# A Novel Nonlinear PID Controller for Improved Performance and Robust Control of Nonlinear Systems

Stefanos Charkoutsis

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

21st March 2025

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

1-DoF = 1 degree of freedom

2-DoF = 2 degree of freedom

2D_PID = 2 degree of freedom PID

CSTR = Continuous Stirred Tank Reactor

FOPID = Fractional Order PID

FOPTD = First Order Plus Time Delay

FONLPID = Nonlinear Fractional Order PID

GA = Genetic Algorithm

GB = Giga Byte

IMC = Internal Model Control

MIT = Massachusetts Institute of Technology

NG = Negative Gain

NLPID = Nonlinear PID

NmP = Non-minimum Phase

NCSTR = Nonlinear CSTR

PID = Proportional, Integral, and Derivative (Controller)

PSO = Particle Swarm Optimization

RAM = Random Access Memory

SOPTD = Second Order Plus Time Delay

SSD = Solid State Drive

SP_PID = Smith Predictor PID

T2_PID = NLPID steady state gain value substitution to PID

# List of Variables

A function is symbolised with a lower case letter and a variable inside brackets, such as $f(t)$. Transfer functions are symbolised as a function using a capital letter with complex variable $s$, for example, $P(s)$. The symbol for a signal norm is $||\cdot||$, while the symbol used for the absolute value is $|\cdot|$.

| Greek Letters | |
|---|---|
| $\Delta(s)$ | Uncertainty Transfer Function |
| $\epsilon$ | Feedback Error |
| $\lambda$ | Delay Variation |
| $\tau$ | Time Delay |

| Symbol | Variable Description |
|---|---|
| $a$ | Nonlinear proportional function gain parameter |
| $b$ | Set-point proportional weight |
| $c$ | Set-point derivative weight |
| $c_1$ | PSO Tuning Parameter |
| $C(s)$ | Controller Transfer Function |
| $d_1$ | Input Disturbance |
| $d_2$ | Output Disturbance / Sensor Bias |
| $f(\cdot)$ | Objective Function |
| $Gbest$ | Global best Objective Function Evaluation |
| $G_p(s)$ | Second Order Pade Delay Transfer Function Approximation |
| $i$ | Iteration index |
| $j$ | Particle Index |
| $k_p$ | Proportional Gain |
| $k_i$ | Integral Gain |
| $k_d$ | Derivative Gain |
| $k_0$ | Nonlinear proportional function gain parameter |
| $k_1$ | Nonlinear Integral function gain parameter |
| $k_2$ | Nonlinear Derivative Function Gain Parameter |
| $k_3$ | Nonlinear Derivative Kick Filtering Gain Parameter |
| $N$ | Derivative Filtering Parameter |
| $\%OS$ | Percentage Overshoot |
| $P(s)$ | Plant Transfer Function |
| $q_i$ | Denominator Polynomial Coefficients |
| $r$ | set-point |

| | |
|---|---|
| $sat(\cdot)$ | Saturation Function |
| $s$ | Laplace Complex Variable |
| $t$ | Time |
| $t_a$ | Additive Uncertainty Time Lag |
| $t_m$ | Multiplicative Uncertainty Time Lag |
| $t_s$ | Settling Time |
| $t_r$ | Rise Time |
| $T(s)$ | Delay-free Plant Transfer Function |
| $u$ | System Plant Input |
| $V$ | PSO particle Velocity |
| $W(s)$ | Multiplicative Uncertainty Weighting Transfer Function |
| $x(\cdot)$ | System States |
| $x_A$ | Steady State |
| $x_B$ | Steady State |
| $x_C$ | Steady state |
| $X$ | PSO Particle Position |
| $y$ | System Output |
| $z_i$ | Numerator polynomial coefficients |

# Abstract

The Proportional, Integral, and Derivative (PID) controller is the most common control algorithm in industry because of its simplicity, well-understood behaviour, and ease of design. PID is a linear controller with trade-offs between performance and robustness that cause performance compromises and a limited operating region in highly nonlinear industrial systems, which require iterative tuning for each operating region. This process is time-consuming, and the literature indicates that nonlinear PID controllers are a better alternative. However, a nonlinear PID controller that is simple, model-free, and easy to implement is required with an extensive analysis of stability.

This thesis proposes a novel Nonlinear PID (NLPID) controller using a unique set of nonlinear gain functions that can improve performance and robustness and eliminate step input derivative kicks, making the controller more energy efficient. The proposed controller is tuned using a Particle Swarm Optimization (PSO) algorithm with an objective function prioritising fast performance with minimum overshoot. An indicative stability analysis has also been conducted through extensive simulations to justify the constraints region, which allows for the determination of stable control gain parameters.

The proposed NLPID controller is simulated for a Nonlinear Continuous Stirred Tank Reactor (NCSTR) model with saturation and disturbances at various operating regions. The proposed controller is also benchmarked against the conventional PID, two degrees of freedom PID, and Smith predictor PID controllers in the three linearised dynamics, which are, a First Order Plus Time Delay (FOPTD), a Negative Gain Second Order Plus Time Delay (NG-SOPTD), and a Non-minimum Phase SOPTD (NmP-SOPTD) systems. The benchmarking results show that the proposed NLPID controller improves the performance in all nominal systems, and improves robustness against parametric, additive, and multiplicative uncertainties.

In summary, the proposed NLPID controller improves performance and robustness, expanding the operating region of PID in nonlinear systems, using the proposed unique set of nonlinear gain functions. The proposed controller provides an alternative control algorithm to the literature that is model-free, nonlinear, and supported with an indicative stability analysis. Future work can be done to expand the stability analysis with a rigorous mathematical approach for linearisations and to a class of nonlinear systems and potentially include an observer to improve robustness.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Acknowledgements

In 2008, Greece went through an economic crisis that caused huge damage to families all across the country, my family being one of them. Surmounting debts and loans taken during the economic bubble our recently built home was finalised in 2010, when we moved in to only live there for 2 years. Because my parents knew that living in Greece would not be possible. The UK was our stepping stone to a new life. My father fought hard to find jobs and finish projects to earn enough money for us to move to the UK, which finally, after extremely hard work and sweat, he managed to achieve. Fast-forward to the future, we reached the UK to make our new life, I was 15 years of age at the time, I missed my friends, my sister, and my family in Greece, and had no one I knew but my parents and my brother, I was scared and lonely, but never alone. My parents went through a lot, my mother got cancer in 2013, which she was lucky to find early, my father got cancer twice, once in 2018, which was surgically removed from his oesophagus and again in 2021. However, this time it came to stay. I am utmost grateful to my family because after all they have gone through they stay positive and they are there to shine the light when it gets dark. I was always lucky to have this family and lucky enough that I knew what I wanted to do. My parents could see my passion and they supported me for all these years in achieving my goals, no matter what, through thick and thin, not only by providing me with food and shelter and all the help I needed but also by giving me the opportunity to make a new life here in the UK. Consequently, my life was simple. I knew what I had to do. The least that I could do. Take this opportunity and maximise its potential. Therefore, to my family, I love you and I dedicate this work to you, for it would not have been possible without you. I am grateful to my family for always supporting me in my goals. They have been the bedrock of my life's pursuits and they have shown their support in silence, daily, by all they provided me with and all the sacrifices they made.

I would also like to thank Liverpool John Moores University for making this project possible and providing this incredible opportunity to propel my career and academic achievements forward. In addition, I would like to thank LJMU for the pro-vice-chancellor scholarship that supported and funded my studies. Moreover, I would like to say a special thank you to my lead supervisor Dr Mohamed Kara-Mohamed not only for providing me with academic support but also for supporting me throughout my personal struggles, and for being a great mentor to me in my professional life. Finally, I would like to say thank you to my friend and colleague Dr Andrew Spiteri for his great help and support in proofreading my thesis and help me build up my work.

# Commemorations

**Because it was space that inspired all my pursuits, I dedicate a quote from the great educator and scientist Carl Sagan, inspired by the first image of Earth that humans ever saw, taken by Voyager I, more than 4 billion miles away.**

> To my mind, there is perhaps no better demonstration of the folly of human conceits than this distant image of our tiny world. To me, it underscores our responsibility to deal more kindly and compassionately with one another and to preserve and cherish that pale blue dot, the only home we've ever known.
>
> *Carl Sagan*

# List of Publications

## List of Journals

Charkoutsis, S. and Kara-Mohamed, M. (2023a). 'A Particle Swarm Optimization Tuned Nonlinear PID Controller with Improved Performance and Robustness for First Order Plus Time Delay Systems'. In: *Results in Control and Optimization* 12, p. 100289. ISSN: 2666-7207. DOI: 10.1016/j.rico.2023.100289. [Accessed: 10th Oct. 2023].

— (n.d.). 'A PSO Tuned Robust Novel NLPID Controller with Application to Nonlinear Systems. (Submitted, Awaiting Results).' In: *Lecture Notes in Electrical Engineering - ICINCO - 2023*. Springer.

## List of Conferences

Charkoutsis, S. and Kara-Mohamed, M. (2023b). 'Towards a Novel Nonlinear Pid Controller Tuned with Particle Swarm Optimization with Improved Performance for First Order Plus Time Delay (FOPTD) Systems.' In: *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics*. Vol. 2, pp. 25–33. DOI: 10.5220/0012173600003543.

# Chapter 1

# Introduction

The Proportional, Integral, and Derivative (PID) controller is a ubiquitous controller in industry (Åström and Hägglund, 1995). The PID controller is commonly used because of its simplicity, performance, and the ability to control systems without requiring knowledge of the system dynamics (Åström and Hägglund, 1995). In addition, the PID controller is a feedback control strategy that is useful in any single-input, single-output system, and it has the flexibility to work in synthesis with any other control strategy or architecture (Åström and Hägglund, 1995). This flexibility allows for improved control and robustness in various systems, which has enlarged the literature on PID control (Åström and Hägglund, 1995).

This intuitive design is an advantage in linear systems, however, in nonlinear systems, the controller design is no longer intuitive, and the engineers rely heavily on analytical tools and heuristic methods to establish the design. Hence, plenty of algorithms and approaches that can tune a PID controller have been developed with little performance improvements (Åström and Hägglund, 1995; O'Dwyer, 2009; Abushawish, Hamadeh and Nassif, 2020). The PID controller also possesses a single-degree-of-freedom structure that can focus either on performance or regulation of the system (Åström and Hägglund, 1995; Åström, 2000; Chen, J., Fang and Ishii, 2019). This also builds an additional problem to its tuning, requiring trade-offs in performance and robustness (Åström and Hägglund, 1995; Åström, 2000; Chen, J., Fang and Ishii, 2019). These limitations of PID control are then amplified by nonlinear systems and it becomes more complicated to achieve improved performance (Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Bernstein, 2022; Chaturvedi, Kumar, N. and Kumar, R., 2023).

## 1.1. Motivation

Nonlinear dynamics are more complex because they possess multiple equilibrium points, each with distinct and independent qualitative local properties, meaning that a general solution cannot be established (Hale and LaSalle, 1963; Slotine and Li, 1991; Khalil, 2002). Other nonlinear properties include chaos and limit-cycles, which can cause problems in control systems if left untreated (Hale and LaSalle, 1963; Khalil, 2002). Linear systems do not suffer from these problems, because of the superposition property that allows one to understand the global behaviour of the system from the local set of dynamics, which makes the system more predictable (Hale and LaSalle, 1963; Khalil, 2002). However, nonlinear systems describe the nature of the system more closely since linear systems are only approximations of small regions of nonlinear systems (Hale and LaSalle, 1963; Slotine and Li, 1991; Khalil, 2002). For that reason, linear controllers provide a satisfactory response to nonlinear systems if the desired control is near the operating point and does not stave away from it and enter a different region. If this occurs, the control system can be rendered unstable or observe a major deterioration in performance. To resolve this using linear controllers, different tunings are established for each operating region, which requires extensive tuning trials and analysis (Khalil, 2002; Sinha and Mishra, 2018; Bernstein, 2022). This approach requires linearisation of the nonlinear dynamics near an equilibrium point and as a consequence, requires robust control design to maintain stability near that operating region (Khalil, 2002; Sinha and Mishra, 2018; Bernstein, 2022). However, the PID controller suffers from trade-offs in performance and robustness and experiences major deterioration in performance due to the prioritization in robustness. This requires improved performance while maintaining robust control of nonlinear systems (Khalil, 2002; Sinha and Mishra, 2018; Bernstein, 2022).

**Motivating Example:**

One highly nonlinear system that is commonly seen in industry is the Nonlinear Continuous Stirred Tank Reactor (NCSTR). This system is widely seen in the chemical industries and its system dynamics are used as a benchmarking model in nonlinear control (Colantonio et al., 1995; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Chaturvedi, Kumar, N. and Kumar, R., 2023). The NCSTR model possesses multiple equilibrium points and engineers resort to linearising the system dynamics to the desired equilibrium points, which then allows the engineer to design a linear control system (Colantonio et al., 1995; Krishna et al., 2012). Because of its complex dynamics other control methods, such as sliding mode control, feedback linearization, and model predictive control become more complicated to implement (Harmon Ray, 1981; Colantonio et al., 1995). Hence, the most common control approach for the NCSTR system is using the PID controller for a set of linearised dynamics (Colantonio et al., 1995; Krishna et al., 2012). As a result, multiple tuning methodologies have been developed for the PID controller to maintain fast regulation and achieve improved performance, with little improvements made (O'Dwyer, 2009; Krishna et al., 2012). Because of the little improvements made and the single degree of freedom structure of the PID controller, the tuning and control problem is commonly focused on the system regulation rather than performance (Krishna et al., 2012; So and Jin, 2018; Sinha and Mishra, 2018; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Chaturvedi, Kumar, N. and Kumar, R., 2023). Due to these issues amplified by the complex NCSTR dynamics and its frequent use in industry, it's effective control has been an on-going problem (So and Jin, 2018; Sinha and Mishra, 2018; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Chaturvedi, Kumar, N. and Kumar, R., 2023).

The dimensionless nonlinear CSTR model is used to show the classical nonlinear control problem where the dynamics are mathematically represented as follows (Harmon Ray, 1981; Colantonio et al., 1995; So and Jin, 2018; Sinha and Mishra, 2018):

$$\begin{aligned} \dot{x}_1(t) &= -x_1(t) + D_a(1 - x_1(t))e^{\left[\dfrac{x_2(t)}{1 + x_2(t)/\gamma}\right]} + d_1(t), \\ \dot{x}_2(t) &= -(1 + \beta)x_2(t) + HD_a(1 - x_1(t))e^{\left[\dfrac{x_2(t)}{1 + x_2(t)/\gamma}\right]} + \beta u(t) + d_2(t), \\ y(t) &= x_2, \end{aligned} \tag{1.1}$$

where $H$ is the heat of reaction, $D_a$ is the Damkohler constant, $\gamma = E/(R \cdot T_f)$, $E$ is the activation energy, $R$ is the gas constant, $\beta$ is the heat transfer coefficient, $T_f$ is the temperature of the inlet.

The example dimensionless NCSTR system possesses three equilibrium points that have different stability properties (Harmon Ray, 1981; Colantonio et al., 1995; So and Jin, 2018; Sinha and Mishra, 2018). The control problem that is solved for the NCSTR system is to control the system so that the states change from the equilibrium point $x_A$ to $x_B$. Then the second change is from equilibrium point $x_B$ to $x_C$. That is $x_A \rightarrow x_B \rightarrow x_C$. The three equilibrium points possess a completely different set of dynamical properties. This means that linear controllers require different tuning to control the system for each equilibrium point. The desired scenario is to control the system, regardless of the equilibrium point with stability and robust performance results using a single tuning. To accomplish this, the desired control approach must have adaptability for each equilibrium point, must be model-free, and maintain the simplicity of use.

An alternative approach to control nonlinear systems is to use adaptive gains to establish a nonlinear PID control form that is consistent and adaptive across all operating regions, without the need for re-design (So and Jin, 2018; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). One simple method of adaptive gains is to incorporate nonlinear functions that determine the gains of the PID controller, which can work for a larger variety of systems when adaptive control methodologies provide plant-specific design (Zhang and Mao, 2017; Frank, 2018; Slama, Errachdi and Benrejeb, 2019). Hence, there is an advantage to having a nonlinear control algorithm that is not model-dependent and can generate adequate performance and robustness for nonlinear systems with ease. For that reason studying the nonlinear gains PID control strategy is useful, due to the well-known model-free nature of the PID controller and due to the adaptive nature of nonlinearity to change the control gains in response to the system in an easy-to-understand approach.

## 1.2. Aims and Objectives

This thesis aims to develop a practical novel nonlinear PID controller that provides improved performance to nonlinear systems with robustness against uncertainty and disturbances.

This thesis comprises a selection of objectives that aim at filling the identified gaps in the literature as follows:

1. To design a novel NLPID controller.
2. To tune the parameters of the NLPID gains to meet specifications.
3. To test the proposed controller in a nonlinear system.
4. To benchmark the proposed controller against conventional control methods in linear and nonlinear control problems.
5. To test the robustness of the proposed controller against various uncertainties.

## 1.3. Thesis Contributions

1. A novel nonlinear model-free PID control algorithm has been proposed and benchmarked that shows improved performance and robustness in nonlinear systems.
2. An indicative stability analysis has been established for the proposed nonlinear PID controller to justify its effectiveness and robustness.
3. The proposed nonlinear PID controller improves key PID limitations such as derivative kicks, gain adaptation, and windup.
4. The proposed controller extends the operating region of the conventional PID controller in the NCSTR problem so that it is not bounded within a linearised operating region.

## 1.4. Thesis Outline

**Chapter 2:** An overview of the recent literature on feedback systems, PID control, its limitations, and alternative control methods. In addition, this chapter presents the background of nonlinear systems and nonlinear controllers. Extensive literature on adaptive and nonlinear PID control is also shown. Finally, the gap in the literature is identified together with the scope and impact of this research.

**Chapter 3:** The introduction of a novel nonlinear PID controller is proposed in this thesis as an alternative control method. An extensive simulations-based indicative stability analysis of the proposed controller is also shown in this chapter. The improvements that the proposed NLPID controller provides to the fundamental limitations of PID are also shown.

**Chapter 4:** The performance, disturbance rejection, and operating region capacity of the proposed controller are shown for the nonlinear CSTR model. In addition, the benchmark against the PID, two-degree-of-freedom PID, and Smith Predictor PID controllers is shown for various plant models. The plant models included are a First Order Plus Time Delay (FOPTD), a Negative Gain Second Order Plus Time Delay (NG-SOPTD), and a Non-minimum Phase SOPTD (NmP-SOPTD) model. Finally, the stability regions of the proposed controller are also shown for each plant model.

**Chapter 5:** Robustness simulations against various types of uncertainty are shown for the proposed NLPID controller. The NCSTR model is simulated against parametric uncertainty and incremental load disturbances to show the robustness of the proposed controller. The FOPTD, NG-SOPTD, and NmP-SOPTD systems are simulated against parametric, additive, and multiplicative uncertainties. This shows an extensive simulation-based testing of the proposed NLPID controller for its robustness to deal with various uncertainties.

**Chapter 6:** Discussion, conclusions, and future work chapter shows a discussion of the results, the performance, and robustness capabilities of the proposed NLPID controller. The benchmarking results of the proposed controller are discussed together with the limitations that have been observed. In addition, the conclusions are summarized for the overall thesis and how this research has answered the key research aims and the contributions to the literature gap, proposed in Chapter 2. Finally, future work is discussed and the potential for future plans that can improve the proposed NLPID controller.

# Chapter 2

# Literature Review

## 2.1. Introduction

The advancements in modern technology have an increasing need for feedback control with the future direction towards intelligent and autonomous systems. Control theory is a well-established and multi-disciplinary research field that studies the interplay of feedback systems in an autonomous system. For that reason, control theory has tools and methods available to promote industrial automation and advance autonomous technology. However, the nonlinear nature of advanced technologies requires new developments and research gaps in nonlinear and adaptive control. This means an increasing need for more intricate and developed control algorithms that better deal with uncertainty, nonlinearity, and time delays. This chapter shows the literature review conducted on the well-established background of feedback control theory and control limitations. The limitations of feedback control in industry-facing uncertainties, the limitations of linear control, and inherent nonlinear dynamics are also discussed. Different industrial methods that deal with those control problems are also shown with a rigorous appeal to their limitations. This literature review establishes the background of the thesis and the noticed gaps in the literature that the contributions of this thesis are to fulfil.

## 2.2. Feedback Control

Feedback control theory is a long and well-established theory within the literature for both linear and nonlinear systems. Figure 2.1 shows a graphical representation of a classical simple feedback loop with the lines connecting the different blocks representing system signals.

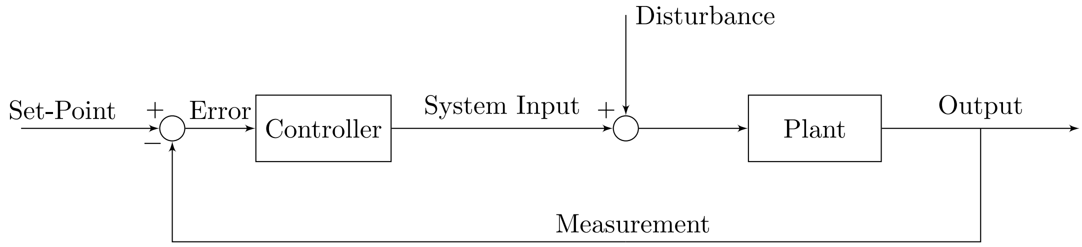



**Figure 2.1:** A classical feedback control strategy.

The feedback control strategy relies on sensor measurements, providing superior control response without assuming perfect system knowledge and relying on ideal conditions for its successful function. The ability of feedback control to deal with uncertainties and disturbances is why feedback is such a widespread control methodology (Chen, J., Fang and Ishii, 2019; Bernstein, 2022). Feedback mitigates uncertainties in models and environmental conditions through sensor measurements that inform the controller to make better decisions. However, a limitation present in feedback are the inherent noisy signals in sensor measurements and inconsistencies in the model parameters that can change the dynamics of industrial systems (Boubaker, 2013; Mangera, Pedro and Panday, 2022; Bernstein, 2022). In essence, feedback control makes systems insensitive to external disturbances and to parameter variations in the individual systems (Bernstein, 2022). A direct consequence of feedback is that stable systems can become unstable, and unstable systems can become stable (Bernstein, 2022). This consequence has inquired theorists and mathematicians to develop stability tools to understand and analyse how parameter variations and uncertainties may render the system unstable or deteriorate its performance (Åström and Hägglund, 1995; Isidori, 1995; Khalil, 2002; Chen, J., Fang and Ishii, 2019; Bernstein, 2022). This brings the design of any feedback loop into an iterative process of analysis and design until the response converges to as close of a solution as desired by the specifications (Slotine and Li, 1991; Khalil, 2002). As a result, the fundamental problem of feedback is to answer; *how can one design a feedback control strategy influenced by model uncertainties and unpredictable disturbances to maintain stability and desired performance* (Bernstein, 2022)?

The control of feedback systems can be conducted in two major strategies, the error-based controllers, and the model-based controllers (Cheng, L., 2021; Bernstein, 2022). The error-based controllers have the advantage that they do not rely on the plant dynamics and as a consequence, they are reactive control systems that are robust and can maintain stability for large uncertainties and disturbances. However, they lack in performance in certain cases, for which model-based controllers can be superior (Konstantopoulos and Baldivieso-Monasterios, 2020; So, 2021; Bernstein, 2022). It is rarely the case that models are certain and disturbances do not occur, as a result, model-based controllers rarely achieve the promised performance and are highly sensitive to disturbances and uncertainties (Boubaker, 2013; Konstantopoulos and Baldivieso-Monasterios, 2020; So, 2021; Bernstein, 2022). This is a limitation when applied to advanced technologies, which require advanced control strategies that are practical and model-free. This increase in control demand and with the advancements of information systems, there is an increasing need for nonlinear and adaptive control methodologies that can increase the operational range and robustness of feedback systems (Boubaker, 2013; Iqbal et al., 2017; Rehman, Petersen and Pota, 2017; Mo and Farid, 2019; Konstantopoulos and Baldivieso-Monasterios, 2020; Bernstein, 2022).

However, it is also necessary to know the limitations of any control system that is designed to achieve a specific feedback response. There is well-documented literature on the design limitations of any feedback controller, which is predominantly restricted by plant dynamics. More specifically, the major concerns are when plants possess one or a combination of unstable poles, unstable (positive) zeros, and time delays (Åström and Hägglund, 1995; Chen, J., Fang and Ishii, 2019). In addition, there is a fundamental performance vs robustness trade-off to be considered that any control design cannot avoid (Åström and Hägglund, 1995; Chen, J., Fang and Ishii, 2019; So, 2021; Bernstein, 2022). These limitations are inherent to both linear and nonlinear systems. To improve performance, adaptive and nonlinear control methods are commonly used, which can change the loop design according to the conditions, to prioritise robustness during steady-state and prioritise performance during the transient state (Iqbal et al., 2017; Rehman, Petersen and Pota, 2017; Mo and Farid, 2019; So, 2021; Bernstein, 2022).

**Robustness in Feedback Control**

Stability robustness is a measure of how much uncertainty can the control system be able to accommodate before reaching instability (Skogestad and Postlethwaite, 2001; Frank, 2018). The more robust a control system is, the better it can handle uncertainty (Skogestad and Postlethwaite, 2001; Frank, 2018). In addition, performance robustness is defined as a measure of the controller's ability to accommodate uncertainty and maintain close performance as that defined by the specification (Skogestad and Postlethwaite, 2001; Frank, 2018). It can be seen that the two definitions of robustness have to deal with how much uncertainty the controller can accommodate to achieve a reliable performance or stability (Skogestad and Postlethwaite, 2001; Frank, 2018). To achieve such a measure, one must be able to model or predict the expected type of uncertainty and consider it as part of the analysis process. The different types of uncertainties are (Skogestad and Postlethwaite, 2001):

- Model Uncertainty,
- Parametric Uncertainty,
- Sensor and Actuator Failure,
- Physical system constraints, and
- Changes in control objectives.

The different methods of modelling and representing parametric and model uncertainties are well-developed and established within the literature. It is customary to model the uncertainty in parameter estimations using parametric uncertainty, while unmodelled dynamics are commonly modelled as additive or multiplicative or different forms of uncertainty (Skogestad and Postlethwaite, 2001; Normey-Rico and Camacho, 2007; Zhang and Mao, 2017; Frank, 2018; Parnianifard, Fakhfakh et al., 2021).

## 2.3. Nonlinear Systems

Nonlinear systems are systems whose output response is not linearly dependent on the inputs provided into the system. This means that nonlinear systems are more difficult to predict and understand when compared to linear systems that have linearly dependent inputs and outputs. Natural systems are inherently nonlinear, as a result, the mathematical description of such systems possesses nonlinear dynamics. The mathematical models describe systems of interest using a set of ordinary differential equations with respect to their time-dependent or state-dependent behaviour. The study of these systems begins from the autonomous nonlinear equations, which form the pure dynamical properties without considering external inputs to the system. The state-space representation of autonomous nonlinear systems can be shown as (Hale and LaSalle, 1963; Khalil, 2002):

$$\dot{x}(t) = f(x(t)),$$
$$x(0) = x_0.$$

The solution of the nonlinear system (2.1) is admitted under a set of initial conditions $x_0$ that determine the future states of the system, called the initial value problem. As a result, system solutions must have continuous dependence on the initial conditions, globally or locally in a defined continuous neighbourhood. In order for the nonlinear system Eq.(2.1) to have global continuous dependence on initial conditions, the nonlinearity $f(x)$ must be Lipschitz continuous (Hale and LaSalle, 1963; Khalil, 2002). Lipschitz continuity is defined as follows (Khalil, 2002):

**Definition 2.3.1** (Lipschitz Continuity)**.** A nonlinear function $f(\cdot)$ is Liprschitz continuous if it satisfies the condition:

$$||f(y) - f(x)|| = K||y - x||. \tag{2.1}$$

For some constant $K$, called the Lipschitz constant, and $f(0) = 0$

Lipschitz is a broader definition of continuity and is necessary for the existence of solutions of nonlinear initial value problems. Finding the solution to nonlinear systems is complex and many can have multiple solutions that do not satisfy the convenient superposition principle that linear systems possess (Hale and LaSalle, 1963; Khalil, 2002). As a result, mathematicians and theorists have developed mathematical tools that can determine the general behaviour of the system solutions, based on the knowledge of the system (Hale and LaSalle, 1963; Khalil, 2002). A key element of nonlinear systems is to determine the states that if an initial condition is defined in such state, then the future states of the system naturally remain static. These points are called equilibrium points and the mathematical definition of an equilibrium point is stated as follows (Khalil, 2002):

**Definition 2.3.2** (Equilibrium Points)**.** The equilibrium point $x_{eq}$ of a nonlinear system, defined as Eq.(2.1), is a state that satisfies the following equation:

$$\dot{x}(t) = f(x_{eq}) = 0. \tag{2.2}$$

From the above definition, it can be seen that a nonlinear function $f(\cdot)$ may admit multiple equilibrium points $x_{eq}$, which satisfy the equation. This is a major drawback in nonlinear systems, since each equilibrium point may possess different dynamical properties, which can cause various types of unpredictable, complex, and periodic behaviour (Hale and LaSalle, 1963; Khalil, 2002). To analyse the system behaviour and to visualise the equilibrium points, phase-plane analysis can be used for both linear and nonlinear systems, which can help intuitively understand the system behaviour (Hale and LaSalle, 1963; Khalil, 2002; Frank, 2018). The main limitation of phase-plane analysis is that it works for two dimensional systems and cannot represent higher-order dynamics, which limits its application (Hale and LaSalle, 1963; Khalil, 2002; Frank, 2018). Hence, stability tools are useful methods for establishing the general solution trends near equilibrium points to determine the overall behaviour of the system (Hale and LaSalle, 1963; Khalil, 2002). To state the stability criteria of different systems it is critical to first define stability properties of an equilibrium point. The stability definitions are as follows:

**Definition 2.3.3** (Stability (Khalil, 2002))**.** The equilibrium point at the origin of a system defined as Eq.(2.1) is:

- stable, if for each $\epsilon > 0$, there is a $\delta = \delta(\epsilon) > 0$ such that:

$$||x(0)|| < \delta \Rightarrow ||x(t)|| < \epsilon, \forall t \geqslant 0. \tag{2.3}$$

- unstable, if it is not stable.
- asymptotically stable, if it is stable and $\delta$ can be chosen such that:

$$||x(0)|| < \delta \Rightarrow \lim_{t \to \infty} x(t) = 0. \tag{2.4}$$

The above definitions of stability determine the mathematical properties of the solutions near the equilibrium states that can aid in understanding the overall system behaviour without directly solving the system (Hale and LaSalle, 1963; Slotine and Li, 1991; Khalil, 2002). Stability criteria are then based on these fundamental definitions with different requirements and constraints imposed to show stability or asymptotic stability (Hale and LaSalle, 1963; Slotine and Li, 1991; Khalil, 2002). The key difference between the two being that asymptotically stable systems are stable but also convergent on the equilibrium state (Hale and LaSalle, 1963; Slotine and Li, 1991; Khalil, 2002). The most used criterion to determine stability or asymptotic stability is the Lyapunov stability criterion that is a sufficient condition for autonomous nonlinear systems, such as Eq.(2.1).

Lyapunov theory utilises energy-type functions, such as $V(x(t))$, to determine stability. Energy functions show that the system trajectories defined by the solutions near a stable equilibrium have decreasing energy over time. Meaning that a stable equilibrium does not provide additional energy into the system. As a result, Lyapunov stability is an overarching stability theorem that has many other stability criteria developed using that energy principle, such as, small gain theorem, passivity and dissipativity, but also for specific nonlinearities the Circle and Popov criteria (Slotine and Li, 1991; Khalil, 2002). The Circle and Popov criteria are also called in the literature as hyper-stability criteria. This is because they are conservative sufficient conditions of stability, meaning that they predict instability prematurely (Maddi, Guessoum and Berkani, 2014). Hence, resulting in a control design that is restricted to smaller region of feedback stabilising controller gains (Slotine and Li, 1991; Khalil, 2002; Maddi, Guessoum and Berkani, 2014).

## 2.4. Nonlinear Control Theory

Nonlinear controllers are increasingly necessary for the effective control of complex systems and for improving the operating region (Khalil, 2002; Iqbal et al., 2017; Mo and Farid, 2019; Konstantopoulos and Baldivieso-Monasterios, 2020). Most nonlinear controllers use the nonlinear dynamics that are underlying within the mathematical model of the system, with the majority generating a control algorithm that relies on the Lie derivatives (Khalil, 2002; Iqbal et al., 2017; Mo and Farid, 2019; Konstantopoulos and Baldivieso-Monasterios, 2020). However, such nonlinear control algorithms are extremely demanding for the hardware and they complicate the process of discretization for hardware implementation (Iqbal et al., 2017; Konstantopoulos and Baldivieso-Monasterios, 2020). In addition, for feedback linearization and Lie derivative approach control algorithm development, nonlinear systems must have a nonlinear control affine model, which is described by system Eq. (2.5)-(2.6) (Mo and Farid, 2019).

$$\dot{x}(t) = f(x(t)) + g(x(t))u, \tag{2.5}$$

$$y(t) = h(x(t)). \tag{2.6}$$

Feedback linearization is an effective and powerful control methodology that fully linearises the dynamics of the system, without limiting the operating region of the system (Slotine and Li, 1991; Khalil, 2002; Mo and Farid, 2019). However, to conduct feedback linearization, it requires high order Lie derivative computations and it is necessary that the involutivity condition holds for the system dynamics (Slotine and Li, 1991; Khalil, 2002; Mo and Farid, 2019). Moreover, feedback linearisation is sensitive to external disturbances and noisy signals, which has significant implications for practical control systems (Slotine and Li, 1991; Khalil, 2002; Mo and Farid, 2019). This sensitivity to disturbance is because feedback linearization works for nonlinear control affine systems, which transform the control problem into a linear control problem through the following control affine transformation (Slotine and Li, 1991; Khalil, 2002; Mo and Farid, 2019):

$$u(t) = \frac{1}{g(x)}(v - f(x)). \tag{2.7}$$

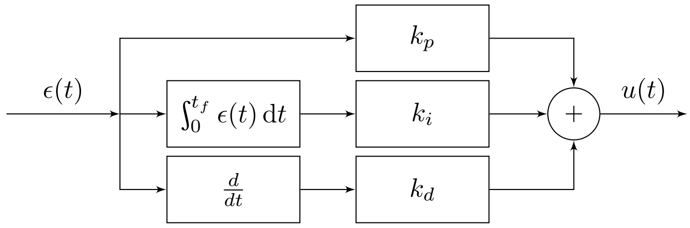
This means that the nonlinear function $g(x)$ or in the case that it is a matrix $G(x)$ must be invertible, which is an additional condition of the feedback linearization method. This means that $f(x)$ and $g(x)$ are both continuous and differentiable. Using the control input Eq.(2.7) cancels the nonlinearities, by utilising the system model and then defining a linear controller $v$. If the system model is slightly different to the feedback linearization, it can render the system unstable (Slotine and Li, 1991; Khalil, 2002; Frank, 2018; Mo and Farid, 2019).

The concept of adaptation is a very powerful nonlinear control scheme and is useful in establishing a control algorithm that works for a large range of operating points (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Mo and Farid, 2019; Slama, Errachdi and Benrejeb, 2019; Cheng, L., 2021). Adaptation prescribes gains dynamically through utilising system identification that models the plant dynamics and accordingly assigns the gains of the controller using an objective function criteria (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Mo and Farid, 2019; Slama, Errachdi and Benrejeb, 2019; Cheng, L., 2021). Adaptive control harnesses data collection technologies to establish the gains actively making it a powerful tool for control of nonlinear systems, with a well-developed literature and theory (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Mo and Farid, 2019; Slama, Errachdi and Benrejeb, 2019; Cheng, L., 2021; Pesce, Colagrossi and Silvestrini, 2023). The drawback in adaptive control are that it requires a two-fold of the control structure that complicates the problem, which are gain adaptation and system model identification (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Slama, Errachdi and Benrejeb, 2019; Pesce, Colagrossi and Silvestrini, 2023).

