = s^{\alpha}\mathcal{L}f(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1}(D^{k}f(t))(0) \tag{2.36}$$

where $\mathcal{L}f(s) = F(s)$, denotes the Laplace transform of $f(t)$ [26]. This equation incorporates integer-order initial values $(D^{k}f(t))(0)$, which aligns with physical states in problems like viscoelastic deformation [16]. Compared to the Riemann-Liouville derivative, the Laplace transform is given by:

$$\mathcal{L}\{{}_{0}D_{t}^{p}f(t);s\} = s^{p}F(s) - \sum_{k=0}^{n-1} s^{k}[{}_{0}D_{t}^{p-k-1}f(t)]_{t=0} \tag{2.37}$$

where $n-1 \leq p < n$, and it also requires initial conditions involving fractional derivatives $D^{p-k-1}f(0)$. The usage of this definition lacks direct physical interpretation, limiting the definition's utility despite its theoretical elegance [16].

Related to Mittag-Leffler functions, the Laplace transform plays a fundamental role in deriving analytical solutions to linear fractional differential equations. The parameters from the function, $E_{\alpha,\beta}(z)$, which generalizes the exponential function, appears widely in such solutions [12, 15]. The Laplace transform of this pair is expressed as:

$$\mathcal{L}\{t^{\beta-1}E_{\alpha,\beta}(\lambda t^{\alpha})\}(s) = \frac{s^{\alpha-\beta}}{s^{\alpha}-\lambda} \tag{2.38}$$

This equation provides the foundation for solving equations like $D^{\alpha}y(t) + ay(t) = f(t)$. This connection allows the extension to the fractional relaxation and oscillation phenomena,

where Mittag-Leffler functions replace exponentials in describing memory-dependent decay or oscillatory modes. The ability to reduce these problems to algebraic manipulations in the s-domain reflects its classical role while accommodating fractional calculus non-local dynamics [12, 15].

The usage of Laplace transform in fractional calculus extends to cutting-edge domains like fractional-order control and anomalous transport. In control theory, fractional PID controllers can be expressed as $C(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu$ leverage Laplace-domain tuning to enhance robustness [27].

2.3 Fractional Controller

Fractional order controller derive their power from the mathematical properties of fractional calculus, particularly its ability to capture memory effects and hereditary phenomena [16]. The transition from classical PID to fractional $\mathrm{PI}^\lambda\mathrm{D}^\mu$ controllers represents more than just an addition of two more variables, it fundamentally changes how we describe system dynamics [9, 27]. When compared to traditional controller that rely on integer order differential equations, fractional controllers employ operators that integrate the entire history of the system state, providing a more accurate model with long-range dependencies. Moreover, they are less sensitive to parameter changes of a controlled system and controller. The mathematical formulation of a fractional PID controller can be written as:

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s^\lambda} + K_D s^\mu, \ (\lambda, \mu \geq 0) \tag{2.39}$$

where λ and μ represent the fractional order of integration and differentiation respectively [27]. With the additional of these two tuning parameters, not only the system get finer control over its response, but also, the fractional operators naturally accommodate systems with memory effects [16]. In general, the fractional order typically ranges from 0 to 2 (Figure 2.1), where within this range:

- If $\lambda = 1$ and $\mu = 1$, the controller reduces to a PID controller;
- If $\lambda = 0$ and $\mu = 1$, it becomes a PD controller;
- If $\lambda = 1$ and $\mu = 0$, it functions as a PI controller;
- If $\lambda = 0$ and $\mu = 0$, it simplifies to a P controller;
- If these two parameters λ and μ were not integers it would result into having the equivalent fractional order controller $PI^\lambda D^\mu$.

In FOPID, the integral and derivative operations become a part of fractional order. Aside from the classical three parameters tuning on classical PID, the two new parameters

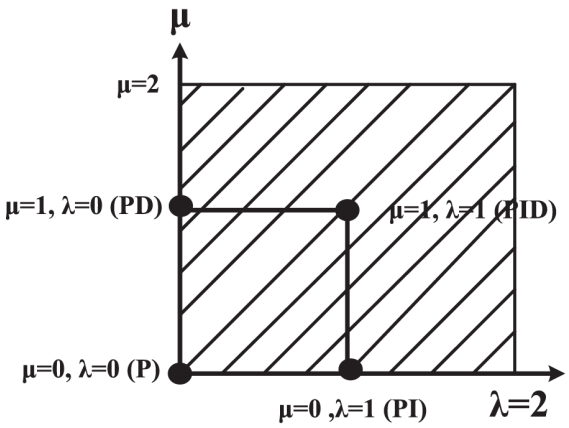


Figure 2.1: Fractional PID converge [9]

must be configured. To determine the optimal values for all parameters, $K_p, K_i, K_d, \lambda, \mu$, while satisfying user-defined performance criteria, a five-dimensional parameter optimization is necessary [9].

2.3.1 Evolution of Fractional-Order Controllers

2.3.1.1 Bode's transfer function

The usage of fractional calculus in control systems began with Bode, with a application on feedback amplifiers [27], where he introduced the transfer function:

$$G(s) = \left(\frac{\omega_{cg}}{s}\right)^{\alpha} \tag{2.40}$$

where ω_{cg} is the gain crossover frequency and the phase margin is given by:

$$\varphi = \pi - \frac{\alpha\pi}{2} \tag{2.41}$$

This function, also known as Bode's ideal loop transfer function, provides a phase margin that remains invariant under gain variations. Thus, it also ensures that the Nyquist plot is a straight line through the origin, leading to a solid robustness against system uncertainties and parameter variations. Bode's transfer function serves as an alternative reference model for control design. The fractional-order nature of this system allows for a smoother phase response and more flexible tuning compared to integer-order controllers [27, 28].

2.3.1.2 CRONE controller

The ***C**ommande **R**obuste d'**O**rdre **N**on **E**ntier* (CRONE) methodology represents a cornerstone of fractional-order control systems. Contrary to classical PID controllers, CRONE

leverages fractional calculus to achieve robust performance in the face of system uncertainties, making it useful for applications that require high precision and adaptability [14, 27, 29]. Some of the main features of this technique are:

- Use of unit feedback;
- Procedure based on frequency domain;
- Continuous or discrete control of **M**ultiple **I**nput **M**ultiple **O**utput (MIMO) systems with disturbances.

There are three generations of CRONE controllers.

The first CRONE generation is specifically designed for system where the plant exhibits gain-like disturbances and maintains a constant phase around a frequency range. Its transfer function is expressed as:

$$C(s) = C_0 s^{\alpha} \tag{2.42}$$

with C_0 representing the gain constant and α denotes the fractional order. This structure provides robustness to gain variations while assuring system stability margins. The exponent α, enables precise shaping of the frequency response, being really valuable in applications where parameter uncertainties are predominantly gain-related [27, 29].

However, as stated by Bode, the design of single-loop absolutely stable amplifiers with varying gains, it is considered a robust controller when it ensures an open-loop transfer function characterized by a constant phase and a constant gain slope (in dB/decade) over a useful frequency range (Figure 2.2). This principle creates the basis for the creation of the second generation CRONE controller [27, 29].

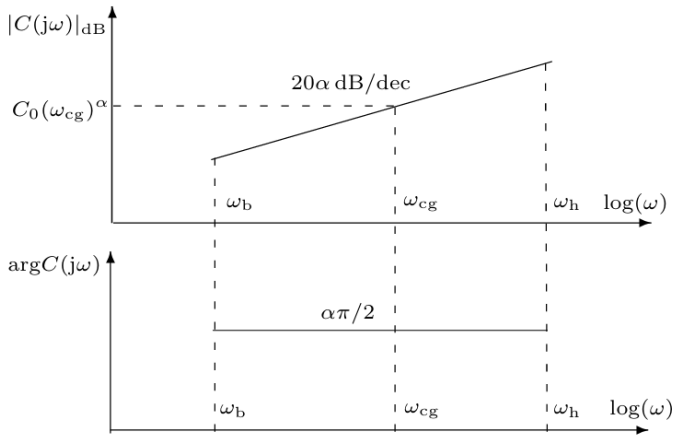


Figure 2.2: Bode plot of CRONE controller first generation [27]

Thus, when addressing plants with no constant phase characteristics, the CRONE controller second generation shows superior performance. This development was made

when, while observing many real world systems, they exhibit phase variations that could not be adequately managed by simple gain compensation. The root innovation of the second generation lies in its open-loop transfer function, which is made to enforce a constant phase within a specified frequency band (ω_b, ω_h)(Figure 2.3):

$$F(s) = C(s)G(s) = \left(\frac{\omega_{eg}}{s}\right)^{\alpha} \tag{2.43}$$

where, ω_{eg} is the gain crossover frequency, and α as the fractional order that determines the phase performance. The term $s^{-\alpha}$ works as a fractional-order integrator, ensuring that the open-loop phase remains at $-\alpha\pi/2$ across the frequency band of interest. This is crucial for keeping robustness in systems where the plant phase varies due to uncertainties or non-linearities [27].

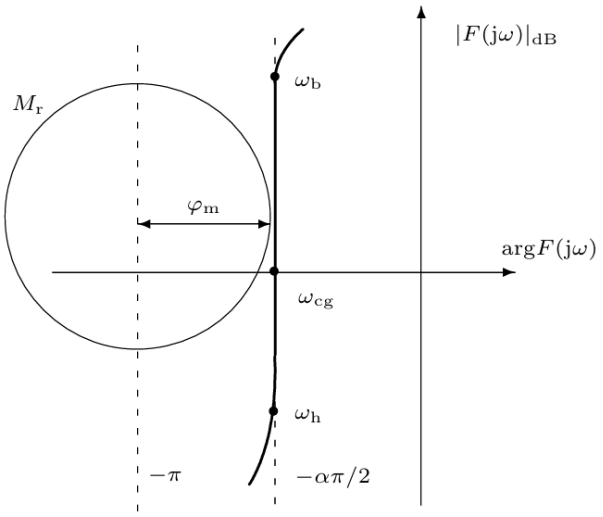


Figure 2.3: Open-loop Nichols chart for the CRONE second generation [27]

Lastly, the CRONE controller of third generation extends the principles of fractional-order control to be used on even more complex systems, such as those with multiple frequency-dependent uncertainties or non-linearities. This generation is characterized by its ability to shape open-loop response across multiple frequency bands, offering incomparable flexibility and robustness [27, 30–32]. The open-loop transfer function is given by:

$$F(s) = k \prod_{i=1}^{n} \left(\frac{1 + s/\omega_{b_i}}{1 + s/\omega_{h_i}}\right)^{\alpha_i} \tag{2.44}$$

with k as a gain constant, w_{b_i} and w_{h_i} define the i-th frequency band, and α_i is the fractional order associated with that band. This multi-band structure allows the controller to independently address ambiguity in different frequency regions, making it convenient for highly complex and heterogeneous systems.

2.3.1.3 Podlubny Fractional Controller Design

The fractional PID controller developed by Podlubny, generalizes the conventional PID structure through the incorporation of non-integer orders of integration and differentiation. As mentioned on subsection 2.3, the Equation 2.39 represents the transfer function of a $PI^{\lambda}D^{\mu}$ controller. This formulation gives additional parameters that allow more precise control of systems with fractional-order dynamics. With that said, when $\lambda = \mu = 1$, the controller reduces to a classical PID form [5, 9, 16]. The key for the development of this controller was the application of Mittag-Leffler functions, which serve as the fractional equivalent of exponential functions in integer-order systems, where the Laplace transform of the Mittag-Leffler is present on Equation 2.38 [15]. The function enable analytical solutions for fractional-order differentiations equations, providing a means to analyse the stability and of performance of FOPID systems. It was established that the condition to guarantee the stability of a fractional-order system requires that the argument of all poles satisfy:

$$|arg(poles)| > \alpha\pi/2 \tag{2.45}$$

extending the Nyquist stability criteria to the fractional domain [5, 9, 16].

2.3.2 Stability of Fractional Order Systems

The stability analysis of fractional-order systems represents a fundamental departure from the classical control. At the core of this analysis lies the sectorial stability condition, which state that a fraction system with characteristic equation $s^{\alpha}+a = 0$ is asymptotically stable, if and only if all poles satisfy the condition mentioned on Equation 2.45 [16, 33]. This condition generalizes the familiar left-half-plane stability requirement of integer-order systems, to a sector in the complex plane whose angle depends on the fractional order α.

For a system with fractional order $\alpha = 0.5$, the stable region comprises all complex numbers whose phase angles satisfy $|\theta| > 45$, Figure 2.4,creating a stability sector that is more restrictive than the classical left-half-plane but less restrictive than what might be initially expected.

The mathematical foundation for this stability criterion can be derived through the properties of the Mittag-Leffler function, which gives as the natural extension of the exponential function to fractional calculus. The behaviour of the Mittag-Leffler function $E_{\alpha}(-t^{\alpha})$ shows that the system response will decay to zero if and only if the sectorial condition is satisfied. This relationship between pole location and time domain behaviour is captured by the equation:

$$y(t) = t^{\alpha-1}E_{\alpha,\alpha}(-at^{\alpha}) \tag{2.46}$$

where the parameters directly reflect the fractional order of the system. This criteria thus ensures that this response function exhibits the necessary decay properties [16, 33].

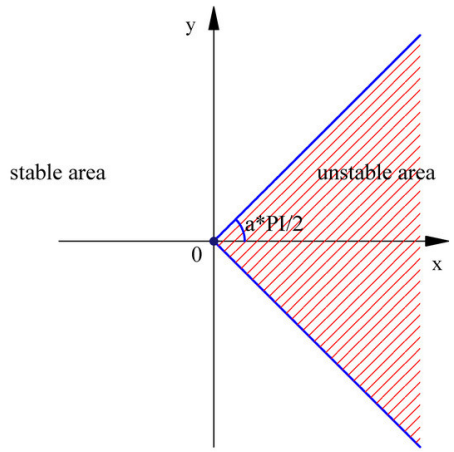


Figure 2.4: Stability region for linear fractional-order systems [34]

Compared to classical integer-order results, several important distinctions appear. While integer-order systems enjoy the simplicity of a binary stability condition, where poles must be in the left half-plane $\Re e(p_i) < 0$, fractional systems, on the other hand, introduce a gradation of stability margins that depend constantly on the fractional order α [5, 27].

This performance has meaningful implications for system robustness. While a fractional system can maintain stability with poles with $\alpha = 0.7$, that would create instability in an integer-order context. This proves the robustness of properly designed fractional controllers to certain perturbations [27, 35]. The phase contribution of fractional operators also creates distinctive time-domain characteristics, where stable systems may show persistent yet decaying oscillations described by:

$$y(t) \sim t^{-\alpha} \sin(\omega t + \varphi) \tag{2.47}$$

These oscillatory modes, sometimes called as asymptotic oscillations, represent a fundamental difference from integer-order system behaviour and can be advantageous in certain control applications [27, 35].

In terms of controller design, the stability criteria implies that a fractional PID controllers has to be tuned first, to achieve desired performance specifications and, secondly, ensure that the close-loop poles satisfy the sectorial condition. The additional degrees of freedom added by the fractional orders, λ and μ, allow for more distinct trade-off between stability margins and performance objectives when compared with integer-order controllers [27, 32].

Lastly, even after all the significant advances in fractional stability, important challenges remain unresolved. The study of non-linear fractional systems lacks general stability criteria when compared to Lyapunov's direct method for integer-order systems. Time-delay fractional systems also present another challenge unresolved, where the stability analysis must account for both the fractional dynamics and the delay effects. The equation that

can describe such systems takes the form:

$$s^{\alpha} + ae^{-\tau s} = 0 \tag{2.48}$$

where, the stability boundaries form complicated surfaces in the parameter space. Recent studies, has developed numerical algorithms to map these stability boundaries, but analytical results remain limited. Related to numerical implementation, these challenges also insist, particularly regarding the reliable computation of stability margins for high-order fractional systems. Not only the truncation of infinite series approximations, but also the finite precision of digital implementations can lead to artificial stabilization or destabilization effects that has to be carefully managed in practical applications [27, 36, 37].

2.4 Classic Controllers

The usage of classic controllers is crucial in engineering and science, being a fundamental piece in several applications. Given its importance in these areas, it is imperative to be well-versed in both theory and practical implementation of the control system. The control element of a controller, is responsible for processing the error signal and generating the corresponding control signal. Its main function is to keep the controlled variable as close as possible to its desired value under the following conditions:

1. Minimizing the steady state error to ensure accuracy;
2. Reducing the settling time for a faster response;
3. Meeting transient response criteria, such as minimizing overshoot to prevent instability.

As the growth of industrial and technological systems tends to get more complexity, the mathematical representation of such systems requires a large set of equations. The classical controllers, which normally address **S**ingle **I**nput **S**ingle **O**utput (SISO), become inappropriate for systems that require MIMO.

Alternatively, modern control theory is based on the time domain only using state space representation of differential equations. It simplifies the design process by simulating real world systems more precisely. However, the system stability remains a crucial factor, since the effectiveness of a controller depends on how well the model represents the actual system. If the differences between the model and the real system are considerable, stability issues may arise when the controller is implemented [2, 38].

2.4.1 The Feedback Principle

The Feedback control constitutes a cornerstone principle in the analysis, operation and design of dynamic systems. The fundamental basis of the feedback principle is both elegant

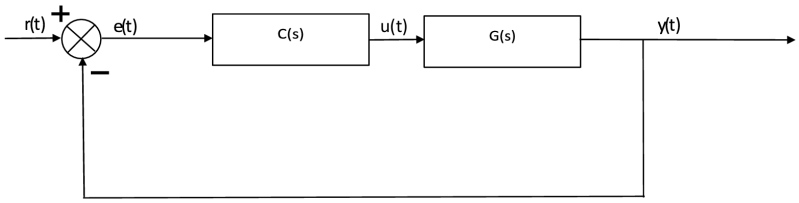


Figure 2.5: Block diagram of system with feedback [41]

and powerful, where through continuous measurement of a system's output and comparison with a desired reference, autonomous corrective actions are generated to regulate system behaviour despite parametric uncertainties and external disturbances [38–40]. The essential components of a feedback control system comprise:

1. A reference input $r(t)$ specifying the desired system behaviour;
2. The plant $G(s)$ representing the system dynamics;
3. A feedback loop creating the error signal: $e(t) = r(t) - y(t)$

where $y(t)$ represents the output value. The error signal is processed by a controller $C(s)$ to produce the control input $u(t)$ that drives the plant towards the desired behaviour (Figure 2.5).

The power of the feedback control is displayed in its ability to modify fundamental system properties. With the appropriate design, the feedback mechanism can stabilize inherently unstable plants, improve transient response characteristics, and reduce sensitivity to parameter variations. The closed-loop transfer function can be written as:

$$T(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{2.49}$$

This formulation reveals how feedback introduces new pole locations that determine system stability and performance characteristics. Practical implementation of feedback control requires careful consideration of several critical aspects. Sensor noise amplification presents a fundamental limitation, particularly in systems employing derivative action. Actuator saturation represents another practical constraint that can lead to integrator wind-up in PID controllers. Time delays in the control loop may destabilize otherwise well-designed systems, necessitating special compensation techniques [38–40].

The fundamental advantage of feedback control lies in its capacity to achieve robust performance without requiring perfect system knowledge. With the continuous adaptation based of the observed behaviour rather than relying solely on predictive models, feedback systems can maintain desired operation across a range of conditions. The adaptive capability allows feedback control to be indispensable in modern engineering applications where uncertainty and variability are inherent characteristics of the operating environment [38–40].

2.4.1.1 On-Off Control

The on-off control, also known as two-position control, represents the simplest form of feedback regulation. The law control operates discontinuously, changing between two discrete output states:

$$u(t) = \begin{cases} u_1, & \text{for } e(t) > 0 \\ u_2, & \text{for } e(t) \leq 0 \end{cases} \tag{2.50}$$

where u_1 and u_2 represents the maximum and minimum value state. The inherent characteristic of this control method is the creation of constant oscillations about the setpoint value, with amplitude and frequency determined by system dynamics [2].

The common refinement involves implementing hysteresis to prevent control chattering:

$$u(t) = \begin{cases} u_1, & \text{for } e(t) > e_{high} \\ u_2, & \text{for } e(t) < e_{low} \end{cases} \tag{2.51}$$

The introduction of this modification allows for a dead-band region (Figure 2.6(A)) that reduces switching frequency while maintaining acceptable regulation performance. The hysteresis width presents a design trade-off between control precision and actuator longevity (Figure 2.6(B)).

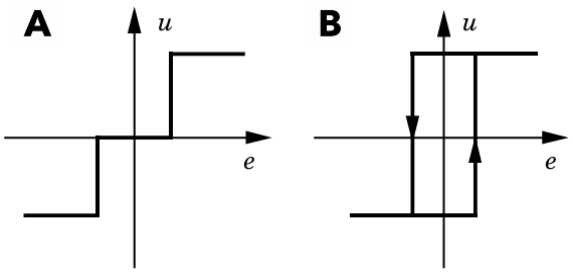


Figure 2.6: (A) On-off control with dead zone and (B) hysteresis implementations [42]

The effectiveness of on-off control depends significantly on the controlled system's time constants. Systems such as thermal process, with slow dynamics, naturally filter the oscillations, making this approach particularly suitable. While offering implementation advantages, the technique shows inherent limitations in precision-critical applications due to its discontinuous nature. The absence of proportional response to error magnitude restricts its use in systems requiring tight tolerance maintenance or rapid dynamic response [2].

2.4.2 PID Control

The simplicity understanding of this controllers ends up being also their flaw, since it limits the system's range that they control with satisfaction. The PID controller operates by combining three control actions, designated as, proportional, integral and derivative. Each of these terms plays a important role in shaping the closed-loop system's response,

affecting the performance metrics such as speed, stability, accuracy and robustness. The effectiveness of PID control hinges not only on the inclusion of these terms, but also on their proper turning and balancing relative to process dynamics [2, 39, 40]. The general time-domain representation of PID controller can be written as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{2.52}$$

where u is the control signal and e is the error in steady state, obtainable by $e(t) = r(t) - y(t)$. The three terms K_p, K_i and K_d represents the gain of each control actions [2, 39].

2.4.2.1 Proportional term

The Proportional component creates a control action directly proportional to the obtained error:

$$u_P(t) = K_p e(t) \tag{2.53}$$

This term enhances the system's immediate responsiveness. A higher proportional gain amplifies the control action for a given error, allowing a faster response. Nevertheless, if the gain is too high, it can induce oscillations or instability into the system. Notably, the proportional term alone cannot eliminate the steady-state error in certain systems [2, 39, 40].

2.4.2.2 Integral

The Integral component represents the accumulation of past errors over time:

$$u_I(t) = K_i \int_0^t e(\tau)d\tau \tag{2.54}$$

In certain cases, the usage of proportional control alone leads to results where the system reaches an equilibrium allowing the control signal to stabilize, but permitting a constant steady-state error to persist. To eliminate this residual error, the controller can be made to generate an increasing control signal as long as the error remains non-zero. This principle is what defines the integral gain action [39, 40, 42].

Unlike the proportional control, the Integral term does not contribute to the system's responsiveness but continuously corrects accumulated errors. While this ensures zero steady-state error, it also introduces a tendency toward oscillations if the gain value is not properly tuned. In processes with significant time delays or rapid dynamics, it is a must account for the delayed correction gain to avoid destabilizing the system [39, 40, 42].

2.4.2.3 Derivative

The Derivative component predicts the future trend of the error by computing its rate of change:

$$u_D(t) = KT_d \frac{de(t)}{dt} \tag{2.55}$$

As mentioned before, the Integral control produces a corrective action that persist even after the error vanishes, leading to oscillation in some cases. To mitigate this, the controller should anticipate when the error is approaching zero. One effective is designing the controller to respond to the rate of change of the error, being this the basis of the Derivative term [39, 42, 43].

With the usage of the three terms, the PID controller can be expressed in the Laplace domain as:

$$U(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) E(s) \tag{2.56}$$

or as a transfer function:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{2.57}$$

The Equation 2.56 is vital for analyzing the controller's behaviour using classical techniques such as Bode plots, root locus, and Nyquist criteria. The interplay of K_p, K_i, and K_d must be carefully balanced to meet performance goals, such as rise time, settling time, overshoot and disturbance rejection (Table 2.1) [39, 42–44].

Table 2.1: Effects of PID Gains on System Response

Parameter	**Rise Time**	**Overshoot**	**Settling Time**	**Steady-State Error**	**Stability**
Increase K_p	Decreases	Increases	Small Change	Decreases	Reduces margin
Increase K_i	Decreases	Increases	Increases	Eliminates	Reduces
Increase K_d	Small Change	Decreases	Decreases	No effect	Improves

2.4.3 Open-Loop vs Closed-Loop

Control systems distinguishes between two principal architectures that embody different philosophies of system design: open-loop and closed-loop control systems. These approaches differ mainly in their structure, operation principles and performance characteristics, with each suited to specific implementations requirements [2].

2.4.3.1 Open-Loop control systems

The architecture of this control operates without any form of output feedback, generating a control signal solely on predetermined input commands. This relation can be

mathematically expressed as:

$$u(t) = K.r(t) \tag{2.58}$$

where $u(t)$ represents the control signal, $r(t)$ the reference input, and K embodies the system's gain characteristics. This configuration becomes effective in systems that are well-characterized and environmental disturbances remain negligible [2].

The primary advantage of open-loop control lie in its structural simplicity and cost-effectiveness. Systems employing this approach require fewer components, as they eliminate the need for sensors and feedback processing elements. However, the absence of feedback mechanisms imposes significant limitations. Open-loop systems cannot automatically compensate for disturbances or variations in system parameters, making them vulnerable to performance degradation in dynamic environments. Any deviation from expected operating conditions, whether due to external disturbances or internal parameter variations, will result in uncorrected errors that persist throughout system operation [2, 39].

In control engineering, block diagrams are commonly used to represent the flow of signals through different system components. In the case of an open-loop control system, its structure can be visualized using the block diagram of Figure 2.7.

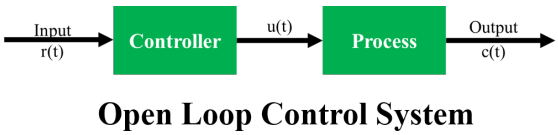


Figure 2.7: Open-Loop block diagram [45]

2.4.3.2 Closed-Loop control systems

Closed-loop control systems incorporate continuous feedback measurements to achieve robust regulation. The fundamental law in this type of systems compares the actual output with the desired reference:

$$e(t) = r(t) - y(t) \tag{2.59}$$

where $e(t)$ represents the error signal and $y(t)$ denotes the measured output like referred previously [2, 39]. The controller then generates corrective actions based on this error signal, leading to the characteristic transfer function of closed-loop system dynamics:

$$C_{(s)} = \frac{G_{(s)}}{1 + G_{(s)}.H_{(s)}} \tag{2.60}$$

This feedback structure, Figure 2.8, provides several critical advantages that make it indispensable for demanding control applications. The constant comparison between desired and actual output, enables automatic disturbance rejection and reduces sensitivity to parameter variations [39].