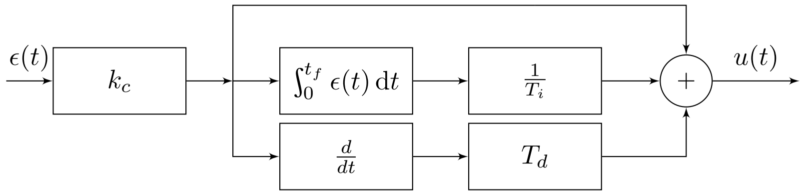
These two problems are difficult to solve and convergence of the system model identification algorithm must also be reassured, while the gain adaptation relies on objective function evaluations that may render the system unstable (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Slama, Errachdi and Benrejeb, 2019; Pesce, Colagrossi and Silvestrini, 2023). Due to the potential for the gains to establish an unstable control, robust adaptive rules have also been studied and are well-established in the literature as it is a well known issue in adaptive control (Zhou and Doyle, 1998; Khalil, 2002; Pesce, Colagrossi and Silvestrini, 2023). In addition, adaptive control requires a large amount of reliable data and the effectiveness of the control algorithm is directly related and has trade-offs between the data reliability and data filter processing time (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Slama, Errachdi and Benrejeb, 2019; Pesce, Colagrossi and Silvestrini, 2023). The adaptation rules and the system model identification algorithms rely on expensive computations that can also take large and expensive hardware to implement in practice (Zhou and Doyle, 1998; Khalil, 2002; Zhang and Mao, 2017; Mo and Farid, 2019; Slama, Errachdi and Benrejeb, 2019; Cheng, L., 2021; Pesce, Colagrossi and Silvestrini, 2023). As a result, although adaptive control is a major corner stone of nonlinear control theory, it is computationally expensive, it is dependent on reliable data and efficient filters. This makes adaptive control the ideal resort for specific applications where the additional cost of hardware is less concerning and priority is made on automation (Zhang and Mao, 2017; Mo and Farid, 2019; Slama, Errachdi and Benrejeb, 2019; Cheng, L., 2021; Pesce, Colagrossi and Silvestrini, 2023).

## 2.5. The PID Controller

The Proportional, Integral, and Derivative (PID) controller is a well-established control algorithm that is used in the majority of control applications (Åström and Hägglund, 1995; Sung and Lee, 1996; O'Dwyer, 2009). It has three gains, which are tunable and have an intuitive effect to the feedback system response, which makes them simple and effective (Åström and Hägglund, 1995; Faccin and Trierweiler, n.d.; O'Dwyer, 2009). PID is an error-based controller that takes the form of linearly combining the past errors (integration), present errors (proportional), and the future estimates of error (derivative). The PID control algorithm can be expressed in many forms, the most commonly used are the series form and the parallel form (Åström and Hägglund, 1995; Sung and Lee, 1996). Figure 2.2 shows the schematic of the parallel and series structures. Figure 2.2a shows the parallel structure form where the three different gains are applied to the feedback error distinctly and the three actions are added in a linear combination. Figure 2.2b shows the series structure with the primary difference being the application of the proportional gain, which is applied as a cascaded gain to the PID controller, influencing all the PID parameters simultaneously.




**(a)** The parallel structure.




**(b)** The series structure.

**Figure 2.2:** The schematic diagram of the parallel and series structures of the PID controller.

In terms of robustness and flexibility, the parallel form PID controller is the most robust and effective form, offering increased flexibility in the tuning (Åström and Hägglund, 1995; Sung and Lee, 1996; O'Dwyer, 2009). This is because the parallel form combines the three error aspects into three linearly independent vectors that decouples the effect of each gain, making it more intuitive and flexible (Åström and Hägglund, 1995; Sung and Lee, 1996; O'Dwyer, 2009). The transfer function and time-domain representations of the parallel form PID control are given, respectively, as follows:

$$K_{\mathrm{PID}}(s) = k_{\mathrm{p_c}} + k_{\mathrm{i_c}}\frac{1}{s} + k_{\mathrm{d_c}}s, \qquad u_{\mathrm{PID}}(t) = k_{\mathrm{p_c}}\epsilon(t) + k_{\mathrm{i_c}}\int_0^{t_{\mathrm{f}}} \epsilon(t)\,dt + k_{\mathrm{d_c}}\dot{\epsilon}(t), \tag{2.8}$$

where $k_{\mathrm{p_c}}$, $k_{\mathrm{i_c}}$, and $k_{\mathrm{d_c}}$ are the proportional, integral, and derivative gains, respectively, $\epsilon(t)$ is the feedback error and $t_{\mathrm{f}}$ is the integration time. These gains serve as weights of importance of the three key aspects of PID control, which mathematically are: proportional, integral, and derivative. These three key aspects possess certain advantages and disadvantages to the control algorithm, but also to the effect of the feedback response. Table 2.1 shows the respective effects on the feedback response, by increasing each PID gain.

**Table 2.1:** Effects of PID gains on feedback response (Åström and Hägglund, 1995).

| | Rise-Time | Settling-Time | Overshoot | Steady-State Error | Stability |
|---|---|---|---|---|---|
| $\uparrow k_p$ | Decrease | Small Increase | Increase | Decrease | Degrade |
| $\uparrow k_i$ | Small Decrease | Increase | Increase | Major Decrease | Degrade |
| $\uparrow k_d$ | Small Decrease | Decrease | Decrease | Small Changes | Improve |

These effects of the gains in the feedback response are intuitive and simple to understand, which also gives the ability for educated guesses and trial and error tuning of the controller. However, there are some additional practical implications of using the integral and the derivative of the feedback error. For example, all actuators have certain limits of operations, this is not seen directly by the controller, since it only knows the feedback error. Hence, when the actuator reaches its operational limits, it saturates and the controller keeps demanding for more. This can cause large actuator degradation and can potentially cause damage. In addition, the integrator in effect is adding more errors, which are not corrected, since the actuator can not handle the appropriate load, which in turn increases the control demand. Effectively, the control signal increases continually and the more error it collects, the longer it takes to reduce the control demand when the control input is corrected.

Another practical disadvantage is that all feedback measurements come from inaccurate and noisy sensors. These sensors essentially cause small but rapid and random variations in the signals, which are amplified by the derivative control. This noise amplification can be seen in the following simple example.

**Example 2.5.1.** Assume a noise described by the following sinusoidal function:

$$n(t) = Asin(\omega t). \tag{2.9}$$

Take the derivative of $n(s)$ and we have:

$$\dot{n}(t) = A\omega cos(\omega t). \tag{2.10}$$

One can see that the larger $\omega$ is, the larger the derivative of the noise, multiplied by the amplitude of the signal. △

This effectively shows that the amplitude of the derivative is directly proportional to the frequency of the noise. As a result, PID controllers rely on filtering procedures for fully robust PID control, to reduce signal noise. Moreover, this amplification effect also causes large disturbing signals when a rapid step function is implemented as a set-point input. This can also be shown by a simple example of a discrete-time derivative as follows.

**Example 2.5.2.** Suppose a step set point is implemented as an input to the system and the feedback error becomes momentarily for $n = 0$ as:

$$\epsilon_n := \begin{cases} k & \text{if } n > 1 \ , \\ 0 & \text{if } n = 0. \end{cases} \tag{2.11}$$

Take the forward Euler discrete derivative as:

$$\frac{d\epsilon}{dt} = \frac{\epsilon_{n+1} - \epsilon_n}{\Delta t}. \tag{2.12}$$

This then drives the value of the PID derivative to be as follows:

$$k_d \frac{\epsilon_{n+1} - \epsilon_n}{\Delta t} = k_d \frac{k - 0}{1.00001 - 1}, \tag{2.13}$$

$$= k_d \frac{k}{0.00001}, \tag{2.14}$$

$$= k_d k \times 10^5. \tag{2.15}$$

This is a simplified example of a discrete-time derivative, where a simple step function can generate a large derivative signal that causes significant issues to PID control. △

It can be seen from the example that the derivative generates a large derivative-kick signal. Some remedies for the derivative kick are to generate a smooth set-point input that is not a sharp step-function signal, effectively filtering the set-point input. One can see that in practical applications, the derivative gain becomes slightly more complicated in terms of stability, since the larger the derivative gain, the larger the undesirable side effects that come with the derivative action. This means there is an interval of derivative gain value that the controller can render the system unstable, forming a small interval in which the derivative improves the stability margins for practical applications.

The effects of the PID gains on the feedback system response are important to tune the controller, which becomes one of the fundamental issues of PID control design (Åström and Hägglund, 1995; O'Dwyer, 2009; Krishna et al., 2012; Parnianifard, Fakhfakh et al., 2021; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). In terms of feedback response, the PID controller is intuitive and has robust performance against uncertainties. However, it does also have certain disadvantages and limitations that increase the complexity of the problem. Another complication of PID is the single-degree-of-freedom (1-DoF) structure. This effectively means that the PID controller can only be tuned to maximise either set-point response, where overshoot and rise-time are minimised, or optimise for disturbance rejection, where an overshoot is frequently observed with a larger settling-time (Chen, J., Fang and Ishii, 2019; Parnianifard, Fakhfakh et al., 2021; Somefun, Akingbade and Dahunsi, 2021; So, 2021; Joseph et al., 2022). This can be accommodated by introducing more complicated and advanced tuning algorithms that can find a balance between the disturbance rejection and set-point tracking of the PID (Åström and Hägglund, 1995; O'Dwyer, 2009; Abushawish, Hamadeh and Nassif, 2020). The drawback of such algorithms is that usually the control specifications are deteriorated, in order to compensate for the unavoidable compromise in achieving a balance of the trade-offs (Åström and Hägglund, 1995; O'Dwyer, 2009; Abushawish, Hamadeh and Nassif, 2020; So, 2021).

### 2.5.1 PID Tuning Methods

#### Classical Methods

The tuning procedures of PID control have been extensively studied in the past and are a broad research field. The problem of PID tuning is extremely vast and multiple methods have been developed that use different strategies for tuning. The first methods of PID design used heuristics from designer experience and experimentation, using the plant estimated parameters to assign PID gains through a relationship (Åström and Hägglund, 1995; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). The most popular heuristic methods are the Ziegler-Nichols (ZN), the Cohen-Coon (CC), and the Chien-Hrones-Reswick (CHR) tuning methods (Åström and Hägglund, 1995; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). The Ziegler-Nichols and Cohen-Coon methods use the First Order Plus Time Delay (FOPTD) model to tune the PID gains using relations between $k_p, k_i, k_d$ and $k, t_p, \tau$ to establish the tuning (Åström and Hägglund, 1995; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). Although these two approaches are able to compensate against input disturbances, it is challenging to generate a fast set-point tracking without generating oscillatory responses and overshoots (Åström and Hägglund, 1995; Åström and Hägglund, 2004; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). The CHR tuning method can provide improved performance and allows for two different tuning approaches, one that has 0% overshoot and one that has 20% overshoot (Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). It also allows for the capability to focus on either set-point tracking or disturbance rejection (Åström and Hägglund, 1995; Åström and Hägglund, 2004; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). The main issues with these tuning approaches is that they suffer from deteriorated performance but also they are only useful for a specific type of system problems and new tuning rules have to be re-established for every model (Åström and Hägglund, 1995; Åström and Hägglund, 2004; O'Dwyer, 2009; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022).

#### PSO and Optimization Based Tuning Methods

To resolve the problem of having multiple heuristics, various approaches can be used to tune the PID gains automatically, one of which is using an optimization algorithm (Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). Optimization algorithms search for the optimal response according to a certain objective function criteria, that can then provide the best response (Valluru and Singh, 2018; Gomez et al., 2020; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022; Chaturvedi, Kumar, N. and Kumar, R., 2023). There are various optimization algorithms that can be used for that purpose, spanning two different categories, evolutionary algorithms and swarm algorithms (Valluru and Singh, 2018; Gomez et al., 2020; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). The Genetic Algorithm (GA) and the Simulated Annealing (SA) algorithm have previously been used to effectively tune PID controllers (Dangor et al., 2014; Fraga-Gonzalez et al., 2017; Qin et al., 2019). Moreover, there are swarm optimization algorithms such as the Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) algorithms that have been previously used successfully to tune PID controllers (Saxena and Dubey, 2019; Chaturvedi, Kumar, N. and Kumar, R., 2023). There have also been extensive surveys conducted on the optimization based tuning algorithms that have been used in tuning PID controllers. These surveys have found extensive research on Differential Evolutionary algorithms (DE), Cuckoo Search (CS), Bat, Hybrid Bat (HB), and many more, together with comparisons in their effectiveness and convergence speed to the PID tuning problem (Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022).

The Particle Swarm Optimization (PSO) algorithm has received particular attention in the PID tuning literature due to its simplicity and effectiveness in finding optimal solutions (Chaturvedi, Kumar, N. and Kumar, R., 2023; Joseph et al., 2022; Abushawish, Hamadeh and Nassif, 2020; Shaikh and Yadav, 2022). The PSO algorithm has previously been used to tune PID controllers with multiple applications, including SISO and MIMO systems, showing improved performance, when compared to heuristic and manual approaches (Dangor et al., 2014; Joseph et al., 2022). An adaptive PSO algorithm has also been used to tune both linear and nonlinear PID controllers, improving performance of a highly nonlinear system, when compared to other methods (Valluru and Singh, 2018). In addition, the PSO algorithm has been able to provide optimal tuning to a neural-network based PID controller for a nonlinear continuous stirred tank reactor plant, minimizing the mean squared error, when compared to other control approaches of the same system (Chaturvedi, Kumar, N. and Kumar, R., 2023). A study compared different evolutionary-based algorithms as tuning approaches of PID controllers in two cascaded loops for a vibration suspension control system using an electro-hydraulic actuator, where the PSO algorithm showed improved results in comparison to manual and genetic algorithm approaches (Dangor et al., 2014). The PSO algorithm also has the advantage that one can re-define the algorithm so that it finds the Pareto optimal solution of a multi-objective function, becoming useful for the tuning of PID in MIMO systems (Parnianifard, Zemouche et al., 2020). The multi-objective PSO algorithm managed to provide PID tuning showing good performance, in terms of overshoot, rise time, and root mean square error of a highly nonlinear and MIMO quadrotor UAV (Parnianifard, Zemouche et al., 2020). Moreover, the PSO algorithm can be easily combined with other optimization approaches, creating a hybrid algorithm that can improve the weak features of PSO specific to a problem (Valluru and Singh, 2018). A great example of PSO combined with a genetic algorithm surrogate model has been used to design an FOPID controller to control a five-bar linkage robotic arm in a cyber-physical system (Parnianifard, Zemouche et al., 2020).

#### Gain-Scheduling and Miscellaneous Methods

Gain scheduling strategies have also been used effectively to increase the operational range of linear controllers to nonlinear systems, which is also an effective remedy against robustness trade-offs (Åström and Hägglund, 1995; Kapsalis et al., 2020). However, the drawback is that it takes a long and complex design process of continual analysis and re-design with each design restricted to a single operating point, which are then interpolated to define the gain values across the global phase plane (Åström and Hägglund, 1995; Kapsalis et al., 2020). This procedure increases the design time but also the hardware memory size (Åström and Hägglund, 1995; Kapsalis et al., 2020).

Other common approaches improve performance are the adaptive gains or nonlinear function gains PID control, that establish a semi-adaptive controller by using the system dynamics to express the gains dynamically (Slama, Errachdi and Benrejeb, 2019; Chong et al., 2021; Joseph et al., 2022). This approach allows for a more efficient and robust control approach that improves the performance of the control systems with minimal tuning (Slama, Errachdi and Benrejeb, 2019; Chong et al., 2021; Somefun, Akingbade and Dahunsi, 2021; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Joseph et al., 2022). These control architectures also target the common trade-offs that exist in feedback control, such as set-point tracking and disturbance rejection, without requiring separate tunings for each (Slama, Errachdi and Benrejeb, 2019; Chong et al., 2021; Somefun, Akingbade and Dahunsi, 2021; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Joseph et al., 2022). This in turn means that the tuning approach of such control algorithms requires a single tuning and the system is effectively adapting to the system requirements according to the specifications (Zhang and Mao, 2017; Slama, Errachdi and Benrejeb, 2019; Mo and Farid, 2019; Chong et al., 2021; Somefun, Akingbade and Dahunsi, 2021; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Joseph et al., 2022).

### 2.5.2 Control Design Objectives

The objectives of feedback control systems tend to have different design criteria and as a result are split into two categories of problems for design purposes, which possess trade-offs between them (Åström and Hägglund, 1995; Garpinger, Hägglund and Åström, 2014; Frank, 2018; Bernstein, 2022). The two problems are split into the servo control problem and the regulator control problem.

#### Servo Control

The servo control problem is the transient performance and speed to achieve the set-point. The most common control performance criteria for servo problems are the % overshoot ($\%Os$), rise-time ($t_r$), settling time ($t_s$), % undershoot ($\%Us$), and the system input energy consumption (Åström and Hägglund, 1995; Garpinger, Hägglund and Åström, 2014). The way to compute the system input energy is frequently in terms of $L_2, L_\infty, H_2$, and $H_\infty$ norms of the signals or systems.

### Regulator Control

The regulator control problem is the ability of the feedback system to maintain the desired output for any external disturbance, perturbations, or uncertainties. The control criteria for regulator control are the disturbance rejection, gain margin, phase margin, and the $H_\infty$ norm of the sensitivity and complementary sensitivity functions. Many times feedback systems are prioritised to simply follow the regulator problem specifications, without the need for optimal servo performance criteria (Chen, J., Fang and Ishii, 2019; Bernstein, 2022).

### Performance Criteria

The common control performance criteria are the Integral Absolute Error (IAE), the Integral Squared Error (ISE), and the Integral Time Absolute Error (ITAE), which are as follows (Åström and Hägglund, 1995; Garpinger, Hägglund and Åström, 2014):

$$IAE = \int_0^{t_f} |\epsilon(t)| \mathrm{d}t, \tag{2.16}$$

$$ISE = \int_0^{t_f} \epsilon(t)^2 \mathrm{d}t, \tag{2.17}$$

$$ITAE = \int_0^{t_f} t|\epsilon(t)| \mathrm{d}t. \tag{2.18}$$

These control performance criteria can be used to compute any signal size, including the system input signal and mostly the feedback error of the system. In addition, linearly independent combinations of these performance criteria can also be used to define objective functions for optimization problems, with assigned weights to each criterion. This is a simple way to define the optimal control objectives to use optimization algorithms as part of the design process of the feedback controller (Åström and Hägglund, 1995; Garpinger, Hägglund and Åström, 2014; Wu et al., 2019; Parnianifard and Azfanizam, 2020; Joseph et al., 2022).

## 2.6. The 2-DoF PID Controller

A different approach to this problem is a well-established control algorithm that can be implemented along PID, which is a 2-DoF feedback structure that incorporates the PID control algorithm (Alfaro, Vilanova and Arrieta, 2008; Suthar, 2015; Hirahara et al., 2017; Mohan et al., 2019; Schröders et al., 2020; So, 2021). The 2-DoF PID controller offers the advantage of an additional independently tuned loop that can establish the tuning for optimal disturbance rejection and set-point tracking performance simultaneously (Alfaro, Vilanova and Arrieta, 2008; Suthar, 2015; Hirahara et al., 2017; Mohan et al., 2019; Schröders et al., 2020; So, 2021). The idea behind this structure is to combine the ability of feedback to correct for disturbances and uncertainties without assuming perfect system knowledge and the principle of feed-forward, which presumes a perfect system to compensate for the optimal transient response (Alfaro, Vilanova and Arrieta, 2008; Suthar, 2015; Hirahara et al., 2017; Mohan et al., 2019; Schröders et al., 2020; So, 2021). The 2-DoF PID controller has received a lot of attention and has been extensively used, due to its excellent ability to conduct simultaneous set-point tracking and disturbance rejection, with minimal compromises to performance (Taguchi and Araki, 2000; Suthar, 2015; Mohan et al., 2019; Schröders et al., 2020; So, 2021). The drawback of 2-DoF PID controllers, is that they have a larger number of tunable parameters and they offer a multi-objective optimization problem, that requires a Pareto optimal solution (Taguchi and Araki, 2000; Alfaro, Vilanova and Arrieta, 2008; Suthar, 2015; Wang, X. et al., 2018; So, 2021).

Figure 2.3 shows the two degrees of freedom control structure that utilises an additional feedforward loop as a set-point weighting.

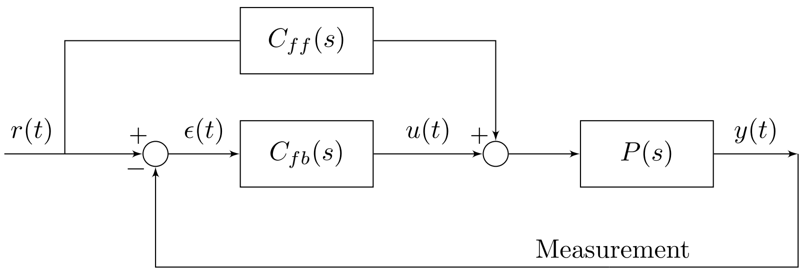



**Figure 2.3:** The 2-DoF PID control structure.

The transfer function of a 2-DoF PID controller is mathematically described as follows (Taguchi and Araki, 2000; Alfaro, Vilanova and Arrieta, 2008; Suthar, 2015; So, 2021):

$$C_{fb}(s) = k_p + k_i \frac{1}{s} + k_d s, \tag{2.19}$$

$$C_{ff}(s) = -k_{p2} - k_{d2} s, \tag{2.20}$$

where $C_{fb}(s)$ is the feedback control structure that is applied to the feedback error, and

$C_{ff}(s)$ is the feed-forward control structure applied to the set-point, which is the set-point weighting function of the 2-DoF structure.

The 2-DoF PID controller also suffers from the same practical disadvantages as the PID controller. The integral part requires anti-windup strategies to eliminate integrator windup in saturated systems. In addition, the derivative of the controller requires filtering to reduce the effects of noise and step functions on the derivative action. The major advantage that the 2-DoF PID structure has over the PID is that it has two independently tuned loops that offer a simultaneous set-point tracking and disturbance rejection response, providing improved robustness.

The 2-DoF PID control structure has shown significant improvements to performance and robustness in both linear and nonlinear systems (Sharma, Gaur and Mittal, 2015; Mohan et al., 2019; Schröders et al., 2020; So, 2021). In addition, it provides an improved system input energy cost when compared against the conventional PID (Mohan et al., 2019; Schröders et al., 2020; So, 2021). Moreover, the drawback with linear control remains with this control strategy, since the linear control of nonlinear systems is limited in operating region, which means that linearization or feedback linearization methods have to be implemented alongside the linear 2-DoF PID controller (Schröders et al., 2020).

## 2.7. Smith-Predictor PID Controller

One of the most common nonlinearities that exist in industrial systems is the time delay (Frank, 2018; Normey-Rico, Santos et al., 2022). Time delay in systems can be caused by multiple sources or side-effects, some of which are the following (Normey-Rico and Camacho, 2007; Frank, 2018; Normey-Rico, Santos et al., 2022).

- Process delays caused by energy, mass, or information transportation.
- Processing time and Sensor delays.
- Actuator response time.
- Apparent time-delay from a series of systems that possess lag dominant dynamics.

These physical properties exist within any system and effectively all industrial systems have some form of delay in their processes. These delay dynamics are most often modelled as a FOPTD or SOPTD system transfer function model of the form (Normey-Rico and Camacho, 2007):

$$P_1(s) = T(s)e^{-\tau s} = \frac{k_p e^{-\tau s}}{(t_p s + 1)}, \tag{2.21}$$

$$P_2(s) = T(s)e^{-\tau s} = \frac{k_p e^{-\tau s}}{(t_{p_1} s + 1)(t_{p_2} s + 1)} = \frac{k_p e^{-\tau s}}{1 + \frac{2\xi}{\omega_n} s + \frac{s^2}{\omega_m^2}}. \tag{2.22}$$

This set of dynamics effectively delays the knowledge of the state to the controller, which causes the feedback error to be that of the previous output. This then causes the controller to have a delayed response to the system, causing a large deterioration in performance and many times even instabilities (Normey-Rico and Camacho, 2007; Frank, 2018; Normey-Rico, Santos et al., 2022). This can be viewed in the frequency domain analysis as a decrease in the system phase. Transforming the delay dynamics into the transfer function form helps improve the understanding of the delay dynamics in the frequency domain, but also simplifies the analytical process. The delay dynamics can be effectively approximated as a transfer function, which is commonly used in control design, using the Pade approximation. The most common Pade approximations are the following first-order and second-order transfer functions, respectively.

$$G_{p_1}(s) = e^{-\tau s} = \frac{2 - \tau s}{2 + \tau s}, \tag{2.23}$$

$$G_{p_2}(s) = e^{-\tau s} = \frac{\tau^2 s^2 - 6\tau s + 12}{\tau^2 s^2 + 6\tau s + 12}. \tag{2.24}$$

Time delay dynamics complicate the design and control procedure, however, the PID controller is a very effective control strategy that is commonly used to control systems with time-delay (Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). A method to compensate time delays has been developed by Otto Smith since 1957, and it has been effectively used in industry to provide improved responses to the systems (Hung, Yu and Cheng, Y.-C., 2004). The Smith predictor uses a model of the expected delay dynamics of the plant and uses these dynamics to estimate the control signal for the nominal system without time delay (Hung, Yu and Cheng, Y.-C., 2004; Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). This method is combined with a classical control algorithm, which is often using the PID controller, making the system more effective. The Smith predictor PID controller has the transfer function form as follows:

$$K_{\mathrm{SP\_PID}}(s) = \frac{K_{\mathrm{PID}}(s)}{1 + K_{\mathrm{PID}}(s)T(s)(1 - G_{p_2}(s))}. \tag{2.25}$$

The major advantage that the Smith predictor provides to systems with time delay is that the classical control algorithm can be tuned much more simply, by using a classical tuning algorithm (Hung, Yu and Cheng, Y.-C., 2004; Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). Effectively, the smith predictor allows for the PID controller to be tuned for the system with no time-delay, making it much more simple and effective. However, the drawback of using Smith Predictor PID (SP_PID) controller is that the predicted time-delay is not always the same as the plant time-delay due to estimation errors and variations in delay (Hung, Yu and Cheng, Y.-C., 2004; Normey-Rico and Camacho, 2007; Normey-Rico and Camacho, 2008; Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). The major drawback of this is that the smith predictor can introduce instability to the feedback system, although the PID controller has been tuned to maintain large gain and phase margins for the delay-free system (Normey-Rico and Camacho, 2007; Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). This is caused because the delay dynamics introduce new robustness criteria, which is the delay margin, which has to be considered (Normey-Rico and Camacho, 2007; Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). Moreover, if the open loop system possesses slow dynamics, the input disturbances are compensated very slowly and the SP_PID controller has a 1-DoF control structure (Hung, Yu and Cheng, Y.-C., 2004; Normey-Rico and Camacho, 2007; Normey-Rico and Camacho, 2008; Sigurd Skogestad, 2018; Frank, 2018; Normey-Rico, Santos et al., 2022). Effectively, the SP_PID controller can only be tuned to have optimal performance or disturbance rejection response for any feedback system. This performance trade-off is worsened as time-delay increases (Normey-Rico, Santos et al., 2022).

Different control strategies have also been developed to improve the response against disturbance rejection, such as a disturbance compensator SP_PID controller or a 2-DoF SP_PID controller structure (Normey-Rico and Camacho, 2008). Moreover, control strategies where a frequency-dependent gain is introduced to give different weights to the prediction and to the control compensation of the feedback system, to reduce the effects of unmodelled time-delay dynamics to the predictor (Hung, Yu and Cheng, Y.-C., 2004; Normey-Rico and Camacho, 2008). To resolve the performance issues different tuning algorithms have also been provided to optimise servo control or regulatory control of delay systems (Normey-Rico and Camacho, 2007; Normey-Rico and Camacho, 2008).

## 2.8. Nonlinear PID Control

Less computationally expensive adaptive control methods can also be achieved by defining variable gains that are expressed in terms of feedback parameters. This effectively computes in real-time a pre-defined relationship between the gain parameters and the feedback variables to establish the tuning. A very common control algorithm that uses this structure is the PID controller, where the gains are expressed as nonlinear functions instead of constants (Zaidner et al., 2010; So and Jin, 2018; Pathak, Bhati and Gaur, 2020; Hua et al., 2020; Chong et al., 2021; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). This is an extremely effective method because PID controllers are well-established in industry, they are effective and simple, which makes them ideal candidates for nonlinear gain adaptation. Nonlinear PID controllers are less reliant on the accuracy of the mathematical model and hence, they are more robust against uncertainty (Zaidner et al., 2010). As a result, in applications where performance and robustness improvements are desirable, nonlinear PID offers a simple alternative to the PID control (Zaidner et al., 2010; Hua et al., 2020). Nonlinear PID has had a large expanse and on-going research literature, where improvements in performance and robustness are seen, without the need for expensive hardware (So and Jin, 2018; So, 2019; Jin and Son, 2019; Pathak, Bhati and Gaur, 2020; Hua et al., 2020; Chong et al., 2021; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021; Son, Bin and Jin, 2021; Liu, T., 2022; Shamseldin, 2023; Chaturvedi, Kumar, N. and Kumar, R., 2023; Sivadasan et al., 2023). This expanse in the research literature is because of its effectiveness in performance and its ease of use in hardware (Zaidner et al., 2010; Hua et al., 2020).

### Nonlinear PID Control with Tracking Differentiators

A method for nonlinear PID control was formulated in 1994 (Han, 1994). The controller was implemented for nonlinear systems and it used nonlinear combination PID control with two tracking differentiators (Han, 1994). Figure 2.4 shows the control scheme developed by Han, using one nonlinear tracking differentiator at the set-point signal to shape the form of the set-point and a nonlinear tracking differentiator in the feedback to estimate the derivative and state (Han, 1994).

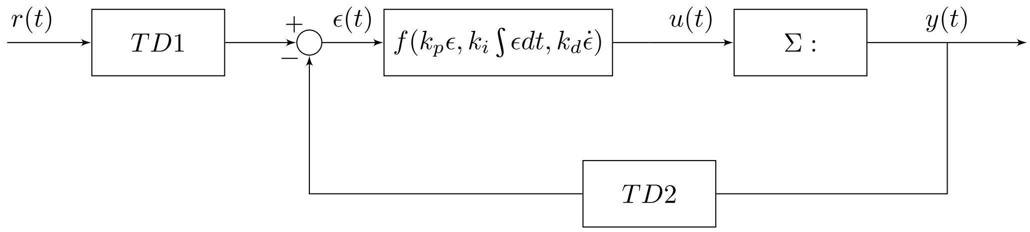



**Figure 2.4:** The system of a nonlinear PID controller using tracking differentiators as depicted in the paper (Han, 1994).

This controller has shown significant improvements in performance and robustness of the control of nonlinear systems with the trade-offs minimised (Han, 1994). This nonlinear PID controller has been used in many applications and research papers, where improvements in performance and robustness have been shown for both linear and nonlinear systems (Rakesh, Satheesh and Thirunavukkarasu, 2014; Kasim and Riyadh, 2016; Valluru, 2017; Liu, T., 2022). A drawback of using the nonlinear tracking differentiator is that it creates a chattering effect and noise reduces the quality of the derivative estimation significantly (Kasim and Riyadh, 2016). This drawback has received some research attention where improved tracking differentiators aim at improving the chattering effect and noise rejection, where this has shown an apparent improvement in the performance of the system (Su, Sun and Duan, 2005; Kasim and Riyadh, 2016). Although this nonlinear PID control algorithm is robust and offers significant improvements in control performance for linear and nonlinear systems, it requires a complex tuning problem (Rakesh, Satheesh and Thirunavukkarasu, 2014; Valluru, 2017; Liu, T., 2022). This complexity in tuning comes from the multi-objective optimization problem that the nonlinear 2-DoF structure of the controller offers, but also from the large number of parameters to be tuned (Rakesh, Satheesh and Thirunavukkarasu, 2014; Valluru, 2017; Liu, T., 2022).

### Lure-type NLPID controllers using Circle and Popov Criteria

There is also a special type of nonlinear control, which can be transformed into the frequency domain analysis that makes their study much simpler. Assuming that a linear system is a feedback interconnected with nonlinear memoryless dynamics (i.e. it only depends on the present state and input) then the system can be transformed into the following definitions (Slotine and Li, 1991; Khalil, 2002; Frank, 2018):

**Definition 2.8.1** (Memoryless Sector Bounded Nonlinearity (Khalil, 2002))**.** A memoryless function $\phi : [0, \infty) \times R^n \to R^n$ is said to belong to the sector:

- $[0, \infty]$ if $u^T \phi(t, u) \geqslant 0$.
- $[K_1, \infty]$ if $u^T [\phi(t, u) - K_1 u] \geqslant 0$.
- $[0, K_2]$ with $K_2 = K_2^T > 0$ if $\phi^T(t, u)[\phi(t, u) - K_2 u] \leqslant 0$.
- $[K_1, K_2]$ with $K = K_2 - K_1 = K^T > 0$ if $[\phi^T(t, u) - K_1 u]^T [\phi(t, u) - K_2 u] \leqslant 0$.

The inequality should hold for all $(t, u)$, in all cases. For any case that the equality does not hold, it is said that the interval is open at that bound of the sector, eg. $(0, \infty)$,$(K_1, \infty)$, $(0, K_2)$, and $(K_1, K_2)$.

**Definition 2.8.2** (Lure System)**.** A system that can be described as a linear function with a nonlinear memoryless function in its feedback, as shown by Figure 2.5, can be described mathematically as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \\ u = -\phi(t, y), \end{cases} \tag{2.26}$$

with a minimal realisation expressed as a transfer function $G(s) = C(sI - A)^{-1}B + D$.

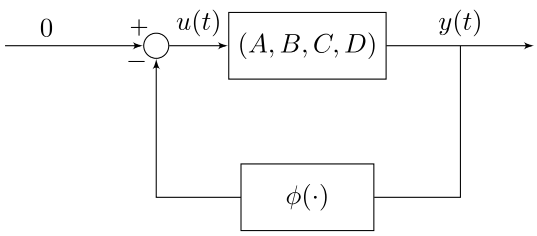



**Figure 2.5:** Schematic of a simple Lure system.

Such nonlinear control cases can be transformed into a linear system, which can be studied under the same tools as linear systems. The most prominent methods of stability analysis are the Popov and Circle criteria (Slotine and Li, 1991; Khalil, 2002). The underlying assumptions that play an important role are that the nonlinearity is memoryless and sector-bounded within a region, which then forms the stability bounds in the Nyquist plots (Slotine and Li, 1991; Khalil, 2002). The Circle theorem is then stated as follows:

**Theorem 2.8.1** (Circle Criterion - Scalar Case (Khalil, 2002))**.** Consider a scalar system of the form Eq. (2.26), where $(A, B, C, D)$ is a minimal realization of $G(s)$ and $\phi \in [a, b]$. Then, the system is absolutely stable if one of the following conditions is satisfied, as appropriate:

- If $0 < a < b$, the Nyquist plot of $G(j\omega)$ does not enter the disk $D(a, b)$ and encircles it m times in the counterclockwise direction, where m is the number of unstable poles of $G(s)$.
- If $0 = a < b$, $G(s)$ is Hurwitz and the Nyquist plot of $G(j\omega)$ lies to the right of the vertical line defined by $Re[s] = -1/b$.
- If $a < 0 < b$, $G(s)$ is Hurwitz and the Nyquist plot of $G(j\omega)$ lies in the interior of the disk $D(a, b)$.

If the sector condition is satisfied only on an interval $[a, b]$, then the foregoing conditions ensure that the system is absolutely stable with a finite domain.