When compared to open-loop control systems, the implementation of closed-loop control adds additional complexity and challenges. The feedback mechanism can lead to potential destabilization of the system if not properly designed, needing rigorous stability analysis through techniques such as root locus method and state-space approaches [2, 40].

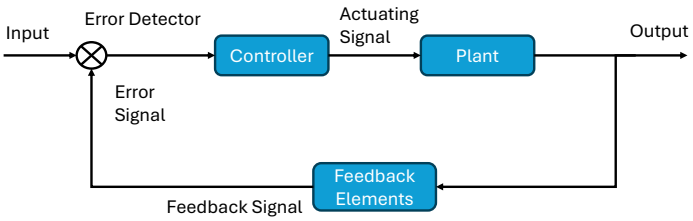


Figure 2.8: Closed-Loop block diagram [46]

The selection between open-loop and closed-loop architectures involves careful consideration of multiple performance and implementation factors. While open-loop systems present simplicity and cost advantages but lack adaptability, in the other hand, the closed-loop systems provide precision and robustness at the expense of greater complexity [38].

The distinction between open-loop and closed-loop control represents a fundamental paradigm in control systems. Open-loop systems serve adequately in predictable, stable environments without any distortions, and closed-loop architectures have become indispensable for applications requiring precision and adaptability [2, 39].

2.5 Importance of Parameter Tuning

The efficiency of control systems rely solely on proper controller tuning, *i.e.*, the process of adjusting the control parameters to achieve desired dynamic behaviour. In case of PID controllers, the main challenge focus on tuning the three fundamental parameters K_p, K_i and K_d. The balance of this parameters is crucial to guarantee the satisfaction of often-competing objectives such as fast response, minimal overshoot and solid disturbance rejection [2, 39, 42].

Throughout the years, tuning methods evolved from methods such as Ziegler-Nichols, to actual model-based optimization techniques. Most of the contemporary solutions incorporate adaptive algorithms and auto tuning capabilities to handle dynamic operating conditions [42].

The consequences of improper tuning leads to several operational deficiencies. The incorrect tuning parameter may either cause harmful oscillations and premature actuator wear, or result in sluggish response and poor disturbance rejection. Between theses extremes, lies the optimal parameter tuning that satisfies the performance criteria without

compromising system robustness [40].

2.6 Ziegler-Nichols tuning method

As one of the most famous techniques in control engineering, the Ziegler-Nichols tuning rules maintained their relevance since their introduction in 1942 [47]. These established two principal approaches for controller parameter tuning, the close-loop tuning approach and the open-loop tuning approach, each with distinct application methods.

2.6.1 Open-Loop Tuning Approach

The open-loop approach analyses the system response under open-loop conditions, typically modelling the process as a **F**irst-**O**rder **P**lus **D**ead **T**ime (FOPDT) system (Figure 2.10). This method tests the system's reaction and involves:

- Conducting open-loop step response tests (Figure 2.9);
- Applying graphical analysis techniques (tangent method) to determine steady-state gain (K_m) (Figure 2.10);
- Using the two-point method to calculate both the process time constant (T_m) and dead time (τ_m);

This variables are calculated by:

$$K = \frac{\Delta y_{ss}}{\Delta u} \tag{2.61}$$

$$\tau_m = \frac{3}{2}(t_2 - t_1) \tag{2.62}$$

$$T_m = (t_2 - \tau) \tag{2.63}$$

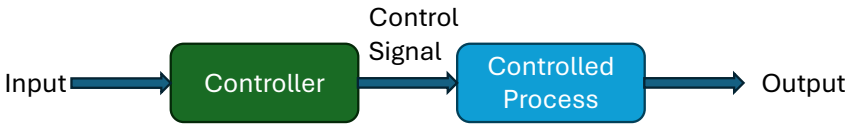


Figure 2.9: Open Loop Control system [48]

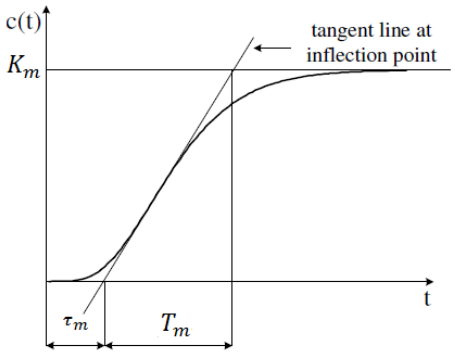


Figure 2.10: Open Loop step response[49]

This method provides a different pathway to establishing the necessary controller parameters without requiring the system to reach critical stability conditions. In one hand, the open-loop method is faster and simple, but, in the other hand, the integral and derivative parameter are merely estimations from the proportional parameter, leading to its values not be suitable for all systems. Each method presents unique advantages and considerations, allowing the selection of the most appropriate approach based on system characteristics and operational constraints [42, 50].

Table 2.2: Open Loop PID parameters

Type of Controller	K_p	T_i	T_d
P	$\frac{T_m}{\tau_m}$	∞	0
PI	$0.9\frac{T_m}{\tau_m}$	$\frac{\tau_m}{0.3}$	0
PID	$1.2\frac{T_m}{\tau_m}$	$2\tau_m$	$0.5\tau_m$

2.6.2 Close-Loop Tuning Approach

This methodology requires a systematic increase of the controller proportional gain until the system starts showing sustained oscillations with consistent amplitude, a state known as critical stability where the system achieves steady oscillations, neither settling nor growing unstable, Figure 2.11 [2]. During this process, there are two crucial variables:

1. the ultimate gain K_u - proportional gain raised until the system reaches a state of sustained oscillations with a consistent amplitude;

2. the ultimate period P_u - the duration required for one complete oscillation cycle;

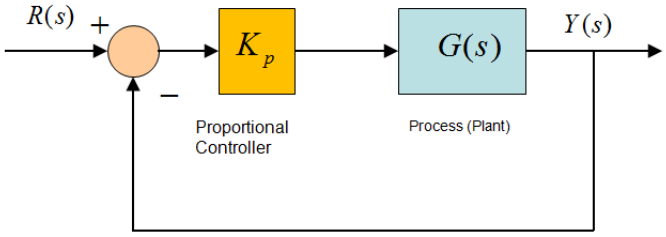


Figure 2.11: Closed loop system with a proportional controller [51]

These measured variables then are used as the basis for calculating the entire PID controller parameters through standardized table (Table 2.3) corresponding to the desired controller type.

Table 2.3: Closed-Loop PID parameters

Type of Controller	K_p	T_i	T_d
P	$0.5K_u$	∞	0
PI	$0.45K_u$	$\frac{1}{1.2}P_u$	0
PID	$0.6K_u$	$0.5P_u$	$0.125P_u$

Although this method takes advantage from requiring only proportional gain adjustments during testing, allowing it to be straightforward to implement, it carries potential risks and takes a long time until the correct ultimate gain value is obtained. The obligation to drive the system to its stability limit may result in undesirable output levels [42, 50].

2.7 Heat Diffusion Systems

The science of thermodynamics studies the quantity of heat transfer as a system goes from one steady state to another, without the consideration of the time required for this process to take place. Nevertheless, in engineering, it is crucial to evaluate the rate of heat transfer [52]. The transfer of energy always occurs from the higher temperature medium to the lower temperature medium, ceasing when both reach thermal equilibrium, *i.e.* the same temperature [52, 53].

Heat transfer can be described as *"the form of energy that can be transferred from one system to another as a result of a temperature difference"*. The science of heat transfer deals with the determination of the rates related to energy transfer [52, 54]. While thermodynamics works with equilibrium states and changes between equilibrium states, heat transfer deals with systems that have low thermal equilibrium, so it is considered a

nonequilibrium phenomenon. Hence, the principal requirement for heat transfer is the existence of a temperature difference. Herewith, the temperature difference is the main force for heat transfer, with the rate of heat transfer fluctuating according to the magnitude of the temperature gradient [52].

Nowadays, it is regularly found heat transfer in engineering systems and other aspects of life, as simple as the human body (Figure 2.12). The human body is in a constant heat rejection to maintain a certain body temperature, tied to the heat rate rejection [52, 53]. Plus, the usage of clothing and heat transfer devices to regulate the heat transfer rate. Devices such as boilers, refrigerators, heaters and furnaces are made mainly on the basis of heat transfer analysis. However, heat transfer faces problems in practice terms that can be distinguished in two groups. The first group, classification problems, deals with determining the heat transfer rate at a specified temperature difference for a given system. The second group, sizing problems, deals with determining the size of a system at a specified rate for a certain temperature difference in order to transfer heat [52, 53].

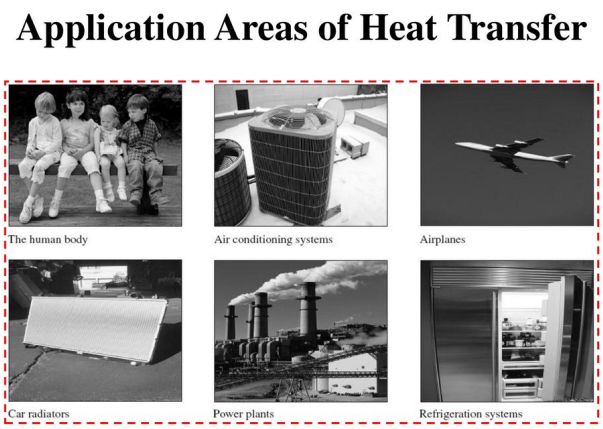


Figure 2.12: Application areas of heat transfer [55]

2.7.1 Heat Transfer Modes

As mentioned before, heat transfer is the science that deals with determining the rates of energy transfer, which stops when both mediums reach the same temperature. The transmission of heat can be done in three methods: conduction, convection and radiation (Figure 2.13). All modes requires the existence of a temperature difference [52–54, 56].

2.7.1.1 Radiation

Thermal radiation is the energy that is emitted by bodies because of their temperature, transported by electromagnetic waves. Unlike conduction and convection, this methods

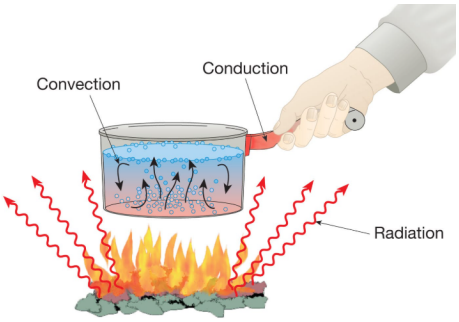


Figure 2.13: Heat Transfer Methods [57]

does not require a material medium, occurring the most efficiently in a vacuum [43, 52, 54, 56]. As shown in the Figure 2.13, radiation emitted by a surface originates from the thermal energy of the matter enclosed within it. The maximum rate of radiation a surface can emit at an absolute temperature T_s (in (Kelvin) K or (Rankine) °R) is determined by the Stefan Boltzmann law

$$\dot{Q}_{emit,max} = \sigma A_s T_s^4 \qquad \text{(W)} \tag{2.64}$$

where $\sigma = 5.67 \times 10^{-8} W/m^2 \cdot K^4$ is the Stefan Boltzmann constant, T_s is the absolute temperature, $\dot{Q}_{emit}$ the rate at which energy is emitted and A_s the area surface. When a surface emits radiation at its maximum rate is called blackbody, *i.e.*, the idealistic surface. When the radiation emitted by this surfaced is named as blackbody radiation. Thus, the radiation emitted by all the non ideal surfaces will be less than the radiation emitted by the idealistic one. It is expressed as:

$$\dot{Q}_{emit,max} = \varepsilon \sigma A_s T_s^4 \qquad \text{(W)} \tag{2.65}$$

with ε being the emissivity of the surface. The emissivity, whose value is in the range of $0 \leq \varepsilon \leq 1$, provides a measure that indicates how efficiently a surface emits energy when compared to a blackbody [52–54, 56]. When two surfaces exchange radiation, the net heat transfer between them is:

$$\dot{Q}_{emit,max} = \varepsilon \sigma A_s (T_s^4 - T_{surr}^4) \qquad \text{(W)} \tag{2.66}$$

where T_{surr} represents the surrounding surfaces. Besides the emissivity property, there is another important radiation property of a surface, which is its absorptivity α, that represents the radiation energy absorbed by the surface, oscillating between the range $0 \leq \alpha \leq 1$. That being said, a blackbody is both a perfect absorber ($\alpha = 1$) and a perfect emitter ($\varepsilon = 1$) [52–54]:

$$\dot{Q}_{abs} = \alpha \dot{Q}_{incident} \qquad \text{(W)} \tag{2.67}$$

2.7.1.2 Convection

The convection mode, is the mode of energy transfer between a solid surface and the adjacent liquid or gas in motion, combining the effects of motion [43, 52, 54]. The faster the motion, the better the heat transfer. This heat transfer method can be classified as forced or natural convection (Figure 2.14). Forced convection happens if the fluid is enforced to flow to a certain direction, accordingly to an external mean, such as a fan or the wind. In terms of natural convection, it happens when the fluid motion is caused by buoyancy forces, caused by the density differences due to the variation of temperature in the fluid [43, 52, 54].

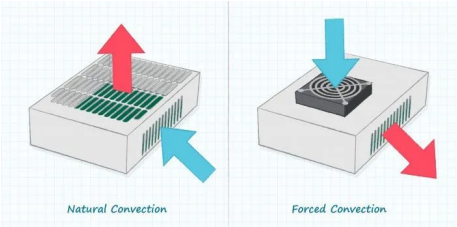


Figure 2.14: Normal and Forced Convection [58]

Regardless of the complexity of convection, the rate of heat transfer during the convection, is proportional to the temperature difference, and it is expressed by Newton's law of cooling:

$$\dot{Q}_{conv} = hA_s(T_s - T_\infty) \tag{2.68}$$

where h is the coefficient of convection heat transfer in $W/m^2 \cdot C$ and T_∞ is the temperature of the oncoming fluid. The coefficient h is not a property of the fluid. In fact, it is an experimentally determined parameter, where its value depends on all variables that affect the convection, such as surface geometry, the natural fluid motion, and properties [52, 53].

2.7.1.3 Conduction

Conduction mode transfers the energy from the more energetic particles of a substance to the less energetic ones, result of the interactions between particles. This mode can occur all states of matter. In liquid and gas state, conduction results of the collisions and diffusion of the molecules during their irregular motion. In case of solids, it is due to the merge of the energy transport by free electrons and molecules vibrations [52, 56].

The amount of energy transfer during a heat conduction rest on the geometry of the medium, as well as its thickness, material and the temperature across the mean. According to the Fourier's law of heat conduction is given by:

$$\dot{Q}_x = -kA\frac{dT}{dx} \tag{2.69}$$

where $\frac{dT}{dx}$ is the temperature gradient in the x direction and k the thermal conductivity. The minus sign is a consequence of energy transfer being in the direction of decreasing temperature [52, 56], when applying a steady conduction through a large plane, with a thickness of $\Delta x = L$. The difference of temperature is given by $\Delta T = T_2 - T_1$. Studies have demonstrated that the amount of heat transfer $\dot{Q}$ trough a wall is doubled and, when the temperature difference is halved the thickness is doubled, Figure 2.15 [52–54]:

$$\dot{Q}_x = kA\frac{T_1 - T_2}{\Delta x} = -kA\frac{\Delta T}{\Delta x} \tag{2.70}$$

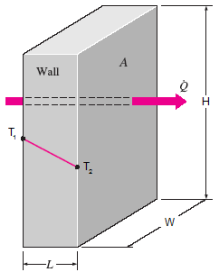


Figure 2.15: Heat conduction trough a wall

2.7.2 General heat equation

The main goal in conduction analysis is to determine the temperature distribution within a material based on the boundary conditions applied to it. In other words, we aim to understand how temperature varies with position inside the material. Once the temperature distribution is known, the conduction heat flux at any point within the material or on its surface can be calculated using Fourier's law. Additionally, other important factors can be derived. Such as, knowing the temperature distribution allows for the assessment of structural integrity by evaluating thermal stresses, expansions, and deflections. It can also help optimize the thickness of insulating materials or evaluate the compatibility of specific coatings or adhesives used with the material [52, 54]. The general heat diffusion equation, in cartesian coordinates, is given by:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + q = \rho c_p \frac{\partial T}{\partial t} \tag{2.71}$$

This equation, also mentioned as the heat equation, provides the key tool to analyse heat conduction. If the thermal conductivity is stable, then:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{2.72}$$

where $\alpha = \frac{k}{\rho c_p}$ represents the thermal diffusivity of the material. Also known as Fourier-Biot equation [59], the Equation 2.72 under certain conditions, can be reduced to follow forms[52, 54]:

Steady state
(Poisson equation)
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} = 0 \tag{2.73}$$

Transient, no heat generation
(diffusion equation)
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{2.74}$$

Steady state, no heat generation
(Laplace equation)
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \tag{2.75}$$

2.8 The First Law of Thermodynamics

The first law, also known as the conservation of energy principle, states that the entire energy of a system is conserved, therefore only changing its form. The principle of conservation can be declared as: *"the net change in the total energy of the system during a process is equal to the difference between the total energy entering and the total energy leaving the system during that process"* [52, 60–62]. With that being said, energy can be transferred from or to a system through heat and mass flow. The total energy of a simple system is a combination of internal, kinetic, and potential energies. For any system undergoing a process, its energy balance can be formulated accordingly:

$$E_{in} - E_{out} = \Delta E_{system} \qquad \text{(J)} \tag{2.76}$$

Energy, as a property, does not change unless the state of the system changes. Consequently, if the system state stays the same throughout the process, $\Delta E_{system} = 0$, simplifying the Formula 2.76 to [52, 60]:

$$\text{Steady, rate form:} \qquad \dot{E}_{in} = \dot{E}_{out} \tag{2.77}$$

When there are no major external influence such as gravity or magnetic, *i.e.* assuming that the system is stationary, the total energy change during a process is solely duo to a change in its internal energy [52, 62, 63].

In terms of heat transfer analysis, the forms of energy that can be transferred by cause of a temperature difference, is typically the focus. In these cases, is useful to write a heat balance and consider the conversion of nuclear, chemical and electric energies into thermal energies as a heat generation. Thus, the energy balance can be expressed as:

$$Q_{in} - Q_{out} + E_{gen} = \Delta E_{thermal,system} \qquad \text{(J)} \tag{2.78}$$

where $Q_{in} - Q_{out}$ represents the net heat transfer and E_{gen} heat generation [52, 62].

2.8.1 Model Heat System

The heat systems is represent by a first order systems which is characterized by an input/output relationship described by a first-order differential equation (Equation 2.79). Such equations include a first-order derivative, reflecting the system's dynamic behaviour [64].

$$\tau \frac{dy}{dt} + y = Kx_{(t)} \tag{2.79}$$

By applying the Laplace transform and considering the initial conditions, Equation 2.79 can be further expanded and simplified as follows:

$$\frac{Y_{(s)}}{X_{(s)}} = \frac{K}{\tau_s + 1} \tag{2.80}$$

where K is the steady state gain. This leads to the derivation of the transfer function for a first-order system.

2.8.1.1 First Order System - Step response

Several key features define the step response of a system. Initially, the output at time zero is assumed to be zero, $y(0) = 0$, under the assumption that the system begins at rest. The steady-state or final value of the response is $y(\infty) = K$, where K represents the output value the system stabilizes to once transient effects have faded [65].

Another critical parameter is the time constant, τ, which indicates the time needed for the system to reach 63% of its final value. The settling time, T_s, refers to the duration required for the output to remain within 98% of its steady state value, typically approximated as four times the time constant, 4τ. Together, these parameters describe the system's response speed and stability when subjected to a step input [65]. In summary, the step response of a first-order system exhibits an exponential rise (Figure 2.16).

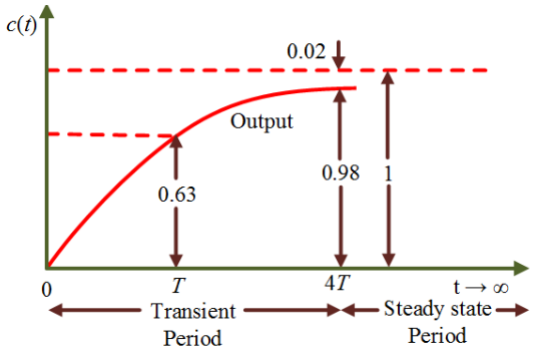


Figure 2.16: First Order step response [66]

2.8.1.2 First Order System with delay

In real-world scenarios, numerous dynamic systems exhibit a delay before responding to an input change. This delay, referred to as dead time, t_d, represents the time interval

between the input action and the first noticeable effect. Dead time is common in systems such as thermal systems [53], where heat diffusion requires time to reach sensors, or chemical processes [67], where reaction times introduce inherent delays.

The transfer function for a first-order system with dead time is similar to Equation 2.80, but it incorporates a time shift represented by an exponential term in the Laplace domain. This exponential term introduces a delay of t_d seconds, causing the system's output to remain unchanged until the delay period has elapsed. Consequently, the transfer function is adjusted as follows:

$$G(s) = \frac{K}{\tau s + 1} e^{-t_d s} \tag{2.81}$$

Moreover, when a unit step is applied to a delayed first-order system, the output in the time domain is expressed as follows:

$$y(t) = \begin{cases} 0, & t < t_d \\ K\left(1 - e^{-\frac{t-t_d}{\tau}}\right), & t \geq t_d \end{cases} \tag{2.82}$$

Before t_d, the output remains zero because the system has not yet started responding. Once $t \geq t_d$, the system exhibits the standard first-order exponential response, but with a time shift of t_d. This delay does not alter the final steady-state value, which remains K, but it extends the time required for the system to reach that value due to the initial response lag (Figure 2.17) [64].

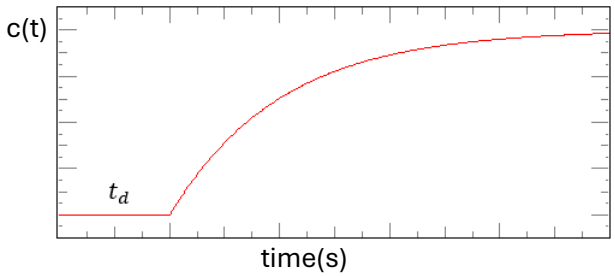


Figure 2.17: First Order step response with time delay [68]

Chapter 3

Controller Development for Heat Diffusion System

3.1 Introduction

This chapter presents the path taken to develop a fractional controller for a heat diffusion system. Initially, in Subection 3.2, a brief explanation of the MATLAB toolbox used is discussed. The Subection 3.3, presents the procedure of obtaining the controller values throughout the toolbox and various performance metrics and optimization algorithms studied. Followed by the Subsections 3.4-3.7, that present all the tests and implementations done to achieve a suitable controller for the studied system. Finally, the Subection 3.8 ends the Chapter with a summary of the tests that showed the best results.

3.2 FOMCON Toolbox

The FOMCON Toolbox for MATLAB is a fractional-order calculus toolbox for system modelling that extends the functionality of the original toolbox **F**ractional-**O**rder **T**ransfer **F**unctions (FOTF). This extended version, provides **G**raphical **U**ser **I**nterfaces (GUIs) and advanced capabilities for model identification in both time and frequency domains. The toolbox features specialized tools for fractional PID controller design and optimization, along with a dedicated *Simulink* block set, making it a complete solution for fractional-order system analysis and control design [69, 70].

The toolbox can be split into four main parts (Figure 3.1):

1. the main system analysis module, *i.e.*, the toolbox core;
2. the identification module;
3. the control design module;
4. the implementation module.

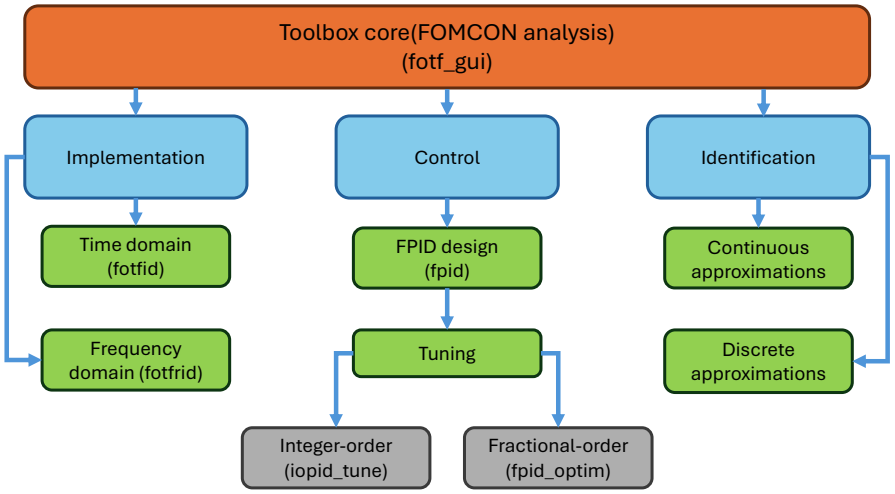


Figure 3.1: FOMCON toolbox structure

The main module (Figure 3.2), called FOFT Viewer is displayed by writing *fotf_gui* into the MATLAB command line. This window is separate into two panels:

- The left panel allows the addition, edition, delete and convert FOTF object;
- The right panel allow the analysis of fractional-order system in frequency domain and in time domain. It also allows the user to check the stability of fractional-order systems. The option *View in console* shows the selected transfer function on the MATLAB command window.

The time domain window, allows the simulation of a step and impulse response of the selected fractional-order system. The frequency domain window, enables the plot of Bode, Nyquist, Nichols and Root Locus diagram of the selected system.

The *Tools* option contains links to the other windows, such as the frequency-domain and time-domain tools and the fractional PID design tool.

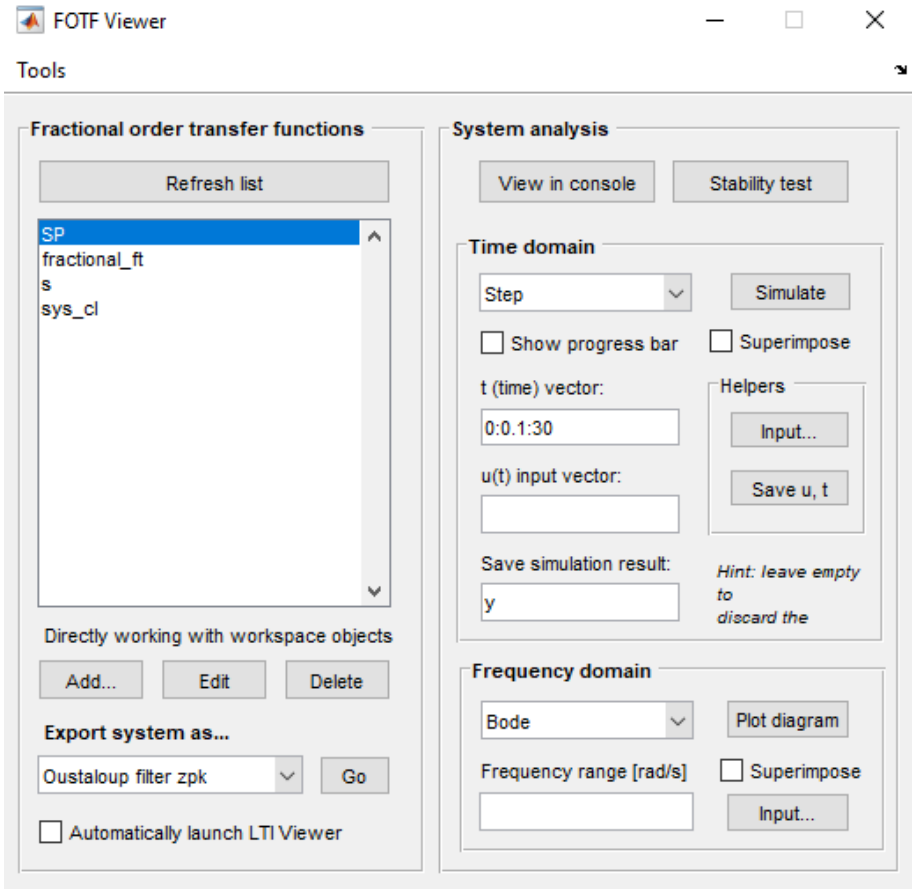


Figure 3.2: fotf_gui command window

3.2.1 Fractional PID Design Tool

For the concept of this controller, the most used tool was Fractional PID Design Tool. This tool can be accessed through *Tools* $\longrightarrow$ *Fractional PID Design* or by writing *fpid* into the prompt MATLAB. This window is split into two panels where the left panel contains the FOPID parameters and the right panel the control system (Figure 3.3). The *Simulate* button allows the visualization of the control system's step response, and *Open-Loop Bode* allows the visualization of both Magnitude and Phase Bode Diagrams (Figure 3.4). The *Import* option, allows to import the *FOPID* controller available on the workspace, which after imported, updates the parameters presented on the left panel. If not, it will assume the default values: 1 for K_p, K_i and K_d, and 0.5 for μ and λ.