Moreover, Popov criteria have a further simplification that the dynamics are of:

$$\dot{x}(t) = Ax + bu, \tag{2.27}$$

$$\dot{\xi}(t) = u, \tag{2.28}$$

$$y(t) = cx(t) + d\xi(t), \tag{2.29}$$

$$u = -\phi(y(t)), \tag{2.30}$$

with $\phi \in [0, b]$, which means that the sector boundedness property of the nonlinear function is further constrained and less flexible (Slotine and Li, 1991; Khalil, 2002). The Circle criterion is a generalised theorem that only requires the nonlinearity to be bounded within two lines as $\phi \in [a, b]$ (Slotine and Li, 1991; Khalil, 2002). This makes the Circle criterion much more applicable and broad in terms of types of nonlinearities. Such system expressions are extremely convenient and are frequently used to express saturation nonlinearity in that form, or the pendulum system can also be expressed in that form (Slotine and Li, 1991; Khalil, 2002). Moreover, linear controllers and linear systems can be combined with a memoryless nonlinear feedback gain, that can improve control performance and the stability properties can be easily studied using the Circle and Popov criteria (Khalil, 2002; Rezaei and Hashemzade, 2016; Son, Bin and Jin, 2021).

A well-established method of stability is using the Lure systems as a ground for formulating a nonlinear-based PID controller. This is to establish a nonlinear gain that is memoryless and sector-bounded, as depicted in Figure 2.6, where the use of the Circle or Popov criteria can establish a stability-proof method and improvements to performance criteria. Based on these stability criteria there have been some nonlinear gains that were studied in their performance improvements and robustness with the following gains studied (Maddi, Guessoum and Berkani, 2014):

$$\phi(\epsilon) = k_0(b)^{\epsilon}, \tag{2.31}$$

$$\phi(\epsilon) = k_0\frac{e^{\alpha\epsilon} + e^{-\alpha\epsilon}}{2} = k_0 cosh(\alpha\epsilon), \tag{2.32}$$

and

$$\phi(\epsilon) = \begin{cases} k_0 e^{\alpha\epsilon} & \text{, if } \epsilon > \epsilon_{max}, \\ k_0 e^{-\alpha\epsilon} & \text{, if } \epsilon < \epsilon_{max}, \end{cases} \tag{2.33}$$

where $\epsilon$ is the feedback error, $\epsilon_{max}$ is a designer-defined maximum error constant value, and constants $k_0, \alpha, b \in \mathbb{R}$. Figure 2.6 shows a schematic diagram of the general nonlinear PID control structure that can be transformed into the classic Lure System.

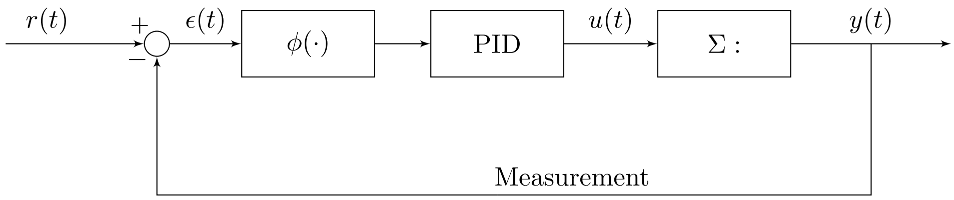



**Figure 2.6:** A schematic of a Lure type nonlinear PID controller.

The nonlinear functions are sector-bounded memoryless gains, used in cascade with a linear PID controller for linear plant models (Maddi, Guessoum and Berkani, 2014). These nonlinear gains were found to have robust stability and provide a strong foundation in theory for the stability analysis of the feedback interconnection (Maddi, Guessoum and Berkani, 2014). These gains were shown to provide improved performance with lower rise-time and settling-time when compared against the linear PID control (Maddi, Guessoum and Berkani, 2014). However, there were oscillatory behaviour and the stability criteria provide a conservative analysis, where better tuning can be established without instability (Maddi, Guessoum and Berkani, 2014).

The nonlinear function described by Eq.(2.32) has been used extensively as a cascaded gain with a combination of different PID control structures due to its foundational method based on established stability tools (Pathak, Bhati and Gaur, 2020; Shamseldin, 2023). Some of these methods include but are not limited to:

1. A fractional order nonlinear PID (FONLPID) control structure has been used with the

nonlinear gain applied to the integral action, which shows significant improvements in performance and robustness of controlling a nonlinear wind energy system (Pathak, Bhati and Gaur, 2020).

2. MIT adaptation rule, where the nonlinear PID controller has a set of nonlinear function gains, using the Eq.(2.32) in addition to a constant gain, which was applied for tracking control of a nonlinear electric vehicle to minimise the energy consumption of electric current and voltage (Shamseldin, 2023).

Both methods showed that they were able to significantly improve the energy efficiency of the system, and improved the rise-time and settling-time of the tracking control problem (Pathak, Bhati and Gaur, 2020; Shamseldin, 2023). However, they have a large number of tuning parameters which makes them highly computationally expensive for tuning purposes (Pathak, Bhati and Gaur, 2020; Shamseldin, 2023).

Another NLPID controller designed on the basis of the Circle criteria tools has been shown in the literature, where the control algorithm is as follows (Son, Bin and Jin, 2021):

$$u(t) = k_p\epsilon(t) + k_i\int_0^{t_f} \epsilon(t)e^{\left[-\dfrac{\epsilon(t)^2}{2(\delta r)^2}\right]}\,\mathrm{d}t + k_d\dot{\epsilon}(t), \tag{2.34}$$

where $\delta r$ is the change in set-point input, $\epsilon(t)$ is the feedback error, $k_p, k_i,$ and $k_d$ are the classic PID gains.

The nonlinear controller has been benchmarked against the linear PID controller using various tuning methods for delay type systems (Son, Bin and Jin, 2021). The nonlinear controller stability and robustness were shown for each delay system using a loop transformation and the circle criteria (Son, Bin and Jin, 2021). The nonlinear controller showed improved performance with lower rise-time and settling-time when compared against Internal Model Control (IMC) PID and conventional PID methods (Son, Bin and Jin, 2021). The controller also has a low number of tunable parameters, which makes it effective and simple to tune (Son, Bin and Jin, 2021). This has also been shown in the literature where simple tuning rules for FOPTD systems have been established using optimisation and internal mode control approaches for the nonlinear controller (Jin and Son, 2019). However, the drawback of the controller is that it tends to have overshoot and oscillatory-type behaviour, where the control performance is not optimised.

#### Model-Based and Lyapunov Design Approach NLPID

Model-based approach design and the use of Lyapunov theory for establishing nonlinear control has been used in the past as a combination with PID control (Hua et al., 2020;

Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). A nonlinear PID controller with an auto-tuning mechanism using a learning approach has also been designed and used for the control of highly nonlinear control of rotorcraft under aggressive manoeuvring (Hua et al., 2020). This control approach has shown significant improvements in performance lowering the settling-time and rise-time (Hua et al., 2020). It has also shown experimental results where the control approach was implemented in the hardware of a rotorcraft (Hua et al., 2020). The nonlinear controller was designed using Lyapunov-based analytical stability analysis proving the asymptotic stability of the controller (Hua et al., 2020). Model-based nonlinear PID control was also used for the control of a nonlinear CSTR system where a modified artificial bee colony algorithm was used (Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). The derived NLPID parameters combined a model-based tuning approach with fuzzy logic fusion and local model PID (Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). The NLPID controller demonstrated improved servo and regulatory control of the nonlinear CSTR system at various operating points (Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). This has expanded the operational range when compared to a linear PID controller and has shown improved results.

The advantage that these methods of control design provide is that they offer a rigorous set of mathematical toolkits that help develop the control algorithm with guaranteed stability approaches (Mo and Farid, 2019; Hua et al., 2020; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). However, the drawback is that they become complicated for the implementation and design from an engineering point of view since they are model dependent control algorithms and the extensive and rigorous mathematical background becomes necessary for control engineers (Hua et al., 2020; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). This may not be ideal for industrial applications and engineers, but the model-based approach provides a robustness drawback due to modelling uncertainties and inaccuracies (Mo and Farid, 2019; Hua et al., 2020; Cheng, L., 2021; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021).

#### Nonlinear Function Gains PID (NLPID)

Nonlinear PID controllers and nonlinear function gains PID controllers are separated in name according to the nature of their design. The former utilises the inherent nonlinearities from the model dynamics that are to be controlled as an aspect of the control system coupled with a linear PID controller. Whereas the latter utilises nonlinear functions to compute the gains of the PID controller in terms of either the feedback error or feedback error rate. Figure 2.7 shows a typical design of a nonlinear gains PID controller where the gains become a set of nonlinear functions which compute in real time the PID gains.

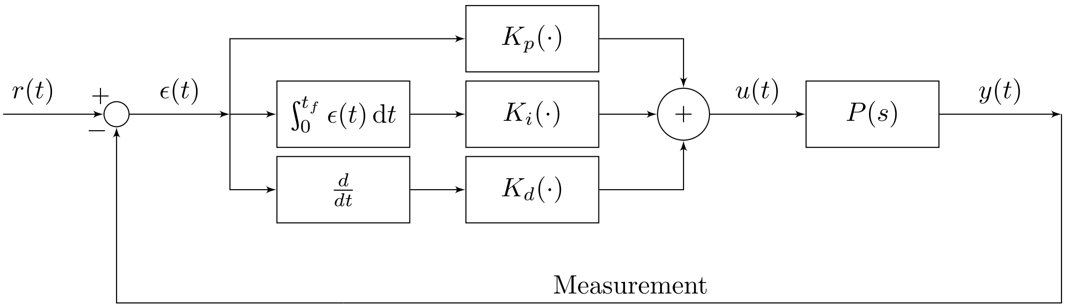



**Figure 2.7:** A schematic of nonlinear function gains PID controller structure.

A nonlinear function gains PID controller has been compared against a Genetic Algorithm (GA) tuned PID controller for a variety of linear systems (Korkmaz, Aydoğdu and Doğan, 2012). The structure of the nonlinear controller was proposed to be as follows (Korkmaz, Aydoğdu and Doğan, 2012):

$$u(t) = K_p(\epsilon) + \frac{K_i(\epsilon)}{s} + K_d(\epsilon)s, \tag{2.35}$$

with the nonlinear gains formulated using the nonlinear functions as (Korkmaz, Aydoğdu and Doğan, 2012):

$$K_p = a_1 + f(\epsilon)a_2, \tag{2.36}$$

$$K_i = b_1 - f(\epsilon)b_2, \tag{2.37}$$

$$K_d = c_1 + f(\epsilon)c_2, \tag{2.38}$$

$$f(\epsilon) = \frac{2}{\sqrt{\pi}} \int_0^{\epsilon} e^{-t^2}\,\mathrm{d}t. \tag{2.39}$$

This nonlinear PID controller showed weakness in establishing strong performance improvements, having a slow settling time and an oscillatory response (Korkmaz, Aydoğdu and Doğan, 2012). The GA-tuned PID controller showed better performance and a comparison against a Ziegler-Nichols tuned PID controller showed that for the benchmarked systems the nonlinear PID controller showed improved performance against it (Korkmaz, Aydoğdu and Doğan, 2012). However, it is well known in the literature that the Ziegler-Nichols tuning procedure is not effective in establishing performance and it is limited to specific systems (Åström and Hägglund, 1995; Åström and Hägglund, 2004; Somefun, Akingbade and Dahunsi, 2021; Joseph et al., 2022). The number of control parameters is high and this requires a larger computational time and effort for the tuning process. Moreover, stability analysis of the nonlinear PID controller was also not established, which forms a limitation in terms of understanding controller behaviour in the feedback loop.

The literature indicates that there is extensive use of nonlinear combinations of the PID parameters and nonlinear control utilising a sector-bounded gain function in cascade with a PID controller to improve performance. However, these control methods do not minimise overshoot and often have large settling and rise times. Another variant of nonlinear PID controllers in the literature has utilised a unique set of nonlinear function gains, where the controller had the following structure (So, 2019):

$$u(t) = K_p(\epsilon) + \frac{K_i(\epsilon)}{s} + \frac{K_d(\epsilon, \dot{\epsilon})s}{1 + T_f(\epsilon, \dot{\epsilon})s}, \tag{2.40}$$

with the proportional and integral nonlinear gain functions represented as (So, 2019):

$$K_p(\epsilon) = k_p(1 - \frac{1}{a_p + (c_p\epsilon)^6}), \tag{2.41}$$

$$K_i(\epsilon) = k_i(\frac{1}{1 + (c_i\epsilon)^6}), \tag{2.42}$$

where $a_p, c_p,$ and $k_p$ are the tunable proportional parameters and $c_i$ and $k_i$ are the integral tunable parameters. In addition, the nonlinear derivative gain is:

$$K_d(\epsilon) = \begin{cases} k_d(1 - \frac{1}{a_d + (c_d\epsilon)^6}) & \text{, if } \epsilon\dot{\epsilon} > 0, \\ k_d(1 - \frac{1}{a_d}) & \text{, elsewhere ,} \end{cases} \tag{2.43}$$

where the parameters $a_d, c_d,$ and $k_d$ are the tunable derivative gain parameters. Finally, the nonlinear function $T_f$ is the nonlinear gain that filters derivative signals in a manner similar to the linear PID control approach that utilises a low-pass filter. The function is mathematically expressed as:

$$T_f(\epsilon, \dot{\epsilon}) = \frac{K_d(\epsilon, \dot{\epsilon})}{NK_p(\epsilon, \dot{\epsilon})}, \tag{2.44}$$

where $N$ is a filtering parameter that is designed as appropriate and usually takes the value of $N = 10$.

This controller utilises the limitations of the PID controller performance and correspondingly shapes the gains as to improve the performance, with a different shape for each action (So, 2019). The NLPID controller was benchmarked against the nonlinear PID controller Eq.(2.35) by Korkmaz and a linear PID controller for an FOPTD and a third order systems (So, 2019). The system also included saturation in the system and the NLPID controller was tuned for the saturated systems using an evolutionary algorithm optimizer (So, 2019). The NLPID controller showed improved performance with faster rise-time and settling-time (So, 2019). Moreover, there were improvements in robustness and noise rejection in the feedback system when compared to the conventional methods. The system also included saturation in the system and the NLPID controller was tuned for the saturated systems (So, 2019). The NLPID controller has also been used for the control of a nonlinear CSTR system, where the NLPID controller was combined with the Tagaki-Sugeno fuzzy logic approach (So and Jin, 2018). The NLPID controller increased the operational range of the linear PID controller for the nonlinear system (So and Jin, 2018). In addition, it showed improved performance with no overshoot and disturbance rejection against input and output disturbances (So and Jin, 2018). However, the drawback of this controller is that it too has a large number of tunable parameters that complicate the tuning procedure and increase computational time (So and Jin, 2018).

This controller circumvents the common limitations of the PID controller by adopting a set of unique nonlinear gains according to the feedback error and error rate (So, 2019). The nonlinear controller was tuned using an evolutionary optimisation algorithm and was benchmarked against the nonlinear PID controller Eq.(2.35) proposed by Korkmaz (So, 2019). The systems simulated in the benchmarking were a First Order Plus Time Delay (FOPTD) system and a third-order system, both including a saturation to the input signal (So, 2019). The NLPID controller showed improved performance with faster rise-time and settling-time (So, 2019). Moreover, there were improvements in robustness and noise rejection in the feedback system when compared to the conventional methods (So, 2019). The nonlinear controller has also been used to control a nonlinear CSTR system with the addition of the Tagaki-Sugeno fuzzy logic approach (So and Jin, 2018). The fuzzy logic implementation of the nonlinear controller showed an improvement in operational range when compared against the conventional PID controller (So and Jin, 2018). In addition, it showed improved performance with no overshoot and disturbance rejection against input and output disturbances (So and Jin, 2018). However, the drawback of this controller is that it has many tunable parameters that complicate the tuning procedure and increase computational time (So and Jin, 2018).

**2-DoF NLPID**

The two-degree-of-freedom control structure has advantages over the single-degree-of-freedom structure that provides simultaneous performance and disturbance rejection to feedback systems. This approach has also been combined with nonlinear function gains for optimizing and improving its performance (Rakesh, Satheesh and Thirunavukkarasu, 2014; Chong et al., 2021). A two-degree-of-freedom NLPID control structure has been utilised recently for the control of highly nonlinear and inherently unstable dynamics of a magnetic levitation system (Chong et al., 2021). The 2-DoF NLPID controller utilised nonlinear functions in the feed-forward, a PI controller in the open loop and a PD controller in feedback as a disturbance compensator combined with nonlinear function gains (Chong et al., 2021). The 2-DoF NLPID controller did not show a sufficient performance in the positioning response of the magnetic levitation system (Chong et al., 2021). However, when the model-based approach was used for the feed-forward solution the positioning response improved and reduced the overshoot (Chong et al., 2021). The disturbance compensator also lowers the sensitivity of the system and improves the robustness (Chong et al., 2021).

## 2.9. Literature Gap and Research Impact

It is well known that PID controllers are implemented in more than 90% of industrial control applications and they are important control algorithms that usually suffer in performance. Improving the capabilities and the performance of the PID controllers can significantly impact a large portion of industrial applications. This, in turn, can increase automation in a time when automatic systems are on the rise. Nonlinear PID control is a simple and robust adaptation method that does not rely on model identification methods and it can be model-free control. Nonlinear systems are more accurate representations of systems and can provide better longevity and performance. It is also easy to implement in hardware making it an effective and efficient control algorithm that applies to industrial applications and can provide a better alternative to the PID control. As a result, literature on the advancements in the study of nonlinear control and the effectiveness of combining nonlinear control strategies with PID were studied to identify the current ground in the field. According to the studied literature and the extensive critical analysis of the materials the gaps in the literature are identified as follows:

- Lack of simple model-free nonlinear control algorithms that can increase the operational region of PID for nonlinear systems.
- Lack of nonlinear PID control approaches addressing the common PID limitations in simultaneous performance and robustness of nonlinear systems.
- Identifying a set of nonlinear functions that adapt the PID parameters to ameliorate the trade-off in performance and robustness in both linear and nonlinear systems.
- Lack of nonlinear PID controllers that are practical in industry and offer a simple tuning problem and the number of tunable parameters kept low.

# Chapter 3

# A Novel Nonlinear PID Controller and a Tuning Algorithm

## 3.1. Introduction

The research literature shows that nonlinear PID controllers become complicated algorithms that are time-consuming to design and difficult to implement in practice. These issues are resolved in this chapter, to meet the research objectives 1-3. Firstly, a novel Nonlinear PID (NLPID) controller is designed to accommodate for the common limitations of the conventional PID controller. The proposed controller structure and the nonlinear functions are shown. The proposed NLPID controller has fewer parameters to tune, which makes it simple and less time-consuming to design. A tuning approach is also shown for the proposed NLPID controller parameters. Moreover, the proposed NLPID controller has potential practical advantages, due to the consideration of the effects of derivative estimation on the controller and the simplicity needed for digital implementation, during design. Finally, some demonstration examples are presented where the practical advantages of the proposed controller when compared to conventional methods are shown.

## 3.2. Problem Statement

The problem considered in this chapter is the design of a novel NLPID controller and a tuning method of the proposed NLPID gains to establish improved control performance of nonlinear systems with minimal trade-offs in robustness. The proposed NLPID controller has independently varying gains that adapt in terms of the feedback error and feedback error rate. The proposed controller also has set-point adaptive gains that match the correct feedback error interval for adaptation and work for all set-point inputs. In addition, it shows potential for mitigating common implementation issues of a conventional PID controller. The equation of the proposed NLPID controller follows a similar format to that of a parallel linear PID controller, the difference being that the gains vary according to nonlinear functions that depend on the feedback error and error rate. The time-domain NLPID controller equation is given by:

$$u_{NLPID}(\epsilon(t), \dot{\epsilon}(t), r(t)) = k_{\mathrm{p}}(\epsilon(t), r(t))\epsilon(t) + k_{\mathrm{i}}(\epsilon(t), r(t)) \int_0^{t_{\mathrm{f}}} \epsilon(t)\,\mathrm{d}t + k_{\mathrm{d}}(\dot{\epsilon}(t), r(t))\dot{\epsilon}(t). \tag{3.1}$$

Figure 3.1 shows the design architecture of the proposed NLPID controller with the separate nonlinear function gains for each PID action.

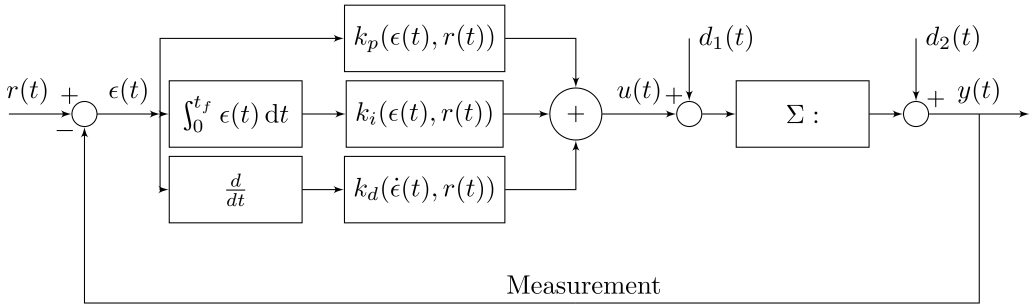



**Figure 3.1:** A schematic of the proposed NLPID controller structure.

The key properties that these functions must hold and have in common are the following:

- The functions must be positive and semi-definite.
- The functions must be even and symmetric along the y-axis.

These two properties are requirements for gain to be applied to the magnitude of the error regardless of the direction. In addition, the two properties are important for the feedback stability that PID controllers require.

The proposed nonlinear parallel structure has been designed due to its flexibility and ability to reduce derivative kicks and have a form of control between all parameters of the PID controller. The nonlinear gains vary according to the feedback error and they scale according to the set-point that enters into the feedback system so that the nonlinearities can act at the time of transient response. The following section is an extensive discussion of the selected nonlinear function gains, their uses, and their advantages.

## 3.3. Nonlinear function gains

The proposed NLPID controller is developed to generate fast set-point tracking with a low overshoot and a fast disturbance rejection. Under these requirements, the conventional PID gains, which most influence the overshoot negatively, are the proportional and integral gains. To reduce the limitations of conventional PID, nonlinear functions are designed to adjust the gains in terms of feedback error and error rate. The proportional gain must be large to produce a fast response, but if it is maintained large during steady-state, then overshoot can occur. As a result, the proportional gain is rapidly reduced near the steady-state to minimise overshoot and maintain a fast response. In addition, a large integral gain can reduce the steady-state errors if maintained large near the steady-state. However, the integral gain can also increase oscillations and reduce the system performance. As a result, the integral gain is low and is rapidly increased near the steady-state. This provides a tuning for fast response and corrects the steady-state errors.

The derivative gain takes a similar form to the integral. However, the derivative gain is in terms of the error rate, so once the error rate becomes rapid, the derivative gain observes the increase and reduces to zero. When the derivative gain reduces to zero, the large error rates are cancelled by the gain to reduce the derivative kicks.

This behaviour of PID control is well known within the literature. As a result, the proposed NLPID controller has been designed with a set of nonlinear functions that have these properties. A nonlinear function that processes such design requirements can be represented as the mollifier function that originates from distribution theory and has not been used in the past in the NLPID control literature. The mollifier function is defined as follows.

**Definition 3.3.1** (The Mollifier Function (Lawrence, 1998)). Define $M \in C^{\infty}(\mathbb{R}^n)$ be the mollifier function and expressed as

$$M(x(t)) := \begin{cases} cexp\left(\dfrac{1}{|x(t)|^2 - 1}\right) & \text{if } |x(t)| < 1, \\ 0 & \text{if } |x(t)| \geqslant 1, \end{cases} \tag{3.2}$$

with a constant $c \in \mathbb{R} > 0$, selected such that $\int_{\mathbb{R}^n} M(x(t))\, dx = 1$.

The mollifier is a $C^{\infty}$ continuous and differentiable function that does not possess any singularities or asymptotes (Showalter, 1994; Lawrence, 1998; Crespi, La Torre and Rocca, 2003). It is also a positive semi-definite even function (Showalter, 1994; Lawrence, 1998; Crespi, La Torre and Rocca, 2003). The mollifier is also contained within low and upper bounds of the interval $0 \leqslant M(x(t)) \leqslant cexp(-1), \forall x(t) \in \mathbb{R}$, with time $t \geqslant 0$ (Showalter, 1994; Lawrence, 1998; Crespi, La Torre and Rocca, 2003). These mathematical properties of the mollifier function are desirable to make the nonlinear gains smooth functions, which has the benefit that the functions and their derivatives are well-defined at all points. In addition, it can be seen that the Mollifier function is mathematically similar to the normal distribution, which can be a potential nonlinear function candidate as a replacement to the Mollifier. However, normal distributions approach zero at infinity, which is a disadvantage since the function is defined as a piecewise function, meaning that using a normal distribution will cause a discontinuity. Hence, a piecwise normal distribution cannot be used without having an undefined point at zero. Moreover, the mollifier function has smoothing properties of input signals, which other functions do not possess. Finally, in this research, the mollifier is adopted to define the nonlinear proportional gain to maximise the effect of the nonlinearity for the minimization of overshoot. The adopted nonlinear gains for the proposed NLPID controller are hence described and shown as follows.

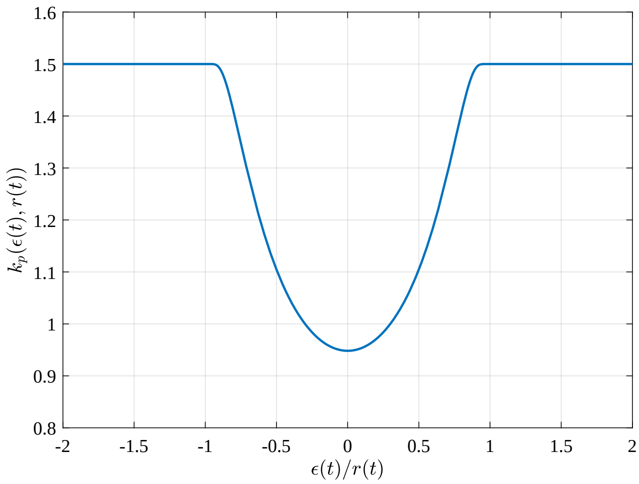
### 3.3.1 Nonlinear Proportional Function Gain

The proportional nonlinear gain is represented as the following function and shown in Figure 3.2:

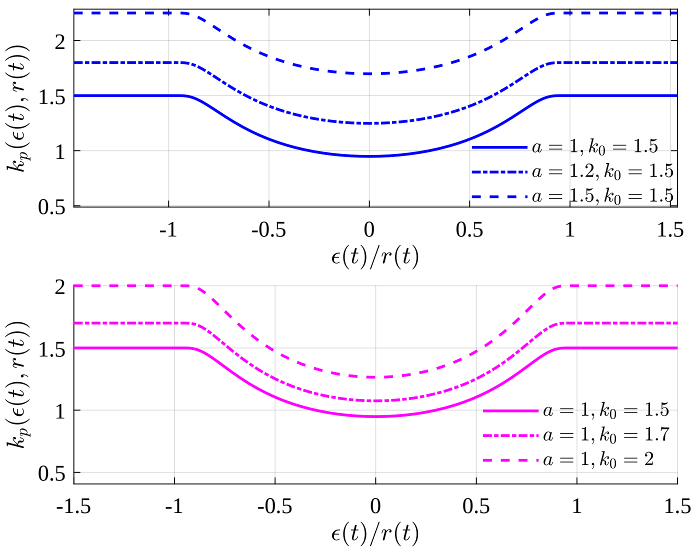
$$k_p(\epsilon(t), r(t)) = \begin{cases} ak_0 - k_0 exp\left(\dfrac{1}{\left|\frac{\epsilon(t)}{r(t)}\right|^2 - 1}\right) & \text{if } \left|\dfrac{\epsilon(t)}{r(t)}\right| < 1,\ r(t) \neq 0, \\ ak_0 & \text{if } \left|\dfrac{\epsilon(t)}{r(t)}\right| \geqslant 1,\ r(t) \neq 0, \\ ak_0 - k_0 exp\left(\dfrac{1}{|\epsilon(t)|^2 - 1}\right) & \text{if } |\epsilon(t)| < 1,\ r(t) = 0, \\ ak_0 & \text{if } |\epsilon(t)| \geqslant 1,\ r(t) = 0, \end{cases} \tag{3.3}$$

where $k_0$ is the proportional constant gain, $a$ is the mean or shift value of the nonlinear function that, together with $k_0$, places the higher gain bounds at either higher or lower values directly related to $a$ and $k_0$. The function is also dependent on the set-point function $r(t)$, which enlarges and shrinks the non-linearity on a one-to-one so that the controller behaves non-linearly in the appropriate error range. A potential change that can be made by the designer is to define the scaling of $r(t)$ to be as desired by altering the right hand side of the inequality and replacing the value of 1 with a tunable parameter. However, as it will be seen later in the definition of the nonlinear derivative gain, this will have a similar effect to the parameter $k_3$, which means that small deviations in the value, will have exponentially large deviations in the peak value of the nonlinear gain, making it in certain systems a disadvantage. Therefore, the chosen approach in this thesis is to design the proportional and integral nonlinear functions without changing this parameter. The benefits to changing it in the nonlinear derivative gain are discussed further in the section.

Figure 3.2a shows the nonlinear proportional gain in terms of the feedback error. This shows the large values of the proportional gains when the feedback error is also large, which helps reduce the error rapidly. Once the system is near steady-state the proportional gain is rapidly reduced. This then provides a fast response and maintains low overshoot. Figure 3.2b shows the effects of varying the gain parameters $a$ and $k_0$ on the proportional nonlinear function gain. It can be seen that the top plots of Figure 3.2b show the effects of varying $a$. It can be seen that the function purely shifts upwards, showing that it is a shifting parameter. The bottom plots of the figure shows the effects of varying $k_0$, in this case, it can be seen that the function also widens as well as shifts upwards, since as $k_0$ varies, so does the product $ak_0$.




**(a)** The tuned nonlinear proportional gain shown for a step set-point function $r(t) = 1$, for values $a = 1$ and $k_0 = 1.5$.




**(b)** Effects of tunable parameters on the proposed nonlinear proportional gain.

**Figure 3.2:** The proposed nonlinear proportional gain and the effects of its parameters on the shape of the function.

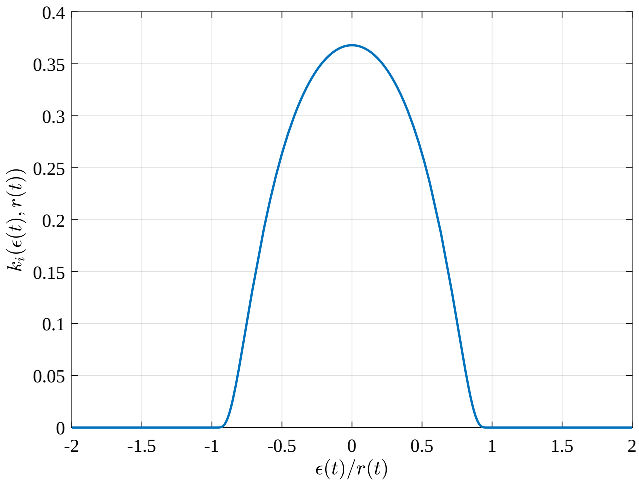
### 3.3.2 Nonlinear Integral Function Gain

The integral nonlinear gain is represented as the following function:

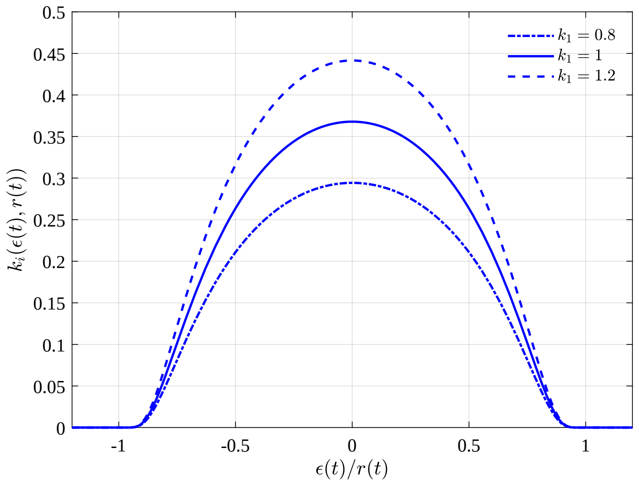
$$k_i(\epsilon(t), r(t)) = \begin{cases} k_1 exp\left(\dfrac{1}{\left|\frac{\epsilon(t)}{r(t)}\right|^2 - 1}\right) & \text{if } \left|\dfrac{\epsilon(t)}{r(t)}\right| < 1,\ r(t) \neq 0, \\ 0 & \text{if } \left|\dfrac{\epsilon(t)}{r(t)}\right| \geqslant 1,\ r(t) \neq 0, \\ k_1 exp\left(\dfrac{1}{|\epsilon(t)|^2 - 1}\right) & \text{if } |\epsilon(t)| < 1,\ r(t) = 0, \\ 0 & \text{if } |\epsilon(t)| \geqslant 1,\ r(t) = 0, \end{cases} \tag{3.4}$$

where $k_1$ is the integral constant that determines the largest value of the integral nonlinear gain. In this case, the set-point function $r(t)$ can also be seen which also enlarges and shrinks the integral nonlinear gain on a scale of one-to-one so that the controller behaves non-linearly in the appropriate error range. Similar to the proportional nonlinear gain a potential change that can be made by the designer is to replace the value of 1 with a tunable parameter to alter the scale by which $r(t)$ enlarges and shrinks the nonlinearity.

The integral gain, shown in Figure 3.3a, is designed so that it starts from a value of zero and increases as the error approaches steady-state, approaching its maximal bounded value. This allows for the integral to error-correct the system during steady state while keeping a low integral value during the transient response, which helps maintain low overshoot. Figure 3.3b shows the effects of varying the parameter $k_1$ on the integral nonlinear function gain. It can be seen that as $k_1$ increases, the peak value of the function also increases, directly affecting the overall 'height' of the function.




**(a)** The tuned nonlinear integral gain shown for a step set-point function $r(t) = 1$, for value $k_1 = 1$.




**(b)** Effects of tunable parameter $k_1$ on the proposed nonlinear proportional gain.

**Figure 3.3:** The proposed nonlinear integral gain and the effects of its parameters on the shape of the function.

### 3.3.3 Nonlinear Derivative Function Gain

Finally, the derivative nonlinear gain is represented as the following function:

$$k_d(\dot{\epsilon}(t), r(t)) = \begin{cases} k_2 exp\left(\dfrac{1}{\left|\frac{\dot{\epsilon}(t)}{r(t)}\right|^2 - k_3^2}\right) & \text{if } \left|\dfrac{\dot{\epsilon}(t)}{r(t)}\right| < k_3,\ r(t) \neq 0, \\ 0 & \text{if } \left|\dfrac{\dot{\epsilon}(t)}{r(t)}\right| \geqslant k_3,\ r(t) \neq 0, \\ k_2 exp\left(\dfrac{1}{|\dot{\epsilon}(t)|^2 - k_3^2}\right) & \text{if } |\dot{\epsilon}(t)| < k_3,\ r(t) = 0, \\ 0 & \text{if } |\dot{\epsilon}(t)| \geqslant k_3,\ r(t) = 0, \end{cases} \tag{3.5}$$

where $k_2$ is the derivative constant that increases the maximum derivative value, $r(t)$, is the set-point function which also enlarges and shrinks the nonlinearity accordingly in a similar behaviour to the previous nonlinear gains, this way all nonlinearities work in synchronization according to the input to maximize the effect of the nonlinearities during the steady-state. The constant $k_3$ is the filtering constant which is a design value determined by the designer according to the amount of derivative the controller needs to control the system.

The derivative gain, shown in Figure 3.4, is similar to the integral gain, with the only difference being the filtering design constant $k_3$. This constant can be useful for the PID limitations. In this case, the input to the nonlinear function is the error rate instead of the feedback error. The nonlinear derivative gain function uses only the error rate as its input because the nonlinear function tracks the changes of feedback error and acts as a predictive function. The derivative of the error considers future actions, the nonlinear gain considers those future actions to determine the value of the nonlinear derivative gain. This helps the controller easily identify the point of steady-state, where the derivative gain is maximized for increased damping, minimizing overshoot, while becoming zero at error rate values higher than the filter constant $k_3$. The constant $k_3$ changes the range at which the nonlinearity operates and defines the zero points of the nonlinear gains. That means that the control designer can freely adjust the function and simultaneously eliminate derivative kicks.