The left buttons are related to the system's controller. The *Export controller* exports the controller with the defined parameters to the actual MATLAB Workspace. The *Realize Controller* opens a new window, that allows the visualization of the Bode Diagrams in both continuous-time and discrete-time approximations (Figure 3.5). The approximations on continuous-time that can be chosen are: Zero-pole gain, Transfer function and State-space. The available discretization methods are: Zero-order hold, Linear interpolation,

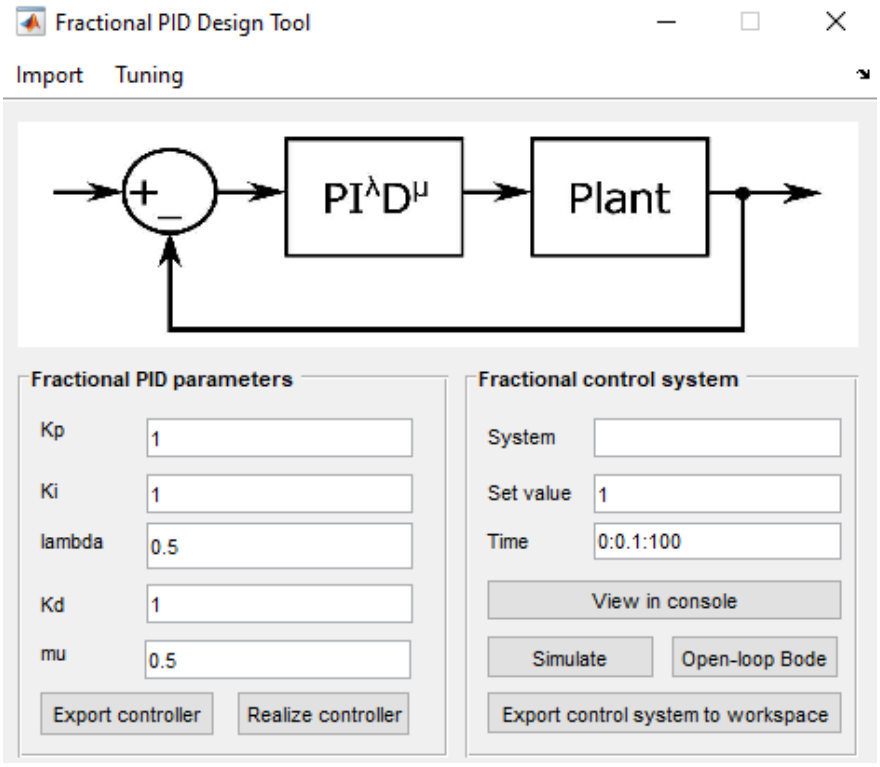


Figure 3.3: Fractional PID Design Tool [71]

Impulsive-invariant, Bilinear (Tustin) and Matched pole-zero. The fractional parameters K_p, K_i, K_d, μ and λ. The wb and wh represents the frequency ranges of order 10, with N being the approximation order.

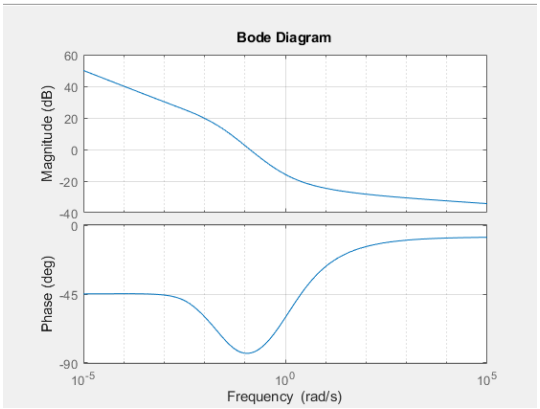


Figure 3.4: Bode Diagrams visualization trough the Open-Loop Bode button

Now related to the *Tuning* option contains two links, the *integer-order PID*, that allows the tuning of a PID controller based on the system and a model type (Figure 3.6). This option only works if the *System* box is filled with the control system's name, in this case, *heat_ sys*. This window is solely dedicated to the parametrization of a PID controller, where

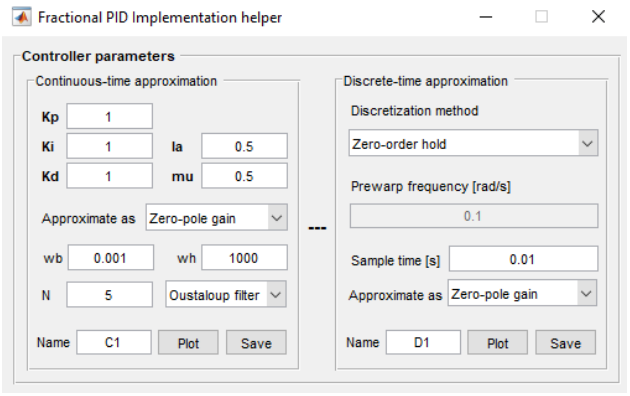


Figure 3.5: FOPID Implementation helper

the tuning method can be choose between: Ziegler-Nichols, Astrom-Hagglund (AMIGO), Chien-Hrones-Reswick 1 setpoint rejection), Chien-Hrones-Reswick 2 (disturbance rejection) and Cohen-Coon. The *Model type* refers to the system model with the following available options: FOPDT, **I**ntegrating **P**lus **D**ead **T**ime (IPDT) (in this option the parameter **T** is disable) and **F**ractional-**O**rder **I**ntgrating **P**lus **D**ead **T**ime (FOIPDT). The model parameters represents:

- **K** - Process gain;
- **L** - Time delay of hte system;
- **T** - Time constant of the first-order system;

The other link, *Optimize*, opens a new window that allows the optimization of the FOPID controller based on the system model. This window will be further explained in detail since it was the most used tool during the concept of the fractional controller.

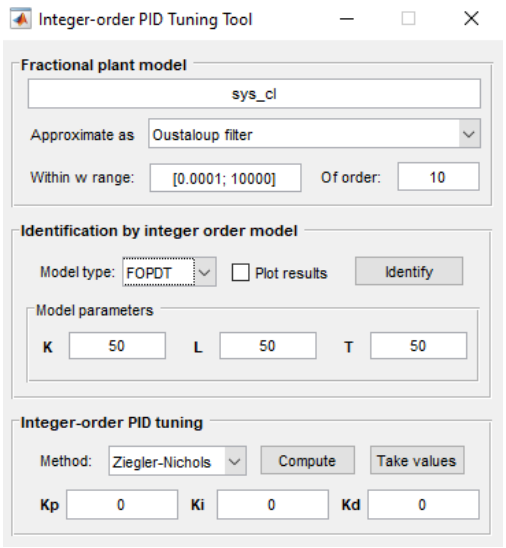


Figure 3.6: Integer-order PID Tuning Tool

3.3 FOPID Controller

The control of a heat system can be represented through a block diagram with a controller and the heat system transfer function (Figure 3.7) with feedback. A good parametrization of the controller parameters guarantees that, independently of the input value, it will present a precise response.

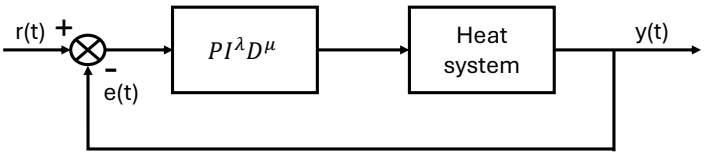


Figure 3.7: Block diagram of $\text{PI}^{\lambda}\text{D}^{\mu}$ controller

The heat system is a transfer function of a first-order system with dead time:

$$G(s) = \frac{K}{\tau s + 1} e^{-t_d s} \tag{3.1}$$

The parameters values applied to $G(s)$ are based on [20], $(K, \tau, t_d) = (0.1, 200, 30)$. Bearing this in mind, the optimization of the $\text{PI}^{\lambda}\text{D}^{\mu}$ controller was the first step, with the usage of the optimization tool mentioned on the previous Subsection 3.2. The first values used on the controller were simply placeholders, to test the optimization tool. Through the command *fpid*, the Fractional PID design Tool is shown in Figure 3.3. Firstly, is added the system's name into to the textbox, followed by the importation of the controller values. Hereafter, it is selected the *Optimize* link, that will lead for a new window (Figure 3.8).

In this window, there is a set of functionalities essential for the fopid parameter tuning. The left panel 1 does the identification of the actual system's name and type, and allows the variation of frequency range as well the order.

The left panel number 2, shows the parameters values of the controller imported on the previous window, if not, the values are set to it default value. The constraints table, allows the definition of the minimum value and maximum value where the optimizer can fluctuate the values, until it finds the optimal values for the current system. The panel number 3 grants the possibility to change the simulation time and the time step.

The right panel is constituted by a set of performance and optimization settings. For the controller development and study, the focus will be around the optimization algorithm and performance metric options.

Without further ado, the first test is on the optimization algorithms while maintaining the default performance metric, **I**ntegral **S**quare **E**rror (ISE). The optimization for each algorithm has the same starting parameters and constraints.

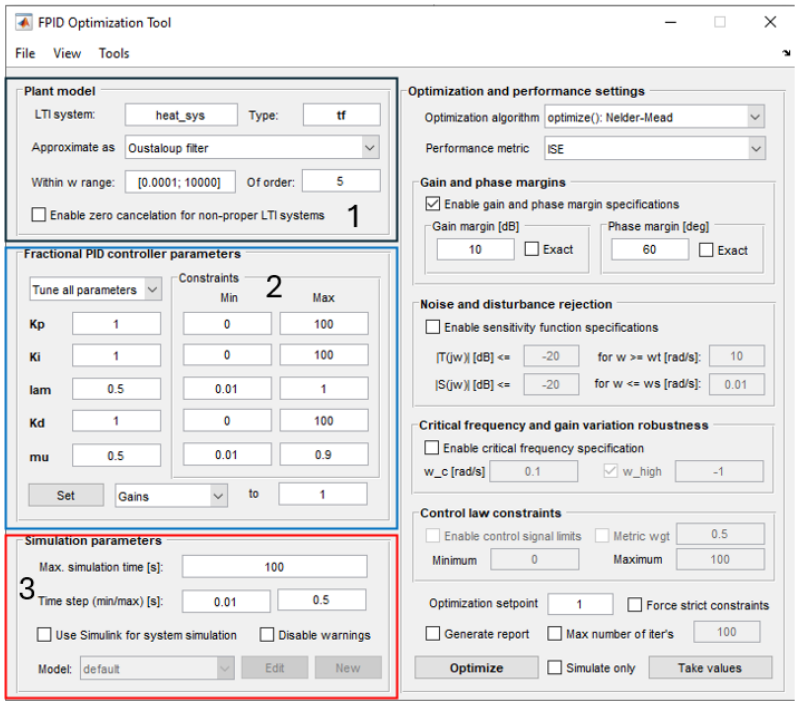


Figure 3.8: Optimize tool window

These values come from a first optimisation test using a set of parameters from the same article mentioned above ($\{K_p, K_i, \lambda, K_d, \mu\} = \{18.07, 0.5315, 0.7, 153.59, 0.8\}$, where λ and μ were randomly chosen). The available algorithms are:

- Nelder-Mead;
- Interior-point;
- SQP;
- Active-set;

The focus of this test is to observe each ones of the algorithm's step response, and verify which gave the best answer based on their optimized parameters (Figure 3.9 and Table 3.1).

Table 3.1: Parameter results for each tested algorithm

Algorithm	K_p	K_i	λ	K_d	μ
Nelder-Mead	45.8672	0.2949	1	99.9993	0.4170
Interior-point	23.385	0.1633	1.122	123.88	0.5815
SQP	7.6906	2.3202	0.580	131.01	0.3968
Active-set	16.782	0.05732	1	78.226	1

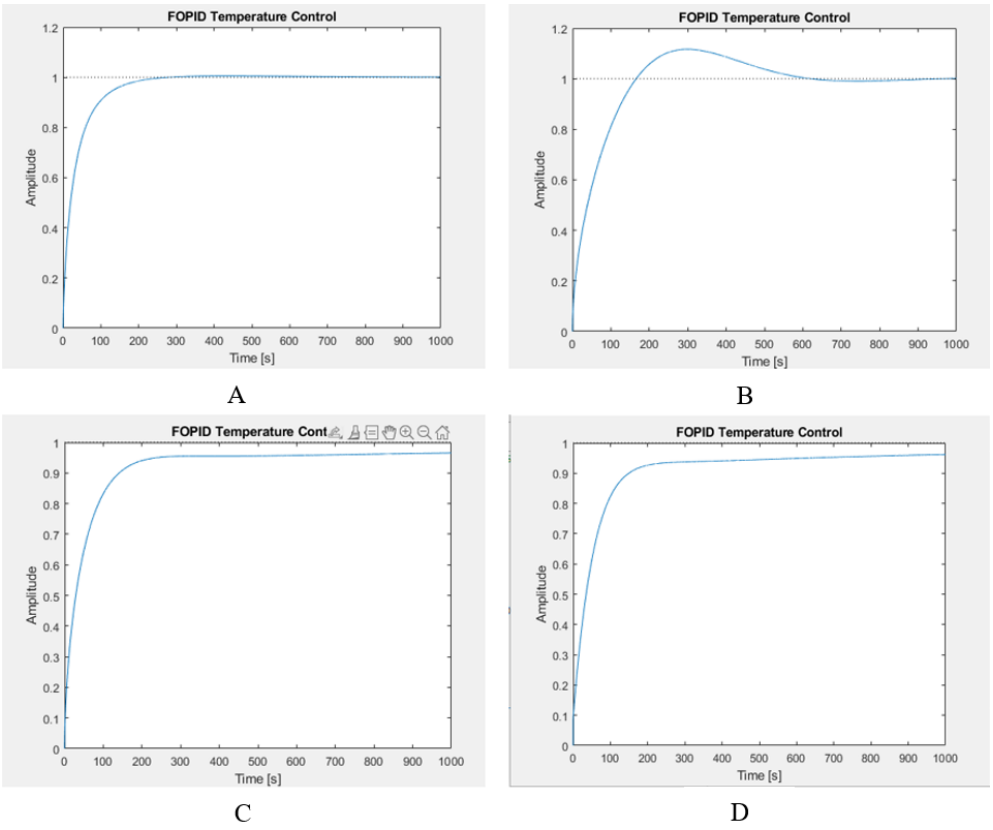


Figure 3.9: Step response of each algorithm: (A) Nelder-Mead, (B) Interior-point, (C) SQP, (D) Active-set

The algorithms SQP and Active-set show a slow settling time, where in a window of 1000 s, they did not reach the input value. Relatively to Nelder-Mead and Interior-point algorithms, both reach the input value, with the Interior-point algorithm showing a slight overshoot. Thereby, the Nelder-Mead is the chosen algorithm, since presents no overshoot and reaches the input value. Now, the next test will be around the performance metrics accessible in the toolbox, these being given by:

- **I**ntegral **S**quare **E**rror (ISE)

$$\int_0^T e(t)^2 dt \tag{3.2}$$

- **I**ntegral **T**ime **S**quare **E**rror (ITSE)

$$\int_0^T t.e(t)^2 dt \tag{3.3}$$

- **I**ntegral **A**bsolute **E**rror (IAE)

$$\int_0^T |e(t)| dt \tag{3.4}$$

- **I**ntegral **T**ime **A**bsolute **E**rror (ITAE)

$$\int_0^T t.|e(t)| dt \tag{3.5}$$

These performance metrics are used to improve the system's response, by lowering the closed-loop integral error [72]. As mentioned previously, the selected algorithm was Nelder-Mead, since it was the one who got the best step response. Following the same goal, this test will verify the performance metrics and see which one has the best step response. Thus, after realising the tests, the performance metric ISE gave the best response (Figure 3.10), since it had the lowest iterations, simulation count and performance index during the optimization process (Table 3.2).

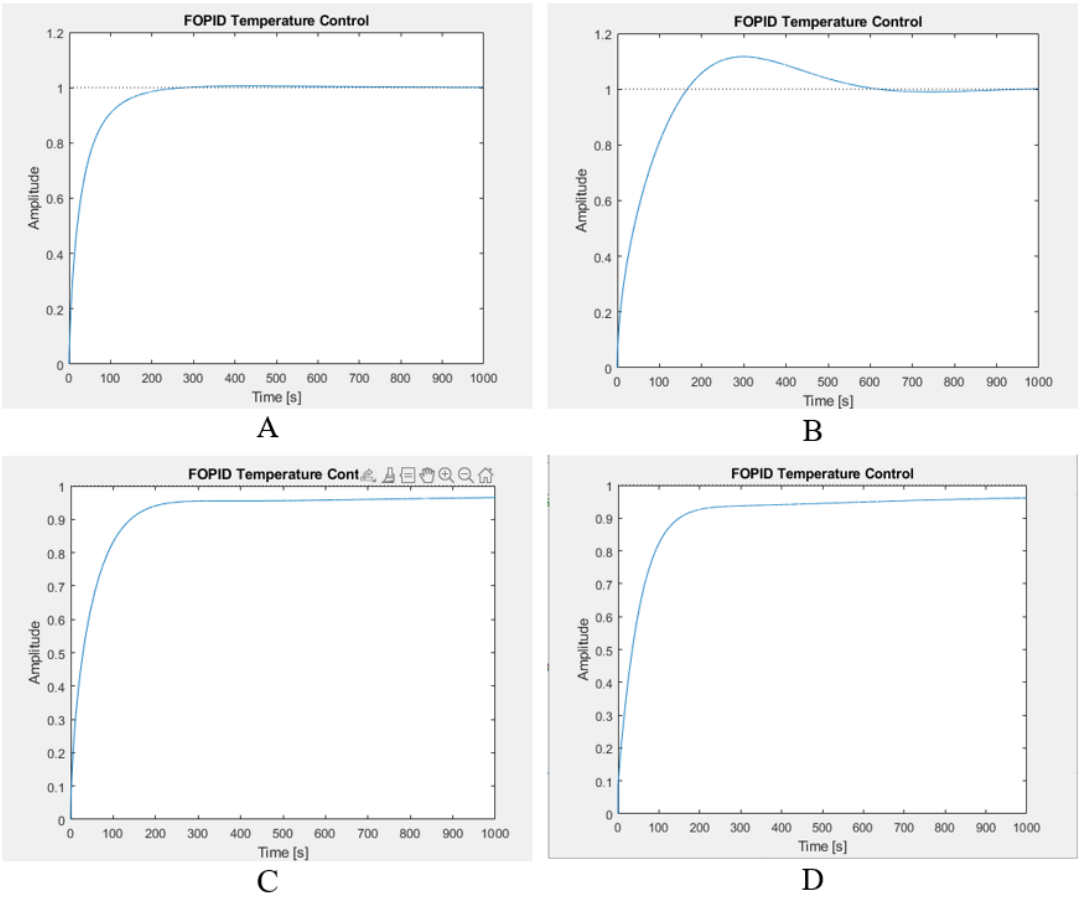


Figure 3.10: Step response of each performance metric: (A) ISE, (B) IAE, (C) ITSE, (D) ITAE

Table 3.2: Comparison between performance metric's value obtained during the Optimization Process

Performance Metric	Iterations	Performance Index	Simulation Count
ISE	293	46.865	503
IAE	348	52.230	613
ITSE	374	803.056	658
ITAE	403	1204.390	693

Therefore, the chosen algorithm and performance index was the Nelder-Mead with ISE, since it showed the best results in terms of step response and optimization performance. The controller parameters obtained by these process is $\{K_p, K_i, \lambda, K_d, \mu\} = \{45.867, 0.294, 1, 99.9993, 0.417\}$. Thus, this will be the parameters used for the following metrics tests.

With the controller parameters obtained, it was made a test with the performance metrics again, to observe the error variation according to the applied metric. In this test, the performance metrics were applied both to $PI^{\lambda}D^{\mu}$ and PID controller in order to compare the difference between them. For a fair comparison, the PID controller parameters are set to the same values as those of the fractional controller. The performance metric values were calculated for both instantaneous error and cumulative error. Hence, the Equations 3.2-3.5 were applied into the workspace to afterwards plot the results and obtain the value of the errors.

Table 3.3: Comparison of cumulative error metrics for FOPID and PID controllers

Controller	**ISE**	**ITSE**	**IAE**	**ITAE**
FOPID	15.278	322.646	39.429	3289.3
PID	44.361	1178.5	65.826	4027.2

By inspecting the Table 3.3, it is noticeable that the ISE has the lowest value for both controllers, proving to be the best performance metric in terms of cumulative error. More, it is clear that the fractional order controller improved the system response, when compared with PID controller, presented the smaller error values for all the index studied, Table 3.3.

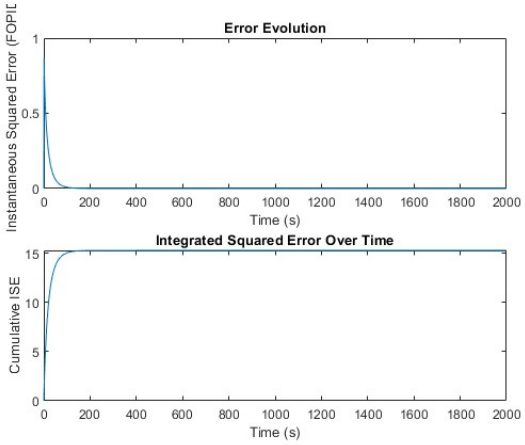
Relatively to instantaneous errors, as they are errors that occur at a given instance of time, instead they were plotted. For analysis purposes, the cumulative error was also plotted to verify the time each error takes to stabilizes, Figure 3.11 for FOPID and Figure 3.12 for PID controllers.

Thus, through the Figure 3.11 it is noticeable certain aspects:

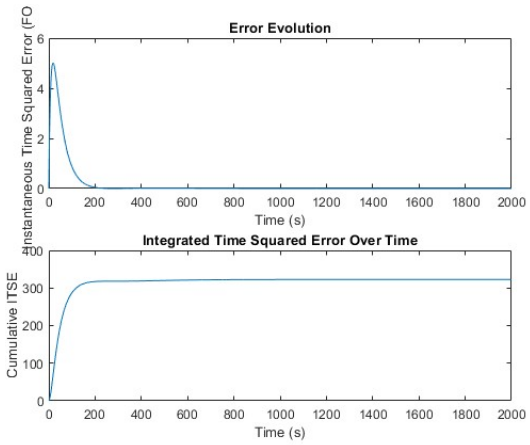
- **ISE** and **IAE** - even though their error evolution is practically equal, it is on the cumulative error where is the difference. The ISE metric stabilizes faster, when the time is 144 s, and IAE metric only at the time 846 s;
- **ISE** and **ITSE** - relatively to the error evolution, the most noticeable difference stands on the peak value and the rise time, with ISE being $(time, error\ \ value) = (1, 0.869)$ and ITSE $(time, error\ \ value) = (18, 5.006)$. Also, the ITSE metric takes longer until it reaches close to zero. The cumulative error, the only remarkable difference is their stabilization value, where ITSE is far superior than the ISE metric what is perceptible as this index includes the duration of the simulation;

- **IAE** and **ITAE** - having the same differences between ISE and ITSE error evolution described previously, the difference that stands the most between IAE and ITAE metric, is the cumulative error. Not only the ITAE value is hundred times higher, but also takes longer to stabilize.

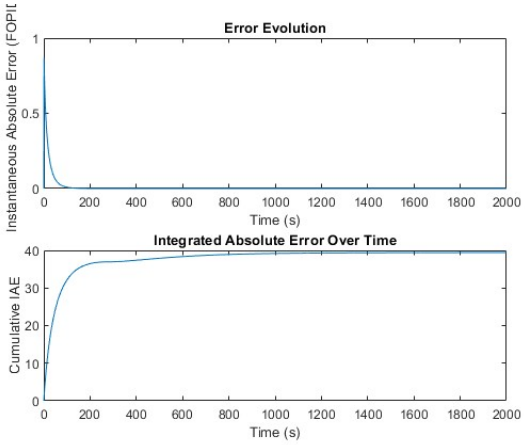
- **ITSE** and **ITAE** - as was the case when comparing the ISE with the IAE,, the error evolution of ITSE and ITAE is equal. Relatively to the cumulative error, the ITAE value is ten times greater than the ITSE metric and also takes longer to stabilize as mentioned previously.