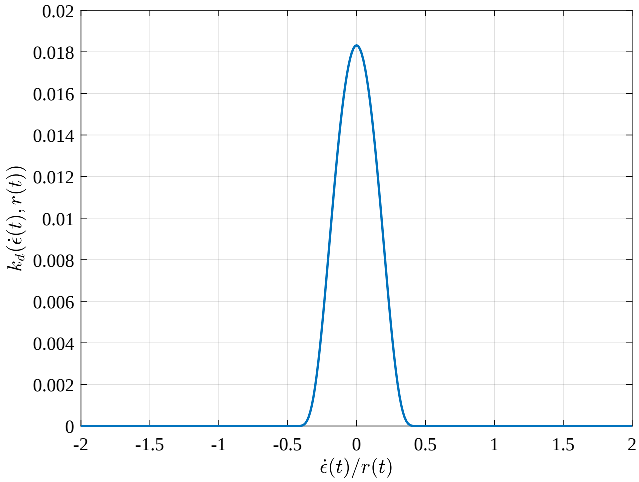
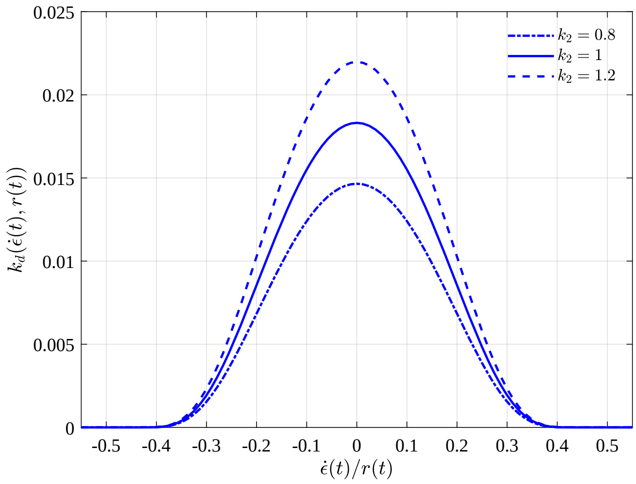
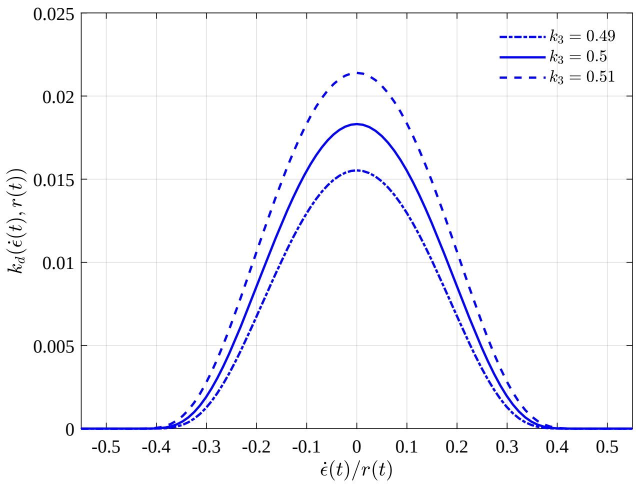



**Figure 3.4:** The tuned nonlinear derivative gain shown for a step set-point function $r(t) = 1$, for values $k_2 = 1$ and $k_3 = 0.5$.

Figure 3.5 shows how variations in the parameters $k_2$ and $k_3$ affect the derivative nonlinear function gain. Figure 3.5a shows the nonlinear function under variation of the parameter $k_2$, where it can be seen that the effect to the function is the same as the parameter $k_1$ for the integral nonlinear function. It makes the function 'taller', increasing its peak value without changing the width. Figure 3.5b shows the effects on the function when varying the parameter $k_3$. The figure shows that the effects of parameter $k_3$ are two-fold, it inreases the peak, similar to $k_2$ but at a much larger scale compared to the changes made in $k_3$. In addition, it widens the function, however, this effect is difficult to visually capture, since small variations in $k_3$ will exponentially increase the function height, while it will linearly increase the function width.




**(a)** Effects of the parameter $k_2$, when $k_3 = 0.5$, on the proposed nonlinear proportional gain.




**(b)** Effects of the parameter $k_3$, when $k_2 = 1$, on the proposed nonlinear proportional gain.

**Figure 3.5:** The proposed nonlinear derivative gain and the effects of its parameters on the shape of the function.

### 3.3.4 Unique Features of the Proposed NLPID Gains

The proposed nonlinear derivative gain has been designed with the pre-meditated intent to eliminate derivative kicks, as this is a well-known limitation of PIDs. This can be shown via a simple mathematical example and an explicit simulation. Firstly, a mathematical example is shown that captures this inherent property of the nonlinear derivative function before proceeding to the simulation case.

**Example 3.3.1.** Assume a controller that can perfectly track a set-point input and assume a unit step set-point input described in discrete form by the following function:

$$r(t) := \begin{cases} 1 & \text{if } t \geqslant 1 \ , \\ 0 & \text{if } 0 \leqslant t < 1. \end{cases} \tag{3.6}$$

The feedback error of a control system can be defined as:

$$\epsilon(t) = r(t) - y(t) \tag{3.7}$$

Take the forward Euler discrete derivative as:

$$\frac{d\epsilon(t)}{dt} = \frac{\epsilon(t + \Delta t) - \epsilon(t)}{\Delta t}. \tag{3.8}$$

This then drives the value of the PID derivative to be as follows:

$$k_d(\dot{\epsilon}(t), r(t))\frac{\epsilon(t + \Delta t) - \epsilon(t)}{\Delta t}. \tag{3.9}$$

Assuming a sampling frequency of 10 Hz, hence the time step is $\Delta t = 0.1$ the derivative of the error at time $t' = 1 - \Delta t$, which can then be computed by starting with the forward Euler Eq.(3.8) and substituting the time value $t' = 1 - \Delta t$ that the derivative is computed at, and proceeding as follows:

$$\frac{\epsilon(t' + \Delta t) - \epsilon(t')}{\Delta t} = \frac{\epsilon(1) - \epsilon(1 - \Delta t)}{\Delta t} \tag{3.10}$$

At this moment of time, the output is equal to zero, since the time $t'$ is the moment that the first non-zero input is inserted into the feedback system, hence, $r(1 - \Delta t) = y(1 - \Delta t) = 0$, consequently, it continues as:

$$\frac{\epsilon(1)}{\Delta t} = \frac{1}{0.1} = 1 \times 10, \tag{3.11}$$

which means that the derivative gain becomes $k_d(\dot{\epsilon}(t), r(t)) = 0$, since $\left|\frac{\dot{\epsilon}(t)}{r(t)}\right| = \left|\frac{1 \times 10}{1}\right| > k_3$ for any $k_3 < 10$. The value of $k_3 < 10$ is reasonable for practical examples, since the largest practical value for $k_3$ would be of unit value $k_3 = 1$ as it will be seen from examples in later chapters. This simple example shows how the proposed nonlinear derivative function gain eliminates derivative kicks, when step set-point functions are introduced to the system. △

Then, an explicit simulation of this benefit has been conducted, where the derivative action of the proposed NLPID controller and a constant gain PID are compared against each other in a basic first order system:

$$P(s) = \frac{1}{s+1} \tag{3.12}$$

The tuning of the proposed NLPID and linear PID, without derivative filtering, are the same in this example as that of the FOPTD system in Chapter 4, since the two systems are similar. Figure 3.6 shows the derivative action comparison between the proposed nonlinear derivative gain function and the constant derivative gain action, showing that the proposed nonlinear function entirely eliminates the derivative kick.

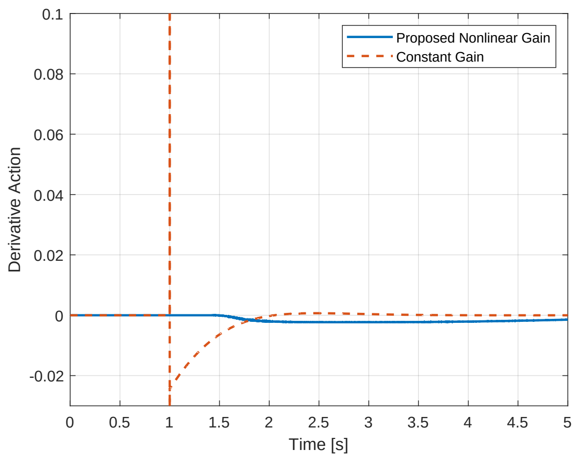



**Figure 3.6:** The derivative action comparison of the proposed nonlinear derivative function gain compared against that of a constant derivative gain.

This advantage can in turn improve robustness of the proposed controller and reduce actuator degradation from such large signals.

An additional advantage of the proposed nonlinear gains is that they are adapting according to the set-point. The effect of a changing set-point to the NLPID gains is shown in Figure 3.7 where the larger the set-point becomes, the wider the nonlinearities are, preserving the design constants, such as the maximum value of the gains, the minimum value of the gains, and so that the nonlinearities are active within the range $-\epsilon_{max} \leqslant \epsilon \leqslant \epsilon_{max}$.

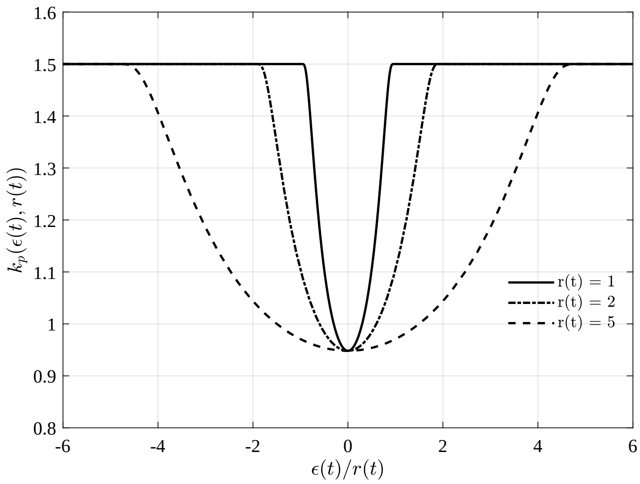



**Figure 3.7:** The tuned nonlinear proportional gain as it adapts to a new step set-point function of values $r(t) = 1, 2$, and 5.

### 3.3.5 Particle Swarm Optimization Algorithm Tuning the Proposed NLPID Controller

The tuning of the proposed NLPID parameters $k_0, k_1, k_2$, and $a$ are conducted using the objective function and optimization problem designed with the Integral Time Absolute Error (ITAE) performance measure and the settling time of the system as:

$$\begin{aligned} \underset{k_0, k_1, k_2, a}{\text{minimize}} \quad & f(t, \epsilon(t), t_s) = \int_0^{t_f} t|\epsilon(t)|\, \mathrm{d}t + t_s \\ \text{subject to} \quad & k_{min} \leqslant k_0, k_1, k_2 \leqslant k_{max}, \\ & a_{min} \leqslant a \leqslant a_{max}, \end{aligned} \tag{3.13}$$

where $t_s$ is the settling time, $\epsilon(t)$ is the feedback error, and $t_f$ is the final time. The objective function has been selected to prioritize the simultaneous minimization of the overshoot and settling time. This is effective in improving transient response, however, it is not very practical for disturbance rejection. The objective function also has the potential to be adopted with weightings to give different priorities to the optimization algorithm. In this case, it has been selected to give equal importance and hence has not been given different weights. Moreover, other objective functions can work well without having the trade-offs between performance and robustness, however, the goal of the selected function is to show that due to the nonlinear gains, when optimally tuned for performance they also show robustness. The parameter constraints specify the region of stable control and the optimization algorithm has a smaller search space with less likelihood of trapping inside local optima. The MATLAB function of the objective function is shown in Appendix B. The PSO algorithm is programmed using MATLAB programming language, shown in Appendix A and has the following process (Wang, D., Tan and Liu, L., 2018):

**Algorithm 1** Particle Swarm Optimization (PSO) pseudo-code

**procedure** PSO ALGORITHM
    Generate n number of random position particle vector and q number of tuning parameters in each particle vector ${}_qX_0^n$ in the range $[k_{min}, k_{max}]$ and $[a_{min}, a_{max}]$ for $a$
    Assume initial velocity vector ${}_qV_0^n = 0$

    **for** j=1 to n, **do**
        Simulate model in Simulink
        Compute $f(t, \epsilon(t), t_s) = \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s$
        find the minimum value of $f(t, \epsilon(t), t_s)$
        Initialise a random number $r_1^j$ in the range $[0, 1]$
        Compute ${}_qV_1^j = |{}_qV_1^j +_q r_1^j c_1(Gbest_1 -_q X_1^j)|$
        Compute ${}_qX_1^j = |{}_qX_1^j +_q V_1^j|$
    **end for**

    **for** j=1 to n, **do**
        **for** i=2 to m, **do**
            **if** $X_1^j \geqslant 2$ **then**
                Re-initialise a random number in range $[k_{min}, k_{max}]$ for $k_0, k_1, k_2$
                Re-initialise a random number in range $[a_{min}, a_{max}]$ for $a$
            **else**
                $k_0 =_1 X_{i-1}^j$, $k_1 =_2 X_{i-1}^j$, $k_2 =_3 X_{i-1}^j$
            **end if**
            **if** $X_1^j \geqslant a_{max}$ OR $X_1^j \leqslant a_{min}$ **then**
                Re-initialise a random number in range $[a_{min}, a_{max}]$ for $a$
            **else**
                $a =_4 X_{i-1}^j$
            **end if**
            Simulate model in Simulink
            Compute $f(t, \epsilon(t), t_s) = \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s$
            find the minimum value of $f(t, \epsilon(t), t_s)$
            **for** q=1 to p, **do**
                Initialise a random number $r_i^j$ in the range $[0, 1]$
                Compute ${}_qV_i^j = |{}_qV_i^j +_q r_i^j c_i(Gbest_i -_q X_i^j)|$
                Compute ${}_qX_i^j = |{}_qX_i^j +_q V_i^j|$
            **end for**
        **end for**
    **end for**
**end procedure**

where

- $i$ is the iteration index, $j$ is the particle index, and $q$ is the particle vector element index, $n$ is the total number of particles generated,
- ${}_qX_i^j$ is the $j^{\text{th}}$ position particle, which is a vector that contains the tuning parameters as elements with each $q^{\text{th}}$ parameter and each iteration $i$,
- ${}_qV_i^j$ is the velocity vector for each iteration $i$, $j^{\text{th}}$ particle, and $q^{\text{th}}$ parameter,
- ${}_qr_i^j$ is the stochastic variable that changes for every iteration and lies in the range $[0, 1]$, and
- *Gbest* is the position particle with the minimum objective function evaluation across the iterations.

If the new position ${}_qX_{i+1}^j$ is outside the specified range of values, then these specific new particles are re-initialized within the pre-specified range. The parameter $c_1 = 1.3$ is a tuning parameter taken from research surveys on PSO tuning (Wang, D., Tan and Liu, L., 2018).

The Particle Swarm Optimization (PSO) algorithm is used to tune the proposed NLPID controller. The PSO algorithm is well known to be an effective stochastic optimization algorithm, with fast convergence, without using derivatives. However, the downside is that it is easy for PSO to fall to a local minimum (Wang, D., Tan and Liu, L., 2018; Parsopoulos and N.Varsatis, 2002; Clerc and Kennedy, 2002). To overcome this challenge, two steps have been considered to lower the possibility of such an occurrence. Firstly, the particles are limited within a range of specified values. Secondly, the personal best value of each particle, also known as cognition, is not considered in this case. The global, also known as social intelligence, is used, taking the social best objective value to make it more difficult to fall at a local optimum.

This process is a modification of the original particle swarm optimization, which included the history of the minimum objective value for each particle, in this case only the social best values are considered.

## 3.4. Indicative Stability and Constraints Justification

To define the parameter constraints a stable control region study has also been conducted to determine the feasible search regions. Simulations have been conducted for three linear systems to determine the stability regions and the stable tunings for $k_0, k_1,$ and $k_2$. This is to establish a simulation-based comparison between the theoretical stability results and the results shown from the simulations. This indicates that the theoretical results are conservative compared to the stability region of the simulations, consequently, the theoretical results can provide a guaranteed indicative stability analysis for the three linear systems. The simulations have been conducted by changing only the parameters $k_0, k_1,$ and $k_2$ with the parameters $a$ and $k_3$ unchanged from the tuned values as shown in this chapter. The initial approach employed a brute force grid-based method of conducting the simulations by ranging both values of $k_0$ and $k_1$ or $k_0$ and $k_2$ for all possible points in a defined rectangular grid. Due to brute force grid-based simulations, the simulation took a long time to capture the stability regions, which could range between 40 to 55 minutes. As a result, the grid-based computation approach is reduced by only ranging the value of $k_0$ and using a bisection algorithm to determine the value of $k_1$ and $k_2$, which reduces the time taken for the simulation to determine the stability region between 13 to 25 minutes. The simulations have been conducted for the region defined by the PSO optimisation-based tuning constraints to capture a comparison between the theoretical and simulation results in the tuning search space. To speed up the simulations, the relative tolerance has been reduced to $10^{-4}$ with the shape preservation and zero-crossing detection disabled, and fast-restart is enabled. In addition, the simulations have been conducted for a longer simulation time of 150 seconds to capture the full dynamics and have a more certain determination of instability. This helps eliminate cases of pure instability and capture cases of marginal stability or large overshoots, which take longer to settle to the reference point. To determine instability, the Bounded-Input Bounded-Output definition is used and the system input and output are detected with a threshold of $\pm100$ used to confirm the boundedness of the signals.

## 3.5. Concluding Remarks

The proposed NLPID controller extends the capabilities of PID control and mitigates its common limitations by proposing a new set of nonlinear gains. The new set of nonlinear gains shows flexibility, they can enlarge and shrink the range of nonlinearity, by taking into consideration the set point. The proposed NLPID controller has fewer tunable parameters that makes the tuning process simpler and less time consuming. The nonlinear derivative function allowed for the elimination of any derivative kicks. This is a crucial advantage since tuning can take the majority of the time of the designer.

The extensive simulation-based indicative stability analysis of the proposed NLPID controller has also been shown for the feedback control of linearised dynamics. Moreover, the indicative stability analysis provides a simulation-based constraints justification of the tunable parameters. This contributes to the support of the proposed NLPID controller design and tuning.

# Chapter 4

# Applications and Benchmarking of the Proposed NLPID Controller

## 4.1. Introduction

This chapter is focused on the possible control applications of the proposed NLPID controller and to benchmark its performance against conventional control algorithms. A control application of the proposed NLPID controller is shown, using the dimensionless Nonlinear Continuous Stirred Tank Reactor (NCSTR), a highly nonlinear system. The control performance of the proposed NLPID controller is shown for all the operating regions under actuation limits. In addition, the proposed NLPID controller is benchmarked using a set of linear plant dynamics that are generally a common outcome of the linearization process to most nonlinear models, namely, a First Order Plus Time Delay (FOPTD), the Negative Gain Second Order Plus Time Delay (NG_SOPTD), and the Non-minimum Phase SOPTD (NmP_SOPTD). The proposed NLPID controller is benchmarked against the conventional PID, two-degree of freedom PID (2D_PID), and Smith Predictor PID (SP_PID) controllers to show the effectiveness of the proposed NLPID controller in transient performance and disturbance rejection. The tuning approach of the conventional controllers is presented in this chapter explicitly to ensure a fair comparison. Finally, the system input energy of the feedback system is shown as an indication of internal stability.

## 4.2. Problem Statement

This chapter is broken down into two main problem areas:

1. Application of the proposed controller to nonlinear systems, using the motivating example as a case scenario.
2. Benchmarking of the proposed controller against conventional methods.

Consequently, in this section, the two main focus areas are defined as follows.

**Nonlinear Case Study**

The motivating example of this thesis has been the NCSTR system, which has a set of highly nonlinear dynamics commonly seen in chemical industries. The proposed NLPID controller is designed and tuned for the NCSTR system to show the robustness of the proposed controller at multiple operating regions of the system. The performance of the proposed controller is analysed in simulation. The system model under consideration is the dimensionless NCSTR model with the following mathematical form (Colantonio et al., 1995; Harmon Ray, 1981; Sinha and Mishra, 2018; So and Jin, 2018):

$$\begin{aligned}
\dot{x}_1(t) &= -x_1(t) + D_a(1 - x_1(t))exp\left(\frac{x_2(t)}{1 + x_2(t)/\gamma}\right) + d_1, \\
\dot{x}_2(t) &= -(1+\beta)x_2(t) + HD_a(1 - x_1(t))exp\left(\frac{x_2(t)}{1 + x_2(t)/\gamma}\right) + \beta sat(u(t)) + d_2, \\
y(t) &= x_2(t).
\end{aligned} \tag{4.1}$$

Analysis conducted within the literature of the dimensionless NCSTR model represented in Eq.(4.1), showing that the system has three steady states $x_A = [0.144, 0.886]$, $x_B = [0.447, 2.752]$, and $x_C = [0.765, 4.705]$ determined for the nominal parameters of the system where $H = 8$, $D_a = 0.072$, $\gamma = 20$, $\beta = 0.3$ (Colantonio et al., 1995; Harmon Ray, 1981). An open-loop analysis can show the stability properties of the system steady-states and expand in more detail what is already known to the literature. Figure 4.1 shows the evolution of the states of the NCSTR system, with initial conditions near the steady state $x_A$. The figure shows that the initial state converges towards the steady state $x_A$ showing that it is a stable steady state.

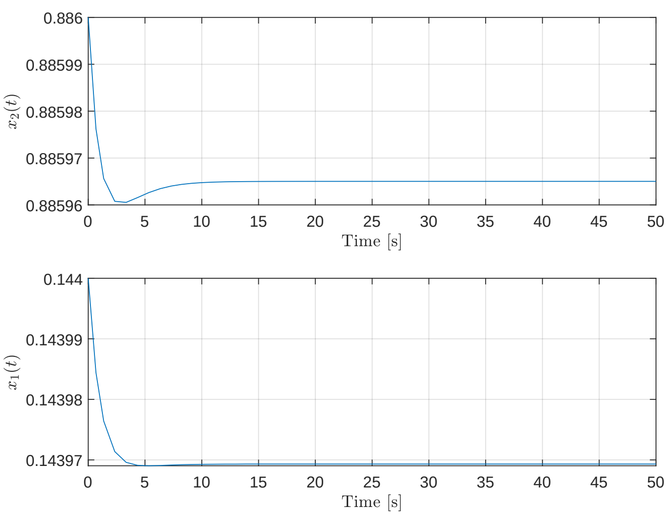



**Figure 4.1:** The open-loop study of the steady state point $x_A$.

Figure 4.2 shows the evolution of the system with initial conditions near the steady state point $x_B$. The figure shows that it tends towards the steady state $x_C$ and away from $x_B$. This shows that $x_B$ is an unstable point that the open-loop system cannot reach the state $x_B$.

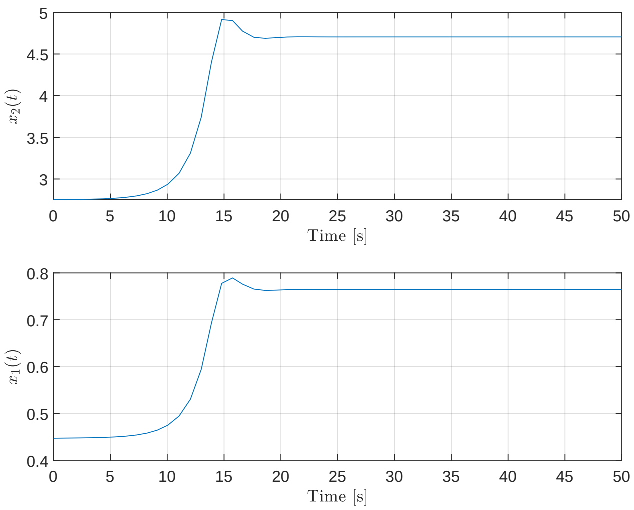



**Figure 4.2:** The open-loop study of the steady state point $x_B$.

Figure 4.3 shows the evolution of the open-loop NCSTR system when it has initial conditions near the state $x_C$. The figure shows that the open-loop response converges towards the point $x_C$ showing a stable steady state at $x_C$.

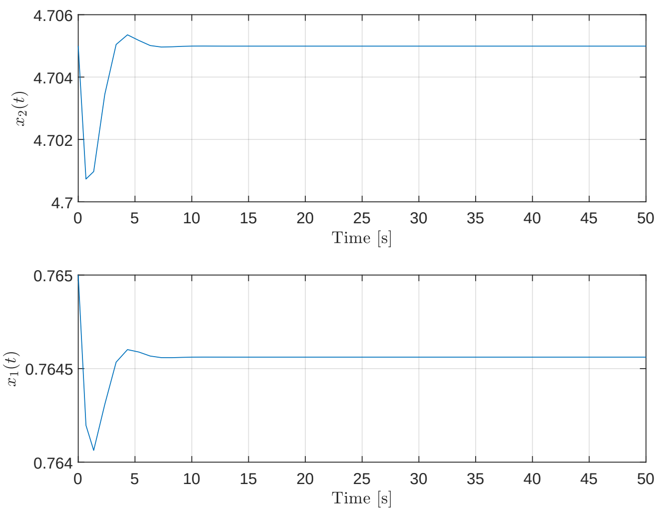



**Figure 4.3:** The open-loop study of the steady state point $x_C$.

The analysis within the literature has also shown that $x_A$ and $x_C$ are stable steady states, while $x_B$ is an unstable steady state (Colantonio et al., 1995; Harmon Ray, 1981).

**Benchmarking Case Studies**

Industrial systems are highly nonlinear and possess a variety of subsystems, actuators, and sensors that all contribute to the nonlinear behaviour. The proposed NLPID controller is simulated for the NCSTR system to determine the effectiveness of the proposed controller in a highly nonlinear scenario. However, conventional controllers are most commonly tuned for linear plants. Consequently, benchmarking of the proposed controller against the conventional methods is done for the linearised system under a desired operating condition. The linearisation constrains the system within the operating condition and it is desirable to design the control algorithm to be robust against uncertainties. Figure 4.4 shows a block diagram of such an industrial control problem using a feedback system that contains disturbances and sensor biases. The control objective is to design the controller $C(s)$ for the plant $P(s)$ to achieve a desired performance and maintain stability for a large uncertainty. The proposed NLPID controller is benchmarked against the conventional methods by simulating common industrial case scenarios.

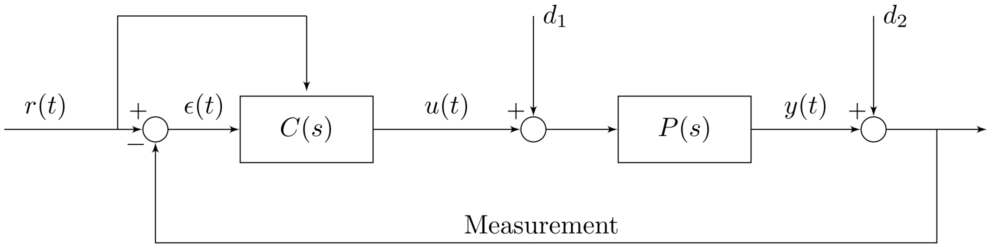



**Figure 4.4:** The schematic block diagram of the control system with both the input and output disturbances.

A general description of common linear model dynamics in industrial systems takes the mathematical form of a Second Order Plus Time Delay (SOPTD) system, which is also commonly seen in linearised dynamics of the NCSTR system (Krishna et al., 2012). The transfer function description of a SOPTD system is as follows (Krishna et al., 2012):

$$P(s) = \frac{z_1 s + z_0}{q_2 s^2 + q_1 s + q_0} e^{-\tau s}. \tag{4.2}$$

The linearised dynamics under investigation are:

1. The First Order Plus Time Delay (FOPTD) system,
2. The Negative Gain Second Order Plus Time Delay (NG_SOPTD) model, and
3. The Non-minimum Phase Second Order Plus Time Delay (NmP_SOPTD) model.

These commonly represent different NCSTR operating region linearisations. Moreover, the dynamics possess various dynamical behaviours that are used to test the limits of the proposed control algorithm. These include delays, first and second-order, negative gain, and non-minimum phase dynamics that can also represent a large set of industrial systems not limited to NCSTR.

The controllers that are benchmarked against the proposed NLPID controller are the traditional PID, Two-degrees-of-freedom PID (2D_PID), and Smith-Predictor PID (SP_PID) controllers. The following control design criteria are used for the benchmarking across all three systems:

- Minimization of overshoot.
- Minimization of rise time and settling time.
- Fast disturbance rejection to input and output disturbances.

Using these control design criteria, the conventional control algorithms are tuned using the tools and methods that are available to the practitioner described in what follows.

The $L_2$ norm of the controller output is also computed for the nominal value of the system. The bounded input bounded output signals can indicate internal stability of the system. The $L_2$ norm can compute the total system input energy produced and if the $L_2$ is finite, that means the system has a bounded output for every bounded input. This provides satisfactory evidence of internal stability, as well as energy usage. The values below are the computed $L_2$ norms using the following equation for the system input signal before applying the input disturbance.

$$L_2(u(t)) = \sqrt{\int_{t_0}^{t_f} (u(t)^2)} \tag{4.3}$$

The mean squared error (MSE) is used to determine the signal variance of the benchmarked controllers, showing a measure of how much the error signals vary across the simulation time for each controller. This is a useful benchmarking measure as it provides information as to how much the inputs vary, which is an important factor for actuator safety and overall signal quality. The formula for computing the MSE measure is shown in Eq 4.4 as follows:

$$MSE = \frac{1}{n} \int_{t_0}^{t_f} \epsilon(t)^2 \tag{4.4}$$

where $n$ is the number of data points, which is dependent on the timestep and total simulation time.

## 4.3. Simulation Execution Methodology

The computer that is used to conduct the research has a quadcore Intel i7-6700HQ processor with 16GB RAM and a 250 GB SSD memory card. The operating system of the computer is a 64-bit Windows 10. MATLAB/Simulink R2021a software version is installed under the academic license and is used to conduct the simulations. A variable step-size solver is used so that it is automatically selected by the software as is best fit for the problem. In the cases of the simulations, the (Runge-Kutta) ODE45 solver is selected with a relative tolerance of $10^{-8}$. The solver used for the second order plus time delay systems is the ODE23tb, which uses the trapezoidal rule + backward differentiation formula to solve stiff differential equations. The solver relative tolerance used in those cases was also $10^{-8}$. The solver reduced the execution time of the system simulations and was able to solve the equations accurately. To speed up the process of the repetitive simulations during tuning and parametric uncertainty response analysis, the 'Fast Restart' approach was used in Simulink to conduct repeated simulations without repeatedly compiling the same model.

## 4.4. Application of the Proposed NLPID Controller to the NCSTR System

The system model under consideration is the dimensionless NCSTR model with the following mathematical form (Colantonio et al., 1995; Harmon Ray, 1981; Sinha and Mishra, 2018; So and Jin, 2018):

$$\begin{aligned} \dot{x}_1(t) &= -x_1(t) + D_a(1 - x_1(t))exp\left(\frac{x_2(t)}{1 + x_2(t)/\gamma}\right) + d_1, \\ \dot{x}_2(t) &= -(1+\beta)x_2(t) + HD_a(1 - x_1(t))exp\left(\frac{x_2(t)}{1 + x_2(t)/\gamma}\right) + \beta sat(u(t)) + d_2, \\ y(t) &= x_2(t). \end{aligned} \tag{4.5}$$

The input saturation is used to represent a fully open and fully closed inlet to the NCSTR system and pose limits to the system input for a realistic and practical representation.

$$sat(u(t)) := \begin{cases} u_{min}, \text{if } u(t) < u_{min}, \\ u(t), \text{if } u_{min} \leqslant u(t) \leqslant u_{max}, \\ u_{max}, \text{if } u(t) > u_{max}, \end{cases} \tag{4.6}$$

where $u_{min} = -5$ and $u_{max} = 5$ adopted from (So and Jin, 2018).

The control problem of the NCSTR plant is to regulate the system while the output transitions from the stable steady state $y_A = 0.886$ into the unstable steady state $y_B = 2.752$. Then, regulate the system for the second transition from the unstable steady state $y_B$ to the stable steady state $y_C = 4.705$. Hence, the desired output is $0.886 \to 2.752 \to 4.705$. The three steady states possess second order with minimum phase dynamics and different stability properties (Colantonio et al., 1995). The simultaneous state disturbances $d_1 = d_2 = 2\%$ of $r(t)$ are also implemented as constant step function disturbances. These values represent practical disturbances of constant values in the $x_1$ and $x_2$ states of the NCSTR to test the robust regulation of the system. It must be noted that the state $x_2$ is observable while the state $x_1$ is not observable. As a result, the disturbance $d_1$ applied in the $x_1$ state affects the state $x_1$ which in turn affects the observable state $x_2$ negatively and by a large factor of $H = 8$. Hence, the value of 2% of $r(t)$ has been selected for this study as an arbitrarily large disturbance to test the robustness of the controller without causing system instabilities due to the large factor $H$.

### Design and Tuning of the NLPID Controller

The Particle Swarm Optimization (PSO) tuning algorithm is used to design the NLPID controller parameters. The filtering parameter is selected as $k_3 = 1$ to reduce derivative kicks and reduce overshoot by increased derivative action. The search space boundaries have been suggested by the authors for the NCSTR system. These constraints can be changed by the designer based either on prior knowledge about the system, the controller, or any other design preference related to the case at hand. The steady state $y_B$ is unstable and it increases the sensitivity of the system. Hence, the constraints of the algorithm are considered for high sensitivity to maintain stability. The search space boundaries are defined as $k_{min} = 0, k_{max} = 10, a_{min} = 0,$ and $a_{max} = 2$.

The PSO algorithm searched for the nonlinear gain parameters $k_0, k_1, k_2,$ and $a$. The tuned values of the proposed NLPID controller are shown in Eq.(4.7).

$$k_0 = 7.4703, k_1 = 8.8245, k_2 = 4.0618, k_3 = 1, a = 1.7382 \tag{4.7}$$

### NCSTR Control Performance Simulation Results

The nominal dynamics simulations are conducted to show the performance of the proposed NLPID controller for the control of the NCSTR system. The simulations are conducted for 60 seconds to perform two consecutive transitions between the steady states $y_A \rightarrow y_B \rightarrow y_C$. The initial conditions of the system are the steady state values of the point $x_A$. Then the system is regulated for the transition to $x_B$ and then to $x_C$. The first transition is conducted from set-point $y_A \rightarrow y_B$ at the 5 second time mark and the second transition from set-point $y_B \rightarrow y_C$ occurs at the 30 second time mark. The disturbances $d_1 = d_2 = 2\%$ of $r(t)$ acting on the states are step functions implemented at the 15 and 40 second time marks, respectively.

Fig. 4.5 shows the time variation of the nonlinear gains of the proposed NLPID controller for the simulation of the NCSTR system for all operating regions. The figure shows the adaptive ability of the proposed NLPID controller. Moreover, the figure indicates how the nonlinear proportional gain has a low initial value, increases with the step execution and then rapidly decreases to reduce the overshoot. In addition, the nonlinear integral gain is shown to have a high value, then drops as the step is executed and provides a rapid increase near the steady state to eliminate the steady-state error. Finally, the nonlinear derivative gain shows a high initial derivative action, which then suddenly drops during the step changes to eliminate derivative kicks and increases during steady-state to improve system performance.

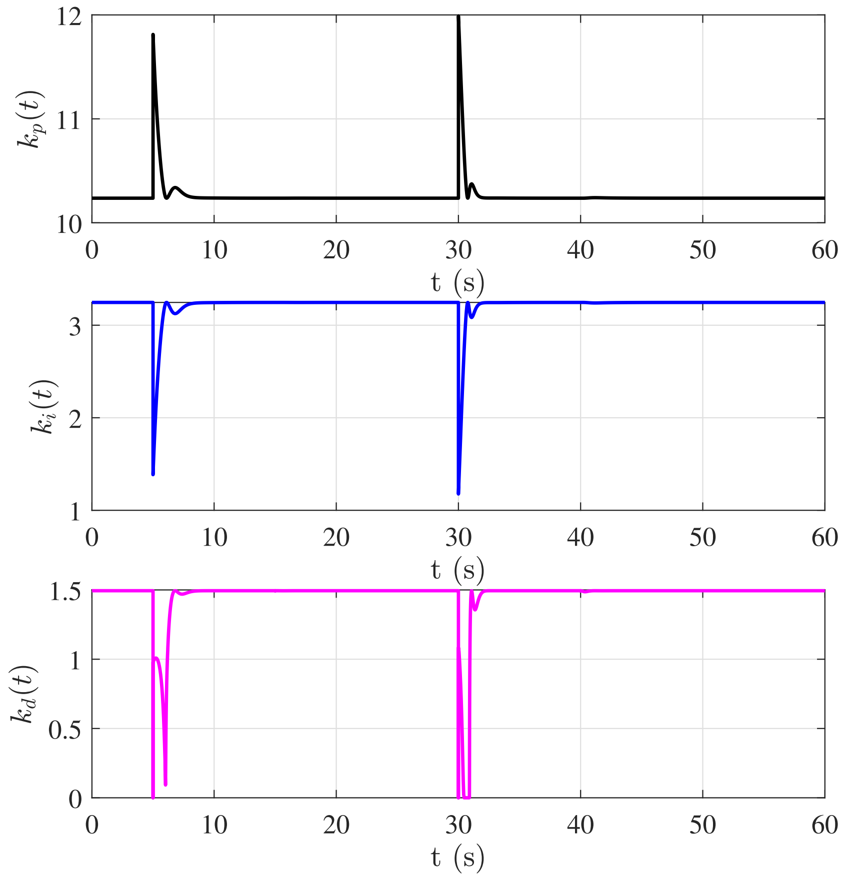
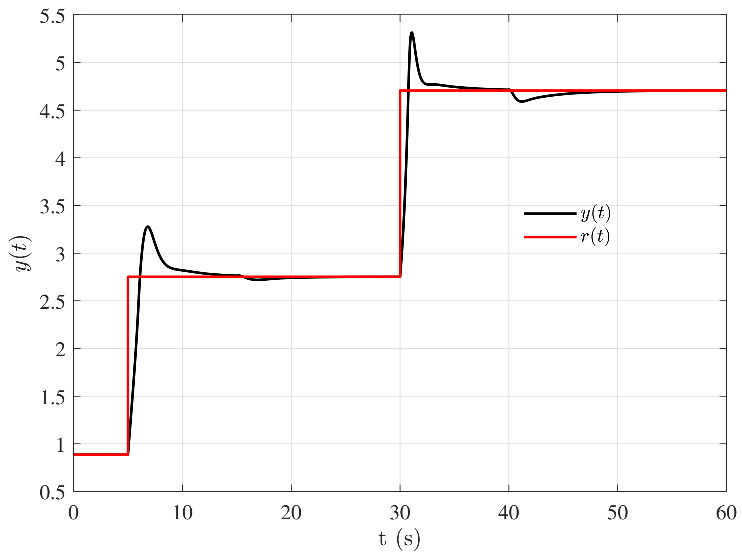
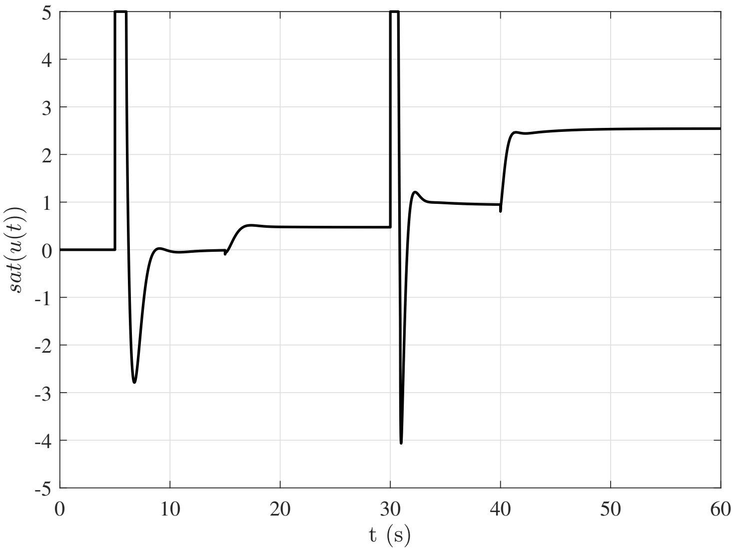



**Figure 4.5:** The time response of the proposed NLPID controller tuned gains to the NCSTR system simulation.

Fig. 4.6 shows the regulation of the control system for the output transition of the NCSTR system from $y_A$ to $y_B$ and from $y_B$ to $y_C$. The output response, shown in Fig. 4.6a, indicates that the proposed NLPID controller produces a fast response with an overshoot of 18.9%, a rise time and settling time of 0.89 and 11.88 seconds, respectively. In addition, the transient response from $y_B$ to $y_C$ produces an overshoot of 15.4% with a rise time and settling time of 0.53 and 9.74 seconds, respectively. The output $y_B$ is unstable and it is shown that the proposed NLPID controller regulates the system and manages to reject disturbances stably for both output transitions and provide a fast transient response. Moreover, it can be seen that the disturbances generate an undershoot that the controller must regulate. According to Eq. (4.5) the undershoot is introduced because of the disturbance $d_1$ that impacts the state $x_1$, which reduces the output $y = x_2$ due to the negative sign. As a result, due to the large factor $H = 8$, the undershoot introduced by $d_1$ is larger than the overshoot introduced by $d_2$ making the resultant effect an undershoot dominated by the impact of $d_1$. Fig. 4.6b shows the system input of the NCSTR. It can be seen that the proposed NLPID controller momentarily reaches saturation to generate a fast performance in both output transition cases. This indicates that the proposed NLPID controller can generate a fast response and regulate the system behaviour under saturation limits without windup or instabilities.




**(a)** The transient response of the NCSTR system using the proposed NLPID controller to transition from operating point $y_A$ to $y_B$ then to $y_C$.




**(b)** The saturated system input to the NCSTR model from the proposed NLPID controller for the operating point transition from $y_A$ to $y_B$ then to $y_C$.

**Figure 4.6:** The servo and regulator performance of the proposed NLPID controller in the NCSTR system for the operating point transition from $y_A$ to $y_B$ then to $y_C$.

The proposed NLPID controller has been simulated controlling an NCSTR plant with a stable closed loop, where the plant model includes input saturation, which is commonly seen in practice. The proposed controller shows applicability to practical problems that commonly require linearisation and multiple design iterations, indicating future practical potential. The advantage of the proposed controller that is demonstrated with this case scenario is that it can handle multiple operating conditions using a single tuning, while the conventional approach would require a multi-step process of linearisation, analysis, and tuning for multiple models. Finally, although the proposed controller has not been compared against gain scheduling in this scenario, the results demonstrate that nonlinear function gains can be a simpler approach, requiring less tuning and analysis iterations for the NCSTR system. Next, to further illustrate this advantage of the proposed controller, benchmarking case scenarios are defined using common linearisations of the NCSTR system, where the proposed controller is benchmarked against the conventional controllers.

## 4.5. Benchmarking Case-Studies

The applicability of the proposed controller to nonlinear systems has been shown using the motivating example as a case scenario. The next problem area focuses on showing a comparison of the proposed NLPID controller to provide fast control performance to linear systems, using common model linearisations of the NCSTR model as case scenarios. To indicate explicitly the fair comparison of the controllers, the tuning approach taken for this problem is shown with the benchmarking results following there after for each linear plant.

### 4.5.1 Tuning Methodology for the Conventional Controllers

PID control research has the difficulty and unfortunate disadvantage that many control comparisons are unfair and improved results can be achieved by spending more effort on tuning (Åström and Hägglund, 1995; Atherton and Majhi, 1999; Åström, 2000; Åström and Hägglund, 2001). To ensure a fair comparison the conventional control algorithms are tuned using the tools available to the practitioner. MATLAB PID tuning algorithm has been used for various problems by both researchers and practitioners, showing effective results with ease of use (Reddy et al., 2011; Gahinet, Chen, R. and Eryilmaz, 2013; Gomes et al., 2016; Scherlozer, Orsini and Patole, 2016; Wang, L., 2020). MATLAB tuner has also been reported in research to provide an effective tuning method for diverse problems, including nonlinear systems, systems with delays, systems with non-minimum phase dynamics, and all the linear models (Reddy et al., 2011; Gahinet, Chen, R. and Eryilmaz, 2013; Gomes et al., 2016; Scherlozer, Orsini and Patole, 2016; Wang, L., 2020).

MATLAB PID tuner is used to design the conventional control algorithms due to its reported ability to adequately tune the controller for various systems. Various tuning trials have been conducted, some of which adequately fit the design criteria, with the design that has minimum overshoot and fast response selected as the successful design candidate. This explicitly shows the common design difficulties faced in the conventional control approaches that are used by practitioners in industry and how the proposed NLPID controller provides an improved alternative.

### MATLAB PID Tuner

MATLAB PID tuning algorithm linearises the plant system at the operating point, then tunes the PID controller according to the linearised model, at that operating point. This algorithm works by parameterizing the controller based on the designer's pre-specified value of the cross-over frequency and the phase margin of the controller (Gahinet, Chen, R. and Eryilmaz, 2013). The cross-over frequency is directly related to the open-loop system bandwidth, which is directly related to the speed of the response and uses the phase margin to design the robustness of the controller (Gahinet, Chen, R. and Eryilmaz, 2013).

The parameterization of the controller allows the designer to directly visualize the response according to the set design criteria, which can be changed in real time. This makes MATLAB tuner a simple, effective, and easy-to-learn tuning method. The parameterization of the controller used by the algorithm can be written as (Gahinet, Chen, R. and Eryilmaz, 2013):

$$C(s) = \frac{\omega_c}{s}\left(\frac{sin(\phi_z)s + \omega_c cos(\phi_z)}{\omega_c}\right)\left(\frac{sin(\beta)s + \omega_c cos(\beta)}{sin(\alpha)s + \omega_c cos(\alpha)}\right), \tag{4.8}$$

where $\omega_c$ is the frequency at which the magnitude of the open-loop response $Y(s) = K_{\text{PID}}(s)P(s)$ first crosses the 0 dB line, and angles $\phi_z, \alpha$, and $\beta$ vary between 0 and 90 degrees, with a total phase shift provided by the PID controller at frequency $\omega_c$ given by (Gahinet, Chen, R. and Eryilmaz, 2013):

$$\Delta\phi = \phi_z + \beta - \alpha. \tag{4.9}$$

In addition, MATLAB tuning algorithm allows for prioritization in robustness, set-point tracking, or a balance of both, which is adopted as the designer requires, and it is also applicable for tuning the 2D_PID control algorithm (Gahinet, Chen, R. and Eryilmaz, 2013; Wang, L., 2020).

### Selection Criteria

The different tuning trials are conducted using MATLAB tuning algorithm. The tuning trials are conducted via manual trial and error by varying the open loop phase margin, but also cutoff frequency to define the open-loop bandwidth. The selection criteria for the conventional controllers is based on the tuning that produces the minimal overshoot, rise time, and settling time. This in effect will also optimise for the objective function used to tune the proposed NLPID controller. Hence, this way a fair approach to tuning is taken by taking the same priorities for all of the controllers.

### 4.5.2 The FOPTD Case Study

An arbitrary FOPTD system is considered for the first benchmarking scenario to start with a more fundamental model. It is mathematically shown in the form of a transfer function, for the values $z_1 = 0, z_0 = 1, q_2 = 0, q_1 = 1,$ and $q_0 = 1$ as follows:

$$P(s) = \frac{1}{s+1}e^{-s}, \tag{4.10}$$

The conventional PID, 2D_PID, and SP_PID controllers are designed and tuned for the FOPTD model. To ensure a fair comparison between the conventional control algorithms and the proposed NLPID controller the design specifications are kept the same for all controllers, and MATLAB PID tuning tool is used to ensure the specifications are met.

#### Design and Tuning of the NLPID Controller

The tuning of the proposed NLPID controller is conducted using the PSO algorithm with the following objective function:

$$\begin{aligned} \underset{k_0,k_1,k_2,a}{compute}\ f(t,\epsilon(t),t_s) &= \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s \\ subject\ to\ 0 \leqslant & k_0, k_1, k_2 \leqslant 2 \\ 0.5 \leqslant & a \leqslant 2, \end{aligned} \tag{4.11}$$

where $t_s$ is the settling time, $\epsilon(t)$ is the feedback error, and $t_f$ is the final time.

The PSO algorithm searched for the nonlinear gain parameters $k_0, k_1, k_2$, and $a$ that reduce settling time, overshoot, and transient response as per the design constraints. The algorithm performs a randomised particle search, evaluating the objective function for each particle in the search space, selecting the one with the least value. The objective function values can potentially 'jump' due to the gain parameters crossing the specified boundaries and the parameters being randomly reset within their pre-specified range. The gain parameter $a$ must be tuned so that the overall proportional nonlinear gain is always positive for all feedback error values. The lowest value that the nonlinear proportional gain can achieve is when the system has reached steady state, hence $\epsilon(t) = 0$. This means that it is more efficient that the gain parameter $a$ follows the rule:

$$k_p(0, r(t)) = ak_0 - k_0 exp(-1) \geqslant 0 \implies a \geqslant exp(-1) = 0.3679 \tag{4.12}$$

Suppose the parameter does not follow that inequality. In that case, the search space is inefficient, and the search algorithm may take longer to tune the controller, and as the system will be unstable in the cases where $a \leqslant 0.3679$, the objective function evaluations will be extremely high, and hence, ignored. Therefore, the lowest bound of the parameter $a$ is 0.5 to include a small margin of difference. The upper bounds of the search space in all parameters have been selected based on trial and error to improve tuning speed.

Table 4.3 shows the parameters of the nonlinear controller gains as determined by the searching algorithm. The parameter $k_3$ is simpler to tune since it has predictable changes to the controller, effectively affecting only the amount of derivative action. Hence, it is unnecessary to use the PSO algorithm for tuning this parameter. Instead, it is determined by the designer after the rest of the controller parameters are tuned. The designer can choose a value for $k_3$ based on experience or based on the knowledge of the system. For example inherently integral systems would require PD control and having a $k_3$ of at least 1, would be beneficial. On the contrary, a low value of $k_3 = 0.5$ is used for systems where PI control usually suffices. Consequently, since FOPTD systems are well-known to be adequately controlled by a PI controller, since there is less need for a derivative action and a value of $k_3 = 0.5$ would be appropriate.

Figure 4.7 shows how the nonlinear gains vary with the feedback error of the system. The proportional gain starts from its maximum value where it then drops as it approaches the steady state value, converging to a specific gain. The integral gain starts from zero and increases rapidly during the transient response, settling to a converged value as the system reaches a steady state, providing error correction. It can be seen that the derivative starts from zero, eliminating any derivative kicks and noisy signals, while it increases as the system approaches steady-state providing the necessary speed of the system to eliminate overshoot.

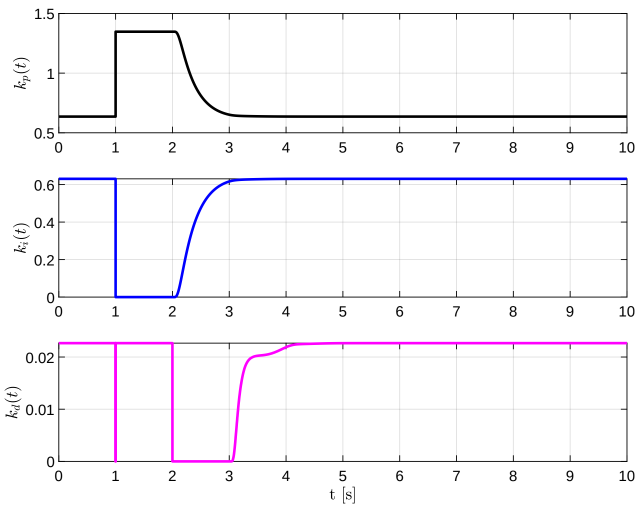



**Figure 4.7:** The tuned nonlinear gain values response to the FOPTD system simulation.

The performance of the proposed NLPID controller is further examined by using the steady-state values of the nonlinear function gains in a PID controller. The derivative filter of the PID controller is known to be $k_d \frac{N}{1-\frac{N}{s}} = k_d \frac{Ns}{s-N}$, which shows that the larger $N$ is, the larger derivative action produced. The smaller $N$ is, the smaller the derivative action produced. Consequently, the steady-state values of the proposed NLPID gains are directly substituted into the PID controller, using a large derivative filter parameter value $N = 100$ consistently for all of the systems. This tuning is provided in Table 4.1 as T2_PID and the gains $a, k_0, k_1, and, k_2$ are computed for all example cases and the FOPTD is uniquely used as an example to show that explicitly. However, to reduce the length of the chapter and avoid repetition it will not be shown explicitly in the other case studies. The design methodology for the T2_PID is defined as follows:

$$\begin{cases} k_{p_4} = ak_0 - k_0 exp(\frac{1}{0-1^2}) = ak_0 - k_0 exp(-1) = 0.6965 \times 1.9344 - 1.9344 \times exp(-1) \\ = 0.6357 \\ k_{i_4} = k_1 exp(\frac{1}{0-1^2}) = k_1 exp(-1) = 1.7142 exp(-1) = 0.6306 \\ k_{d_4} = k_2 exp(\frac{1}{0-k_3^2}) = 1.2373 exp(\frac{-1}{0.5^2}) = 0.0227 \end{cases}$$

The tuned parameters are summarised in Table 4.1.

**Table 4.1:** The tuned control parameter values used for the FOPTD benchmarking simulations.

| Controller | PSO Tuning Parameters |
|---|---|
| NLPID | $k_0 = 1.9344, k_1 = 1.7142, k_2 = 1.2373, k_3 = 0.5, a = 0.6965$ |
| T2_PID | $k_{p_4} = 0.6357, k_{i_4} = 0.6306, k_{d_4} = 0.0227, N_4 = 100$ |

The constraints of the optimisation algorithm have been verified to be stable based on the stability regions of each parameter, which has been established after extensive simulation trials. In addition, the constraints allow for the optimisation algorithm to find the desired performance tuning faster with less local optimal convergence.

**Tuning and Design of the PID, 2D_PID, and SP_PID Controllers**

The benchmarking uses the MATLAB PID and 2D_PID controllers, including derivative filtering, to reduce the derivative kick effects and improve control performance. MATLAB 2D_PID controller contains the set-point weighting parameters as a percentage of the input that contributes to the proportional and derivative actions. The transfer function expressions of MATLAB PID and 2D_PID are, respectively, as follows:

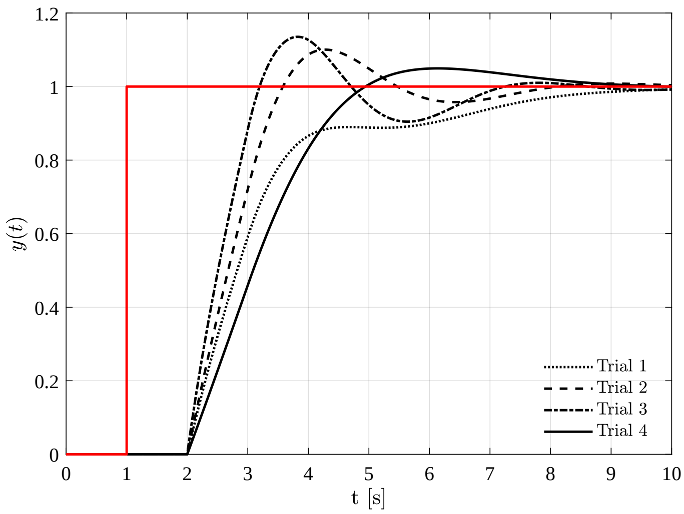
$$K_{\mathrm{PID}}(s) = k_{\mathrm{p}} + k_{\mathrm{i}}\frac{1}{s} + \frac{k_{\mathrm{d}}N}{1-\frac{N}{s}}, \tag{4.13}$$

$$K_{\mathrm{2D\_PID}}(s) = k_{\mathrm{p}}(br-y) + k_{\mathrm{i}}\frac{1}{s} + \frac{k_{\mathrm{d}}N}{1-\frac{N}{s}}(cr-y), \tag{4.14}$$

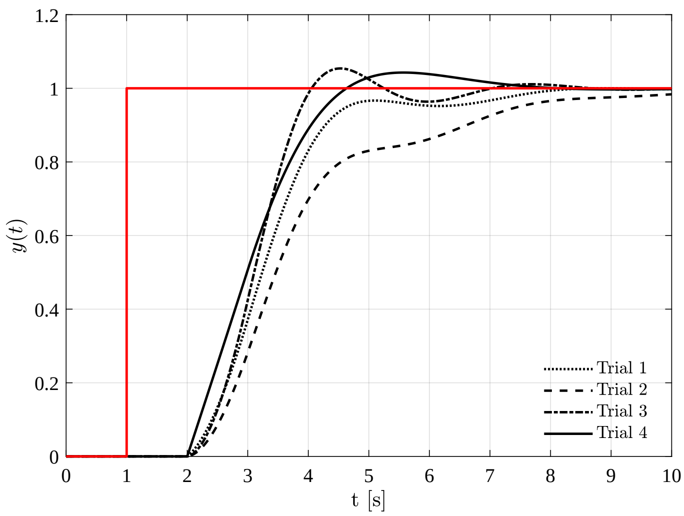
where $k_{\mathrm{p}}, k_{\mathrm{i}}$, and $k_{\mathrm{d}}$ are the proportional, integral, and derivative gains respectively, $N$ is the filtering parameter, which represents the inverse of the time constant of the filter, and in the case of 2-DoF control, $b$ and $c$ are set-point weightings. These are the respective parameters to be tuned by the algorithm according to the performance criteria. As can be seen, the PID controller has 4 parameters to be tuned, including the filter, whereas the 2D_PID controller has 6 parameters, including the set-point weighting.

A sequence of tuning trials have been conducted for each of the conventional controllers with 4 tuning trials explicitly shown. The 4 trials that are shown have been selected to show the common trade-off issues in the PID design when the goal is to minimise overshoot, settling time, and rise time. One of the four trials is selected based on the three aforementioned criteria, while also taking into consideration the inevitable trade-offs. The tuning trial that is selected, it can be purely on the basis of the objective function evaluation, however, it is not always clear that this is the best approach. Hence, it is beneficial to, in some cases, decide based on scrutiny and experience. When the trade-offs are significant, the tuning is selected based on minimising two of the criteria, while compromising the third. The two that can be minimised can depend on the system dynamics and can be quite varied, which is where experience can become useful.

The tuning trials conducted are shown in Figure 4.8. The PID and 2D_PID controllers can achieve a response containing 0.88% and 0% overshoot, respectively, a rise time of 2.36 and 2.01 seconds, and settling time of 5.42 and 5.07 seconds, respectively. If increased, overshoot and oscillations are presented for both controllers.




**(a)** MATLAB PID tuning trials for the FOPTD system.




**(b)** MATLAB 2D_PID tuning trials for the FOPTD system.

**Figure 4.8:** Different tuning trials for both the PID and 2D_PID controllers using MATLAB control system toolbox PID tuning algorithm.

An alternative method for controlling FOPTD systems is the Smith Predictor PID (SP_PID) controller. SP_PID controllers provide an improvement to the PID controller specific to delay systems, where the delay dynamics are predicted and then a PID controller is also used and tuned. Research indicates that this method provides improved results as compared to PID control in delay systems (Normey-Rico and Camacho, 2007; Zerong and Zhigang, 2017; Frank, 2018; Gnanamurugan and Senthilkumar, 2018; Normey-Rico, Santos et al., 2022). This provides a fair comparison and extends the simulations to more complex and industry-used control systems. With the use of the Smith predictor, one can achieve a faster response with a minimal overshoot that can improve the PID controller with a time-delay prediction step. The transfer function that represents the SP_PID controller is:

$$K_{\text{SP\_PID}}(s) = \frac{K_{\text{PID}}(s)}{1 + K_{\text{PID}}(s)T(s)(1 - G_p(s))}. \tag{4.15}$$

The SP_PID controller design is formulated using MATLAB PID controller as described by Eq. (4.13) with the FOPTD plant model as described by Eq. (4.16).

$$\begin{aligned} P(s) &= T(s)e^{-\tau s},\ \tau = 1\ seconds, \\ T(s) &= \frac{z}{b_1 s + 1},\ k = 1,\ t_n = 1\ seconds, \end{aligned} \tag{4.16}$$

where $z$ is the open loop plant gain , $b_1$ is the lag, and $\tau$ is the time delay.

The time delay $e^{-\tau s}$ is approximated by the second order Pade transfer function $G_{\text{p}}(s)$ for the design of the SP_PID controller. The general second-order Pade transfer function of time delay is as follows:

$$G_{\text{p}}(s) = \frac{\tau^2 s^2 - 6\tau s + 12}{\tau^2 s^2 + 6\tau s + 12}. \tag{4.17}$$

The different MATLAB tuning trials of the SP_PID controller are shown in Figure 4.9. The SP_PID controller has a rise-time of 1.22 seconds, settling time of 6.24 seconds, and an overshoot of about 3.4%, showing oscillatory response. The overshoot of the SP_PID controller can be reduced, however at the cost of also reducing rise time and settling time. The SP_PID controller produces a non-smooth rise time, which can negatively affect the system input and may produce technical difficulties in practice.

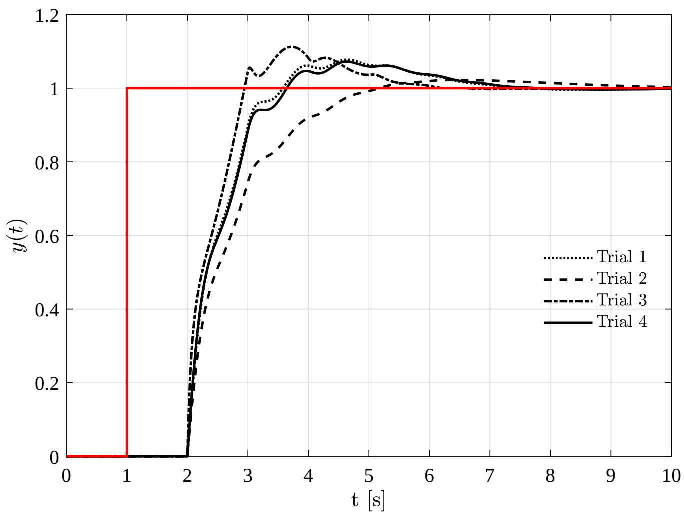



**Figure 4.9:** MATLAB SP_PID tuning trials for the FOPTD system.

The tuning trials in this section show the adequacy of MATLAB tuning algorithm, but also it indicates the limitations of the PID, 2D_PID, and SP_PID controllers at different tuning paradigms for the FOPTD system. After extensive tuning trials, efforts were made to increase controller speed and reduce overshooting.

Table 4.2 shows the objective function evaluations at each tuning trial for the conventional controllers to indicate on the selection criteria. Trial 4 has the lowest objective function evaluation for the PID and 2D_PID controllers, consequently they were chosen on the bases of objective costs and the performance criteria that they have the lowest rise time and settling time of all other tuning trials. Trial 4 of the SP_PID controller does not have the lowest objective function value, however, that is because it has the best compromise between overshoot and settling time.

**Table 4.2:** The objective function evaluations for the individual tuning trials of the conventional controllers for the FOPTD benchmarking simulations.

| | $f(t, \epsilon(t), t_s)$ | | | |
|---|---|---|---|---|
| Controller | Trial 1 | Trial 2 | Trial 3 | Trial 4 |
| PID | 174.3512 | 118.5859 | 175.2632 | 88.0333 |
| 2D_PID | 551.1609 | $1.7733 \times 10^3$ | $1.3580 \times 10^3$ | 100.2722 |
| SP_PID | 310.9030 | 362.1999 | 355.8497 | 317.6669 |

Table 4.3 shows the tuned parameters of each controller together with their respective objective function evaluations. It can be seen that the proposed NLPID controller has a similar objective function evaluation as the 2D_PID controller. The PID controller shows to have the lowest objective function value, while the SP_PID and T2_PID controllers have the largest objective function evaluations.

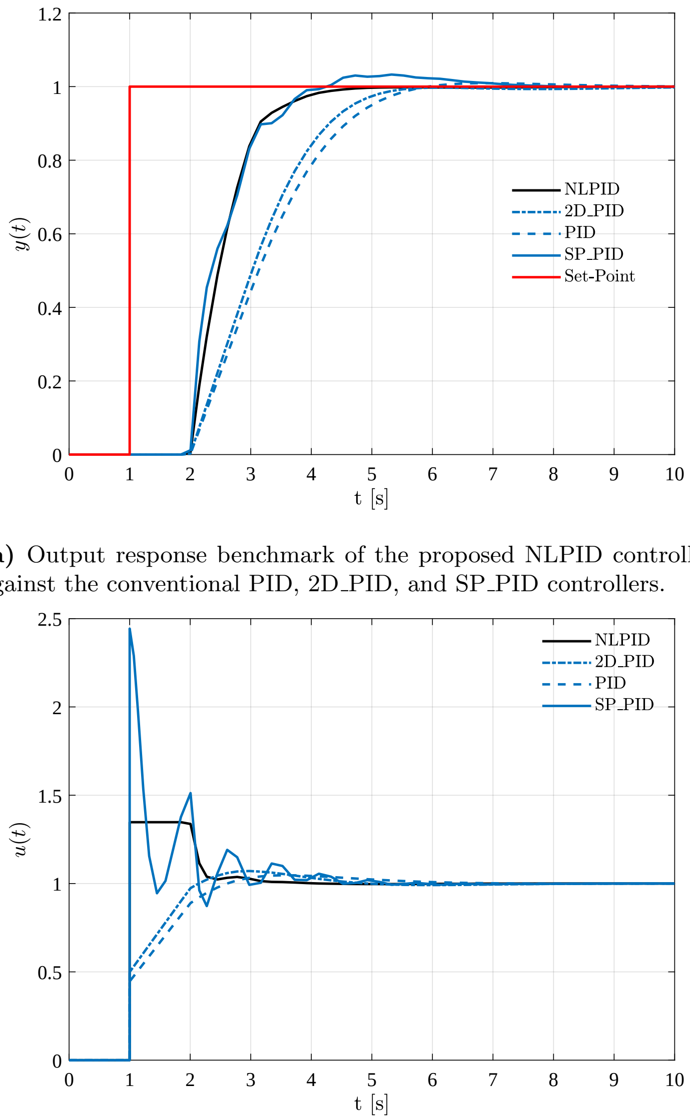
**Table 4.3:** The tuned control parameter values used for the FOPTD benchmarking simulations.

| Controller | PSO Tuning Parameters | $f(t, \epsilon(t), t_s)$ |
|---|---|---|
| NLPID | $k_0 = 1.9344, k_1 = 1.7142, k_2 = 1.2373, k_3 = 0.5, a = 0.6965$ | 107.75 |
| | MATLAB Tuning Parameters | |
| PID | $k_{\mathrm{p}_1} = 0.4458, k_{\mathrm{i}_1} = 0.4422, k_{\mathrm{d}_1} = 0, N = 0$ | 88.0333 |
| 2D_PID | $k_{\mathrm{p}_2} = 0.5308, k_{\mathrm{i}_2} = 0.4743, k_{\mathrm{d}_2} = 0, N_2 = 0, b = 0.9400, c = 0$ | 100.2722 |
| SP_PID | $k_{\mathrm{p}_3} = 1.4089, k_{\mathrm{i}_3} = 2.1239, k_{\mathrm{d}_3} = 0.4227, N_3 = 2.4471$ | 317.6669 |
| T2_PID | $k_{\mathrm{p}_4} = 0.6357, k_{\mathrm{i}_4} = 0.6306, k_{\mathrm{d}_4} = 0.0227, N_4 = 100$ | 595.47 |

In the next section, the benchmarking of the proposed controller against conventional control algorithms is shown, with the system input energy computed as a comparison for benefits to actuation and for internal stability.

#### Controller Benchmarking to Servo and Regulator Performance

It can be seen by Figure 4.10a that the proposed NLPID controller outperforms all the conventional control methods and produces no overshoot with the fastest rise time and settling time of 1.07 and 4.06 seconds, respectively. This outperforms both the PID and 2D_PID. Figure 4.10a shows that the conventional controllers produce a response with minimal overshoot, however, they have a significantly slower settling and rise time of up to 2.24 seconds difference. The proposed NLPID controller improves the transient response speed while maintaining no overshoot. Figure 4.10b shows the system input to the plant model for each of the benchmarked controllers. The NLPID controller shows bounded signals, eliminating the derivative kicks. However, the SP_PID controller shows a large sharp input that deteriorates the controller performance and can degrade actuators. In contrast, the PID and 2D_PID controllers show a slow response with a smooth input to the process model.



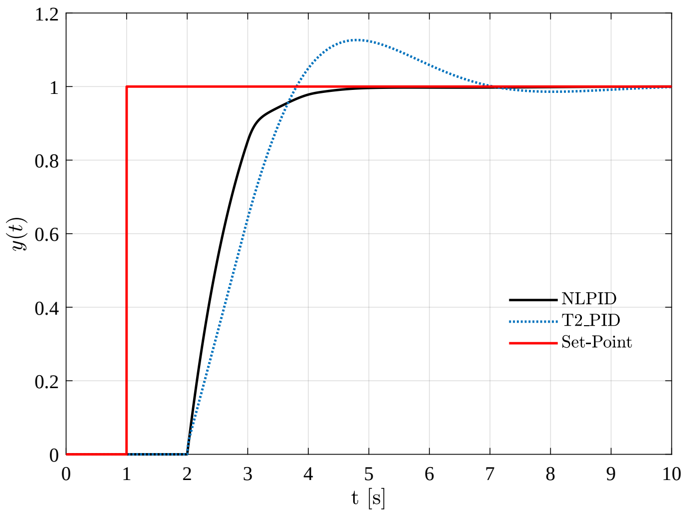

**(a)** Output response benchmark of the proposed NLPID controller against the conventional PID, 2D_PID, and SP_PID controllers.



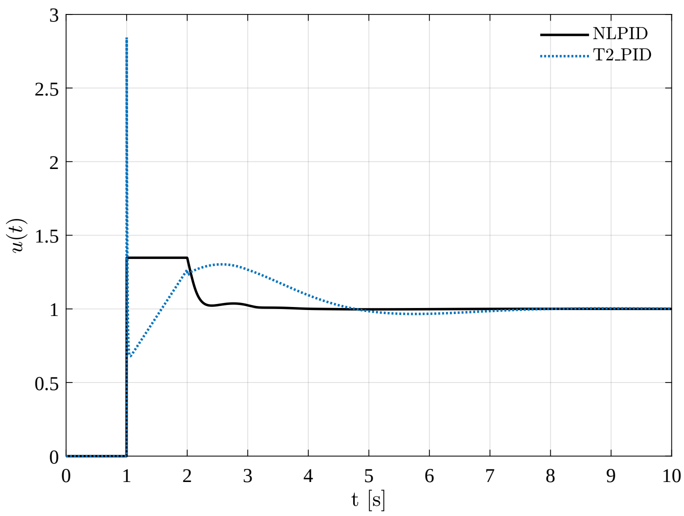
**(b)** System input signals of the benchmarked controllers.

**Figure 4.10:** Performance comparison of the proposed NLPID controller for step set-point function against the conventional PID, 2D_PID, and SP_PID control of the FOPTD system.

Figure 4.11a shows that the T2_PID controller generates an overshoot of 13.6% with a rise time and settling time of 1.34 and 6.42 seconds, respectively. This also indicates the limitations of PID control where once the performance is optimized for fast disturbance rejection it provides a fast transient response with an overshoot. Figure 4.11b shows the system inputs for the proposed NLPID and T2_PID controllers. It can be seen that the T2_PID controller produces a derivaive kick when the set-point is active. Both controllers are internally stable and provide a smooth response.




**(a)** Output response benchmark of the proposed NLPID controller against the nonlinear gains steady-state values substitute PID tuning.