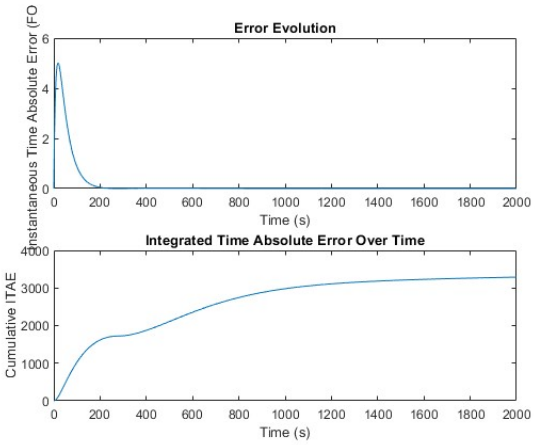
(A) Step error response using ISE



(B) Step error response using ITSE



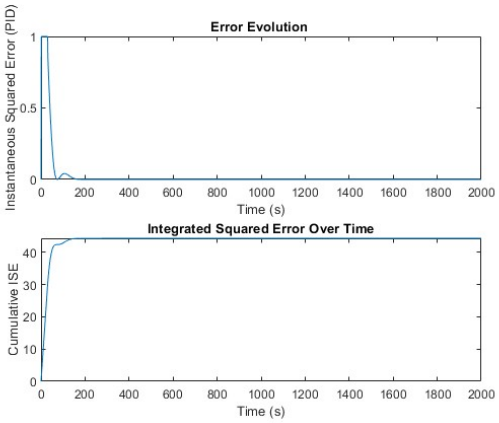
(C) Step error response using IAE



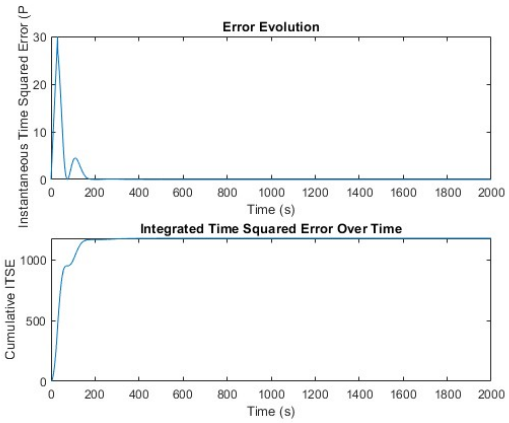
(D) Step error response using ITAE

Figure 3.11: Step error response for different performance metrics for FOPID controller

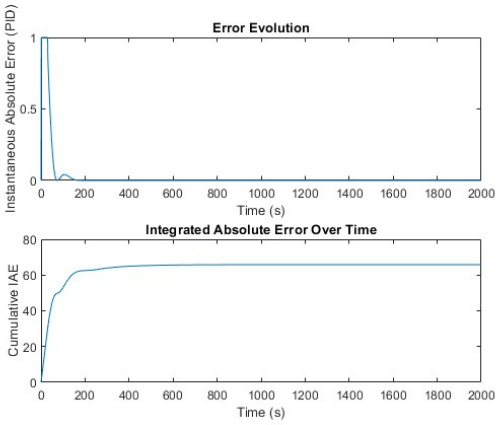
It was verified that the FOPID controller performance metrics comments drawn, are also applicable for PID, since most of the differences are similar.



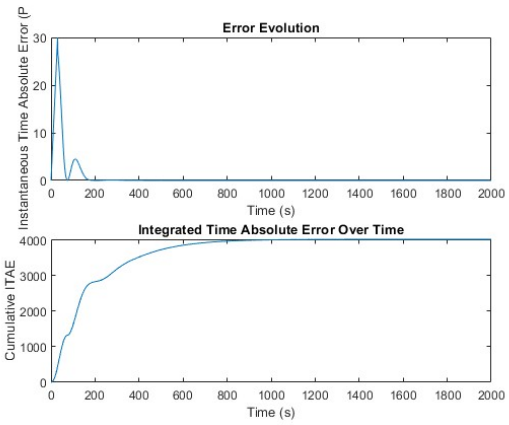
(A) Step error response using ISE



(B) Step error response using ITSE



(C) Step error response using IAE



(D) Step error response using ITAE

Figure 3.12: Step error response for different performance metrics for PID controller

In addition to these performance metrics, it was also made a new test to regression metrics, with these being [73]:

- **M**ean **A**bsolute **E**rror (MAE) - measures the average absolute difference between predicted and actual values. It is straightforward to understand and not heavily affected by outliers, where $\hat{y}$ is the predicted values:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \; where \;\; y_i - \hat{y}_i = e \tag{3.6}$$

- **M**ean **S**quared **E**rror (MSE) - similar to MAE but squares the errors, penalizing larger errors more heavily:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.7}$$

- **M**ean **A**bsolute **P**ercentual **E**rror (MAPE) - error is presented as a percentage, useful for understanding performance relative to actual values. Can be unstable when the value is near zero:

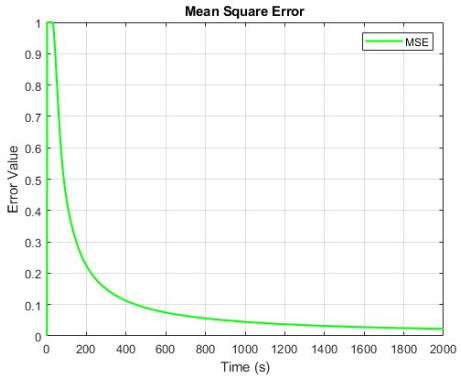
$$\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{max(\varepsilon, |y_i|)} \tag{3.8}$$

  where ε is a small constant to prevent division by zero, usually set between 10^{-3} and 10^{-2} according to the system's scale. For this study case it was needless.

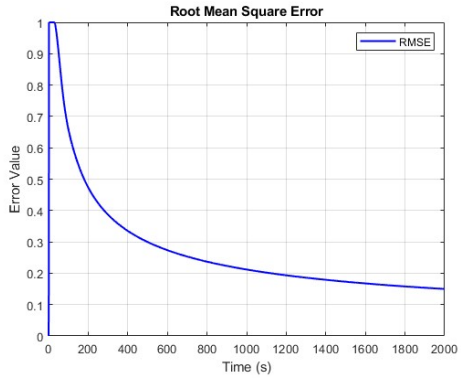
- **R**oot **M**ean **S**quared **E**rror (RMSE) - square root of MSE, maintaining the penalization to larger errors keeping the unit consistent with the data:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.9}$$

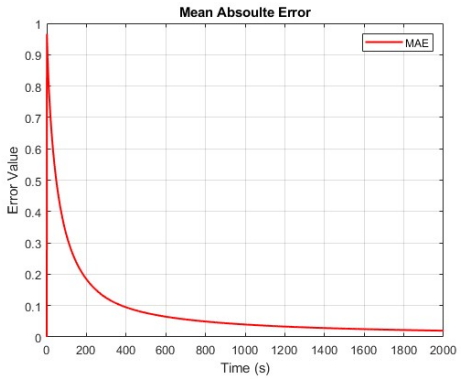
Figures 3.13 and 3.14 present the error response for different regression metrics, for FOPID and PID controllers, respectively:



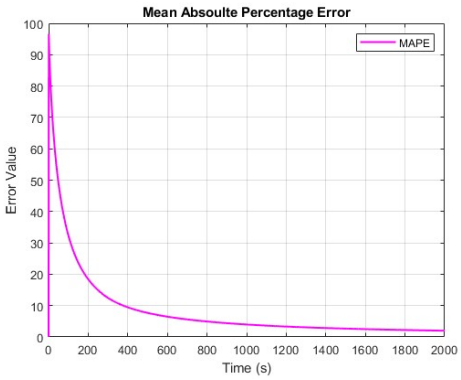
(A) Step error response using MSE



(B) Step error response using RMSE

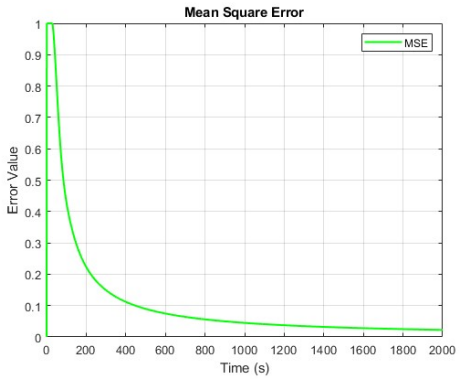


(C) Step error response using MAE

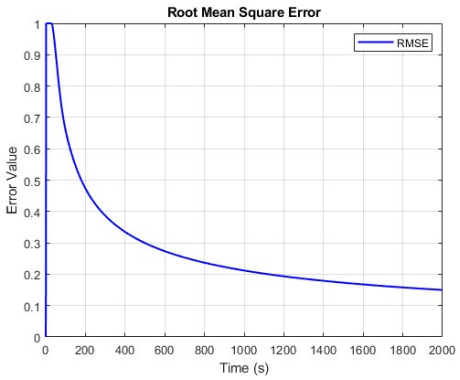


(D) Step error response using MAPE

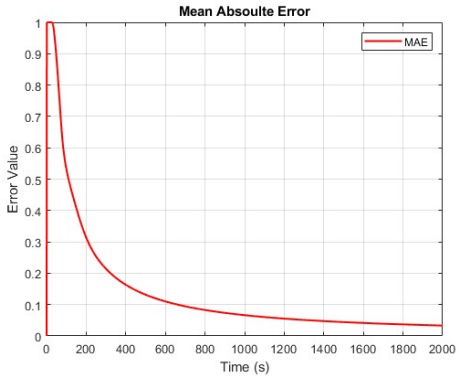
Figure 3.13: Step error response for different regression metrics for FOPID controller



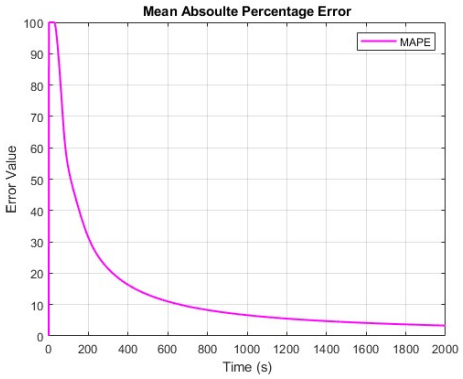
(A) Step error response using MSE



(B) Step error response using RMSE



(C) Step error response using MAE



(D) Step error response using MAPE

Figure 3.14: Step error response for different regression metrics for PID controller

Graphically, the evolution of MSE and MAE over time appeared nearly identical (Figures 3.13 and 3.14). To complement the visual analysis, the final values of both MSE and MAE were computed over the entire simulation period (Table 3.4).

Table 3.4: Comparison of regression metrics for FOPID and PID controllers

Controller	**MSE**	**RMSE**	**MAE**	**MAPE**
FOPID	0.0079	0.088	0.02	1.995
PID	0.0224	0.1497	0.0331	3.314

These final values provides a concise indicator of overall accuracy on the regression metrics. Lastly, it is important to note that there is no possible comparison between these set of metrics since they serve different purposes: the performance evaluate systems dynamic, while the regression metrics focus on the accuracy of prediction across samples.

3.4 Simulink Simulation

Following the evaluation of performance and regression metrics, a Simulink model was developed to simulate the dynamic behaviour of the system under closed-loop control configurations (Figure 3.15). The controller parameters used for the following tests were $\{K_p, K_i, \lambda, K_d, \mu\} = \{45.867, 0.294, 1, 99.9993, 0.417\}$, which are the values resulting from the experiments carried out previously, having responded well to the proposed objectives, namely, obtaining faster system responses in transient regime (minimising the effect of delay on the system response), smaller overshoot, and faster steady-state response.

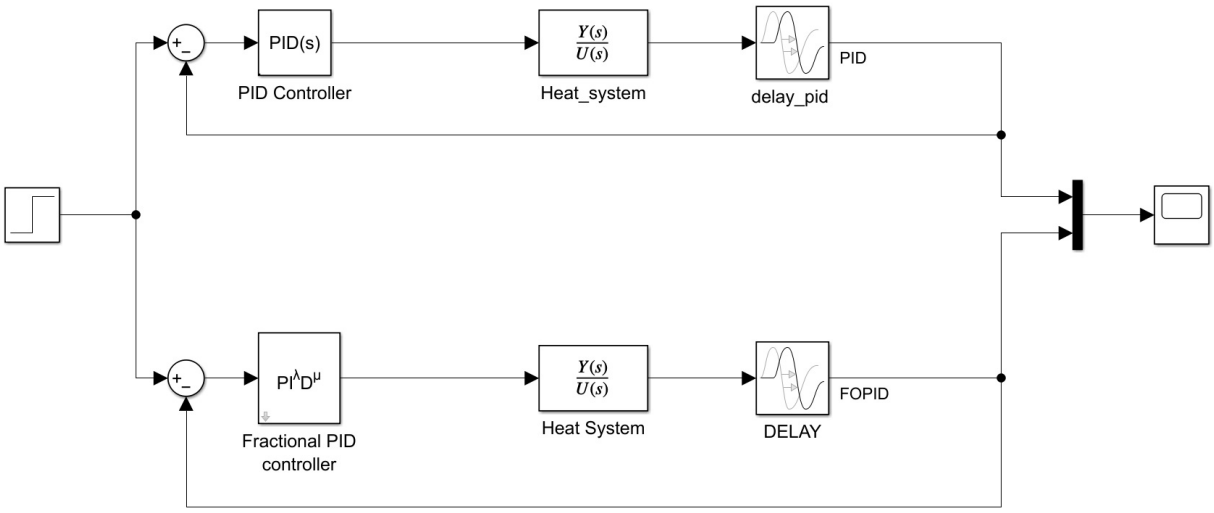


Figure 3.15: Block Diagram of the PID and FOPID controller performance for comparasion applied to the heat diffusion system

The first experiment focused on observing the system behaviour in a closed-loop control and comparing the step responses of FOPID with PID controller. To represent the heat diffusion system is necessary the *Heat_ system* transfer function and a delay block. The simulation time was two thousand seconds and the step block defined with an amplitude of one. By observing Figure 3.16, both signals present overshoot, with the FOPID being the highest. Despite FOPID having a faster rise time, it takes longer time to stabilize when compared to the PID controller (Table 3.5).

Table 3.5: Comparison of time metrics for FOPID and PID controllers

Controller	**Rise Time (tr)**	**Peak Time (tp)**	**Overshoot**(%)	**Settling Time (ts)**	**Steady-State Error** (e_{ss})
FOPID	22.821	81.074	30.1	222.515	0
PID	35.930	107.793	20.3	178.847	0

To overcome this situation, the first step was the addition of non-linearity block - NL (Figure 3.17), since it can help with robustness and stability of the system.

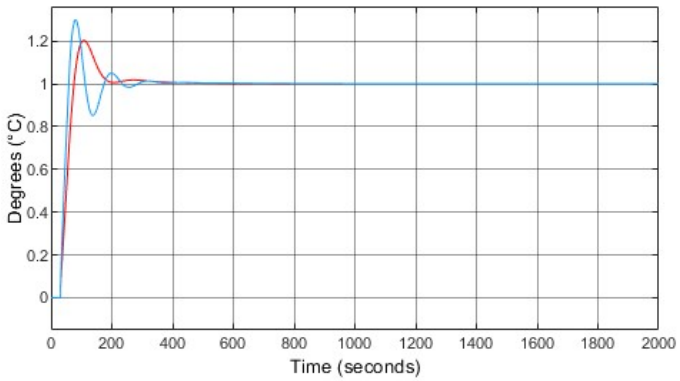


Figure 3.16: Step response of the PID (red) and FOPID (blue) controller for the block diagram of Figure 3.15

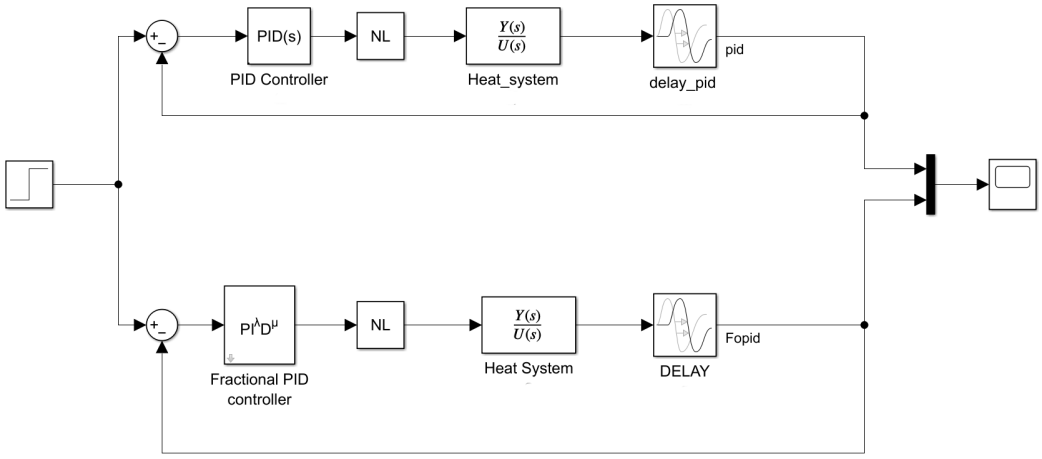


Figure 3.17: Block Diagram of the PID and FOPID controller with the addition of the non-linearity block (NL)

Each one of these non-linearities will be explained in the following subsections. Furthermore, the non-linearities blocks tested were:

- Saturation;
- Dead zone;
- Relay;
- Backlash.

3.4.1 Saturation

The first block tested was the saturation (Figure 3.18), where it limits the input signal to a specified upper and lower bound that, when exceeded saturates the signal at a minimum or maximum value. The block was tested for following ranges: 0-25%, 0-50%, 0-75% and 0-100% (Figure 3.19).

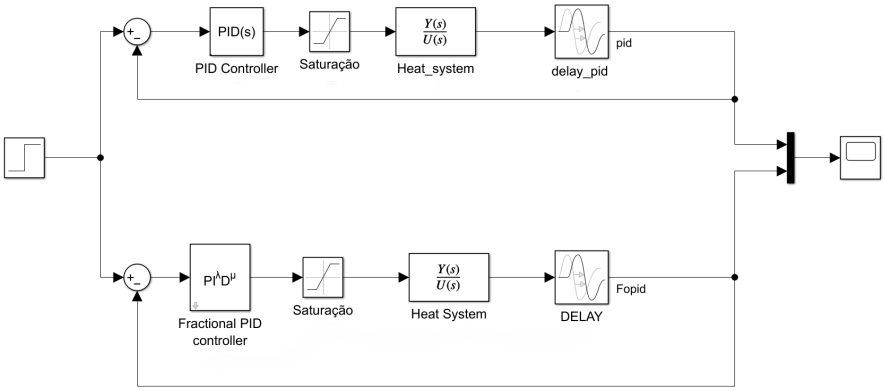
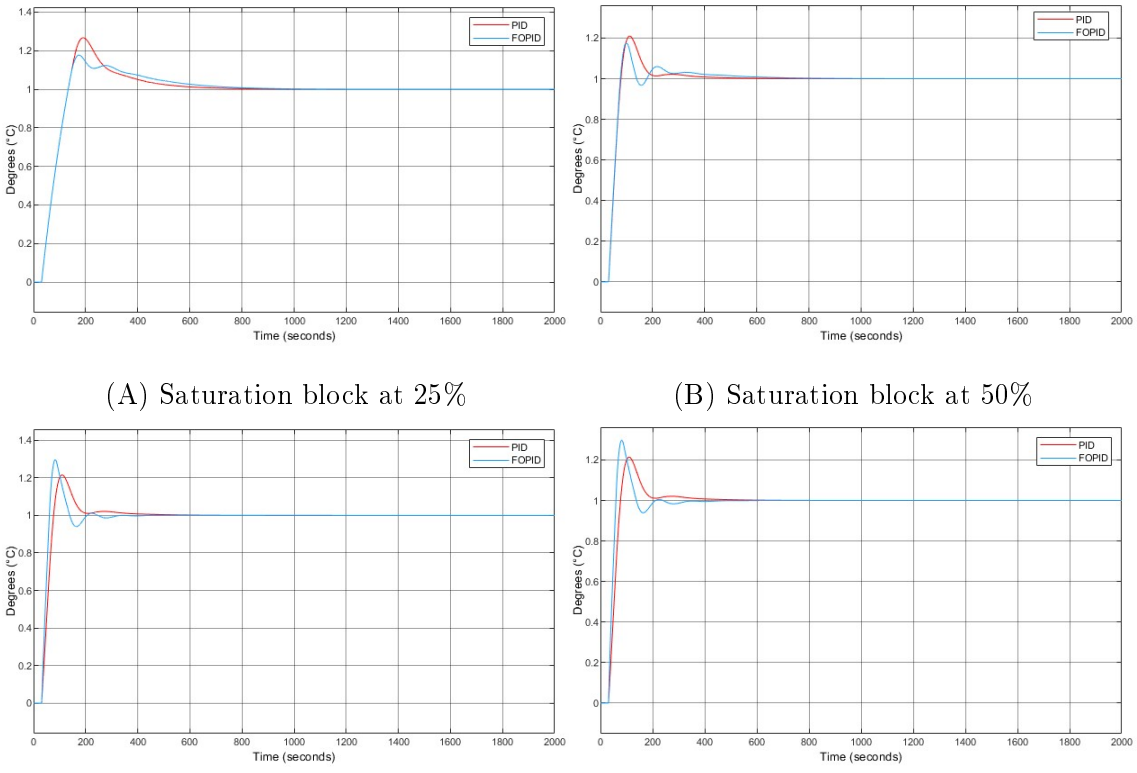


Figure 3.18: Addition of the Saturation block to the system



(A) Saturation block at 25%

(B) Saturation block at 50%

(C) Saturation block at 75%

(D) Saturation block at 100%

Figure 3.19: Step response to different saturation values

Upon analysing the results, it was confirmed that the values above 75% of the saturation block, the FOPID has lower settling time, rise time and peak time. At values below 50%, the FOPID overshoot is smaller than PID, but has an increase on settling time. Moreover, the FOPID rise time become identical to PID controller (Table 3.6). These results lead to conclude that the fractional order controller improved the response in the transient regime, anticipating rise time, peak time, when compared to the integer PID.

The peak rise at 25% occurs due to the signal entering an oversaturated stage. For lower saturation values. as demonstrated in the Figure 3.20, with 1, 2 and 3 being the step response for the saturation levels at 20%, 15% and 10% respectively, the step response ends up with a large settling time. The small increase on the overshoot when compared to a saturation level of 50% in the Figure 3.19, is consequent of the signal getting close to an oversaturated stage, leading to a signal cut-off, as showed at a saturation level of 10%.

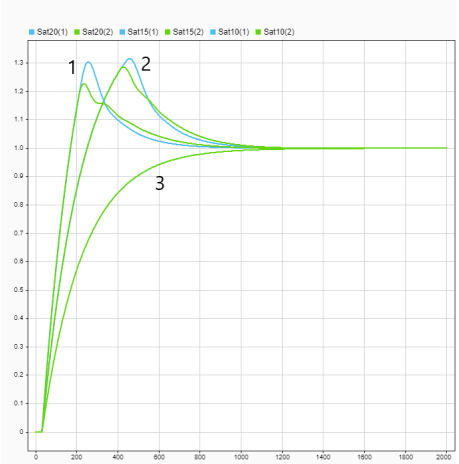


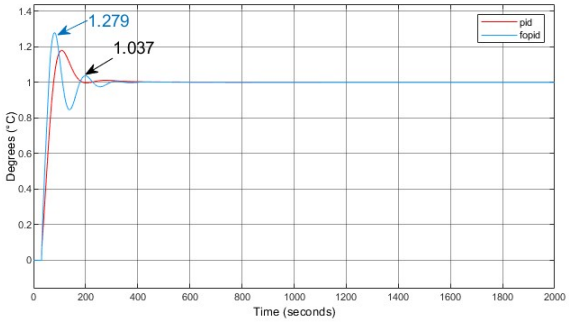
Figure 3.20: Comparison of lower saturation values (green represents FOPID and blue represents PID)

Table 3.6: Comparison of time metrics under different saturation configurations

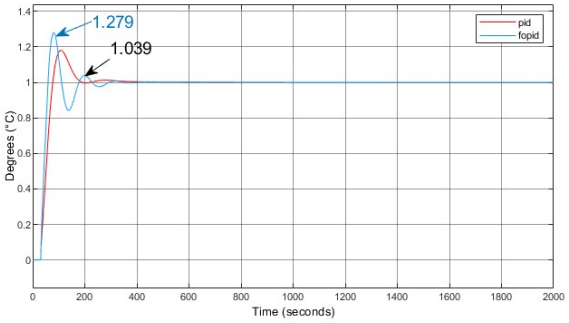
Controller	**Sat. (%)**	**Rise Time (tr)**	**Peak Time (tp)**	**Overshoot (%)**	**Settling Time (ts)**	**Steady-State Error (e_{ss})**
FOPID	100	23.081	81.774	30.0	195.695	0
	75	23.759	83.313	29.6	193.013	0
	50	35.651	99.348	17.8	417.082	0
	25	81.092	173.702	17.7	643.340	0
PID	100	35.439	110.440	21.5	289.305	0
	75	35.433	109.783	21.5	289.310	0
	50	36.534	111.905	21.1	299.797	0
	25	81.095	191.841	26.6	528.506	0

3.4.2 Dead Zone

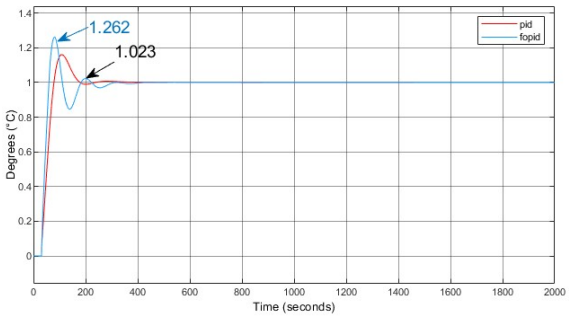
Afterwards, it was tested the dead zone block, where this non-linearity allows to specify a range that ignores small inputs within it. As with saturation block, it was chosen certain ranges: 0.8-1.2, 0.5-1.2, 0.8-2.2 and 0.8-3.2. Figure 3.21 shows the results found.



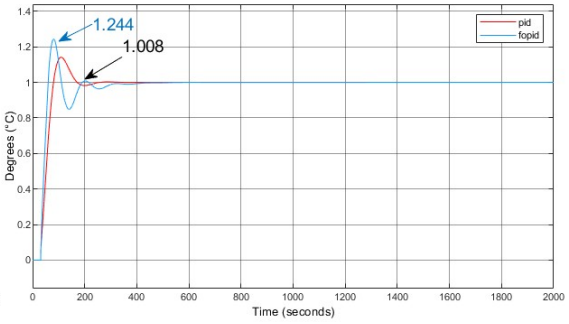
(A) Dead zone range within 0.5 and 1.2



(B) Dead zone range within 0.8 and 1.2



(C) Dead zone range within 0.8 and 2.2



(D) Dead zone range within 0.8 and 3.2

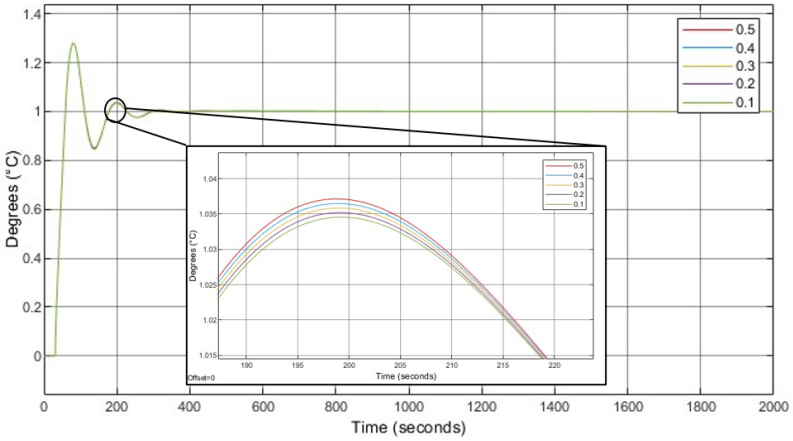
Figure 3.21: Step response to different dead zone ranges

The Figure 3.21 shows that the higher upper limit is, the peaks of step response will lower. The decrease in the minimum limit from 0.8 to 0.5 is attributed to the fact that the difference in the step response is zero within the range of 0.8 and 0.5. For limits below 0.5 the difference in step response is minimal (Table 3.7).