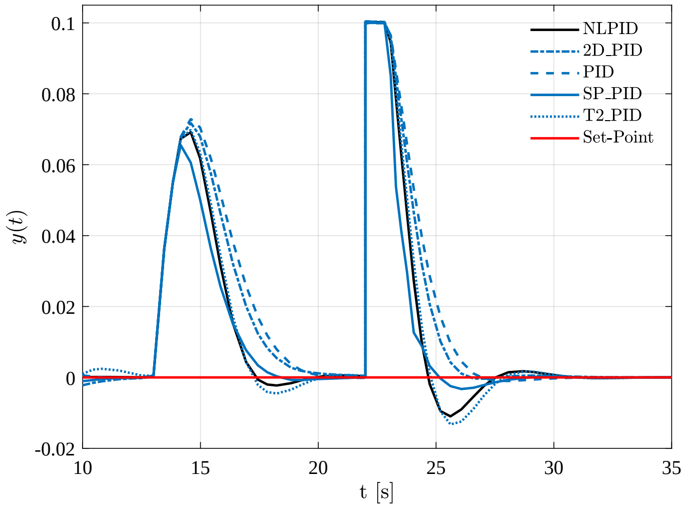



**(b)** System input of the benchmark.

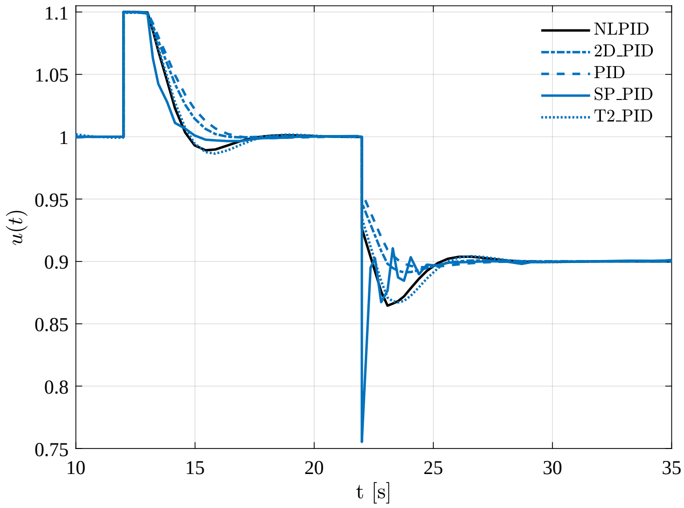
**Figure 4.11:** Performance comparison of the proposed NLPID controller against the nonlinear gains steady-state values substitute PID tuning for the FOPTD system.

The proposed NLPID controller has shown its ability to outperform the conventional control algorithms in the transient response of the FOPTD system. The benchmark testing for disturbance rejection of the FOPTD system is conducted using the input and output disturbances as 10% of the set-point value, applied to the system input at 12 seconds time mark and in the system output at 22 seconds time mark, as shown by Figure 4.4.

Figure 4.12 shows the response to disturbance rejection by the benchmarked controllers. Figure 4.12a indicates the deviation of the steady-state value produced by the disturbances. The figure indicates that the proposed NLPID controller outperforms SP_PID and T2_PID producing a faster settling time for the output disturbance rejection and outperforms all controllers for input disturbance rejection. Figure 4.12b indicates that the SP_PID disturbance rejection comes with a cost of more vibratory system input, as compared to the proposed NLPID control.




**(a)** Output response of both input and output disturbance rejection of the proposed NLPID controller benchmark against the conventional PID, 2D_PID, SP_PID, and T2_PID controllers.




**(b)** System input response of the benchmarked controllers to input and output disturbance rejection.

**Figure 4.12:** Disturbance rejection benchmarking of the proposed NLPID controller against the conventional PID, 2D_PID, SP_PID, and T2_PID controllers for the FOPTD system.

Using Eq. (4.3) and Eq.(4.4), the control signal energy and feedback error variance are computed, with the feedback measures and performance comparison summarised in the table below as follows:

**Table 4.4:** Performance comparison evaluation summary table for the FOPTD system.

| FOPTD | NLPID | PID | 2D_PID | SP_PID | T2_PID |
|---|---|---|---|---|---|
| $\%Os$ | 0 | 0.88 | 0 | 3.37 | 13.63 |
| $t_r(s)$ | 1.07 | 2.36 | 2.09 | 1.22 | 1.34 |
| $t_s(s)$ | 4.06 | 5.42 | 5.07 | 6.24 | 6.42 |
| $L_2$ | 10.90 | 9.30 | 9.37 | 10.56 | 36.04 |
| $MSE$ | 0.13 | 0.17 | 0.18 | 0.13 | 0.15 |

The proposed NLPID controller shows similar input energy to the SP_PID controller while showing significant improvements in both overshoot and settling time. The T2_PID controller has a larger $L_2$ norm evaluation because of the observable derivative kick. Moreover, all controllers have a similar MSE measure, with the proposed NLPID and the SP_PID controllers having the lowest MSE measure when compared to the PID and 2D_PID controllers. This indicates that the error signal is smooth, without significant variations. Extensive benchmarking simulations conducted under various feedback conditions, which form a Simulation-based Extensive Testing (SET) approach to performance and stability, the proposed NLPID controller shows a finite $L_2$ norm, indicating system internal stability.

The proposed NLPID controller has been compared against the conventional control methods in a FOPTD system linearisation case. The proposed NLPID controller shows performance improvements and maintains fast disturbance rejection with a comparable $L_2$ energy to conventional controllers. However, it can be seen that the control performance and the input signal energy are competing requirements. This indicates that the advantage of the proposed controller is that it improves the performance at very little added energy cost. NCSTR models possess multiple dynamics, with the FOPTD being the most fundamental element of its dynamics. However, it is a less accurate representation of the NCSTR linearised dynamics. As a result, Second Order Plus Time Delay system models are compared including negative gain, non-minimum phase zeros, and second-order dynamics. The next case study is the Negative Gain SOPTD (NG_SOPTD) system, which is a more accurate linearisation of an NCSTR at an equilibrium point.

### 4.5.3 The Negative Gain SOPTD (NG_SOPTD) Case Study

Reverse dynamics are a complex control problem that is frequently seen in industry and can also represent linearisation of the NCSTR model that is common to chemical industries (Colantonio et al., 1995; Sinha and Mishra, 2018; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). The proposed NLPID controller is benchmarked against conventional control methods in this system to show improvements to transient behaviour and disturbance rejection. The model example is a linearization of the NCSTR model at the point

$c_A = 0.0644[lbmol/ft^3], T = 560.77R, T_i = 539.67R$. The model dynamics contain a negative gain, which makes the negative feedback loop into a positive feedback loop, which is unstable. As a result, the control algorithm and the gains become negative, to establish stability of the system and this takes a unique tuning approach for linear controllers. The transfer function of this model is expressed as (Krishna et al., 2012):

$$P(s) = \frac{-0.8775s - 8.774}{s^2 + 2.674s + 10.97} e^{-108.72s}. \tag{4.18}$$

**Design and Tuning of the NLPID Controller**

The filtering parameter of the NLPID controller is $k_3 = 1$ to promote derivative action to reduce the overshoot. The objective function and search algorithm constraints for the tuning are the following:

$$\begin{aligned} \underset{k_0,k_1,k_2,a}{compute}\; & f(t,\epsilon(t),t_s) = \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s \\ subject\ to\; & 0 \leqslant k_0, k_2 \leqslant 3 \\ & 0 \leqslant k_1 \leqslant 8 \\ & 0.3 \leqslant a \leqslant 1, \end{aligned} \tag{4.19}$$

where $t_s$ is the settling time, $\epsilon(t)$ is the feedback error, and $t_f$ is the final time.

The constraints have been defined based on the stability regions of each parameter, which have been established after extensive simulation trials. In addition, the integrator gain has a different constraint range because the system requires larger integral action to eliminate steady-state errors, and the upper bound is derived via extensive trial and error. The parameter bounds for $k_0, k_2$ have been defined to reduce the search region, effectively improving tuning speed. Moreover, the lower bound of $a$ has been defined according to the rule of Eq.(4.12) with a lower value than the constraint because the NG_SOPTD system has improved performance when $a$ is near the lowest possible bound. The lowest evaluation is taken by the PSO algorithm and the respective tuning is used. This approach is an automatic trial and error. The PSO algorithm determined the parameters for the nonlinear controller gains, which are shown in Table 4.5.

Figure 4.13 shows the time variation of the nonlinear gain functions, and it can be seen that the proposed nonlinear gain functions have been mirrored along the x-axis, using negative gain parameters, as appropriate for the system. The nonlinear function gains start from their steady-state values, as the initial error is 0. As the value of the error changes, the nonlinear proportional gain rises to its maximum value and then drops into its steady state value as the error reaches zero again. The nonlinear integral gain drops to zero and then rises to its maximum value at a steady state to correct for any steady-state error. Then, the derivative gain shows a similar behaviour to the nonlinear integral gain, providing a fast response and lowering the overshoot as the error tends towards a steady state. The major observation to make in this case is the value of the nonlinear proportional gain at the steady-state. It can be seen that the proportional gain reaches close to zero, whereas the integral and derivative gains are much larger in magnitude in comparison.

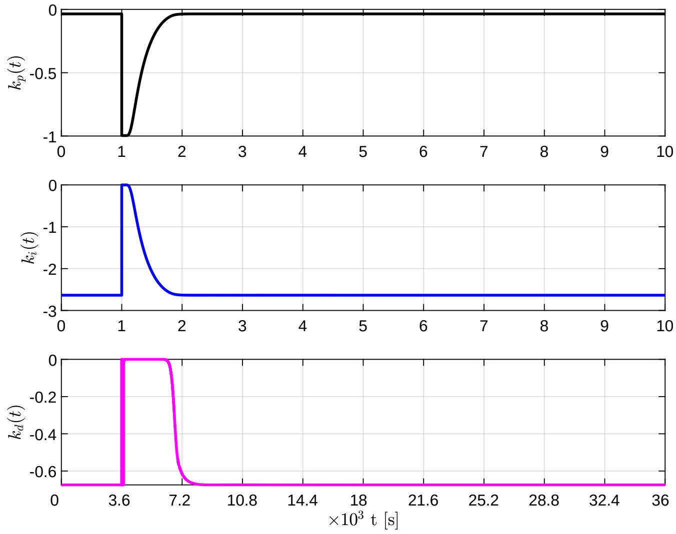



**Figure 4.13:** The tuned nonlinear gain values response to the NG_SOPTD system simulation.

The steady-state values of the nonlinear gains are used to tune a separate PID controller. This tuning can then show whether the nonlinear functions are providing the suggested improvement to the response. The new tuning values are shown in Table 4.5 as T2_PID.

**Table 4.5:** The tuned control parameter values used for the NG_SOPTD benchmarking simulations.

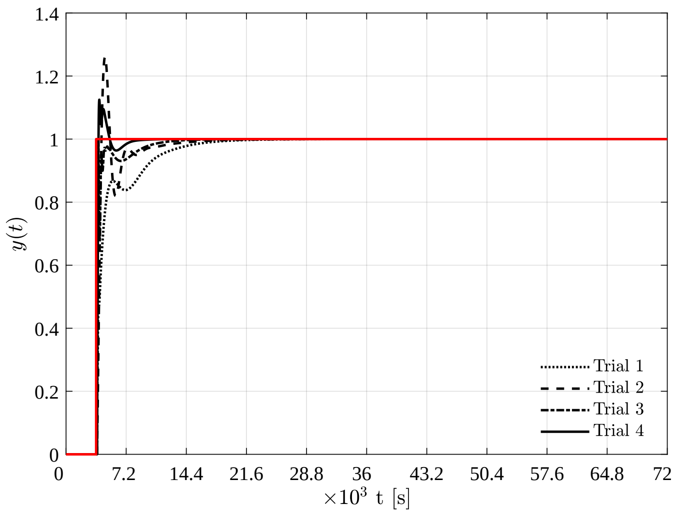
| Controller | PSO Tuning Parameters |
|---|---|
| NLPID | $k_0 = -2.6119, k_1 = -7.1640, k_2 = -1.8336, k_3 = 1, a = 0.3815$ |
| T2_PID | $k_{\mathrm{p}4} = -0.0356, k_{\mathrm{i}_4} = -2.6355, k_{\mathrm{d}_4} = -0.6746, N_4 = 100$ |

The NG_SOPTD system has a large derivative gain when compared to the other systems, as a result, there is a larger impact from derivative kicks that can cause instabilities. To correct that, a large filtering value is used to reduce the likelihood of instability caused by derivative side effects. Hence, the parameter $N_4$ has been selected at a larger value than $k_3$ and it is also much larger than the filtering value of the T2_PID controllers used in the FOPTD and NmP_SOPTD system models. This is also in agreement with the MATLAB tuning algorithm that has tuned the PID, 2D_PID, and SP_PID controllers with a large derivative gain $k_d$ and filtering constant $N$.

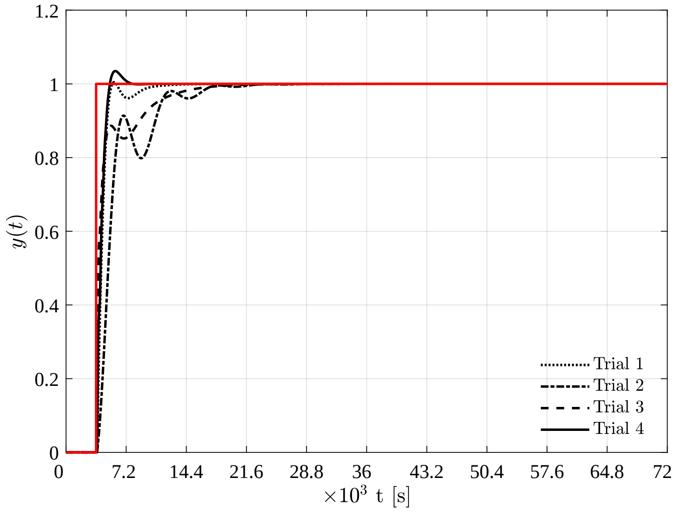
The stability regions of the tunable parameters of the proposed controller $k_0, k_1,$ and $k_2$ for the NG_SOPTD system have been verified to be stable for which the feedback interconnection of the proposed NLPID controller with an NG_SOPTD system remains stable. This also shows that the simulation confirms the optimisation constraints of the PSO algorithm and justifies the search regions of the optimisation algorithm.

#### Tuning and Design of the PID, 2D_PID, and SP_PID Controllers

The conventional controllers are tuned using MATLAB tuning algorithm with their different tuning trials shown in Figure 4.14. Figure 4.14a shows that the PID controller presents a 12.49% overshoot with a rise time and settling time of 144 and $6.95 \times 10^3$ seconds, respectively. The 2D_PID controller shows a lower overshoot of 3.46% with a rise time and settling time of $1.06 \times 10^3$ and $6.65 \times 10^3$ seconds, respectively. Figure 4.14b shows that the 2D_PID controller can achieve a response that contains minimal overshoot and has a similar settling time as the PID controller. However, the 2D_PID controller indicates a slower rise time that when increased, presents an overshoot.




**(a)** MATLAB PID tuning trials for the NG_SOPTD system.




**(b)** MATLAB 2D_PID tuning trials for the NG_SOPTD system.

**Figure 4.14:** Different tuning trials for both the PID and 2D_PID controllers using MATLAB control system toolbox PID tuning algorithm.

An SP_PID controller has also been designed for the NG_SOPTD model. The transfer function that represents the SP_PID controller is:

$$K_{\text{SP\_PID}}(s) = \frac{K_{\text{PID}}(s)}{1 + K_{\text{PID}}(s)T(s)(1 - G_p(s))}. \tag{4.20}$$

The SP_PID controller design is formulated using MATLAB PID controller as described by Eq. (4.13) with the FOPTD plant model as described by Eq. (4.21).

$$\begin{aligned} P(s) &= T(s)e^{-\tau s},\ \tau = 108.72\ seconds, \\ T(s) &= \frac{z_1 s + z_0}{q_2 s^2 + q_1 s + q_0}, \end{aligned} \tag{4.21}$$

where $z_1 = -0.8775, z_0 = -8.774, q_2 = 1, q_1 = 2.674, q_0 = 10.97$. The time delay $e^{-\tau s}$

is approximated by the second order Pade transfer function $G_{\text{p}}(s)$ for the design of the SP_PID controller. The general second-order Pade transfer function of time delay is as follows:

$$G_{\text{p}}(s) = \frac{\tau^2 s^2 - 6\tau s + 12}{\tau^2 s^2 + 6\tau s + 12}. \tag{4.22}$$

MATLAB algorithm is used to tune the SP_PID controller with its tuning trials shown in Figure 4.15. It can be seen that the SP_PID controller indicates a faster rise-time of 781.2 seconds with the presence of 3.73% overshoot and a slower settling time of $9.07 \times 10^3$ seconds compared to the PID and 2D_PID controllers.

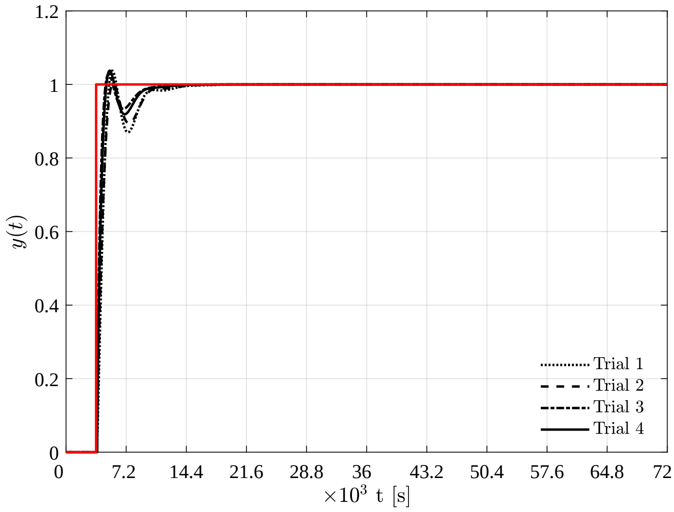



**Figure 4.15:** MATLAB SP_PID tuning trials for the NG_SOPTD system.

Table 4.6 shows the objective function evaluations of every trial for each controller. It can be seen that Trial 4 has the lowest objective function value for the PID controller, however, that is not the case for the 2D_PID and SP_PID controllers. For the 2D_PID controller, Trial 4 was selected on the bases that it has the lowest settling time and rise time at a very small cost of some additional overshoot. Finally, Trial 4 was selected for the SP_PID controller on the same basis that it has the lowest settling time and rise time when compared to all other trials at a bare minimum additional cost and with minimal differences to the other tunings.

**Table 4.6:** The objective function evaluations for the individual tuning trials of the conventional controllers for the NG_SOPTD benchmarking simulations.

| | $f(t, \epsilon(t), t_s)$ | | | |
|---|---|---|---|---|
| Controller | Trial 1 | Trial 2 | Trial 3 | Trial 4 |
| PID | $1.3898 \times 10^3$ | 746.7770 | $1.0200 \times 10^3$ | 688.0489 |
| 2D_PID | $1.0126 \times 10^3$ | $1.0164 \times 10^3$ | $1.1735 \times 10^3$ | $1.5059 \times 10^3$ |
| SP_PID | $2.4564 \times 10^3$ | $2.4069 \times 10^3$ | $2.1556 \times 10^3$ | $2.2183 \times 10^3$ |

The values of the tuned parameters are shown in Table 4.7 for all the designed controllers together with their respective objective function evaluations. It is worth noting that the 2D_PID controller has the lowest objective function evaluation, also showing the lowest settling time and with similar overshoot to the proposed NLPID controller. Moreover, it can be seen that the objective function evaluations of the conventional controllers are similar to the proposed NLPID controller. The T2_PID controller has the largest objective function evaluation, showing that it provides a worsened response in this system.

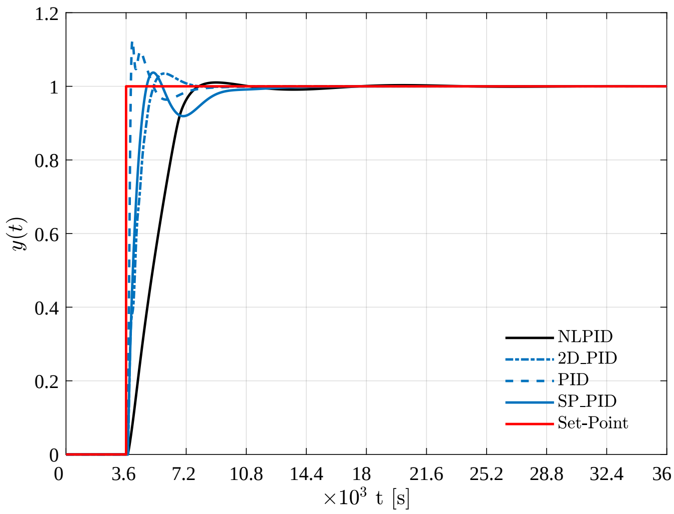
**Table 4.7:** The tuned control parameter values used for the NG_SOPTD benchmarking simulations.

| Controller | PSO Tuning Parameters | $f(t, \epsilon(t), t_s)$ |
|---|---|---|
| NLPID | $k_0 = -2.6119, k_1 = -7.1640, k_2 = -1.8336, k_3 = 1, a = 0.3815$ | $1.8512 \times 10^3$ |
| Controller | MATLAB Tuning Parameters | |
| PID | $k_{\mathrm{p}_1} = -10.2457, k_{\mathrm{i}_1} = -25.8340, k_{\mathrm{d}_1} = -0.7797, N = 23.0091$ | $1.5185 \times 10^3$ |
| 2D_PID | $k_{\mathrm{p}_2} = -7.7668, k_{\mathrm{i}_2} = -12.8018, k_{\mathrm{d}_2} = -1.0504, N_2 = 48.1072,$ | |
| | $b = 0.9083, c = 0.0301,$ | 692.6310 |
| SP_PID | $k_{\mathrm{p}_3} = -3.9580, k_{\mathrm{i}_3} = -8.0589, k_{\mathrm{d}_3} = -0.3268, N_3 = 20.4265$ | $2.2197 \times 10^3$ |
| T2_PID | $k_{\mathrm{p}_4} = -0.0356, k_{\mathrm{i}_4} = -2.6355, k_{\mathrm{d}_4} = -0.6746, N_4 = 100$ | $2.4807 \times 10^3$ |

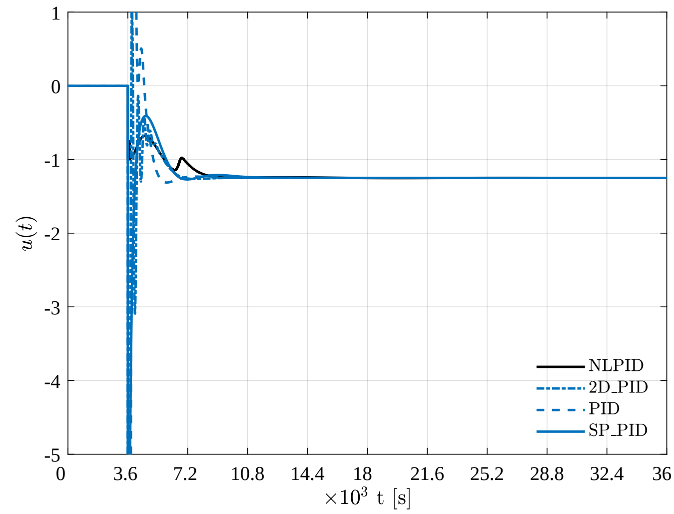
In the following section, the simulations of the controllers to the NG_SOPTD system are shown for the set-point tracking and disturbance rejection, with the computation of the signal energy for each controller to show internal stability and energy consumption.

#### Controller Benchmarking to Servo and Regulator Performance

Figure 4.16a shows that the proposed NLPID controller provides an overshoot of 1.03% and a fast transient response with a rise time and settling time of $2.66 \times 10^3$ seconds and $7.42 \times 10^3$ seconds, respectively. This shows a balance in the transient response when compared to the linear controllers. The proposed NLPID controller manages to reduce the overshoot at the cost of a higher settling time than the PID and 2D_PID controllers. However, it provides an improved response when compared to the SP_PID controller in terms of both overshoot and settling time. Although the PID controller has the shortest rise time, it provides a large overshoot of 12.49%, as expected. The advantages of 2DoF control are presented in this example where the 2D_PID controller significantly reduces the overshoot to 3.46% with a similar settling time as PID. Although the proposed controller has higher rise time and settling time when compared to the conventional controllers, its performance can be improved by weighting the objective function, prioritizing the settling time rather than the overshoot. This will then reduce the response speed at the sacrifice of some overshoot, which can then lead to performance closer to what is achieved by the conventional controllers. Figure 4.16b shows the system input of all the controllers indicating that they have internally stable control, with the NLPID controller providing the cheapest system input.



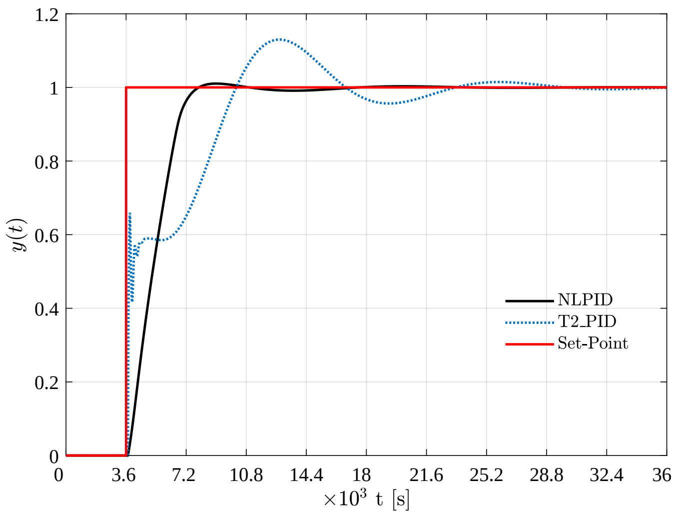

**(a)** Output response benchmark of the proposed NLPID controller against the conventional PID, 2D_PID, and SP_PID controllers.



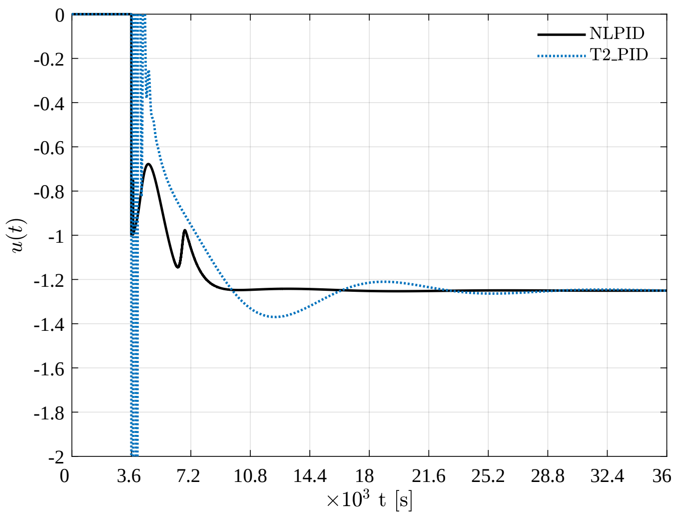

**(b)** System input signals of the benchmarked controllers.

**Figure 4.16:** Performance comparison of the proposed NLPID controller for step set-point function against the conventional PID, 2D_PID, and SP_PID control of the NG_SOPTD system.

The second tuning approach to the PID controller, which uses the steady-state values is benchmarked against the NLPID controller. This attempt is made to show that the proposed performance improvement is dependent on the proposed nonlinear function gains. Figure 4.17a shows the output response of the benchmarking, and it shows that the proposed NLPID controller has a low overshoot with minimal oscillations when compared against the PID controller, which shows an overshoot of 9.77% with rise and settling time of $5.47 \times 10^3$ and $21.2 \times 10^3$ seconds, respectively. In addition, this shows that the steady-state values of the proposed NLPID controller are not necessarily providing adequate system response. Figure 4.17b shows that the T2_PID controller produces larger system input signals, whereas the NLPID controller shows cheaper system input. This indicates that the proposed nonlinear function gains play a key role in the proposed performance improvements, so much so, that if constant gains are used the performance collapses and shows significant deterioration.



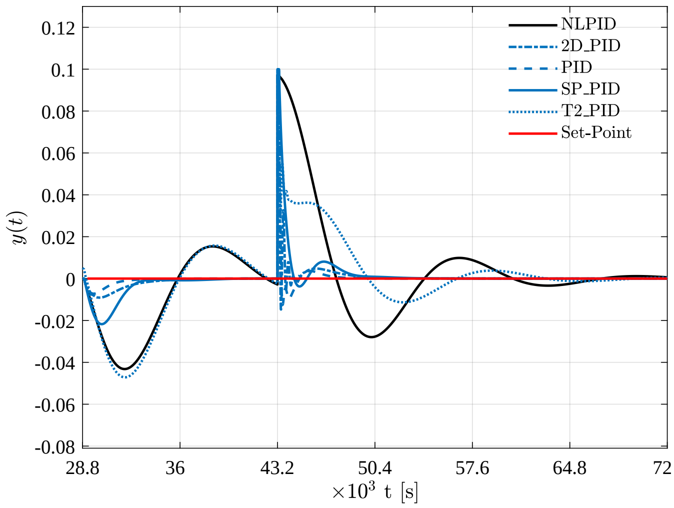

**(a)** Output response benchmark of the proposed NLPID controller against the nonlinear gains steady-state values substitute PID tuning.



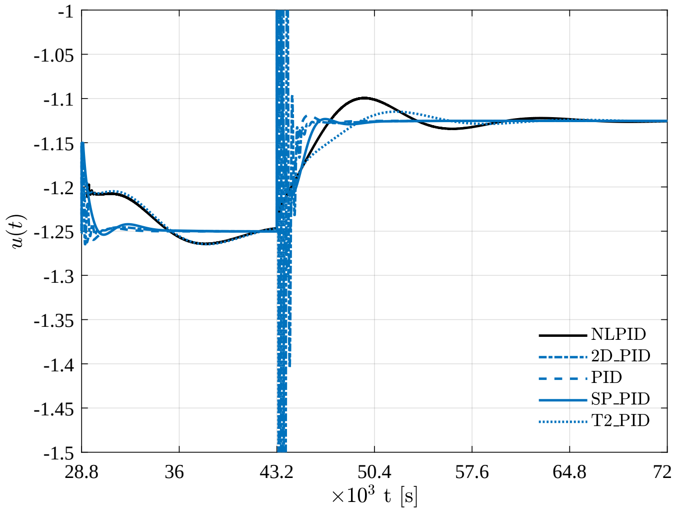

**(b)** System input of the benchmark.

**Figure 4.17:** Performance comparison of the proposed NLPID controller against the nonlinear gains steady-state values substitute PID tuning for the NG_SOPTD system.

To test the system's internal stability and disturbance rejection, input and output disturbances are applied simultaneously with 10% magnitude of the set-point, and they act after settling time, with the input disturbance at $28.8 \times 10^3$ seconds and the output disturbance at $43.2 \times 10^3$ seconds. Figure 4.18a shows that the proposed NLPID controller suffers from poor disturbance rejection in both input and output disturbances, showing the slowest and most oscillatory response than all the conventional control methods. In addition, Figure 4.18b shows the system input signals and it shows that all the controllers are internally stable and similarly the proposed NLPID controller shows the cheapest system input signal response. It indicates that the proposed NLPID controller improves set-point tracking, whereas the disturbance rejection suffers from large oscillations and settling time.




**(a)** Output response of both input and output disturbance rejection of the proposed NLPID controller benchmark against the conventional PID, 2D_PID, SP_PID, and T2_PID controllers.




**(b)** System input response of the benchmarked controllers to input and output disturbance rejection.

**Figure 4.18:** Disturbance rejection benchmarking of the proposed NLPID controller against the conventional PID, 2D_PID, SP_PID, and T2_PID controllers for the NG_SOPTD system.

The disturbance rejection study shows that the objective function defined to tune the proposed NLPID controller is not designed to produce an effective disturbance rejection and the proposed nonlinear function gains are also mostly focused on producing fast transient response rather than disturbance rejection. There are two possible ways that this can be mitigated, on one hand, the nonlinear integral function could increase rather than decrease with increasing error, similar to the nonlinear proportional function, however, that would result in responses that contain large overshoots. The other way, is to introduce a different objective function such as the integral squared error, that could be more effective in finding a tuning that is effective for disturbance rejection.

Table 4.8 shows the $L_2$ gains of the system input signals for all the controllers together with their performance comparison. The proposed NLPID controller has the cheapest response with a significant difference against the PID and T2_PID controllers.

**Table 4.8:** Performance comparison evaluation summary table for the NG_SOPTD system.

| NG_SOPTD | NLPID | PID | 2D_PID | SP_PID | T2_PID |
|---|---|---|---|---|---|
| $\%Os$ | 1.03 | 12.49 | 3.47 | 3.73 | 9.77 |
| $t_r(h)$ | 0.74 | 0.04 | 0.29 | 0.22 | 1.52 |
| $t_s(h)$ | 2.07 | 1.94 | 1.85 | 2.52 | 5.88 |
| $L_2$ | 80.70 | 258.37 | 120.16 | 137.66 | 150.42 |
| $MSE$ | 0.03 | 0.01 | 0.005 | 0.01 | 0.03 |

In the case of NG_SOPTD systems, it is shown that the proposed NLPID controller can provide minimal overshoot with improved performance when compared to the SP_PID controller with the cheapest system input energy. It can be seen that the proposed controller provides significant improvements in overshoot at a cheaper energy consumption. However, the disturbance rejection of the proposed controller provides an oscillatory response compared to the conventional control methods. Consequently, the main advantage of using the proposed controller in negative gain systems is that it can reduce overshoot, effectively reducing the degradation of actuators and improving smoothness in the transient response. However, due to the disadvantage of reduced disturbance rejection, the controller should be used in a controlled environment, such as a manufacturing plant, where disturbances are far less likely to occur. The conventional controllers show improved disturbance rejection. In addition, the PID and 2D_PID controllers have the best rise-time and settling time with low MSE measure also indicating smooth error signals. The PID controller can reduce the rise time, however, it presents the worst overshoot. The benefits of the 2D_PID controller become apparent as it can reduce that overshoot significantly and maintain a similar rise time and settling time as PID. Finally, the controllers have a similar MSE evaluation except for the 2D_PID controller, which has the lowest evaluation, indicating that it has a smooth signal with low variation.

### 4.5.4 The Non-minimum Phase SOPTD (NmP_SOPTD) Case Study

The non-minimum phase dynamics pose a complex control problem that often suffers from design trade-offs. The NmP_SOPTD system is common to many industries and causes difficulties in control tuning (O'Dwyer, 2009; Chen, J., Fang and Ishii, 2019). The NmP_SOPTD system is also common in NCSTR linearised system dynamics since the NCSTR system is an inherently non-minimum phase and second order system (Colantonio et al., 1995; Sinha and Mishra, 2018; Pugazhenthi P, Selvaperumal and Vijayakumar, 2021). These plant dynamics are intrinsically complex and possess servo and regulator design trade-offs for the PID controller since they contain multiple non-minimum phase zeros (Alcántara, Vilanova and Pedret, 2013; Chen, J., Fang and Ishii, 2019). The NmP_SOPTD has second-order dynamics with unstable zeros and a measurement delay of 0.2 seconds. The plant model transfer function is expressed as (Krishna et al., 2012):

$$P(s) = \frac{1 - 1.6s}{s^2 + 2s + 1} e^{-0.2s}. \tag{4.23}$$

#### Design and Tuning of the NLPID Controller

The design of the NLPID controller is accomplished using the PSO algorithm that is described in Section 3.3.5. The derivative gain is not large enough to cause deteriorated performance or instabilities. In addition, the derivative action is needed to reduce the overshoot. Hence, the filtering parameter is selected as $k_3 = 1$ to reduce overshoot. The objective function is the following:

$$\begin{aligned} \underset{k_0,k_1,k_2,a}{\mathit{compute}}\ f(t, \epsilon(t), t_s) &= \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s \\ \mathit{subject\ to}\ 0 \leqslant & k_0, k_1, k_2 \leqslant 1 \\ 0.5 \leqslant & a \leqslant 1.4, \end{aligned} \tag{4.24}$$

where $t_s$ is the settling time, $\epsilon(t)$ is the feedback error, and $t_f$ is the final time.