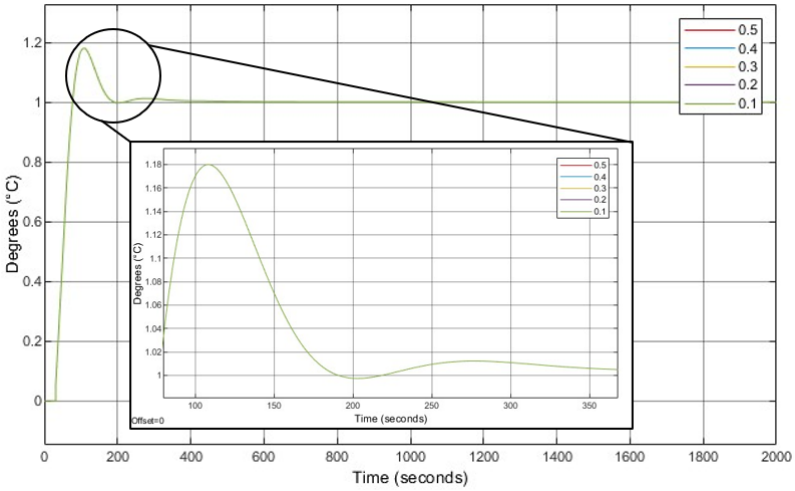
These measurements were made in the same spots as shown in the Figure 3.21 for FOPID, and in the PID controller they were made in the first peak and second peak (Figure 3.22). Relatively to PID, the peak values remained consistent across the tested dead zone range. For the FOPID controller, reducing the lower dead zone value resulted in a marginal decrease in the output, though this effect was not significant. Notably, while both controllers shared identical K_p, K_d, and K_i parameters, the fractional-order parameters, $\lambda = 1$ and $\mu = 0.4170$, in the FOPID architecture were found to change its response compared to the conventional PID controller.

Table 3.7: Comparison of response values for different lower limits of the Dead Zone block

Lower Limits	Controller			
	FOPID		PID	
	1st Peak	2nd Peak	1st Peak	2nd Peak
0.5	1.279	1.037	1.180	1.012
0.4	1.279	1.036	1.180	1.012
0.3	1.279	1.036	1.180	1.012
0.2	1.279	1.034	1.180	1.012
0.1	1.279	1.034	1.180	1.012



(A) FOPID response for for different lower values of deadzone



(B) PID response for different lower values of deadzone

Figure 3.22: Controllers response for different lower limits of the Dead Zone block

Therefore, after carrying out several tests, it was found that for a upper limit superior than 10, the system is delayed and the overshoot that exists got lower (Figure 3.23 and Table 3.8). In the case of the PID it became a small overshoot, and in the case of the FOPID it removed the second peak that the response had.

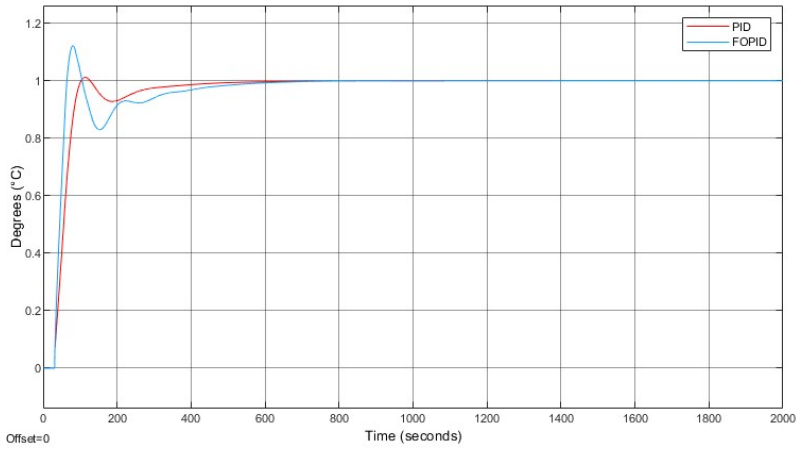


Figure 3.23: Step response of the dead zone block with upper limit of 10

Table 3.8: Comparison of response values for a upper limit of 10 of the Dead Zone block

Controller			
FOPID		**PID**	
1st Peak	**2nd Peak**	**1st Peak**	**2nd Peak**
1.121	0.930	1.012	0.9704

3.4.3 Relay

The next block tested was the relay, it outputs a fixed "on" value or "off" value depending on whether the input crosses a defined threshold, introducing switch-like behaviour. Compared to others non-linearities, the relay requires more parameters for its operation. After some initial tests, it turned out that the relay was not an ideal choice for the system (Figure 3.24).

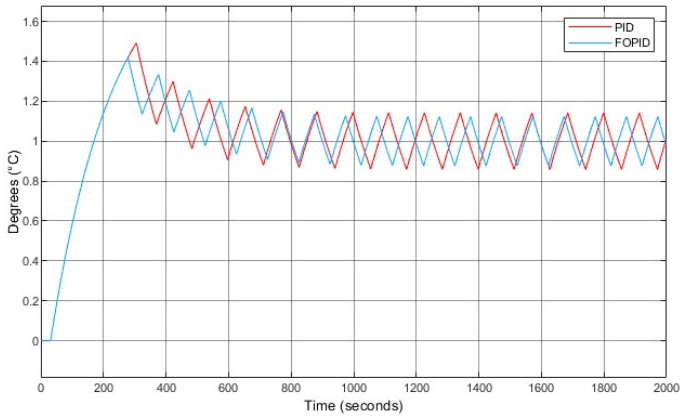


Figure 3.24: Step response with relay block applied

Furthermore, if the step response is narrowly constrained so that it cancels out the waveform generated, the system suffers a huge delay, increasing the system's response by at least twice as long (Figures 3.25). To produce such results the relay block parameters were change to:

- **Switch on point** – 0.99;
- **Switch off point** – 0.98;
- **Output when on** – 10;
- **Output when off** – 0.

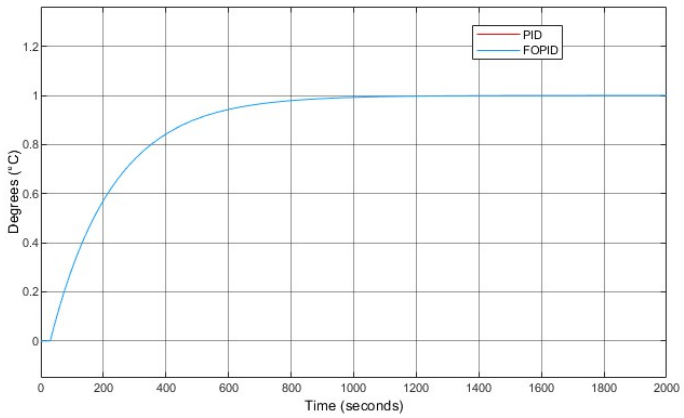
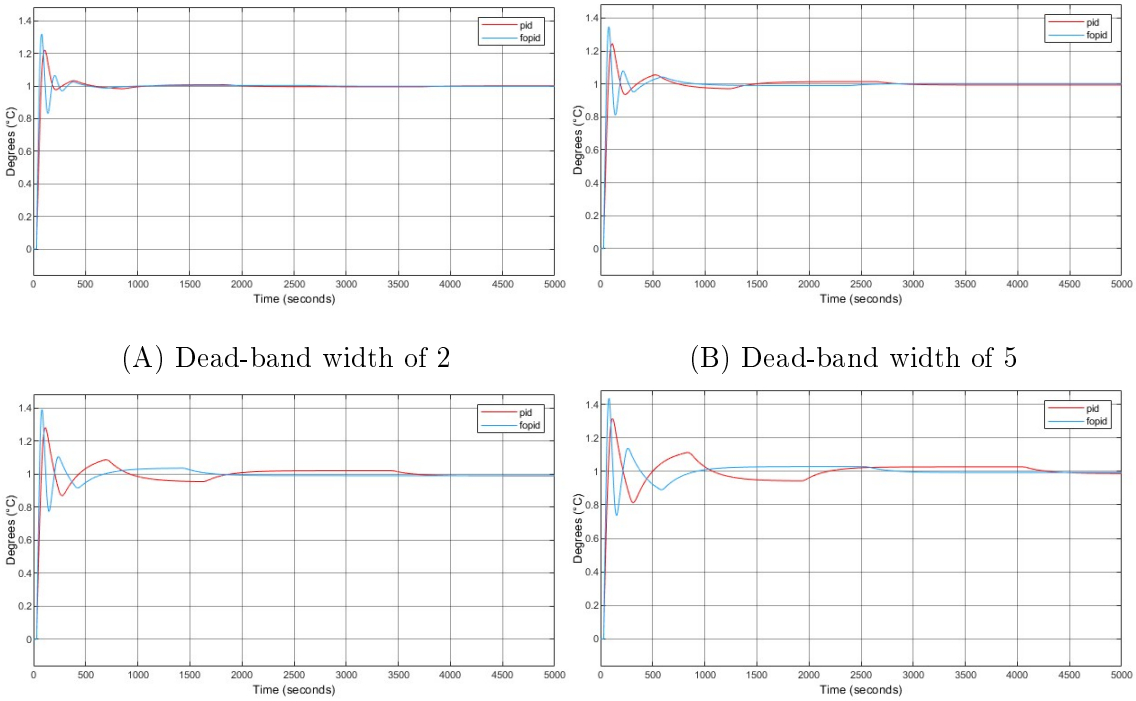


Figure 3.25: Step response with increased delay caused by the relay block (signals overlapped)

3.4.4 Backlash

The backlash block introduces a nonlinear hysteresis effect, where a change in input does not immediately produce a corresponding output unless it exceeds a specified backlash

width. Four values representing the dead-band width were chosen for this controller: 2, 5, 10 and 15. Figure 3.26 shows the step response to these different dead-band widths.



(A) Dead-band width of 2

(B) Dead-band width of 5

(C) Dead-band width of 10

(D) Dead-band width of 15

Figure 3.26: Step response to different dead-band widths

After carrying out the test for the first value (Figure 3.26 (a)), a variation in the PID response was observed toward the end of the simulation, which led to an increase in the simulation time to 5000 s. Upon testing the other values, it was found that the PID controller had difficulty settling in the presence of this non-linearity. Furthermore, the results indicate that the system is sensitive to this type of non-linearity.

In contrast, the FOPID controller performed better, reaching the target value with a 2% error after 1000 s (Table 3.9). Thus, the use of the backlash block resulted in a delayed system response and a distorted output signal.

Table 3.9: Time steady-state stamps comparison between different dead-band widths

Dead-band Width	Time for $\pm$**2%** $C_{ss}(s)$	Time for $C_{ss}(s)$
2	423.173 s	1291.380 s
5	791.693 s	1400.687 s
10	913.492 s	2184.572 s
15	1269.519 s	3165.209 s

Thus, among the non-linearities tested, saturation proved to be the most suitable for the heat diffusion system under analysis. Moreover, unlike the other non-linearities such as relay and backlash, it does not compromise the stability of the response, which can introduce meaningful oscillations or delays. Saturation enables an analysis that more closely reflects the system's real behaviour, maintaining a smooth and predictable response for both the PID and FOPID controllers.

3.5 Anti-windup system

The earlier section discussed and concluded that the saturation block is the most suitable non-linearity for the current system. To further enhance the system's performance, and reach a better response from the heat diffusion system, was implemented a anti-windup system to assess potential improvements in step response. An anti-windup system limits or resets the integrator when the actuator saturates. It is a modification usually implemented in PID controllers to prevent integrator windup [74].

To evaluate the effects of this adjustment, the anti-windup modification was firstly applied to one PI controller while a standard PI controller was retained as a baseline for comparison (Figure 3.27).

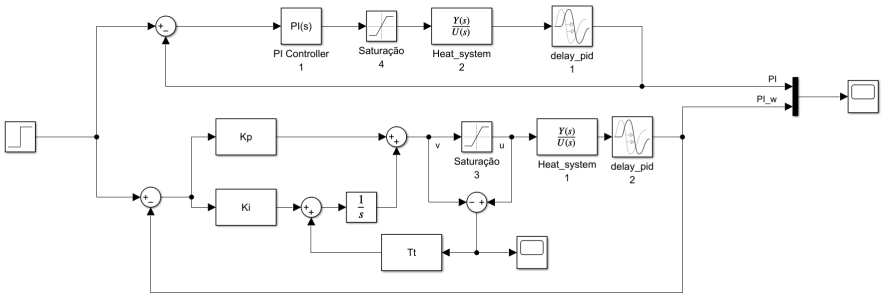
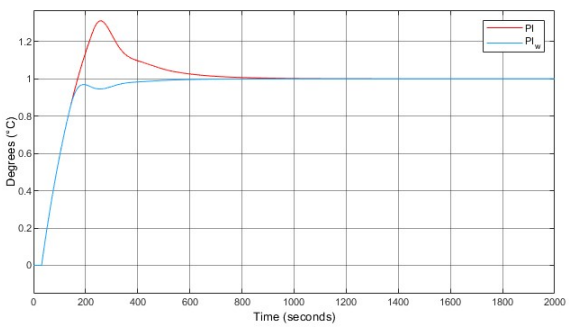


Figure 3.27: Block diagram of a PI controller with anti-windup

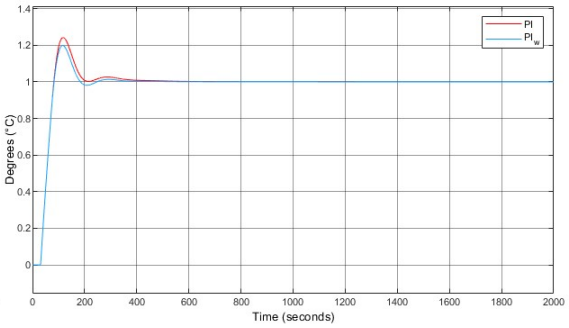
To implement the anti-windup mechanism, a sum block must be added to compute the difference between the saturated and unsaturated controller outputs $(u - v)$. This difference is then divided by the anti-windup time constant (T_t) and fed back to the integral component. The time constant is defined as:

$$T_t = 0.5 \times T_i \tag{3.10}$$

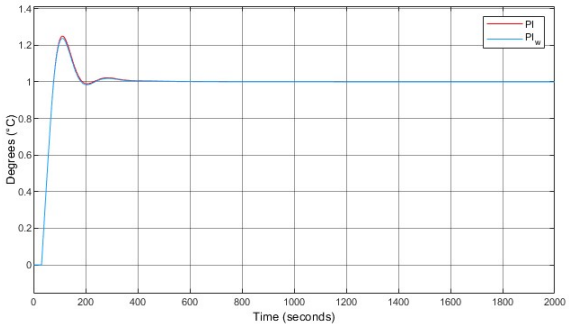
To evaluate the anti-windup system's impact on system response, tests were conducted with different saturation limits: 20%, 45%, 50%, and 80% (Figure 3.28). The change of limits when compared to the subsection 3.4.1 is due to the fact that between 80% and 100% the response is the same and the difference at 20% is more noticeable than at 25%.



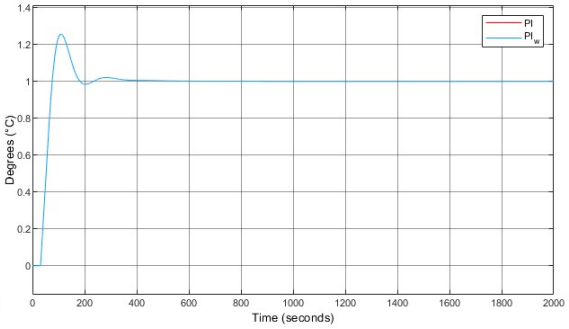
(A) Anti-windup with saturation at 20%



(B) Anti-windup with saturation at 45%



(C) Anti-windup with saturation at 50%



(D) Anti-windup with saturation at 80% (signals overlapped)

Figure 3.28: Step response to different saturation values for PI controller with and without Anti-windup

The analysis that compares the PI controllers with and without windup, demonstrated that the anti-windup mechanism only becomes noticeable at saturation levels below 50%, where the overshoot is mitigated (Table 3.10). For saturations above 20%, the rise time tr and peak time tp are similar for both controllers. When the saturation is 20%, the controller without windup not only presents a shorter tr, but also a higher overshoot when compared to the controller with anti-windup, which the overshoot is non-existent. Regarding the values of overshoot and settling time, these are significantly lower when the windup is implemented, across all saturation levels. However in certain cases, the usage of such low saturation limits may compromise system performance for different input values, for example, for a input with amplitude equal to two (Figure 3.29).

The addition of the component D demonstrated no significant improvements, leading to identical results, which can be consulted on the Appendix A.

Table 3.10: Comparison of time metrics under different saturation configurations for PI controller with and without Anti-windup

Controller	**Sat. (%)**	**Rise Time (tr)**	**Peak Time (tp)**	**Overshoot (%)**	**Settling Time (ts)**	**Steady-State Error (e_{ss})**
PI_W	80	34.845	109.255	25.7	294.5016	0
	50	36.211	111.194	23.8	173.481	0
	45	40.383	115.811	20	174.345	0
	20	111.263	–	–	375.238	0
PI	80	34.845	109.255	25.7	294.5016	0
	50	36.132	111.194	25	303.660	0
	45	40.163	118.254	24.2	326.178	0
	20	109.309	256.444	31.0	634.973	0

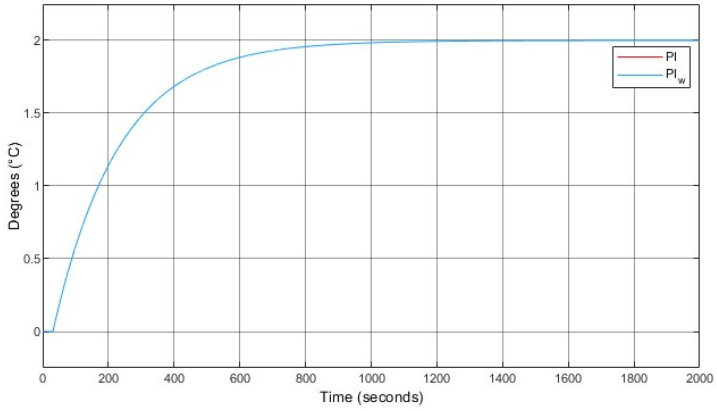


Figure 3.29: Step response of PI, with and without Anti-windup, with 20% saturation for a input of amplitude equal to 2

Because the PI controller has an λ of "1" (value chosen for these experiments), the fractional controller for PI becomes an integer controller, *i.e.*, the same as described above. Thus, only the fractional PID controller, will be analysed, (Figure 3.30) with $\lambda = 1$ and $\mu = 0.417$.

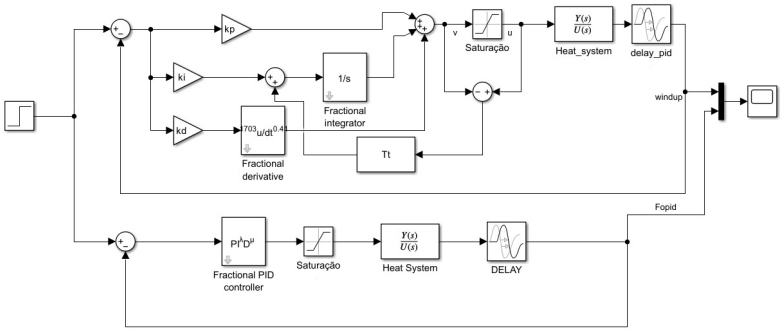


Figure 3.30: Block diagram of a FOPID controller with anti-windup

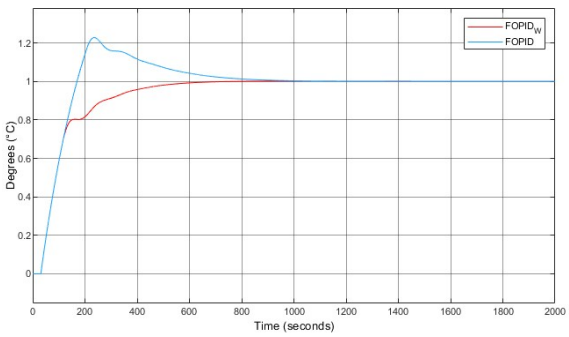
A further test was made to compared the behave between fraction and classical controller when the D component is added (Table 3.11). The fractional controller presents an overshoot and tp much smaller than the classical controller, however, the values of tr and ts are greater than the values obtained for classical controller.

The experimental results demonstrated that, in the case of the fractional controllers with and without anti-windup, it is conclusive that the transient response for the various saturation values is better for the FOPID controller (Figure 3.31) with anti-windup. When compared the PI and PID controller structures, with and without anti-windup, it is conclusive that the introduction of derivative component helped the system reach its final value more quickly, also reducing overshoot.

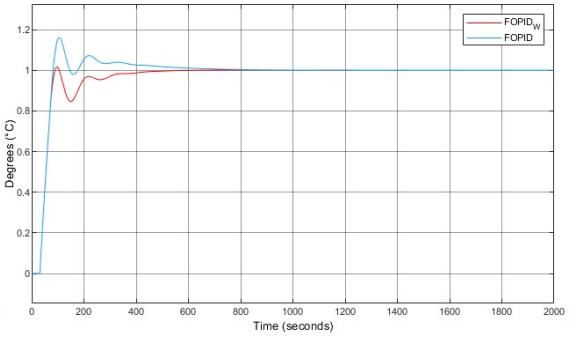
Table 3.11: Comparison of time metrics under different saturation configurations for PID and FOPID controllers with Anti-windup

Controller	λ	μ	**Sat (%)**	**Rise Time(tr)**	**Peak Time(tp)**	**Over shoot (%)**	**Settling Time (ts)**	**Steady -state Error (e_{ss})**
FOPID_W	1	0.417	80	23.918	82.070	27.2	330.665	0
	1	0.417	50	36.280	92.994	6.4	301.313	0
	1	0.417	45	41.498	96.833	1.7	322.367	0
	1	0.417	20	229.418	–	–	497.751	0
PID_W	–	–	80	35.510	109.617	21.3	285.752	0
	–	–	50	36.673	111.813	20.1	185.107	0
	–	–	45	40.779	115.462	16.9	183.524	0
	–	–	20	111.603	–	–	356.403	0

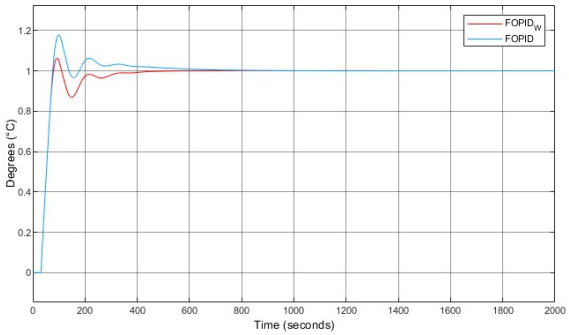
Thus, the addition of the D component in both controllers was found to eliminate the overshoot value and peak time. Notably, at 20% saturation for an input value equal to 2, both controllers converged to similar dynamic behaviour (Figures 3.32).



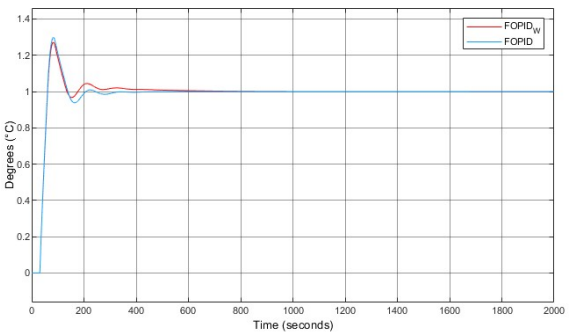
(A) Saturation at 20%



(B) Saturation at 45%



(C) Saturation at 50%



(D) Saturation at 80%

Figure 3.31: Step response to different saturation values for FOPID controller with and without Anti-windup

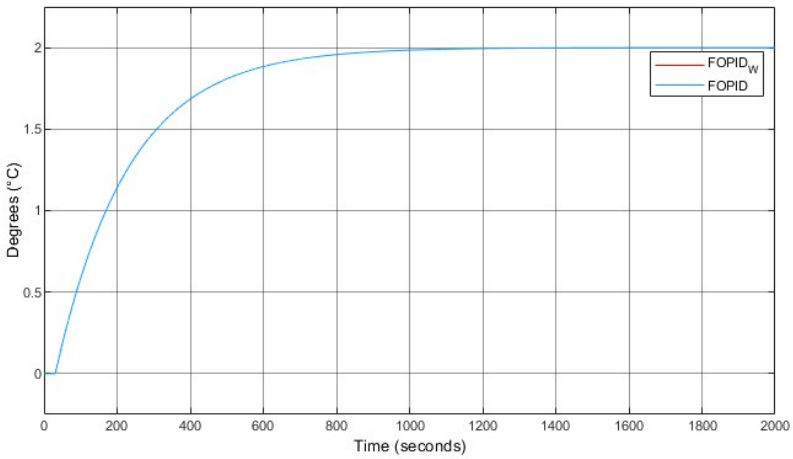


Figure 3.32: Step response of FOPID, with and without Anti-windup, with 20% saturation for a input of amplitude equal to 2 (signals overlapped)

3.6 Smith Predictor

3.6.1 Theoretical Explanation

The Smith Predictor is a control architecture designed to improve the performance of systems with significant time delays. In first-order systems with delays, conventional feedback can lead to slow responses or even instability, since the controller operates with outdated information from the system provoked by the delay. This control architecture uses two fundamental components:

1. A subsystem model that accurately represents the system model;

2. A time delay component;

The controller's unique operation involves direct interaction with the delay-free model component, effectively simulating a delay-free control environment. Simultaneously, the actual output of the delayed process is replaced in the feedback loop by the predicted output of the model, allowing the controller to use real-time estimates of the system state for closing the loop [75, 76].

The Smith Predictor offers three advantages:

- Advanced predictive functionality for anticipating future system states;

- Enhanced transient response characteristics through delay compensation;

- Superior stability maintenance despite substantial dead-time intervals;

The effectiveness is highly dependent on the accuracy of the system model, particularly in estimating the time delay. Model inaccuracies can significantly compromise control performance and, in certain cases, lead to instability. To address these limitations, current implementations integrate adaptive delay estimation techniques and fractional order modelling approaches. These advances improve the robustness while maintaining stability under varying operating conditions [75, 77].

3.6.2 Implementation

With this in mind, the Smith Predictor was implemented in the thermal diffusion system (Figure 3.33). The aim of the implementation was to improve the step response, reducing overshoot and steady state time. The tests were carried out for the following controllers: PI, PID, FOPI e FOPID. The first test focused on comparing the behaviour of each of the classical and fractional controllers, in order to check whether the D component is advantageous or not for the current system. The parameters used for this tests is the same as the previous one, namely $\{K_p, K_i, \lambda, K_d, \mu\} = \{45.867, 0.294, 1, 99.9993, 0.417\}$.