The constraints have been selected on the basis of extensive simulation trials and attempts to tune the proposed NLPID controller. The optimisation constraints provide stable tuning values and refine the region of parameter search space to reduce the chances of the PSO algorithm getting trapped in local optima. The constraint boundaries are selected based on extensive trial and error. It can be helpful to initially assume that $k_0, k_1,$ and $k_2$ share a common boundary, however, this may not always be the case, and different cases have to be tried. The general rule of thumb is that the search space boundaries can be refined to speed up the search. The bounds can be shared among the three gain parameters $k_0, k_1,$ and $k_2$, if the search bound is within the unit distance from the origin. There is no need for further refinement in such scenarios. The lower bound of $a$ has been defined according to the rule of Eq.(4.12) with a small margin of difference. The upper bound of parameter $a$ have been selected based on trial and error. Although the selected value of $a$ might be within the unit distance, it is often helpful for the random search algorithm to extend the search space, as the random values are selected based on a normal distribution, it becomes less likely to initialise random particles near the value of 1 when it is 3 standard deviations away from the mean, hence it can at times miss better tuning. The following upper bound of 1.4 generates a mean of 0.95, meaning that the value of 1, is less than 1 standard deviation away from the mean making it more likely for the random search to re-initialise particles near that value.

The PSO algorithm manages to search for the best tuning value, with some tuning iterations indicating a large objective function evaluation. These are caused by the algorithm re-establishing the tuning after the parameter search has escaped the constraints. The PSO algorithm shows a different behaviour to its tuning when compared to the other systems. This is because the search space is much smaller and the system model is extremely sensitive to the controller gains. Moreover, the non-minimum phase dynamics of the system can cause instabilities when the constraints are surpassed and the optimisation algorithm readjusts the parameters within the constraints.

Figure 4.19 shows the variation with the feedback error across the simulation time of the proposed NLPID nonlinear gains. It can be seen that the proportional nonlinear gain starts from a large proportional gain and produces a fast response, which then quickly drops to the steady-state value. The integral gain starts from zero and rapidly increases to the steady-state value. This increase causes the controller to eliminate any remaining steady-state error while maintaining a low integral response at large errors to maintain a fast response. The derivative gain shows large drops. This is caused by eliminating the derivative kicks when the step functions are introduced. This improved the system performance and improved the system-input signal energy. Finally, the derivative gain shows a rapid increase to a steady state value, that reduces the system overshoot.

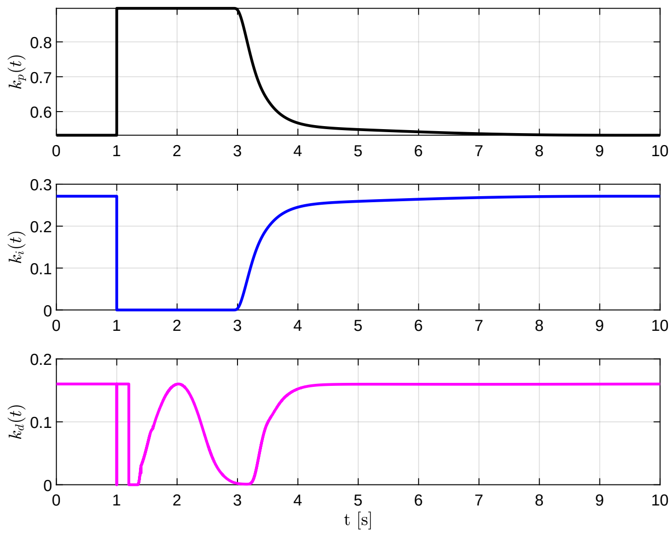



**Figure 4.19:** The tuned nonlinear gain values response to the NmP_SOPTD system simulation.

A second PID controller has been tuned using the steady-state values of the nonlinear gains. This is done to accomplish the difference between the nonlinear functions and the tuning at steady-state. This then isolates the improvements of the proposed NLPID controller to the nonlinear functions selected to solve this trade-off in the response. Table 4.9 shows the tuning values of all the conventional controllers, the proposed NLPID controller and the second tuning of a PID controller T2_PID.

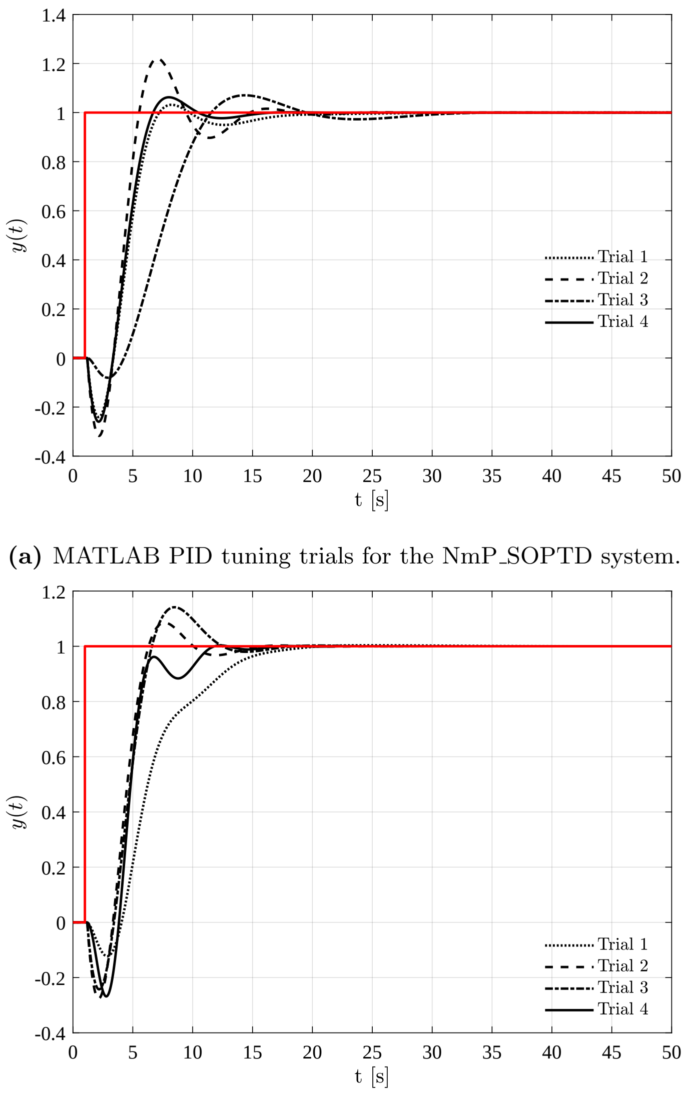
**Table 4.9:** The tuned control parameter values used for the NmP_OPTD benchmarking simulations.

| Controller | Tuning Parameters |
|---|---|
| NLPID | $k_0 = 0.9891, k_1 = 0.7376, k_2 = 0.4354, k_3 = 1, a = 0.9060$ |
| T2_PID | $k_{p_4} = 0.5323, k_{i_4} = 0.2713, k_{d_4} = 0.1602, N_4 = 100$ |

The stability regions for the tunable parameters of the proposed NLPID controller $k_0, k_1,$ and $k_2$ for the NmP_SOPTD. This shows that the tunable parameters $k_0$ and $k_1$ for which the feedback interconnection of the proposed NLPID controller with a NmP_SOPTD system remains stable. The simulations confirm the optimisation constraints and justify the search region defined for the PSO algorithm.

### Tuning and Design of the PID, 2D_PID, and SP_PID Controllers

Figure 4.20a shows the performance of the different tuning trials of the PID and 2D_PID controllers. It can be seen from the figure that the PID controller suffers from slow transient performance to avoid overshoots or a fast response with a large overshoot and undershoot. A similar response is observed for the 2D_PID, with slow transient dynamics. The PID controller tuning trial 4 indicates an overshoot of 6.24% with a rise time and settling time of 2.42 and 9.62 seconds, respectively. The 2D_PID controller tuning trial 4 indicates an overshoot of 0.32% with a rise time and settling time of 1.94 and 10.93 seconds, respectively.




**(a)** MATLAB PID tuning trials for the NmP_SOPTD system.

**(b)** MATLAB 2D_PID tuning trials for the NmP_SOPTD system.

**Figure 4.20:** Different tuning trials for both the PID and 2D_PID controllers using MATLAB control system toolbox PID tuning algorithm.

A SP_PID controller has also been designed for the NmP_SOPTD model. The transfer function that represents the SP_PID controller is:

$$K_{\text{SP\_PID}}(s) = \frac{K_{\text{PID}}(s)}{1 + K_{\text{PID}}(s)T(s)(1 - G_p(s))}. \tag{4.25}$$

The SP_PID controller design is formulated using MATLAB PID controller as described by Eq. (4.13) with the FOPTD plant model as described by Eq. (4.26).

$$\begin{aligned} P(s) &= T(s)e^{-\tau s},\ \tau = 0.2\ seconds, \\ T(s) &= \frac{z_1 s + z_0}{q_2 s^2 + q_1 s + q_0}, \end{aligned} \tag{4.26}$$

where $z_1 = -1.6, z_0 = 1, q_2 = 1, q_1 = 2, q_0 = 1.$

The time delay $e^{-\tau s}$ is approximated by the second order Pade transfer function $G_{\text{p}}(s)$ for the design of the SP_PID controller. The general second-order Pade transfer function of time delay is as follows:

$$G_{\text{p}}(s) = \frac{\tau^2 s^2 - 6\tau s + 12}{\tau^2 s^2 + 6\tau s + 12}. \tag{4.27}$$

Then, MATLAB algorithm is used to tune the SP_PID controller with its tuning trials shown in Figure 4.21. It can be seen that the SP_PID controller indicates a faster rise-time of 2.21 seconds with the presence of a 9.57% overshoot and a slower settling time of 10.15 seconds when compared to the PID and 2D_PID controllers.

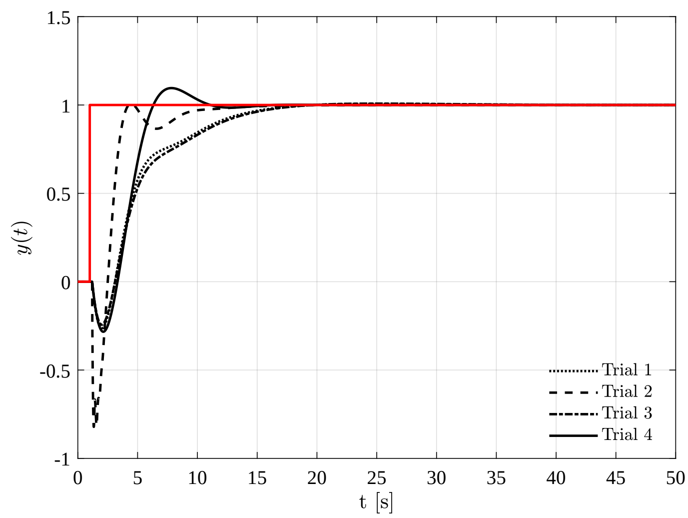



**Figure 4.21:** MATLAB SP_PID tuning trials for the NmP_SOPTD system.

The SP_PID controller outperforms the PID and 2D_PID controllers for the NmP_SOPTD system, showing improved performance with minimal overshoot and undershoot. It can also be seen that it provides the fastest rise time and settling time using the same tuning methodology as PID and 2D_PID.

Table 4.10 shows the objective function evaluations for every trial of each controller. Trial 4 of the PID and SP_PID controllers clearly have the lowest objective function evaluation and as a result they were selected. The 2D_PID controller has a large objective function evaluation for Trial 4, however, it was selected on the basis that it has the lowest overshoot and lowest settling time when compared to the other trials. However, at the cost of a larger undershoot.

**Table 4.10:** The objective function evaluations for the individual tuning trials of the conventional controllers for the NG_SOPTD benchmarking simulations.

| | $f(t, \epsilon(t), t_s)$ | | | |
|---|---|---|---|---|
| Controller | Trial 1 | Trial 2 | Trial 3 | Trial 4 |
| PID | $2.8480 \times 10^3$ | $3.2998 \times 10^3$ | $4.2754 \times 10^3$ | $2.7553 \times 10^3$ |
| 2D_PID | $5.0559 \times 10^3$ | $2.8396 \times 10^3$ | $2.8662 \times 10^3$ | $1.0362 \times 10^4$ |
| SP_PID | $7.1487 \times 10^3$ | $1.0483 \times 10^4$ | $7.1306 \times 10^3$ | $5.9842 \times 10^3$ |

Table 4.11 shows a summary of the tuning values of each controller together with their respective objective function evaluations. It can be seen that the proposed NLPID controller has the largest objective function evaluation, since it has the largest undershoot. The 2D_PID controller has the lowest objective function evaluation, which also has the lowest overshoot and undershoot, when compared to the other controllers. It is also worth noting that the T2_PID controller has a lower objective function evaluation when compared to the proposed controller, as it provides improved response with less settling time and less undershoot.

**Table 4.11:** The tuned control parameter values used for the NmP_SOPTD benchmarking simulations.

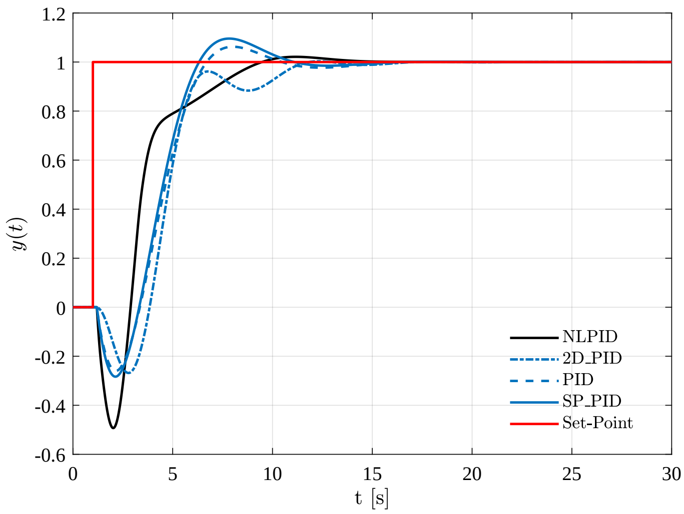
| Controller | Tuning Parameters | $f(t, \epsilon(t), t_s)$ |
|---|---|---|
| NLPID | $k_0 = 0.9891, k_1 = 0.7376, k_2 = 0.4354, k_3 = 1, a = 0.9060$ | $1.4143 \times 10^4$ |
| Controller | MATLAB Tuning Parameters | |
| PID | $k_{p_1} = 0.4128, k_{i_1} = 0.2477, k_{d_1} = 0, N = 100$ | $1.0406 \times 10^4$ |
| 2D_PID | $k_{p_2} = 0.8399, k_{i_2} = 0.3880, k_{d_2} = 0.2908, N_2 = 66.1754,$ | |
| | $b = 0.0154, c = 6.3031 \times 10^{-5}$ | $2.7238 \times 10^3$ |
| SP_PID | $k_{p_3} = 0.4100, k_{i_3} = 0.2803, k_{d_3} = 0, N_3 = 100$ | $5.8092 \times 10^3$ |
| T2_PID | $k_{p_4} = 0.5323, k_{i_4} = 0.2713, k_{d_4} = 0.1602, N_4 = 100$ | $5.9240 \times 10^3$ |

In the following section, the controller benchmarking for the NmP_SOPTD system is shown against the proposed NLPID controller. The benchmarking is conducted for two problems.

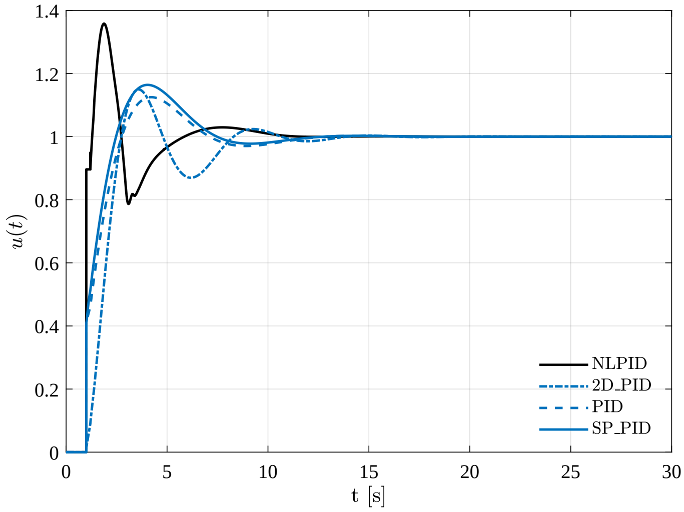
The servo problem and the regulator problem.

### Controller Benchmarking to Servo and Regulator Performance

Figure 4.22a shows the performance of the proposed NLPID controller against the conventional methods for the NmP_SOPTD system. It can be seen that the proposed NLPID controller provides a settling time of 8.61 seconds and rise time of 4.18 seconds with an overshoot of 2.13%. However, it is shown that the proposed NLPID controller provides the largest undershoot when compared to the conventional methods. In addition, it can be seen that the proposed NLPID controller has a sharp change in response speed once it is above 63% of the output between the 3 and 4 seconds time mark, this can also be seen in the Figure 4.19, where between 3 and 4 seconds time mark there is also a rapid change in the nonlinear derivative gain. Near the steady state, the derivative function becomes constant and the value of the derivative becomes zero. Consequently, it can be seen that at this point the derivative action is low and reduces the speed of the response. The SP_PID controller provides the largest overshoot of 9.57% and has a large settling time of 10.15 seconds. The 2D_PID controller has the shortest rise time of 1.94 seconds and simultaneously manages to reduce the overshoot to a minimal 0.32%. Figure 4.22b shows the system input to the NmP_SOPTD plant. It can be seen that all controllers maintain a bounded signal and controllers are internally stable.



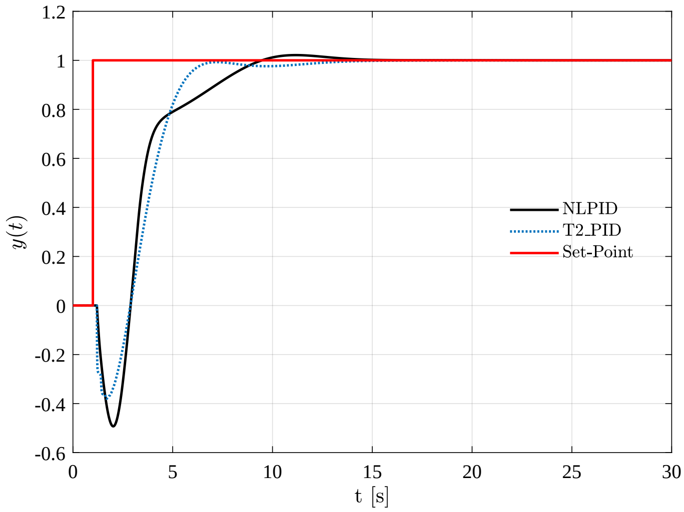

**(a)** Output response benchmark of the proposed NLPID controller against the conventional PID, 2D_PID, and SP_PID controllers.



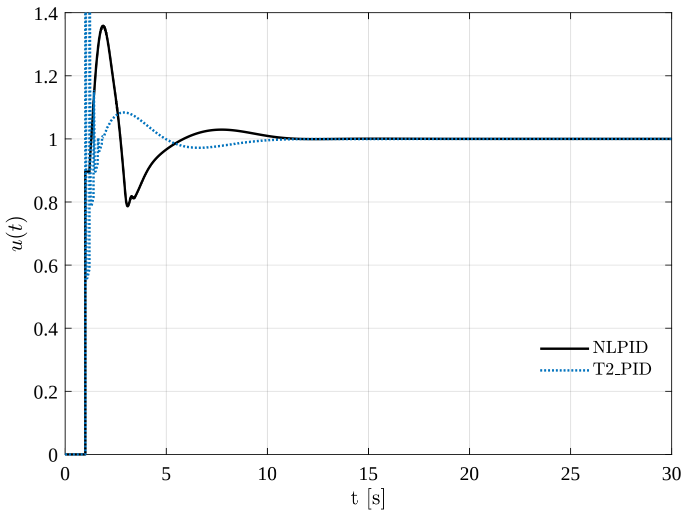

**(b)** System input signals of the benchmarked controllers.

**Figure 4.22:** Performance comparison of the proposed NLPID controller for step set-point function against the conventional PID, 2D_PID, and SP_PID control of the NmP_SOPTD system.

Figure 4.23 shows the response of the benchmarking comparison in the servo control problem between the proposed NLPID controller and the T2_PID controller. Figure 4.23a shows that the proposed NLPID controller has a low overshoot while having a larger undershoot. The proposed controller also shows a larger rise time and settling time. The T2_PID controller shows an overshoot of 0% and an undershoot of 38.9 % with a rise time and settling time of 2.37 and 6.24 seconds, respectively. Moreover, the figures show that the steady state values of the proposed controller can equally improve the system performance, which means that there are cases where there is little performance advantages in using the proposed nonlinear function gains. Figure 4.23b shows the system input response of the proposed NLPID and T2_PID controllers. It can be seen that they both maintain internal stability, producing bounded system input signals.



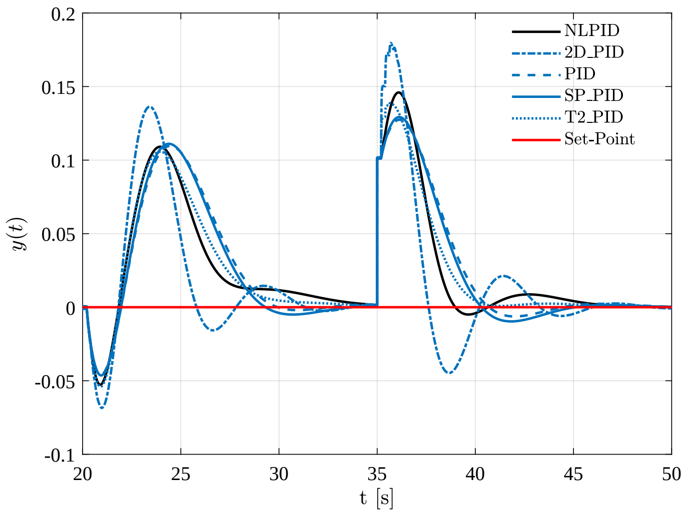

**(a)** Output response benchmark of the proposed NLPID controller against the nonlinear gains steady-state values substitute PID tuning.



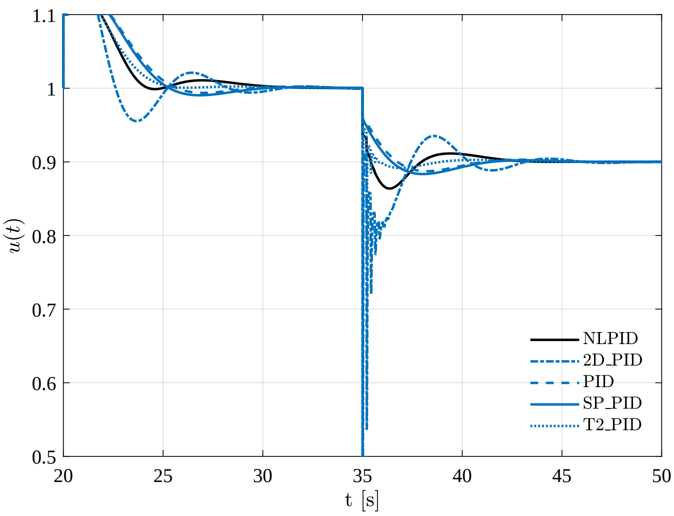

**(b)** System input of the benchmark.

**Figure 4.23:** Performance comparison of the proposed NLPID controller against the nonlinear gains steady-state values substitute PID tuning for the NmP_SOPTD system.

Figure 4.24a shows the response of all the controllers against input and output disturbances. It can be seen that the proposed NLPID controller has fast disturbance rejection that is similar to that of the SP_PID controller. The PID and 2D_PID controllers show a slow disturbance rejection, which shows that the proposed NLPID controller improves on performance and robustness when compared to the conventional methods. Figure 4.24b shows the system input to the plant after disturbances have been applied to the system. It can be seen that all the controllers remain internally stable with a bounded system input.




**(a)** Output response of both input and output disturbance rejection of the proposed NLPID controller benchmark against the conventional PID, 2D_PID, SP_PID, and T2_PID controllers.




**(b)** System input response of the benchmarked controllers to input and output disturbance rejection.

**Figure 4.24:** Disturbance rejection benchmarking of the proposed NLPID controller against the conventional PID, 2D_PID, SP_PID, and T2_PID controllers for the NmP_SOPTD system.

Table 4.12 shows the $L_2$ norm of the system input signals of all the benchmarked controllers together with their performance comparison. All the controllers produce a similar system input energy with the proposed NLPID controller showing the most expensive system input response. It can be seen that the improvements shown in the performance and robustness of the proposed NLPID controller are shown to produce a similar system-input energy as the conventional control methods. This indicates that the proposed nonlinear gains improve the performance and robustness with little to no expense in energy increase.

**Table 4.12:** Performance comparison evaluation summary table for the NmP_SOPTD system.

| NmP_SOPTD | NLPID | PID | 2D_PID | SP_PID | T2_PID |
|---|---|---|---|---|---|
| $\%Os$ | 2.13 | 6.24 | 0.32 | 9.57 | 0 |
| $t_r(s)$ | 4.18 | 2.42 | 1.94 | 2.21 | 2.37 |
| $t_s(s)$ | 8.61 | 9.61 | 10.93 | 10.15 | 6.24 |
| $L_2$ | 63.75 | 60.81 | 58.41 | 61.68 | 73.33 |
| $MSE$ | 0.37 | 0.44 | 0.41 | 0.40 | 0.36 |

In the case of NmP_SOPTD systems, it is shown that the proposed NLPID controller has a loss in performance and the competing requirements between performance and energy consumption are worsened. This is expected by systems that contain non-minimum phase dynamics and the proposed controller suffers from similar design limitations as any other controller in such systems. However, the performance of the proposed controller could be improved via improved tuning. From the results that are seen from this system, the proposed NLPID controller shows no major improvements in $L_2$ or MSE measures and no major performance advantages when compared to the conventional methods, apart from the improvement in overshoot when compared to the PID and SP_PID controllers.

## 4.6. Summary

The proposed NLPID controller has been demonstrated to have practical potential for use in nonlinear systems without requiring linearisation. The performance of the proposed controller shown in the NCSTR case scenario can potentially be improved, however, it is adequate to show that nonlinear gains can be applied as a simpler method to gain scheduling in the demonstrated example, with the proposed nonlinear gains showing great performance potential.

**Application to Nonlinear Systems: The NCSTR Case Study**

The NCSTR system has been utilised as a nonlinear case study to simulate the performance and disturbance rejection of the proposed NLPID controller in a nonlinear setting. The proposed NLPID controller has shown its ability to control the NCSTR system, which is highly nonlinear. The proposed controller has been tested for various operating regions possessing different dynamic and stability properties. This indicates that the proposed NLPID controller is robust in nonlinear systems and maintains the performance with less than 20% overshoot, fast rise-time and fast settling time. Finally, it is shown that the NLPID controller produces a system input energy that is finite and low, showing a practical control energy.

**Benchmarking**

The proposed NLPID controller has been demonstrated for its performance and disturbance rejection. The results were demonstrated by splitting the control into two problems, the servo and the regulator. In both cases, the proposed controller was benchmarked against the conventional control methods for practical feedback systems that include input and output disturbances. The benchmarking has been established by tuning the conventional controllers using the available tools to the practitioner, which in this case has been the MATLAB PID Tuner Toolbox. The proposed NLPID controller was tuned using a Particle Swarm Optimization algorithm. Extensive tuning has been shown for all the controllers to ensure their fair comparison with the main design focus on reducing overshoot, rise time, and settling time.

**FOPTD Case Study**

The proposed NLPID controller has been shown to improve simultaneous transient performance and disturbance rejection with minimal trade-offs under a single-control design. The proposed controller has shown performance improvements against the conventional PID, 2D_PID, and SP_PID methods. In addition, the proposed controller managed to improve the performance and robustness while maintaining similar system input energy to that of the conventional control methods. This is an advantage for the proposed controller since nonlinear controllers can generate large and energy-expensive input signals. Finally, the proposed NLPID controller has been shown to remain internally and externally stable for the FOPTD system.

**NG_SOPTD Case Study**

The proposed NLPID controller has shown that it can provide improved transient performance for the NG_SOPTD case study. However, it has shown that it is limited in disturbance rejection, which means that it requires separate tuning to solve the regulator problem, requiring a trade-off between performance and robustness. The proposed controller has been shown to provide cheaper system input energy, when compared to the conventional control methodologies and is capable of maintaining both internal and external stability of the feedback interconnection.

**NmP_SOPTD Case Study**

The proposed NLPID controller has shown no improvement in either performance or disturbance rejection for the NmP_SOPTD system. The proposed controller provides a similar response when compared to the conventional controllers. However, it is shown that the undershoot is maximal for the proposed controller, while the conventional controllers produce a lower undershoot. The proposed controller has shown internal stability with bounded system input signals that have similar energy and MSE evaluation to the conventional controllers.

## 4.7. Concluding Remarks

The benefits of utilising nonlinear control to design the proposed novel NLPID controller has been shown through simulations for the highly nonlinear NCSTR system example. The results have shown that the proposed NLPID controller is suitable for application in nonlinear systems providing excellent performance for multiple operating regions. In addition, the proposed NLPID controller has achieved the control performance under efficient actuation with minimal reach of the saturation limits. In addition, the proposed NLPID controller has been benchmarked against conventional control methods. This ascertains the suitability of the proposed NLPID controller for industrial and linear systems applications and its performance comparison to the conventional control methods. The indicative analysis of the linear systems has been shown. This justifies the design and tuning of the proposed controller and provides further contribution when compared to the NLPID controllers in the studied literature that lack such contribution. Finally, the proposed NLPID controller has shown its capacity to maintain a similar or lower system input energy to the linear control methods. This indicates the ability of the proposed NLPID controller to utilise similar actuation requirements as linear control methodologies.

# Chapter 5

# NLPID Controller Robustness Against Uncertainty

## 5.1. Introduction

In this chapter, the robustness of the proposed controller is shown under ±5% parametric uncertainty in the NCSTR system model parameters. In addition, the robustness of the proposed NLPID controller is shown for various types of uncertainties in the FOPTD, NG_SOPTD, and NmP_SOPTD systems. Firstly, the structured parametric uncertainty is conducted for the plant models, all of which are simulated for a ±10% parametric uncertainty. The parametric uncertainty of the three linear models is also supported with stability analysis to further the results on the robustness of the controller. The controller is also simulated for unstructured types of uncertainty, such as additive and multiplicative uncertainties, which offer a larger encapsulation of modelling errors.

## 5.2. Problem Statement

Practical control systems possess uncertainty due to inaccuracies in the model that can come from modelling errors and imprecision in the parameter estimations arising from sensor measurements. These sources of error can influence the entire system's behaviour, and if they are large enough, they can also render the control system unstable, if it is not made robust. The linear system models are assumed to possess inaccessible dynamics and hidden modes due to the linearisation process. The modelling inaccuracies are mathematically represented in three different formats, which are parametric, additive, and multiplicative uncertainties. The performance and stability robustness of the proposed NLPID controller are investigated for the three uncertainties in the linear systems. Figure 5.1 shows the schematic of a classical feedback interconnection of the proposed NLPID controller for different plant models. The tuning of the proposed NLPID controller is kept the same as the tuning used in the benchmarking studies to investigate the performance variations when compared to the nominal dynamics.

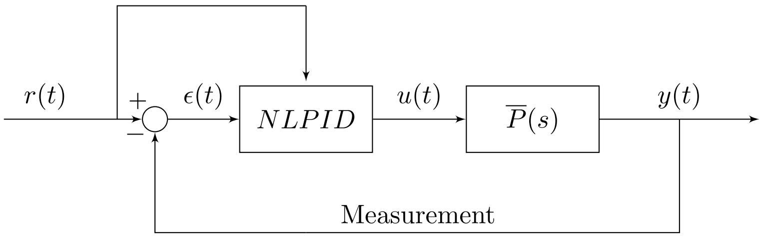



**Figure 5.1:** The schematic block diagram of the feedback interconnection of the proposed NLPID controller and plant model containing uncertainty.

The robustness investigation of the proposed NLPID controller is conducted under pure feedback, assuming perfect sensor measurements and no disturbances are introduced into the system. This focuses mainly on the ability of the proposed NLPID controller to handle uncertainties in modelling. The parametric uncertainty is applied in all the possible parameters of the system models, investigating the system sensitivity against parameter variations. Finally, the additive and multiplicative uncertainties are applied distinctly based on the system models.

## 5.3. Parametric Uncertainty

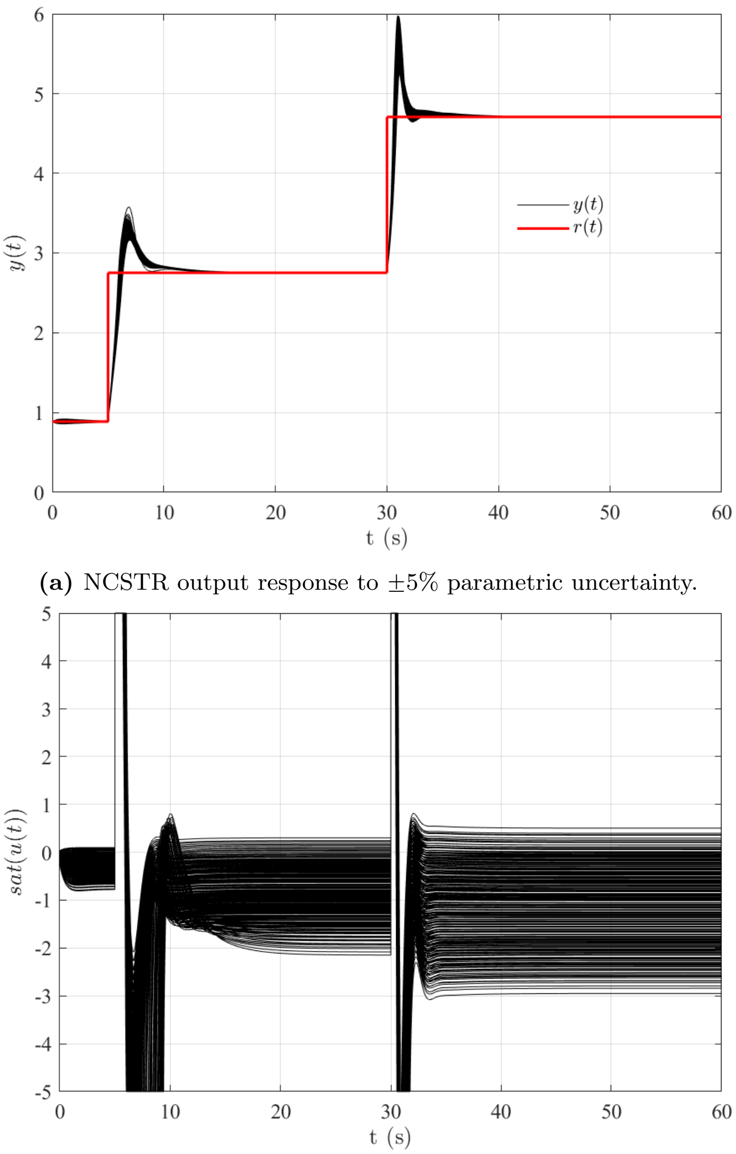
### The NCSTR Model

The uncertainty of the NCSTR mathematical model is represented by a parameter variation of $\pm 5\%$ in all its measurable parameters. This variation simulates the sensitivity of the system to potential measurement errors and shows the capacity of the proposed NLPID controller to mitigate the negative effects on performance. The parametric uncertainty of the dimensionless NCSTR model is mathematically represented as follows Harmon Ray, 1981; Colantonio et al., 1995; So and Jin, 2018; Sinha and Mishra, 2018:

$$
\begin{aligned}
\dot{x}_1(t) &= -x_1(t) + \overline{D_a}(1 - x_1(t))e^{\left[\dfrac{x_2(t)}{1 + x_2(t)/\overline{\gamma}}\right]}, \\
\dot{x}_2(t) &= -(1 + \overline{\beta})x_2(t) + \overline{H}\,\overline{D}_a(1 - x_1(t))e^{\left[\dfrac{x_2(t)}{1 + x_2(t)/\overline{\gamma}}\right]} + \overline{\beta} sat(u(t)), \\
y(t) &= x_2(t),
\end{aligned}
\tag{5.1}
$$

where $7.6 \leqslant \overline{H} \leqslant 8.4$ , $0.0684 \leqslant \overline{D}_a \leqslant 0.0756$ , $19 \leqslant \overline{\gamma} \leqslant 21$, $0.285 \leqslant \overline{\beta} \leqslant 0.315$ which represents $\pm 5\%$ uncertainty in all parameters.