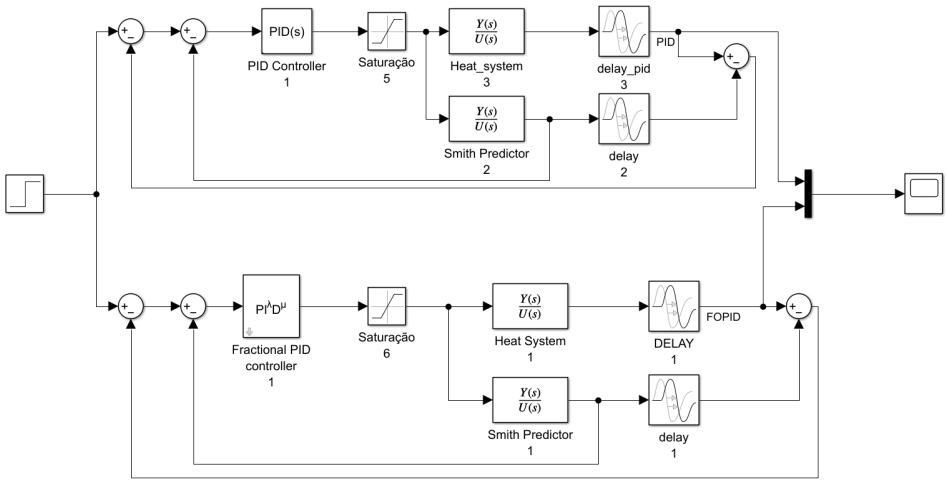
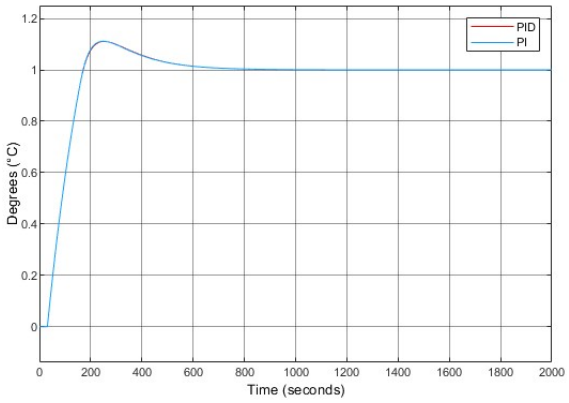


Figure 3.33: Implementation of Smith Predictor

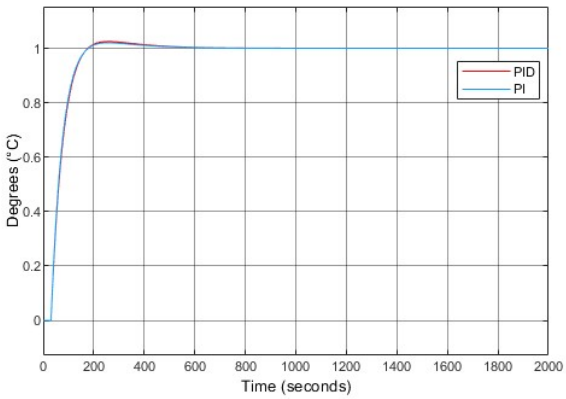
Once the results for the classic controller had been analysed, it was found that the D component did not contribute any advantages when implemented together with the Smith Predictor (Figure 3.34 and Table 3.12). The PI controller has a slightly lower tr, tp and overshoot than the PID controller, but stands out for the large difference in the settling time. Although it is not so noticeable in Figure 3.34, the plots has been zoomed into for clarity, making more clear (Appendix B).

Table 3.12: Comparison of time metrics under different saturation configurations for PI and PID controllers with Smith Predictor

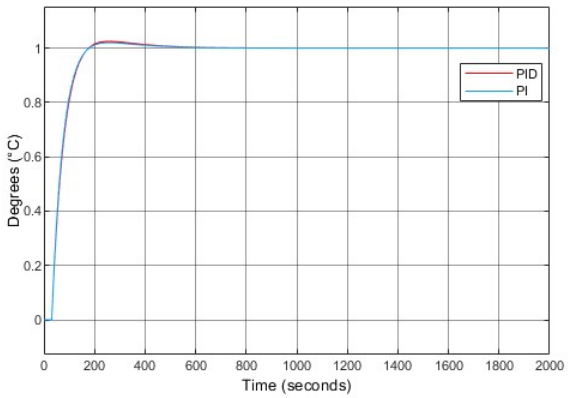
Controller	**Sat. (%)**	**Rise Time (tr)**	**Peak Time (tp)**	**Overshoot (%)**	**Settling Time (ts)**	**Steady-State Error (e_{ss})**
PI	80	84.313	257.556	2	261.937	0
	50	84.313	257.717	2	261.826	0
	45	84.299	255.823	2	263.414	0
	20	109.309	248.666	11.2	550.002	0
PID	80	86.505	257.556	2.5	327.985	0
	50	86.454	258.210	2.5	328.586	0
	45	86.442	258.736	2.5	328.713	0
	20	109.308	251.627	11.1	551.317	0



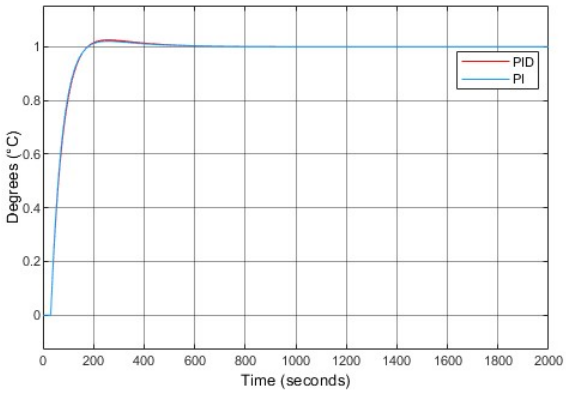
(A) Saturation at 20%



(B) Saturation at 45%



(C) Saturation at 50%



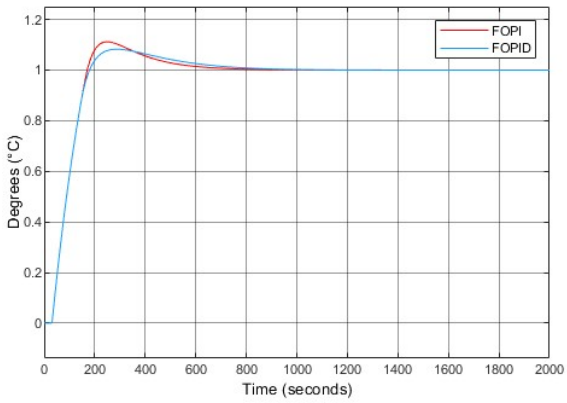
(D) Saturation at 80%

Figure 3.34: Smith Predictor step response to different saturation values on the classical controllers

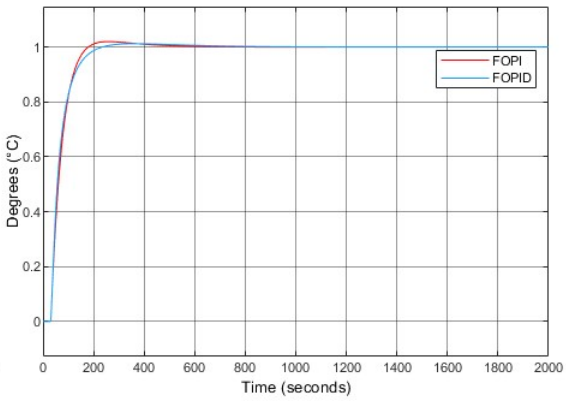
Relatively to the fractional controllers, the implementation of the D component was more noticeable at all the saturation levels selected (Figure 3.34 and Table 3.13). The FOPI controller presents slightly lower tr and considerably lower tp, whereas the FOPID has a lower overshoot and ts, for any saturation level. However at a saturation of 20% the FOPID has higher ts. Therefore, since the goal is to achieve a controller that diminishes the overshoot and achieves the steady state value as quickly as possible, and in order to maintain a fair comparison during the next tests, the controllers selected were PID and FOPID. Thus the following test focused on comparing the FOPID with PID controller.

Table 3.13: Comparison of time metrics under different saturation configurations for FOPI and FOPID controllers with Smith Predictor

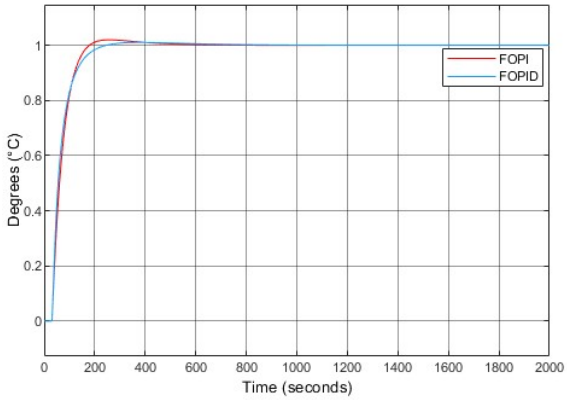
Controller	**Sat. (%)**	**Rise Time (tr)**	**Peak Time (tp)**	**Overshoot (%)**	**Settling Time (ts)**	**Steady-State Error (e_{ss})**
FOPI	80	84.321	258.216	2	261.855	0
	50	84.312	258.187	2	261.985	0
	45	84.300	256.798	2	263.321	0
	20	109.309	250.013	11.2	549.992	0
FOPID	80	91.392	427.223	0.7	204.836	0
	50	89.564	377.906	1.1	192.197	0
	45	89.128	366.252	1.2	187.856	0
	20	109.520	286.938	8.2	646.251	0



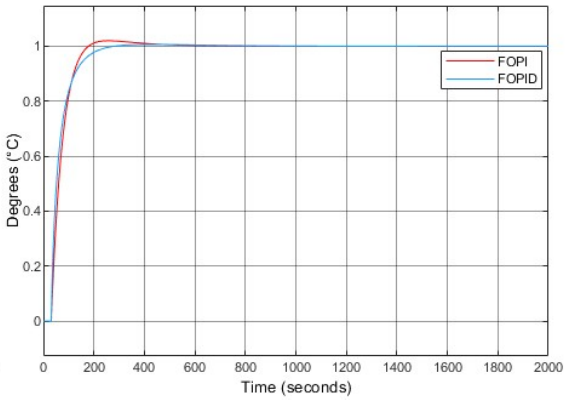
(A) Saturation at 20%



(B) Saturation at 45%



(C) Saturation at 50%



(D) Saturation at 80%

Figure 3.35: Smith Predictor step response to different saturation values on the fractional controller

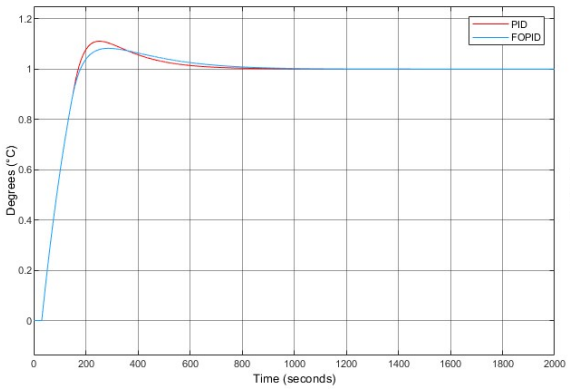
As with the previous system tests, saturation was analysed at the following levels: 20%, 35%, 40%, 60%, 80% and 90% (Figure 3.36 and Table 3.14). The experimental results revealed different performance characteristics between the controllers:

- Through all the saturation values, the FOPID controller has a slight higher rise time than the PID;
- Relatively to the peak time, across all saturation values, the PID has a substantial lower value;
- In terms of overshoot, the FOPID has a remarkable low value comparatively to the PID in all saturation values;
- For saturations above 35%, the FOPID exhibits lower settling time compared to its counterpart;

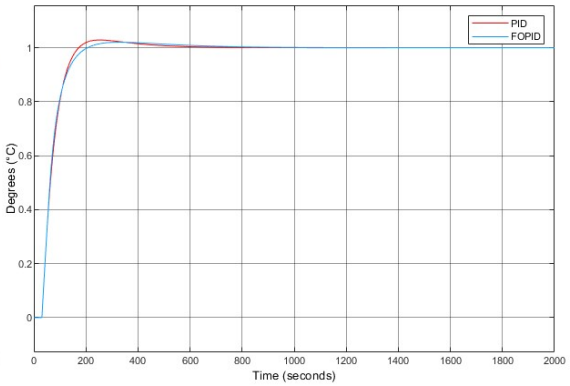
These results demonstrate the superior performance of the FOPID controller in the heat diffusion system compared to its PID counterpart, particularly in terms of overshoot reduction and steady state characteristics, which means better steady state response.

Table 3.14: Comparison of time metrics under different saturation configurations for FOPID and PID with Smith Predictor

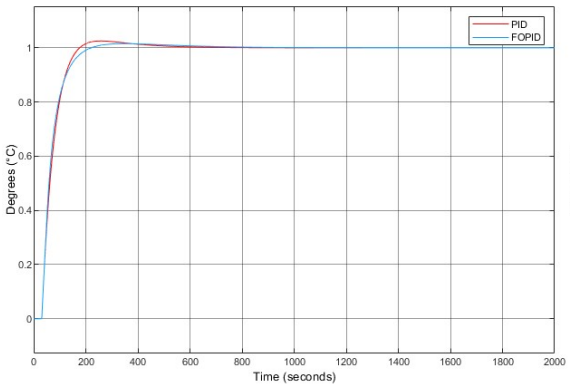
Controller	**Sat. (%)**	**Rise Time (tr)**	**Peak Time (tp)**	**Overshoot (%)**	**Settling Time (ts)**	**Steady-State Error (e_{ss})**
FOPID	90	91.698	432.961	0.6	206.492	0
	80	91.392	427.223	0.7	204.836	0
	60	90.344	403.736	0.8	198.420	0
	40	88.738	347.926	1.5	182.505	0
	35	88.586	328.926	2	357.351	0
	20	109.520	286.938	8.2	646.251	0
PID	90	86.520	258.805	2.5	327.800	0
	80	86.505	257.556	2.5	327.985	0
	60	86.474	256.756	2.5	328.369	0
	40	86.127	258.350	2.5	332.450	0
	35	85.274	252.427	2.9	351.366	0
	20	109.308	251.627	11.1	551.317	0



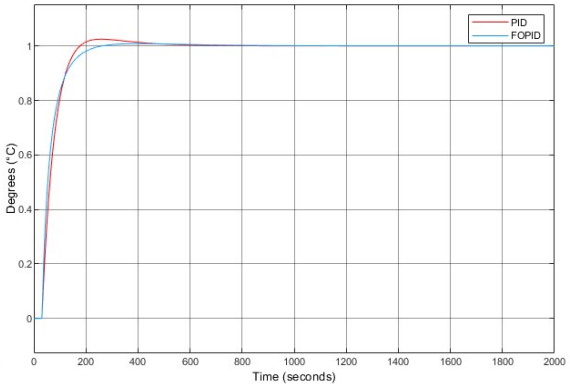
(A) Smith Predictor with saturation at 20%



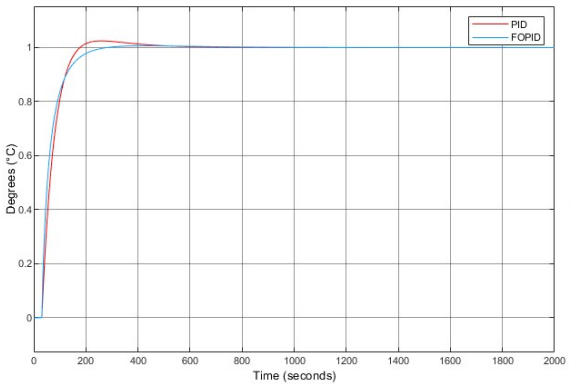
(B) Smith Predictor with saturation at 35%



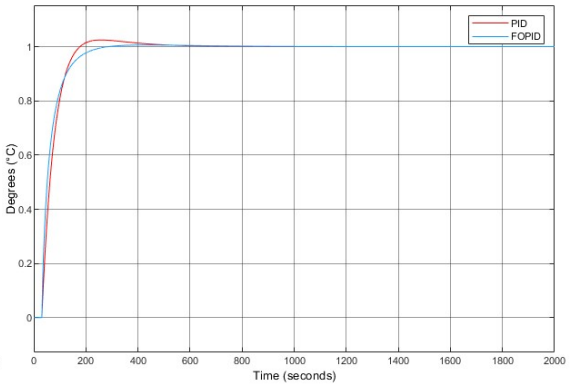
(C) Smith Predictor with saturation at 40%



(D) Smith Predictor with saturation at 60%



(E) Smith Predictor with saturation at 80%



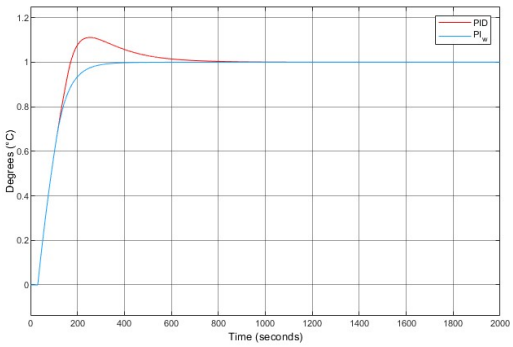
(F) Smith Predictor with saturation at 90%

Figure 3.36: Smith Predictor step response to different saturation values

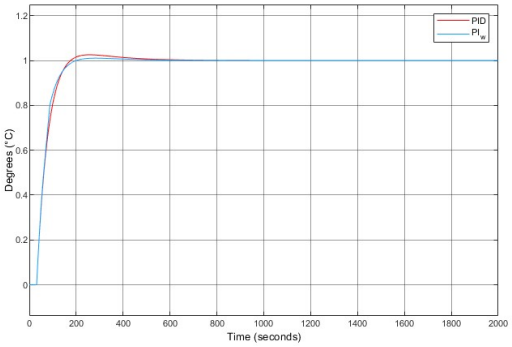
3.7 Anti-windup & Smith Predictor

Based on the methodologies presented in subsections 3.5 and 3.6, an additional study was conducted to evaluate the implementation of both control systems. While the Smith Predictor alone demonstrated excellent step response characteristics for both controllers, its potential performance enhancement through anti-windup integration remains unverified. This study particularly focuses on the FOPID controller, as previous results indicated the anti-windup system's measurable impact on overshoot reduction in this architecture.

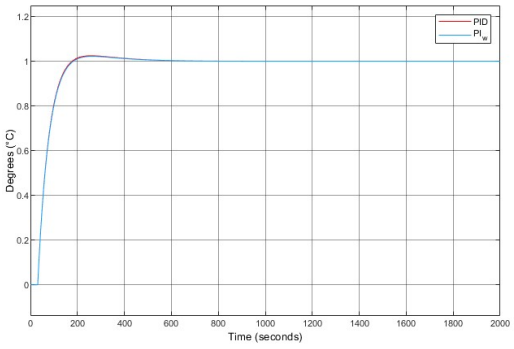
Experimental validation was performed for both controller architectures. The PID controller exhibited predictable behaviour, showing only changes in the step response when combining the Smith Predictor with anti-windup compensation for saturation values below 20% (Figure 3.37).



(A) Saturation at 20%



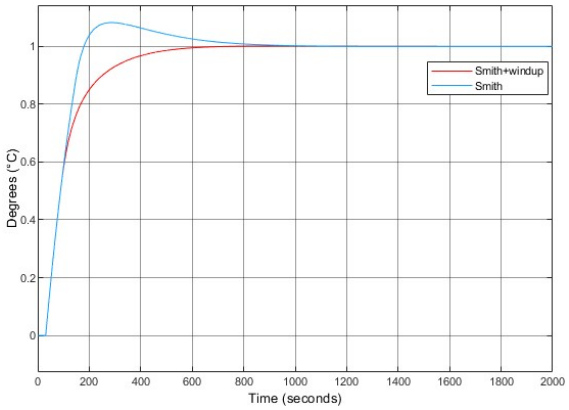
(B) Saturation at 40%



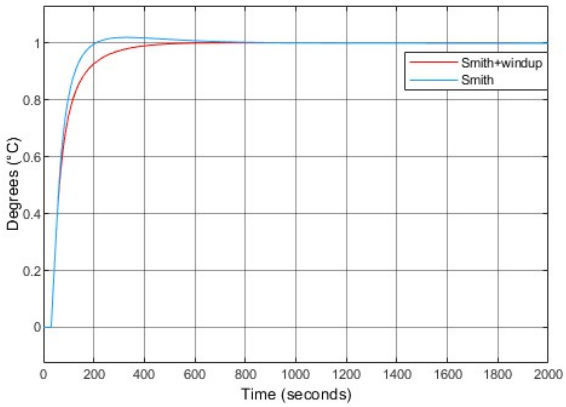
(C) Saturation at 60%

Figure 3.37: Smith Predictor with anti-windup step response to different saturation values

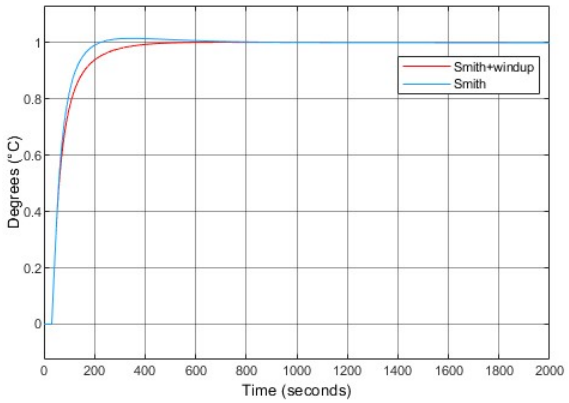
In contrast, the FOPID controller demonstrated consistent performance improvements across all tested saturation values from previous experiments. The hybrid implementation brought two key advantages: faster stabilization compared to stand-alone anti-windup, and just slightly slowed the response time even at lower saturation levels (Figure 3.38).



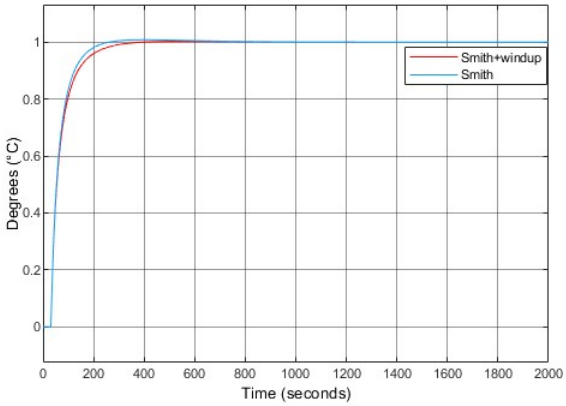
(A) Saturation at 20%



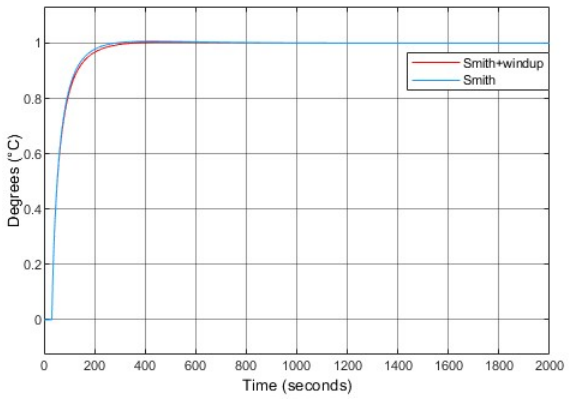
(B) Saturation at 35%



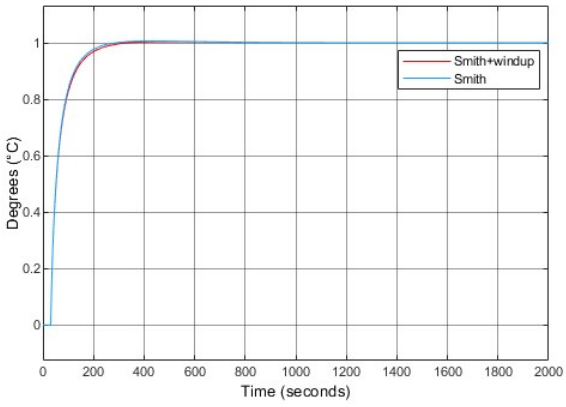
(C) Saturation at 40%



(D) Saturation at 60%



(E) Saturation at 80%



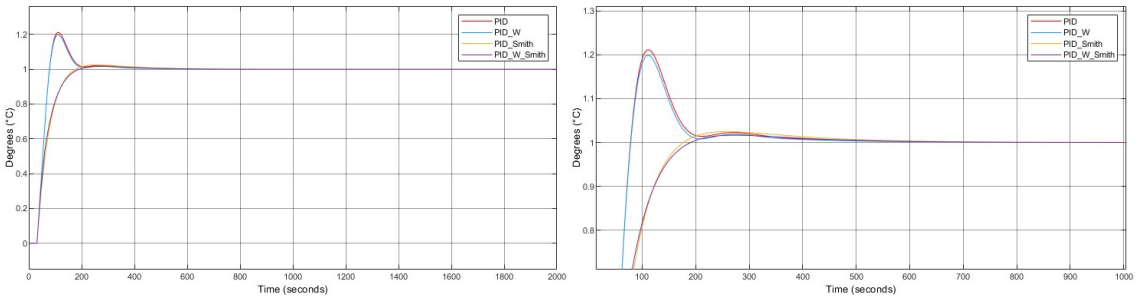
(F) Saturation at 90%

Figure 3.38: Smith Predictor combined with anti-windup step response to different saturation limits

3.8 Result Analysis

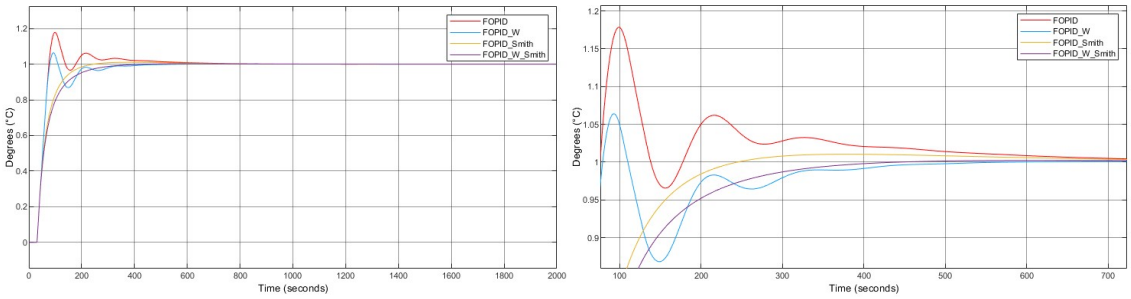
Throughout all the experimental evaluation, enveloping performance metrics, anti-windup system, and Smith Predictor implementation, they allowed the following key findings:

- The ISE metric demonstrated superior performance, with the FOPID controller achieving lower values compared to PID;
- Among regression metrics, MSE showed the better results, where the FOPID exhibited nearly threefold improvement over the PID;
- Saturation emerged as the most effective non-linearity for the presented system, with experimental results confirming its optimal performance;
- The anti-windup system initially displayed almost no utility, Figure 3.39, showing negligible impact on classical controllers. Relatively to the fractional controllers, it contribute with a small overshoot reduction, noticeable in the FOPID architecture, Figure 3.40;
- Smith Predictor implementation brought substantial improvements, not only reducing steady-state response times, but also effectively eliminating overshoot across most saturation levels (Figures 3.39 and 3.40). This implementation was the cornerstone to select which controller to maintain for the following tests;
- The integrated system combining both techniques enhanced step response for both controllers, though with some differences:
  - PID improvements were limited to low saturation values, Figure 3.39;
  - FOPID demonstrated consistent performance gains across all saturation values, Figure 3.40;



(A) Full step response (B) Zoomed-in view of the step response

Figure 3.39: Step response for all PID configurations



(A) Full step response (B) Zoomed-in view of the step response

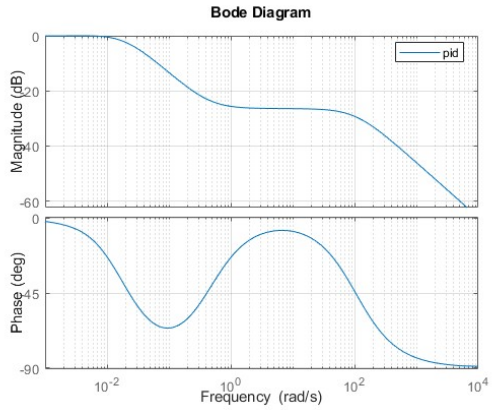
Figure 3.40: Step response for all FOPID configurations

These results demonstrate that for the heat diffusion system, the implementation of an FOPID controller with Smith Predictor and anti-windup system delivers superior performance compared to conventional PID control, improving the transient response of the system.

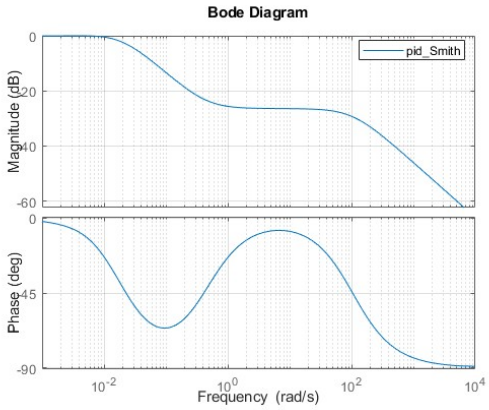
Moreover, there was also made frequency-domain analysis through Bode Diagrams and Geometric Root Locus providing further insight into system behaviour. The Bode Diagrams revealed that all configurations exhibited typical low-pass characteristics. Relatively to the Geometric Root Locus, it further prove the system's stability in both controllers, since all the poles stayed at the left side of the Imaginary axis. The introduction of the Smith Predictor brought no changes on both controllers in the frequency-domain. On the other hand, the addition of the Anti-windup system introduced modifications into both controllers.

In the PID controller, the anti-windup altered the Bode Diagram plot decreasing even further both the Magnitude and Phase (Figure 3.41). It also altered the location of the poles and zeros of the system, namely making them more negative, which leads to greater stability of the system.

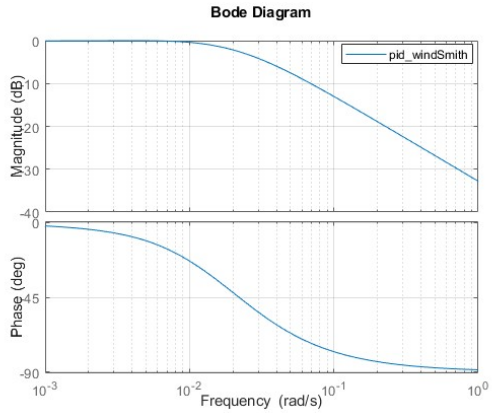
These transfer functions were obtained using the toolbox based on CRONE methodology, mentioned in the subsection 2.3.1.2. Consequently, it also affected the transfer function and the Root Locus of the system (Equations 3.11 and 3.12 and Figure 3.42).



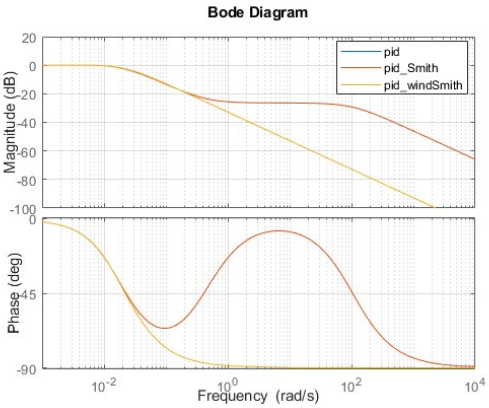
(A) PID controller



(B) PID controller with Smith Predictor



(C) PID controller with Smith Predictor and Anti-windup



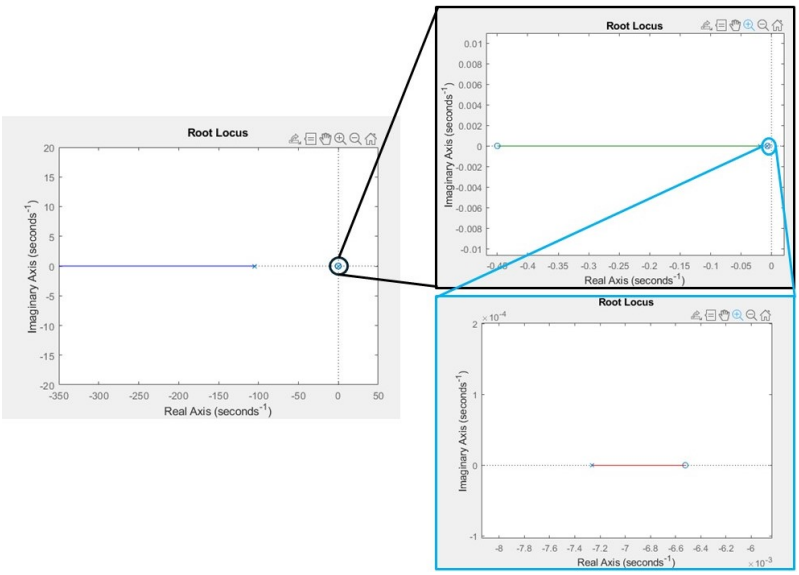
(D) Comparison between all Bode diagrams ((A) and (B) are overlapped)

Figure 3.41: Bode Diagrams for different PID controllers implementations

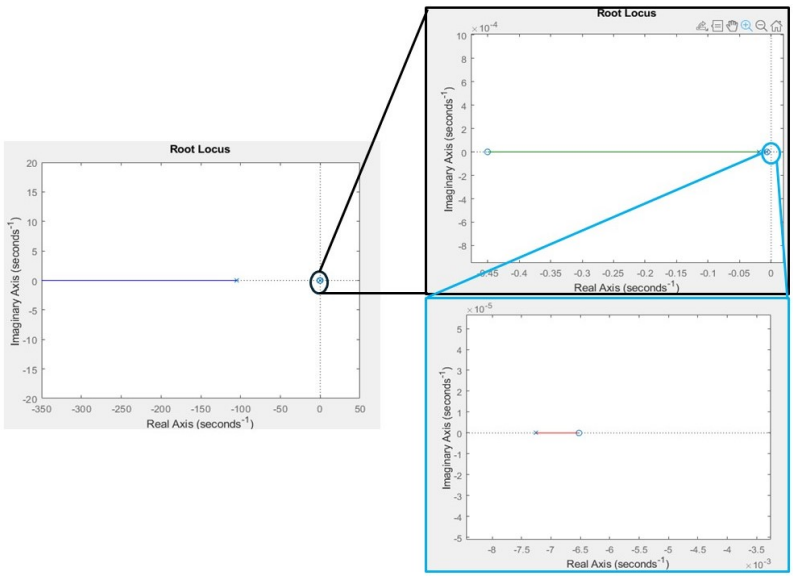
$$TF_{pid}(s) = \frac{5.023s^2 + 2.294s + 0.01475}{s^3 + 105s^2 + 2.794s + 0.01475} \tag{3.11}$$

$$TF_{pid\_windSmith}(s) = \frac{0.02293s + 0.0001475}{s^2 + 0.02793s + 0.0001475} \tag{3.12}$$

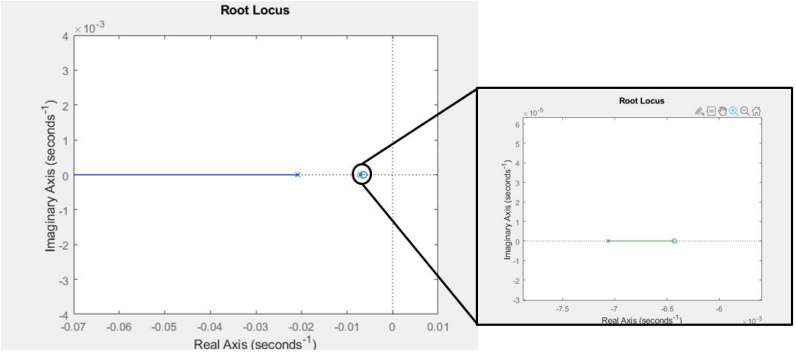
Analysing these transfer functions, its conclusive that the classical controller with Smith Predictor and anti-windup reduces the number of poles and zeros leading to a simpler transfer function.



(A) PID controller



(B) PID controller with Smith Predictor



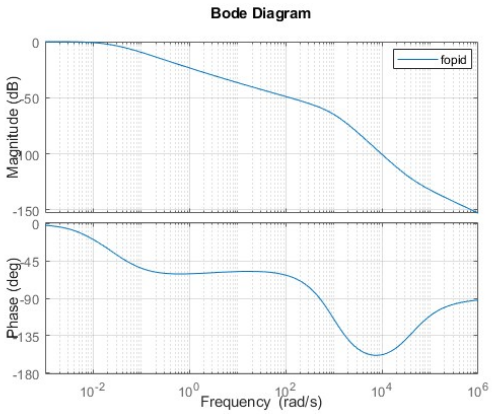
(C) PID controller with Smith Predictor and Anti-windup

Figure 3.42: Root Locus for different PID controllers implementations

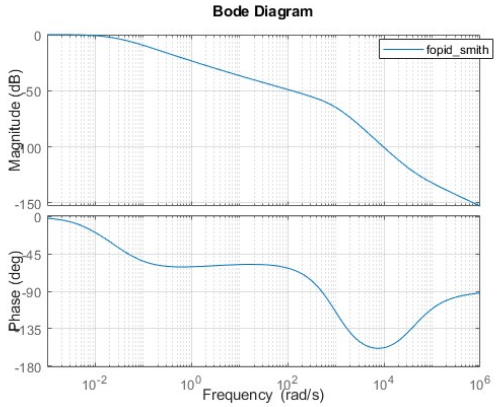
Relatively to the FOPID controller, the Anti-windup system did not alter the Bode Diagram (Figure 3.43). However, the number of poles and zeros and their location suffered a huge difference, changing the Root Locus (Figures 3.44–3.46) and the transfer function (Equations 3.13 and 3.14). Even after this increase the number of poles and zeros, the system remained stable with all poles on the left-half plane. In the next figures, for the analysis of real axis, it is need a special pay attention to the thickness scale (10^{-3} or 10^{-4}).

$$TF_{fopid}(s) = \frac{\begin{aligned}&2.293 \times 10^{-2} s^{14} + 9.595 \times 10^{2} s^{13} + 1.475 \times 10^{6} s^{12} + 6.091 \times 10^{8} s^{11} \\ &+ 7.272 \times 10^{10} s^{10} + 2.527 \times 10^{12} s^{9} + 2.576 \times 10^{13} s^{8} + 7.785 \times 10^{13} s^{7} \\ &+ 7.072 \times 10^{13} s^{6} + 1.958 \times 10^{13} s^{5} + 1.682 \times 10^{12} s^{4} + 4.686 \times 10^{10} s^{3} \\ &+ 4.754 \times 10^{8} s^{2} + 1.968 \times 10^{6} s + 2.629 \times 10^{3}\end{aligned}}{\begin{aligned}&s^{15} + 2.970 \times 10^{3} s^{14} + 3.149 \times 10^{6} s^{13} + 1.400 \times 10^{9} s^{12} \\ &+ 2.332 \times 10^{11} s^{11} + 1.246 \times 10^{13} s^{10} + 1.978 \times 10^{14} s^{9} + 9.132 \times 10^{14} s^{8} \\ &+ 1.232 \times 10^{15} s^{7} + 5.010 \times 10^{14} s^{6} + 6.615 \times 10^{13} s^{5} + 3.207 \times 10^{12} s^{4} \\ &+ 6.362 \times 10^{10} s^{3} + 5.445 \times 10^{8} s^{2} + 2.058 \times 10^{6} s + 2.629 \times 10^{3}\end{aligned}} \tag{3.13}$$

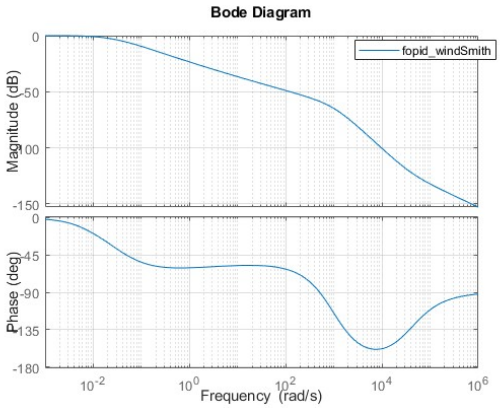
$$TF_{fopid\_windsmith}(s) = \frac{\begin{array}{l} 2.293 \times 10^{-2} s^{24} + 9.765 \times 10^{2} s^{23} + 2.176 \times 10^{6} s^{22} \\ + 1.767 \times 10^{9} s^{21} + 6.515 \times 10^{11} s^{20} + 1.176 \times 10^{14} s^{19} \\ + 1.073 \times 10^{16} s^{18} + 5.023 \times 10^{17} s^{17} + 1.217 \times 10^{19} s^{16} \\ + 1.534 \times 10^{20} s^{15} + 1.010 \times 10^{21} s^{14} + 3.479 \times 10^{21} s^{13} \\ + 6.289 \times 10^{21} s^{12} + 5.980 \times 10^{21} s^{11} + 2.998 \times 10^{21} s^{10} \\ + 7.939 \times 10^{20} s^{9} + 1.114 \times 10^{20} s^{8} + 8.317 \times 10^{18} s^{7} \\ + 3.329 \times 10^{17} s^{6} + 7.240 \times 10^{15} s^{5} + 8.698 \times 10^{13} s^{4} \\ + 5.834 \times 10^{11} s^{3} + 2.137 \times 10^{9} s^{2} + 3.881 \times 10^{6} s + 2.629 \times 10^{3} \end{array}}{\begin{array}{l} s^{25} + 3.713 \times 10^{3} s^{24} + 5.427 \times 10^{6} s^{23} \\ + 3.973 \times 10^{9} s^{22} + 1.549 \times 10^{12} s^{21} + 3.236 \times 10^{14} s^{20} \\ + 3.592 \times 10^{16} s^{19} + 2.098 \times 10^{18} s^{18} + 6.409 \times 10^{19} s^{17} \\ + 1.020 \times 10^{21} s^{16} + 8.455 \times 10^{21} s^{15} + 3.647 \times 10^{22} s^{14} \\ + 8.200 \times 10^{22} s^{13} + 9.645 \times 10^{22} s^{12} + 5.968 \times 10^{22} s^{11} \\ + 1.960 \times 10^{22} s^{10} + 3.466 \times 10^{21} s^{9} + 3.360 \times 10^{20} s^{8} \\ + 1.825 \times 10^{19} s^{7} + 5.666 \times 10^{17} s^{6} + 1.020 \times 10^{16} s^{5} \\ + 1.074 \times 10^{14} s^{4} + 6.582 \times 10^{11} s^{3} + 2.271 \times 10^{9} s^{2} \\ + 3.970 \times 10^{6} s + 2.629 \times 10^{3} \end{array}} \tag{3.14}$$



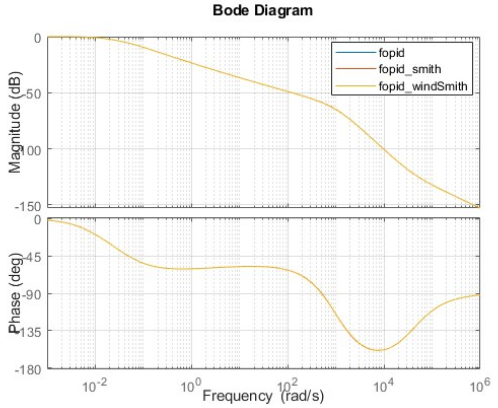
(A) FOPID controller



(B) FOPID controller with Smith Predictor



(C) FOPID controller with Smith Predictor and Anti-windup



(D) Comparison between all Bode diagrams (all signals are overlapped)

Figure 3.43: Bode Diagrams for different FOPID controllers implementations

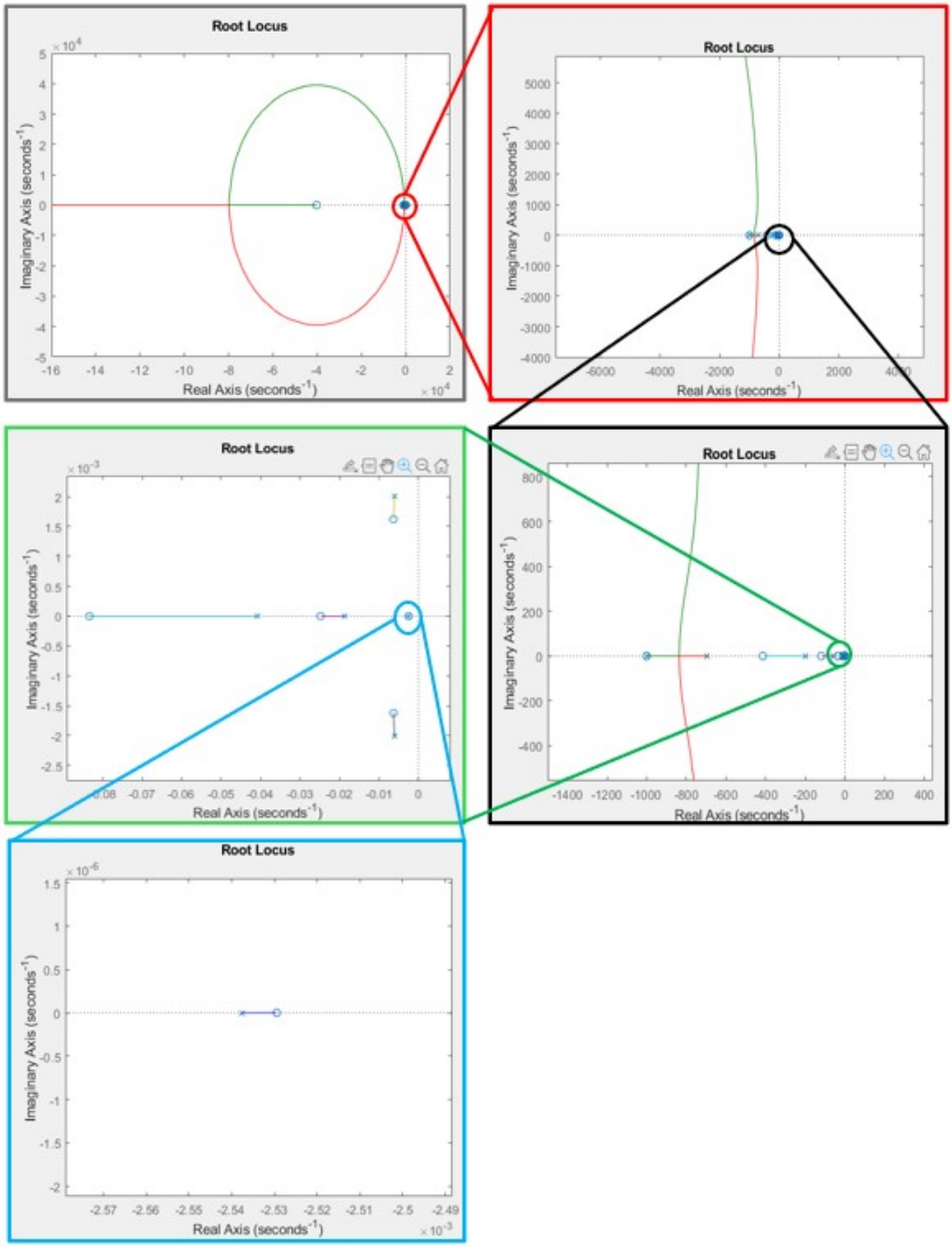


Figure 3.44: Root Locus for FOPID controller

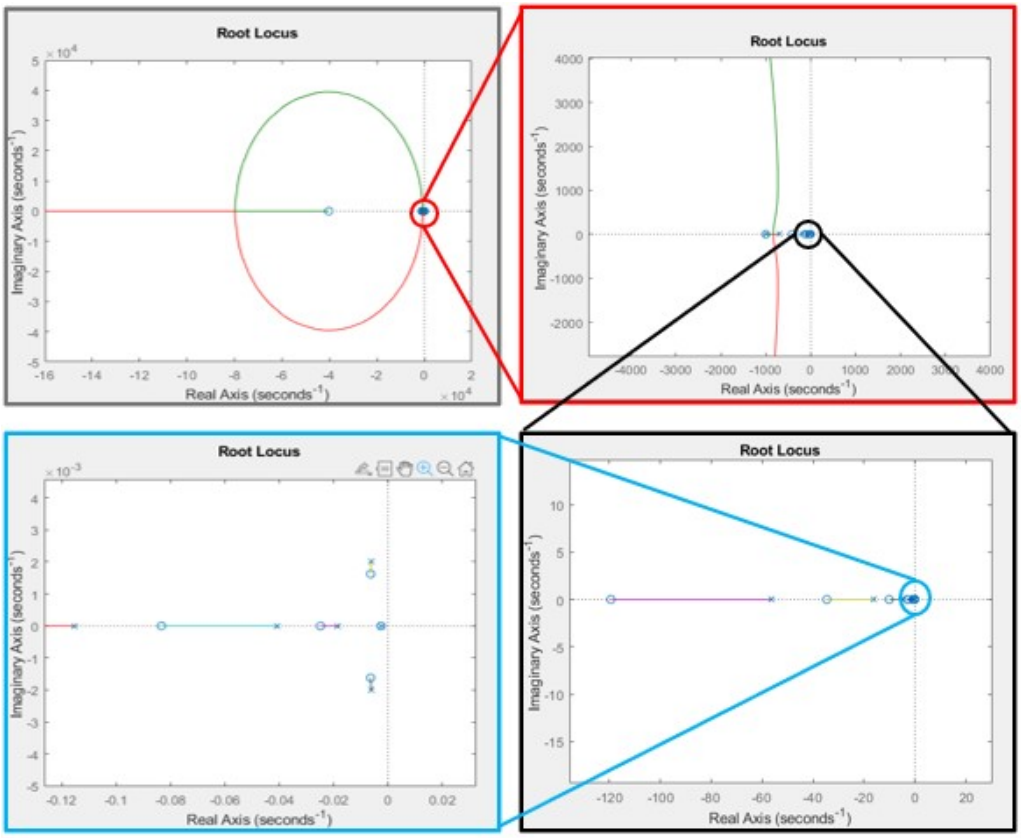


Figure 3.45: Root Locus for FOPID controller with Smith Predictor

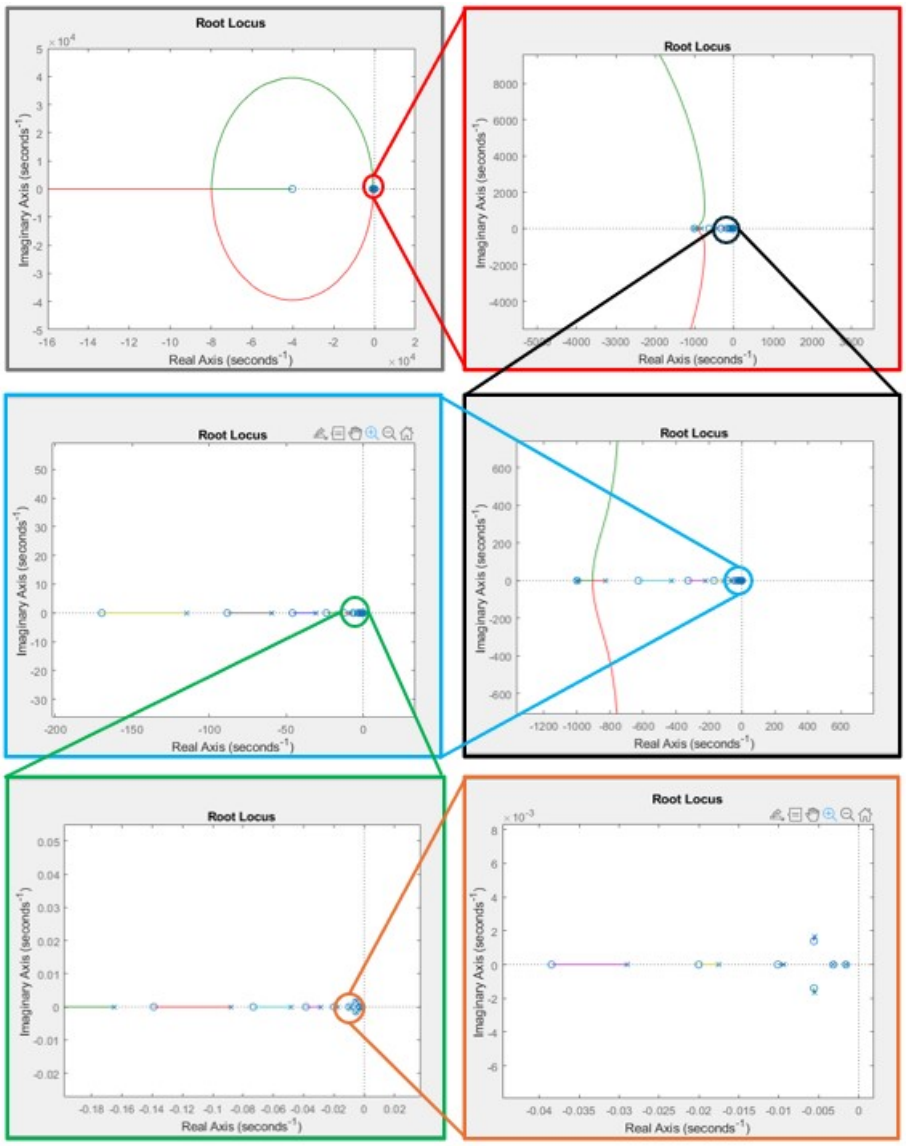


Figure 3.46: Root Locus for FOPID controller with Smith Predictor and Anti-windup

Chapter 4

Conclusion and Future Work

4.1 Conclusion

The aim of this dissertation was to study and develop a fractional order controller for a heat diffusion system with time delays. To this end, simulations were carried out to evaluate its behaviour, using the classical PID controller as a term of comparison. After the tests were completed, it showed that the classical PID controller had difficulty managing the thermal system effectively.

The experiments involved analysing the behaviour of both controllers under identical conditions. The results showed that the fractional controller performed better in terms of stability, response time and system robustness. The great adaptability of the fractional controller results from its additional tuning parameters in the integral and derivative components.

In addition, saturation non-linearity was identified as the most suitable approach for systems with significant delays. Complementary techniques such as the Smith Predictor and anti-windup were integrated, and their combined use substantially improved the system response, particularly for the fractional controller. Further analysis in the frequency-domain, particularly the Bode Diagrams, confirmed enhanced robustness and stability by the FOPID controller. Additionally, the Root Locus highlighted the system stability even more, since all the poles were at the left-half plane.

Overall, all the objectives were successfully achieved. This document affirm that fractional order controllers represent a viable solution for systems with time delay, in particular for the system used for this study, the heat diffusion system.

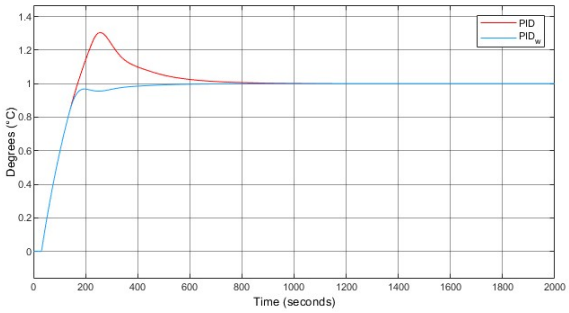
4.2 Future Work

The following list provides some ideas for future works:

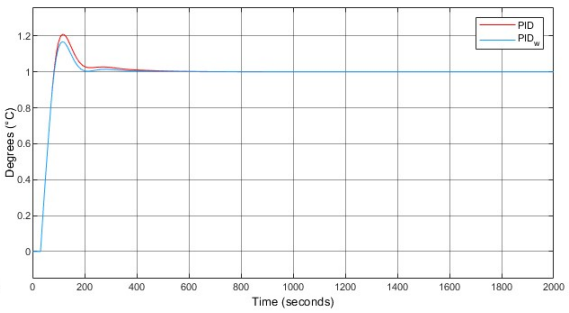
- Make the fractional controller adaptive according to the input;
- Addition of a new controller/algorithm, such as hybrid Fuzzy Controller/PSO;
- Tuning all the parameters of each controller, for each saturation value and algorithm;
- Adaption of the simulation into a real-life scenario;
- Addition of other techniques to further increase the response performance;

Appendix A

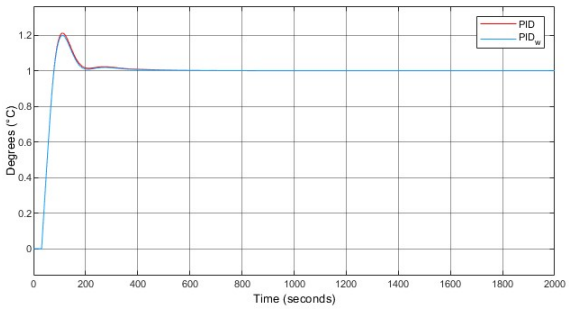
PID controller step response with Anti-windup system



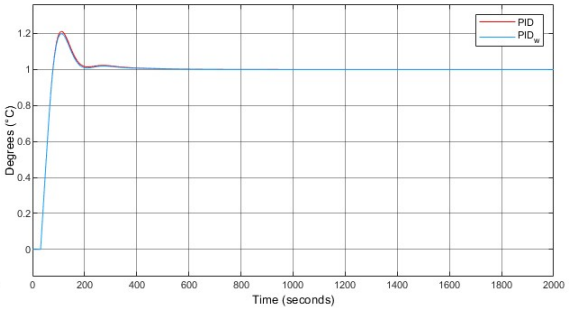
(A) Anti-windup with saturation at 20%



(B) Anti-windup with saturation at 45%



(C) Anti-windup with saturation at 50%



(D) Anti-windup with saturation at 80% (signals overlapped)

Figure A.1: Anti-windup step response to different saturation values on PID controller

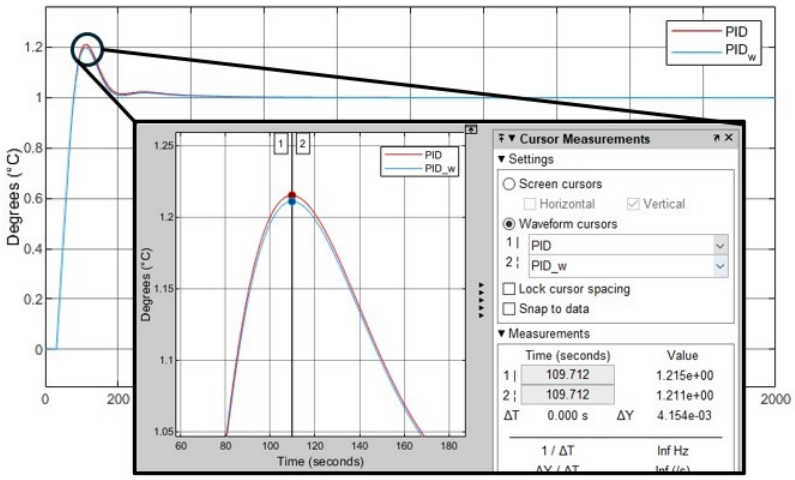


Figure A.2: Aproximation on the overshoot of the Anti-windup step response with 80% saturation for PID

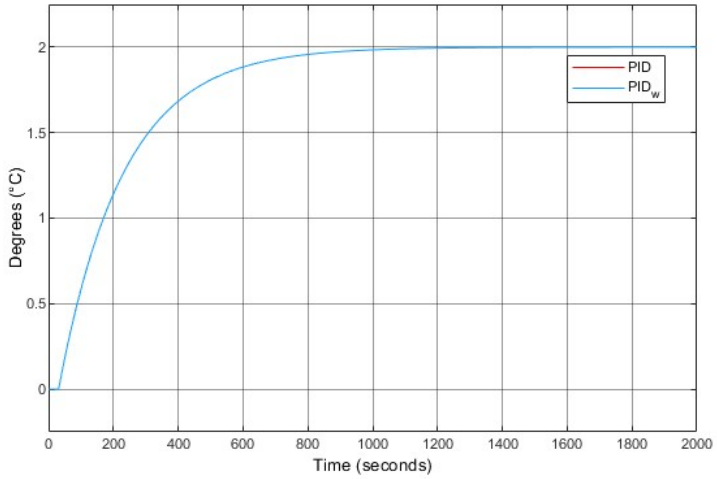


Figure A.3: Step response of PID, with and without Anti-windup, with 20% saturation for a input of amplitude equal to 2 (signals overlapped)

Appendix B

PI and PID controller with Smith Predictor settling time difference proof

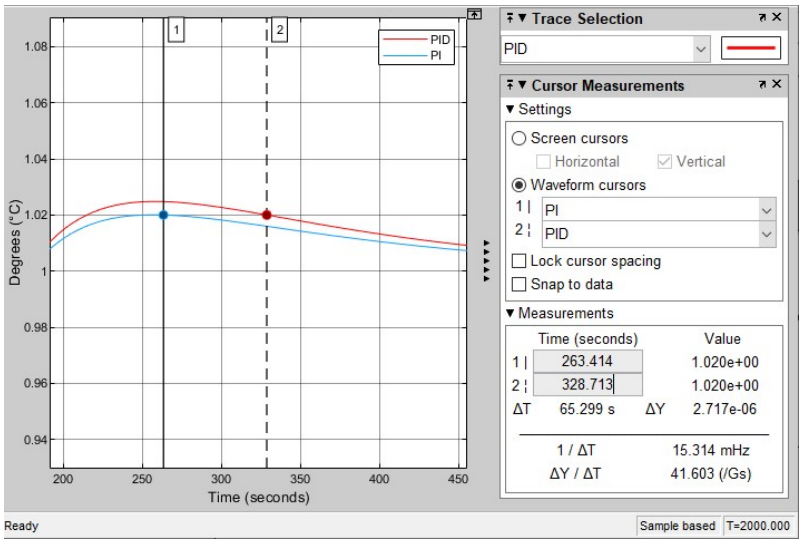


Figure B.1: Settling time difference with saturation at 45%

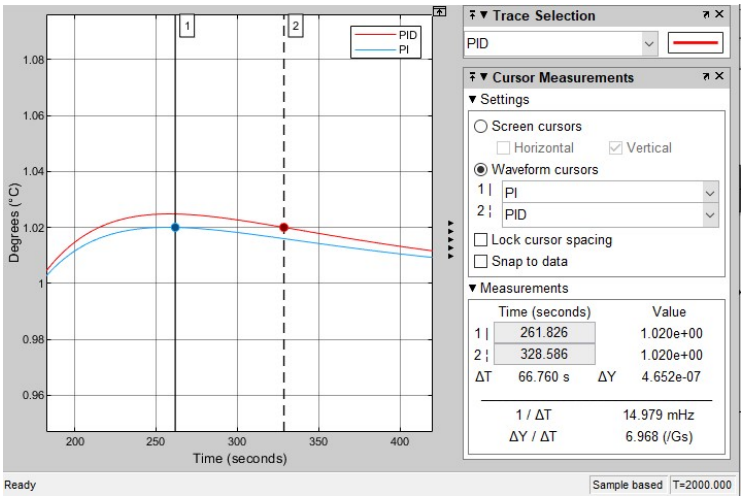


Figure B.2: Settling time difference with saturation at 50%

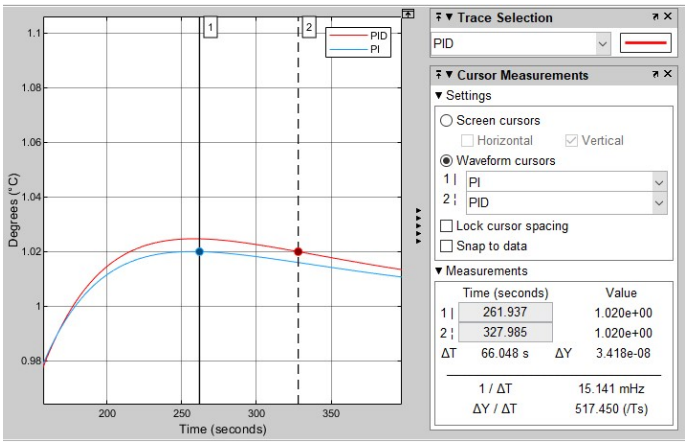


Figure B.3: Settling time difference with saturation at 80%

Bibliography

[1] David G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications*. New York: John Wiley & Sons, 1979.

[2] Katsuhiko. Ogata. *Modern control engineering*. Prentice Hall, 2022. ISBN: 0136156738.

[3] Keith B. Oldham and Jerome Spanier. *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*. Academic Press, 1974.

[4] C. V. Hollot and D. S. Bernstein. "Stability problems in the control of time-delay systems". In: *IEEE Control Systems Magazine* 16.6 (1996), pp. 65–71.

[5] Igor Podlubny. "Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers". In: *IEEE Transactions on Automatic Control* 44.1 (1994), pp. 208–214.

[6] Rafael F. Vázquez and Vicente M. Feliu. "Control of a heat process with a long dead-time using a Smith predictor". In: *ISA Transactions* 50.3 (2011), pp. 477–485.

[7] M. T. Tavares, C. M. Figueiredo, and J. A. T. Machado. "Comparison of classical and fractional controllers performance in thermal systems". In: *Fractional Calculus and Applied Analysis* 17.2 (2014), pp. 449–463.

[8] Ivo Petráš. "Tuning and implementation methods for fractional-order controllers". In: *Fractional Calculus and Applied Analysis* 15.2 (2012), pp. 282–303. DOI: 10.2478/s13540-012-0021-4. URL: https://doi.org/10.2478/s13540-012-0021-4.

[9] Pritesh Shah and Sudhir Agashe. "Review of Fractional PID Controller". In: *Mechatronics* 38 (2016), pp. 29–41. DOI: 10.1016/j.mechatronics.2016.06.005. URL: https://www.sciencedirect.com/science/article/pii/S095741581630068X.

[10] Ying Luo and YangQuan Chen. *Fractional Order Motion Controls*. John Wiley & Sons, 2013. ISBN: 978-1-119-94455-3. URL: https://www.wiley.com/en-us/Fractional+Order+Motion+Controls-p-9781118387719.

[11] Domenico Cafagna. "Fractional Calculus: A Mathematical Tool from the Past for Present Engineers [Past and Present]". In: *IEEE Industrial Electronics Magazine* 1.2 (2007), pp. 35–40. DOI: 10.1109/MIE.2007.901479. URL: https://doi.org/10.1109/MIE.2007.901479.

[12] Shantanu Das (auth.) *Functional Fractional Calculus*. 2nd ed. Springer, 2011. ISBN: 9783642205446.

[13] Stefan G. Samko, Anatoly A. Kilbas, and Oleg I. Marichev. *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach Science Publishers, 1993.

[14] Rafael Martínez-Guerra, Fidel Meléndez-Vázquez, and Iván Trejo-Zúñiga. *Fault-tolerant Control and Diagnosis for Integer and Fractional-order Systems: Fundamentals of Fractional Calculus and Differential Algebra with Real-Time Applications*. Vol. 328. Studies in Systems, Decision and Control. Springer, 2021. ISBN: 978-3-030-62094-3. DOI: 10.1007/978-3-030-62094-3. URL: https://doi.org/10.1007/978-3-030-62094-3.

[15] Rudolf Gorenflo et al. *Mittag-Leffler Functions, Related Topics and Applications*. Springer, 2020.

[16] Igor Podlubny. *Fractional Differential Equations*. Academic Press, 1999.

[17] Pierre Melchior, Alexandre Poty, and Alain Oustaloup. "Frequency Band-Limited Fractional Differentiator Prefilter in Path Tracking Design". In: *Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering*. Ed. by J. Sabatier, O.P. Agrawal, and J.A. Tenreiro Machado. Springer, 2007, pp. 477–493.

[18] Harry Bateman and Arthur Erdélyi. *Higher Transcendental Functions*. McGraw-Hill, 1953.

[19] John W. Hanneken, David M. Vaught, and B. N. Narahari Achar. "Enumeration of the Real Zeros of the Mittag-Leffler Function $E_\alpha(z)$, $1 < \alpha < 2$". In: *Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering*. Ed. by J. Sabatier, O.P. Agrawal, and J.A. Tenreiro Machado. Dordrecht: Springer, 2007, pp. 15–26. ISBN: 978-1-4020-6042-7.

[20] S. Isabel Jesus, J. A. Tenreiro Machado, and S. Barbosa Ramiro. "On the Fractional Order Control of Heat Systems". In: *Intelligent Engineering Systems and Computational Cybernetics*. Ed. by J. A. Tenreiro Machado, B. Pátkai, and I. J. Rudas. Springer, 2009, pp. 375–385. DOI: 10.1007/978-1-4020-8678-6_32. URL: https://doi.org/10.1007/978-1-4020-8678-6_32.

[21] Ricardo Enrique Gutiérrez, João Maurício Rosário, and José Tenreiro Machado. "Fractional Order Calculus: Basic Concepts and Engineering Applications". In: *Mathematical Problems in Engineering* 2010 (2010), 375858, 19 pages. DOI: 10.1155/2010/375858.

[22] Donato Cafagna. "Fractional calculus: A mathematical tool from the past for present engineers [Past and present]". In: *IEEE Industrial Electronics Magazine* 1.2 (2007), pp. 35–40. DOI: 10.1109/MIE.2007.901479.

[23] Francesco Mainardi. "Memory Effects in Linear Viscoelastic Models". In: *Fractional Calculus and Waves in Linear Viscoelasticity*. World Scientific, 2010. Chap. 4, pp. 87–112.

[24] Anatoly A. Kilbas, Hari M. Srivastava, and Juan J. Trujillo. *Theory and Applications of Fractional Differential Equations*. Elsevier, 2006, 97–101 (GL linearity via finite differences).

[25] Kai Diethelm. *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*. Vol. 2004. Lecture Notes in Mathematics. Focuses on computational aspects of nonlocality in Caputo-type operators. Includes MATLAB implementations for memory-dependent systems. Berlin, Heidelberg: Springer, 2010. ISBN: 978-3-642-14573-5. DOI: 10.1007/978-3-642-14574-2.

[26] Dumitru Baleanu et al. *Fractional Calculus: Models and Numerical Methods*. Jan. 2012. ISBN: 978-981-4355-20-9. DOI: 10.1142/10044.

[27] C. A. et al. Monje. "Fractional PID Controllers". In: *Fractional Calculus Applications in Control*. Springer, 2010, pp. 101–120.

[28] S. Manabe. "Fractional-order control for spacecraft attitude stabilization". In: *Journal of Guidance, Control, and Dynamics* 4.2 (1961), pp. 121–128.

[29] A. Oustaloup et al. "Frequency-domain robustness analysis of CRONE controlOustaloup". In: *European Journal of Control* 1.2 (1995), pp. 102–115.

[30] B. Mathieu and A. Oustaloup. "Fractional differentiation in mechatronic systems". In: *Mechatronics* 8.5 (1998), pp. 473–488.

[31] Patrick Lanusse, Rachid Malti, and Pierre Melchior. "CRONE control system design toolbox for the control engineering community: Tutorial and case study". In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 371 (May 2013), p. 20120149. DOI: 10.1098/rsta.2012.0149.

[32] Duarte Valério and José Sá da Costa. "Tuning of fractional PID controllers with Ziegler-Nichols-type rules". In: *Signal Processing* 86.10 (2005), pp. 2771–2784.

[33] Denis Matignon. "Stability results for fractional differential equations with applications to control processing". In: *Computational Engineering in Systems Applications* 2 (1998), pp. 963–968.

[34] Hong Li et al. "Stability Analysis of a Fractional-Order Linear System Described by the Caputo–Fabrizio Derivative". In: *Mathematics* 7.2 (2019). ISSN: 2227-7390. DOI: 10.3390/math7020200. URL: https://www.mdpi.com/2227-7390/7/2/200.

[35] Alain Oustaloup et al. “Frequency-band complex noninteger differentiator: Characterization and synthesis”. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47.1 (2000), pp. 25–39.

[36] Yan Li, YangQuan Chen, and Igor Podlubny. “Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized Mittag-Leffler stability”. In: *Computers & Mathematics with Applications* 59 (2009), pp. 1810–1821. DOI: 10.1016/j.camwa.2009.08.019.

[37] Weihua Deng, Cheng Li, and Jinhu Lü. “Stability analysis of linear fractional differential systems with multiple time delays”. In: *Nonlinear Dynamics* 48 (2007), pp. 409–416. DOI: 10.1007/s11071-006-9094-0.

[38] William J. Palm III. *System Dynamics*. 4th. Fourth edition of the book on system dynamics, covering modeling, analysis, and control of dynamic systems. New York, NY: McGraw-Hill Education, 2020. ISBN: 978-0-07-814005-1. URL: https://www.mheducation.com/.

[39] Graham C. Goodwin, Stefan F. Graebe, and Mario E. Salgado. *Control System Design*. 1st. Upper Saddle River, NJ: Prentice Hall, 2001. ISBN: 978-0139586538.

[40] Hassan A. Yousef. *Power System Load Frequency Control: Classical and Adaptive Fuzzy Approaches*. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2017. ISBN: 978-1-4987-4557-4.

[41] GeeksforGeeks. *Closed-Loop Control System - Operations, Components and Applications*. https://www.geeksforgeeks.org/closed-loop-control-system/. Accessed: 2025-03-22.

[42] Karl J. Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. A foundational text on the theory, design, and tuning of PID controllers. Research Triangle Park, NC: ISA - The Instrumentation, Systems, and Automation Society, 1995. ISBN: 978-1-55617-516-9.

[43] William J. Palm III. *System Dynamics*. 4th. New York, NY: McGraw-Hill Education, 2020. ISBN: 978-0-07-814005-1.

[44] Liuping Wang. *PID Control System Design and Automatic Tuning using MATLAB/Simulink*. 1st. John Wiley & Sons, 2019. ISBN: 978-1119469445. DOI: 10.1002/9781119469414. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119469414.

[45] ElectricalEngineering.XYZ. *Top 10 Open Loop Control System Examples*. https://www.electricalengineering.xyz/top-10-open-loop-control-system-examples/. Accessed: 2025-03-22.

[46] ElProCus. *Closed Loop Control System: Block Diagram, Types & Its Applications*. https://www.elprocus.com/what-is-a-closed-loop-control-system-its-working/. Accessed: 2025-03-22.

[47] J.G. Ziegler and N.B. Nichols. "Optimum Settings for Automatic Controllers". In: (1942).

[48] Electronics Coach. *Open Loop Control System*. https://electronicscoach.com/open-loop-control-system.html. Accessed: 23 March 2025.

[49] G. Prabhakar, Nedumal Pugazhenthi, and Selvaperumal S. "Implementation Analysis of State Space Modelling and Control of Nonlinear Process Using PID Algorithm in MATLAB and PROTEUS Environment". In: *Applied Mechanics and Materials* 592-594 (2014), pp. 2081–2085. DOI: 10.4028/www.scientific.net/AMM.592-594.2081. URL: https://www.researchgate.net/publication/272063126_Implementation_analysis_of_state_space_modelling_and_control_of_nonlinear_process_using_PID_algorithm_in_MATLAB_and_PROTEUS_environment.

[50] P.J. Woolf. *Chemical Process Dynamics and Controls*. Open textbook library. University of Michigan Engineering Controls Group, 2009. URL: https://books.google.pt/books?id=Op87vwEACAAJ.

[51] Toronto Metropolitan University. *9.2 Proportional Control*. https://pressbooks.library.torontomu.ca/controlsystems/chapter/9-2proportional-control/. Accessed: 2025-03-23. 2025.

[52] Yunus A. Çengel. *Heat Transfer: A Practical Approach*. 2nd. McGraw-Hill, 2003. ISBN: 978-0072458930.

[53] John H. Lienhard IV and John H. Lienhard V. *A Heat Transfer Textbook*. 3rd. Phlogiston Press, 2008. URL: http://web.mit.edu/lienhard/www/ahtt.html.

[54] Frank P. Incropera and David P. DeWitt. *Fundamentals of Heat and Mass Transfer*. 4th ed. New York, NY: John Wiley & Sons, 1996. ISBN: 9780471304609.

[55] Yoav Peles. *Chapter 1: Introduction and Basic Concepts*. https://slideplayer.com/slide/12728534/. Accessed: 2025-03-15.

[56] Michael J. Moran and Howard N. Shapiro. *Fundamentals of Engineering Thermodynamics*. 5th. John Wiley & Sons, 2006. ISBN: 978-0470030370.

[57] SimScale. *What Is Heat Transfer (Heat Flow)? Complete Guide*. https://www.simscale.com/docs/simwiki/heat-transfer-thermal-analysis/what-is-heat-transfer/. Accessed: 2025-03-15. 2024.

[58] Jeff Smoot. *DC fans for forced-air convection cooling*. https://www.ednasia.com/dc-fans-for-forced-air-convection-cooling/. Accessed: 2025-03-15. 2021.

[59] T. N. Narasimhan. "Fourier's heat conduction equation: History, influence, and connections". In: *Reviews of Geophysics* 37 (1 Feb. 1999), pp. 151–172. ISSN: 87551209. DOI: 10.1029/1998RG900006.

[60] Claire Yu Yan. *Introduction to Engineering Thermodynamics*. University of British Columbia, 2020. URL: https://open.library.ubc.ca/soa/cIRcle/collections/facultyresearchandpublications/52383/items/1.0420218.

[61] N. A. Gokcen and R. G. Reddy. "The First Law of Thermodynamics". In: *Thermodynamics*. Boston, MA: Springer US, 1996, pp. 37–69. ISBN: 978-1-4899-1373-9. DOI: 10.1007/978-1-4899-1373-9_3. URL: https://doi.org/10.1007/978-1-4899-1373-9_3.

[62] E. Fermi. *Thermodynamics*. Dover Books on Physics. Dover Publications, 2012. ISBN: 9780486134857. URL: https://books.google.pt/books?id=xCjDAgAAQBAJ.

[63] Keith Sherwin. *Introduction to Thermodynamics*.

[64] N. S. Nise. *Control Systems Engineering*. 6th. Hoboken, NJ: Wiley, 2011.

[65] Gene F Franklin, J David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems Sixth Edition*. 2010.

[66] Electrical Workbook. *Unit Step Input Time Response of a First Order Control System*. Accessed: 2025-03-23. 2019. URL: https://electricalworkbook.com/unit-step-response-first-order-control-system/.

[67] Thomas E Marlin. *Process Control Designing Processes and Control Systems for Dynamic Performance 2 nd Edition*. 2015.

[68] Process Control Guru. *First Order Plus Time Delay*. https://processcontrolguru.wordpress.com/2008/11/05/first-order-plus-time-delay/. Accessed: 2025-03-23. 2008.

[69] Aleksandr Tepljakov, Eduard Petlenkov, and Juri Belikov. "FOMCON: Fractional-Order Modeling and Control Toolbox for MATLAB". In: *Proceedings of the 18th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*. Gliwice, Poland: IEEE, 2011, pp. 684–689. DOI: 10.1109/MIXDES.2011.5979736.

[70] Aleksandr Tepljakov, Eduard Petlenkov, and Juri Belikov. "Controller design and implementation with FOMCON toolbox". In: *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Manchester, UK: IEEE, 2013, pp. 455–460. DOI: 10.1109/SMC.2013.83.

[71] MathWorks. *MathWorks – MATLAB & Simulink*. 2025. URL: https://www.mathworks.com/.

[72] Abdoalnasir Almabrok, Mihalis Psarakis, and Anastasios Dounis. "Fast Tuning of the PID Controller in An HVAC System Using the Big Bang–Big Crunch Algorithm and FPGA Technology". In: *Algorithms* 11 (Sept. 2018), p. 146. DOI: 10.3390/a11100146.

[73] Clébio de Oliveira Júnior. *Prevendo Números: Entendendo as métricas R^2, MAE, MAPE, MSE e RMSE*. Publicado no Medium pela comunidade Data Hackers. Dec. 2021. URL: https://medium.com/data-hackers/prevendo-n%C3%BAmeros-entendendo-m%C3%A9tricas-de-regress%C3%A3o-35545e011e70.

[74] Antonio Visioli. "Anti-windup Strategies". In: *Practical PID Control*. London: Springer London, 2006, pp. 35–60. ISBN: 978-1-84628-586-8. DOI: 10.1007/1-84628-586-0_3. URL: https://doi.org/10.1007/1-84628-586-0_3.

[75] Djalil Boudjehem, Moussa Sedraoui, and Badreddine Boudjehem. "A fractional model for robust fractional order Smith predictor". In: *Nonlinear Dynamics* 73.3 (2013), pp. 1557–1563. DOI: 10.1007/s11071-013-0885-9.

[76] Rammurti Meena et al. "Smith-predictor based enhanced Dual-DOF fractional order control for integrating type CSTRs". In: *International Journal of Chemical Reactor Engineering* (2023). DOI: 10.1515/jcre-2021-0216.

[77] Massimiliano Veronesi. *Performance improvement of Smith predictor through automatic computation of dead time*. Tech. rep. Technical Report. Yokogawa Italia, 2003.