Fig. 5.2 shows the output and system input to the NCSTR system to transition from $y_A$ to $y_B$ and then from $y_B$ to $y_C$ under parametric uncertainty. The output response, shown in Fig. 5.2a, indicates that the proposed NLPID controller keeps the NCSTR system stable with fast performance. In addition, the output of the system, $y_B$ is unstable, and the proposed NLPID controller can regulate the system under parametric uncertainty for both output transitions with minimal impact on performance. It is also shown that the parametric uncertainty does not disregulate the system and the proposed NLPID controller maintains stability. Although the system performance has deteriorated due to the parametric uncertainty, it shows small deviations in performance, making stability the main priority. Fig. 5.2b shows that the system input is bounded within the saturation limits for all parametric uncertainties. The proposed NLPID controller generates large system inputs for fast performance, which means that the system input reaches saturation for a period of time. Although the system input saturates, the proposed NLPID controller manages to stabilise the system and provide a fast response with no windup.




**(a)** NCSTR output response to $\pm 5\%$ parametric uncertainty.

**(b)** NCSTR system input due to $\pm 5\%$ parametric uncertainty.

**Figure 5.2:** Output and system input responses to $\pm 5\%$ parametric uncertainty in the NCSTR parameters for the output transitions $y_A \rightarrow y_B \rightarrow y_C$.

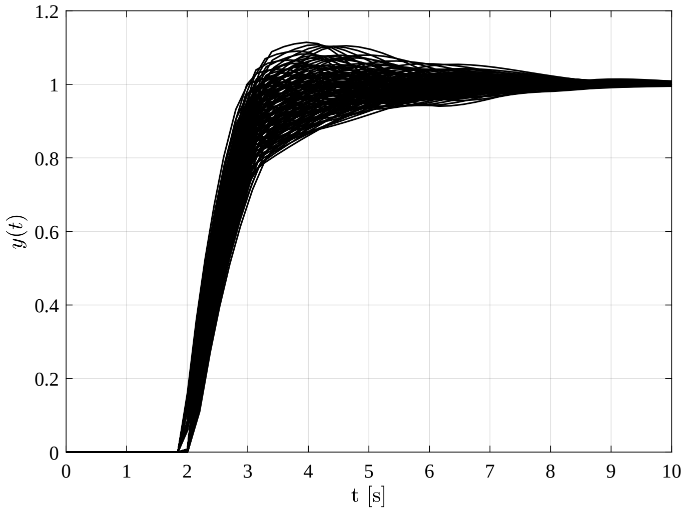
### The FOPTD System

The parametric uncertainty of the nominal FOPTD plant is modelled using Eq. (5.2) in the following transfer function format:

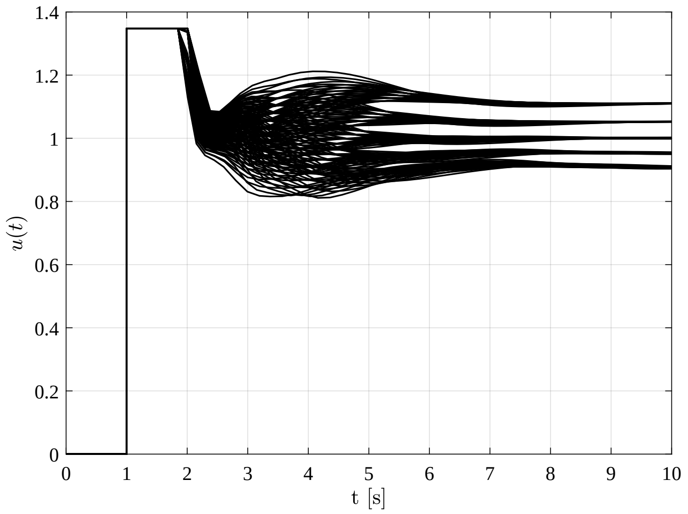
$$\overline{P}(s) = \frac{\overline{z}e^{\overline{\tau}s}}{\overline{b}_1 s + 1}, \tag{5.2}$$

where $0.9 \leqslant \overline{z} \leqslant 1.1$, $0.9 \leqslant \overline{b}_1 \leqslant 1.1$, and $0.9 \leqslant \overline{\tau} \leqslant 1.1$, which models a parameter change of $\pm 10\%$.

Figure 5.3, shows the gain, lag, and delay parametric uncertainty output and system input plots. Figure 5.3a shows how the proposed NLPID controller responds to a large set of $\pm10\%$ variations in gain $z$, lag $b_1$, and delay parameter $\tau$. It can be seen that there are no large variations of overshoot and no instabilities. In the case where the gain, lag, and delay parameters are underestimated, the response shows a maximum overshoot of approximately 10% and a larger settling time. The figure also shows no effect on stability, providing evidence of robust performance and robust stability for the proposed NLPID controller against gain, lag, and delay variations.




**(a)** Output response to parametric uncertainty.




**(b)** System input to parametric uncertainty.

**Figure 5.3:** NLPID controller response to $\pm10\%$ gain, lag, and delay parametric uncertainty in FOPDT system, showing both output and system input responses.

According to the parametric uncertainty study, it can be seen that the proposed NLPID controller shows resilience to parameter variations in a structured model uncertainty. The uncertainty tests indicate that internal stability is maintained across different types of parameter variations with some changes in performance, showing slower settling time, extending from 4 seconds of the nominal plant, up to a maximum of 10 seconds for the extreme variations.

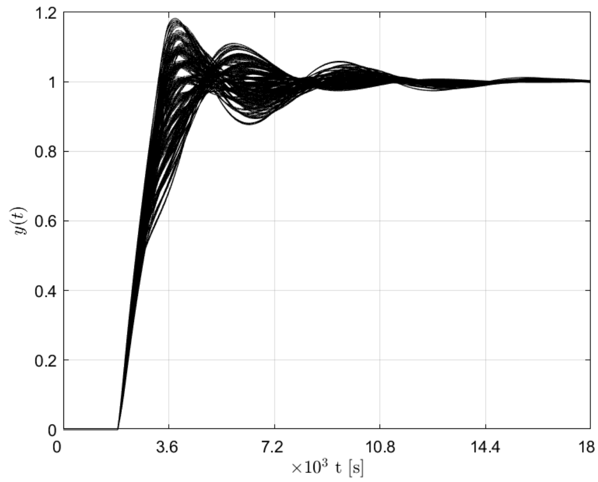
#### The NG_SOPTD System

The parametric uncertainty of the NG-SOPTD model is introduced to show the robustness of the proposed NLPID controller against changes in the value of the parameters. All the parameters of the system are changed by a value of ±10%. The parametric uncertainty model is represented as (Krishna et al., 2012):

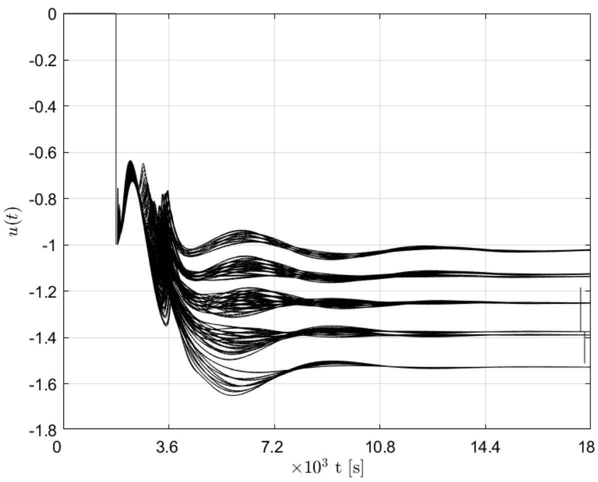
$$\overline{P}(s) = \frac{\overline{z_1}s + \overline{z_0}}{\overline{q_2}s^2 + \overline{q_1}s + \overline{q_0}}e^{-\overline{\tau}s}, \tag{5.3}$$

where all the parameters $-0.79 \leqslant \overline{z_1} \leqslant -0.96, -7.89 \leqslant \overline{z_0} \leqslant -9.65, 0.9 \leqslant \overline{q_2} \leqslant 1.1, 2.41 \leqslant \overline{q_1} \leqslant 2.94, 9.87 \leqslant \overline{q_0} \leqslant 12.07$, and $97.2 \leqslant \overline{\tau} \leqslant 118.8$.

Figure 5.4a shows the output response of the parametric uncertainty of the NG_SOPTD model. According to the figure, it can be seen that the system remains stable and produces a maximum overshoot of approximately 18% and has a deteriorated performance showing longer rise time and settling time. Figure 5.4b shows the system input of the NLPID controller into the NG_SOPTD plant and it can be seen that the signals are bounded and internally stable. The sharp signals observed in the figure are produced by the derivative action due to the oscillatory responses.




**(a)** Output response to parametric uncertainty.




**(b)** System input to parametric uncertainty.

**Figure 5.4:** NLPID controller response to $\pm 10\%$ parametric uncertainty in all the NG_SOPTD system parameters, showing both output and system input responses.

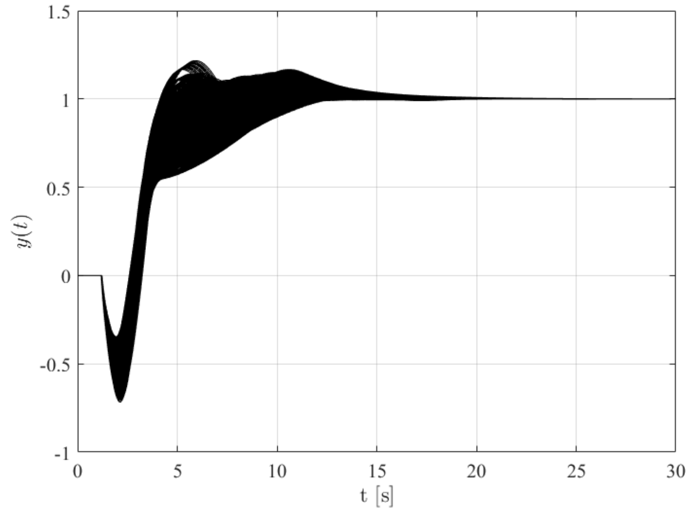
### The NmP_SOPTD System

The parametric uncertainty of the NmP_SOPTD model is introduced to show the robustness of the proposed NLPID controller against changes in the value of the parameters. All the parameters of the system are changed by a value of $\pm 10\%$. The parametric uncertainty model is represented as (Krishna et al., 2012):

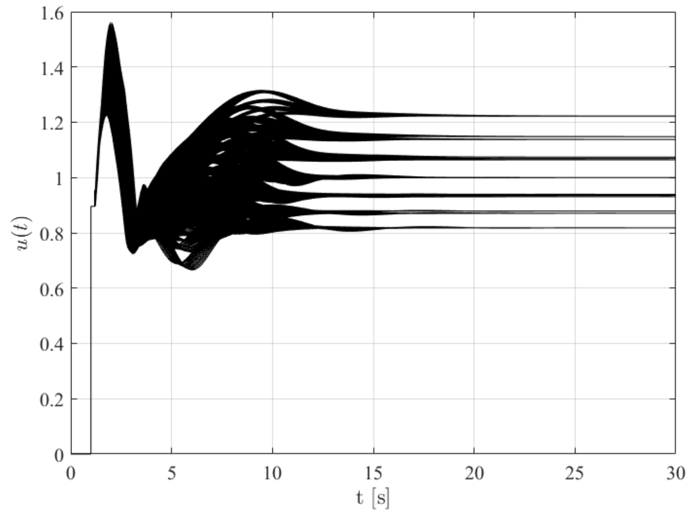
$$\overline{P}(s) = \frac{\overline{z_1}s + \overline{z_0}}{\overline{q_2}s^2 + \overline{q_1}s + \overline{q_0}} e^{-\overline{\tau}s}, \tag{5.4}$$

where all the parameters $-1.44 \leqslant \overline{z_1} \leqslant -1.76, 0.9 \leqslant \overline{z_0} \leqslant 1.1, 0.9 \leqslant \overline{q_2} \leqslant 1.1, 1.8 \leqslant \overline{q_1} \leqslant 2.2, 0.9 \leqslant \overline{q_0} \leqslant 1.1$, and $0.18 \leqslant \overline{\tau} \leqslant 0.22$.

Figure 5.5 shows the response of the feedback interconnection and the proposed NLPID controller system input to the NmP_SOPTD system against ±10% uncertainty in all its parameters. Figure 5.5a shows the output response of the feedback system, indicating that the proposed NLPID controller is robust, maintaining stability and proximal performance with small changes in its rise time, settling time, and overshoot. Figure 5.5b shows the proposed NLPID controller system input under parametric uncertainty, indicating that the system is internally stable, with bounded system input signals.




**(a)** Output response to parametric uncertainty.




**(b)** System input to parametric uncertainty.

**Figure 5.5:** NLPID controller response to ±10% parametric uncertainty in all the NmP_SOPTD system parameters, showing both output and system input responses.

## 5.4. Additive Uncertainty

The proposed NLPID controller has shown its robustness against parametric uncertainty which is a structured type of uncertainty. To extend these results the proposed NLPID controller is simulated against unstructured uncertainties that model additional dynamics. This will further test the robustness of the proposed NLPID controller against additional higher-order dynamics that are common in industrial systems. The tests begin with additive uncertainty, which is modeled using Eq. (5.5). The uncertainty plant $\Delta(s)$ is designed to be additional lag dynamics into the plant model that may not be considered in the modelling process for modelling simplification.

$$\overline{P}(s) = P(s) + \Delta(s) \tag{5.5}$$

The FOPTD system additive uncertainty is modelled under 50% unmodelled lag dynamics. Assuming that $\Delta(s)$ is any arbitrary transfer function, satisfying the condition $||\Delta(s)||_\infty \leqslant 1$ then an arbitrarily large variation of uncertainty is chosen to be (Skogestad and Postlethwaite, 2001):

$$\Delta(s) = \frac{1}{t_a s + 1} = \frac{1}{1.5s + 1}. \tag{5.6}$$

The simulation of the additive uncertainty is conducted using the system as a plant with a minimal realization that can be expressed in the additive form. The worst-case scenario is taken as the primary example for the simulation, depicting additional 50% lag dynamics.

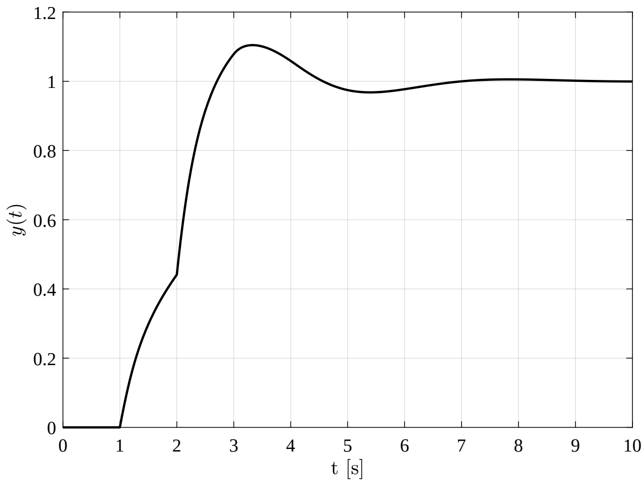
The additive uncertainty of the linear NG_SOPTD and NmP_SOPTD is modelled as an arbitrarily large second order dynamics of uncertainty as follows (Skogestad and Postlethwaite, 2001):

$$\Delta(s) = \frac{0.1}{s^2 + 0.1s + 1}. \tag{5.7}$$

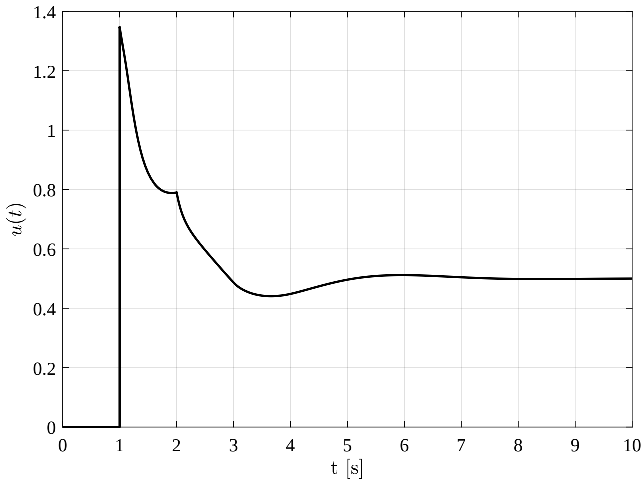
The simulation of the additive uncertainty is conducted using the system as a plant with a minimal realization that can be expressed in the additive form. The worst-case scenario is taken as the primary example for the simulation, depicting arbitrarily large second-order unmodelled dynamics.

#### The FOPTD System

Figure 5.6a shows that the proposed NLPID controller is robustly stable to the additional lag dynamics, with a fast settling time of approximately 6 seconds. Performance deterioration is observed as the plant damping is reduced. The performance of the proposed NLPID indicates that even after an additive uncertainty in the system, the controller maintains stability with a slightly reduced performance, observing an overshoot of approximately 10%. Figure 5.6b shows the system input due to the additive uncertainty. According to the figure, it can be seen that the signal is bounded and hence internally stable.




**(a)** NLPID output response due to unstructured additive uncertainty.

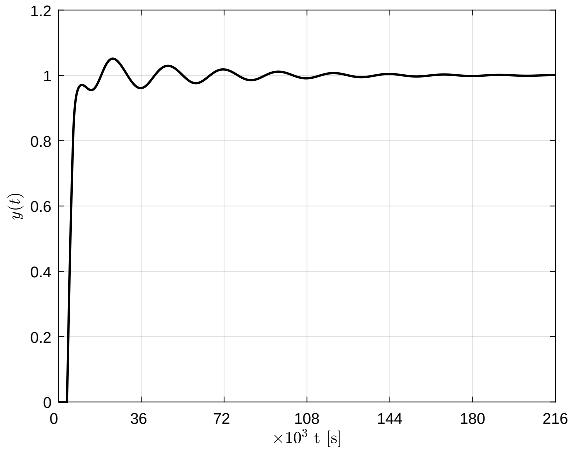



**(b)** NLPID system input due to unstructured additive uncertainty

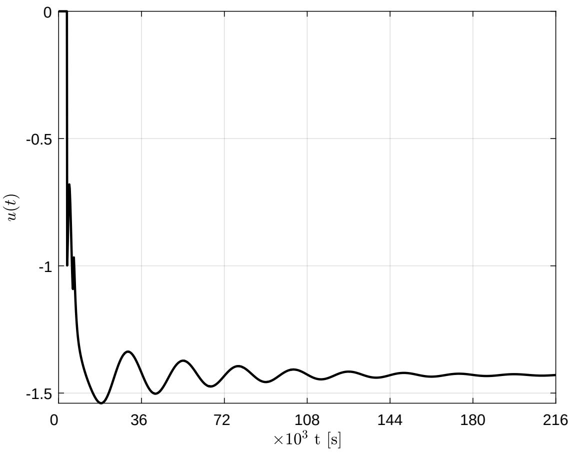
**Figure 5.6:** NLPID controller response to unstructured additive uncertainty in FOPTD system, showing both output and system input responses.

### The NG_SOPTD System

Figure 5.7a shows the output response of the NG_SOPTD model. The figure clearly indicates that the uncertain dynamics have largely increased the settling time of the proposed NLPID controller with the addition of the second order dynamics oscillatory behaviour. Although the system remains stable there is a large deterioration in output performance and the system takes a long time to track the set-point input, which is undesirable. Figure 5.7b shows the system input to the uncertain NG_SOPTD model.




**(a)** NLPID output response due to additive uncertainty.

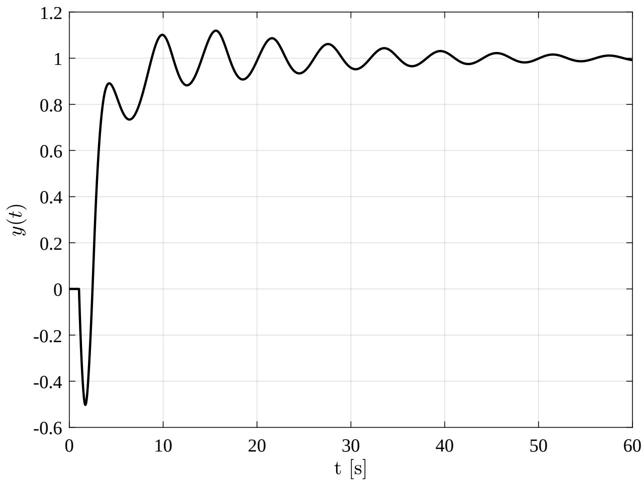



**(b)** NLPID system input due to the additive uncertainty

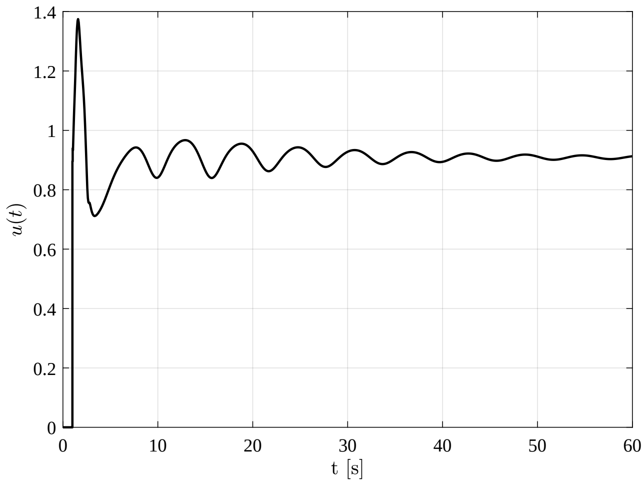
**Figure 5.7:** NLPID controller response to additive unstructured uncertainty in NG_SOPTD system.

### The NmP_SOPTD System

Figure 5.8 shows the output and system input response produced by the proposed NLPID controller for the NmP_SOPTD model under additive uncertainty. Figure 5.8a shows that the proposed NLPID controller is stable under the influence of additive second-order dynamics. It can be seen that the response performance has deteriorated, where the settling time is increased to approximately 55 seconds. The NmP_SOPTD system is also showing robust stability, however, the system shows a large settling time and an undershoot of more than 40%, which shows a large deterioration in performance that is undesirable. Figure 5.8b shows the system input to the plant model. It can be seen that the proposed controller is internally stable with a bounded system input signal. This indicates that the proposed controller is robustly stable for large uncertainty when introducing higher frequency higher order dynamics.




**(a)** NLPID output response due to additive uncertainty.




**(b)** NLPID system input due to the additive uncertainty

**Figure 5.8:** NLPID controller response to additive unstructured uncertainty in NmP_SOPTD system.

## 5.5. Multiplicative Uncertainty

A more common uncertainty model used is multiplicative uncertainty. This uncertainty model provides more information and analysis of the uncertainty dynamics and is commonly used to model delay and gain uncertainty (Skogestad and Postlethwaite, 2001). The multiplicative uncertainty is modeled using Eq. (5.8) as follows:

$$\overline{P}(s) = P(s)[1 + W(s)\Delta(s)]. \tag{5.8}$$

where $\Delta(s)$ being any arbitrary transfer function, satisfying the condition $||\Delta(s)||_{\infty} \leqslant 1$.

The following inequality must hold true for any multiplicative uncertainty, indicating a circle of radius equal to the magnitude of $W(s)$ that the system uncertainty must lie away from the $-1+0j$ point of the Nyquist plot. The multiplicative uncertainty has been considered to be an unstructured uncertainty of delay dynamics equivalent to 20% time delay from the nominal plant. From the above condition, we can determine the weighting dynamics of the uncertainty as follows:

$$\begin{aligned}\left|\frac{\overline{P}(s)}{P(s)}-1\right| &= \left|\frac{\left(\dfrac{z_1 s+z_0}{q_2 s^2+q_1 s+q_0}\right)e^{-(1+\lambda)s}}{\left(\dfrac{z_1 s+z_0}{q_2 s^2+q_1 s+q_0}\right)e^{-s}}-1\right|,\\ &= \left|\frac{e^{-(1+\lambda)s}}{e^{-s}}-1\right|,\\ &= |e^{-\lambda s}-1| \leqslant |W(s)|,\end{aligned}$$

for which when the maximum delay uncertainty of 20% occurs at the value of $[\lambda_{min}, \lambda_{max}] = [0, 0.2]$, that makes the equation into:

$$|e^{-0.2s}-1| \leqslant |W(s)|. \tag{5.9}$$

The weighting function $W(s)$ is the transfer function that has been modelled to contain the worst-case magnitude of the delay uncertainty magnitude. The weighting function that is recommended to fit the uncertain lag dynamics to be modelled as (Skogestad and Postlethwaite, 2001):

$$W(s) = \frac{\lambda_{max} s}{\dfrac{\lambda_{max}}{2}s+1} = \frac{0.2s}{0.1s+1}. \tag{5.10}$$

The magnitude plot can be shown by the red and blue plots, respectively, in Figure 5.9, which shows $W(s)$ estimating the distribution of the worst-case delay uncertainty transfer function magnitude. This forms a more generic unstructured delay uncertainty that can be implemented in more complex controllers to show extensive robustness to a larger set of uncertain dynamics. When compared to parametric uncertainty, which only includes a certain range of values, the degree and structure of the plant dynamics are assumed to be unknown. In this case, unstructured uncertainty allows some flexibility for ignorance in the degree and structure of the dynamics.

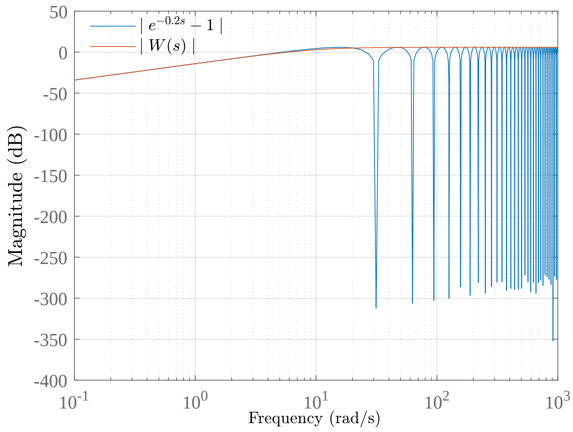



**Figure 5.9:** The bode magnitude response of the uncertainty weighting transfer function and the maximum delay deviation transfer function.

The uncertainty $\Delta(s)$ is chosen to be a transfer function that contains the same poles as the nominal plant, the appropriate transfer function selection is modelled as (Skogestad and Postlethwaite, 2001):

$$\Delta(s) = \frac{1}{t_m s + 1}. \tag{5.11}$$

As a result, the total uncertainty dynamics in the multiplicative form can be represented as:

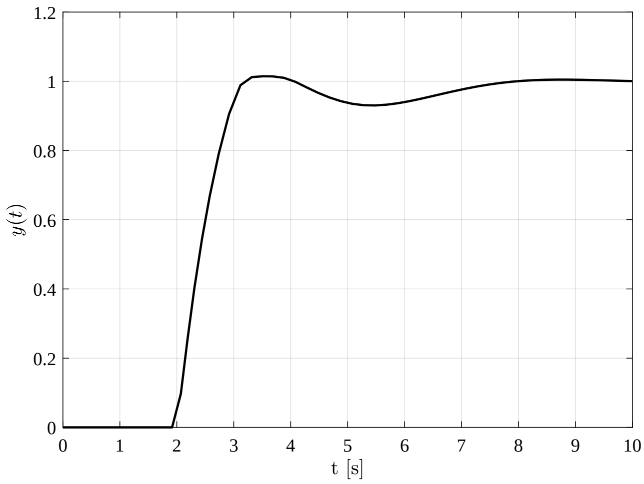
$$W(s) = \frac{0.2s}{0.1s + 1}, \Delta(s) = \frac{1}{s + 1}. \tag{5.12}$$

The uncertainty function $\Delta(s)$ represents the uncertainty in the magnitude and phase dynamics and is implemented according to the weighting function $W(s)$, where $||\Delta(s)||_\infty \leqslant 1$, satisfying the $H_\infty$ condition.

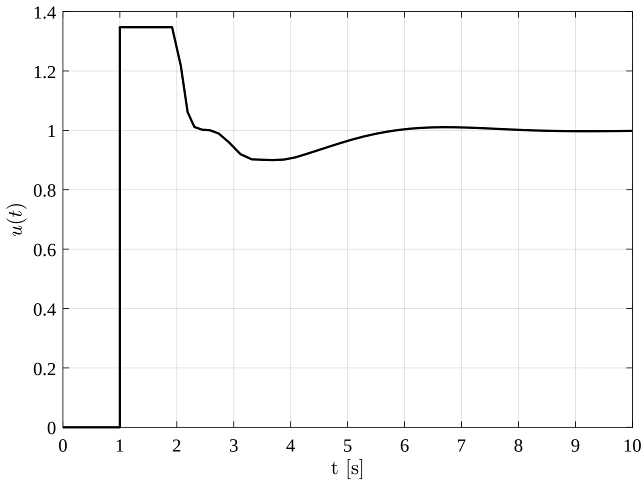
The simulation of the multiplicative uncertainty is conducted under the developed uncertainty. The system is represented as a plant with a minimal realisation that can be expressed in the multiplicative form. The worst-case scenario is taken as the primary example for the simulation, depicting the 20% unmodeled delay dynamics.

### The FOPTD System

Figure 5.10a shows the output response of the proposed NLPID controller, indicating robust stability within a large set of unstructured dynamics of the plant model. It also indicates that the controller suffers from deteriorated performance with an overshoot of less than 10% and a settling time of approximately 8 seconds. Figure 5.10b shows the system input from the proposed NLPID controller into the uncertain plant, indicating internal stability to the uncertainty.




**(a)** NLPID output response due to multiplicative uncertainty.

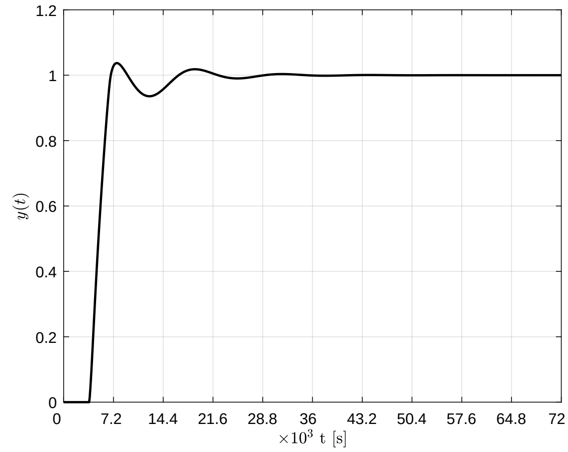



**(b)** NLPID system input due to the multiplicative uncertainty

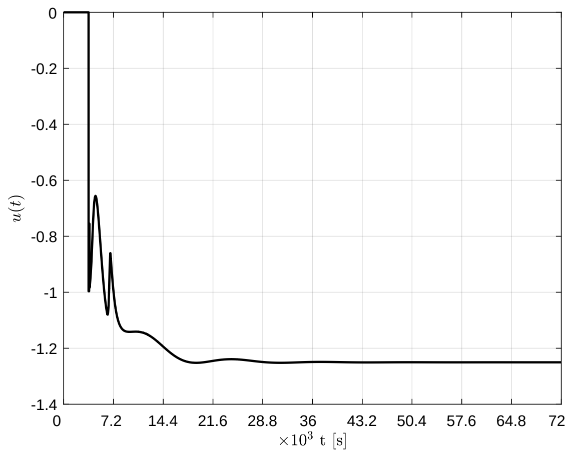
**Figure 5.10:** NLPID controller response to multiplicative unstructured uncertainty in FOPTD system.

### The NG_SOPTD System

Figure 5.11a shows the output response of the multiplicative uncertainty. It is clearly seen that the system has a larger settling time with a larger oscillatory response. Figure 5.11b shows the system input signals and clearly shows that the system remains internally stable.




**(a)** NLPID output response due to multiplicative uncertainty.

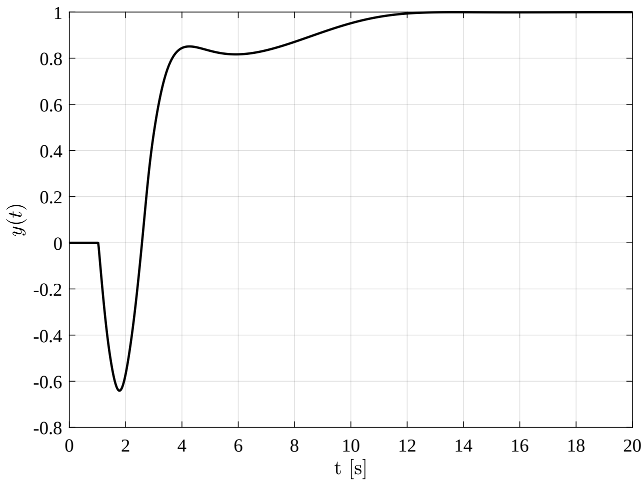



**(b)** NLPID system input due to the multiplicative uncertainty

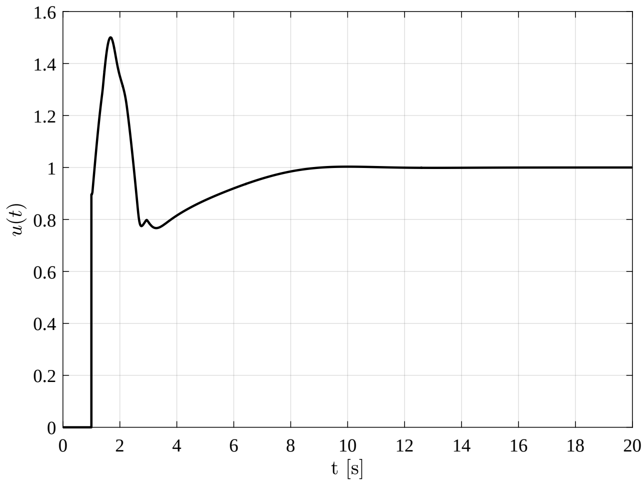
**Figure 5.11:** NLPID controller response to multiplicative unstructured uncertainty in NG_SOPTD system.

### The NmP_SOPTD System

Figure 5.12 shows the response of the proposed NLPID controller to multiplicative uncertainty in the NmP_SOPTD system model. Figure 5.12a shows the output response of the proposed NLPID controller against multiplicative uncertainty. It can be seen that the proposed NLPID controller produces a stable response and it can be seen that it maintains a robust performance against multiplicative uncertainty. Figure 5.12b shows the system input of the proposed NLPID controller. It can be seen that the proposed controller is internally stable and maintains bounded system input signals to large multiplicative uncertainties.




**(a)** NLPID output response due to multiplicative uncertainty.




**(b)** NLPID system input due to the multiplicative uncertainty.

**Figure 5.12:** NLPID controller response to multiplicative unstructured uncertainty in NmP_SOPTD system.

## 5.6. Summary

The proposed NLPID controller has been tested under extensive simulation for various types of uncertainty in nonlinear systems. The proposed NLPID controller shows excellent performance and stability robustness for the various uncertainties.

### The NCSTR System

The proposed NLPID controller has also been investigated for its ability to robustly control nonlinear systems under the NCSTR system example case. A $\pm 5\%$ parametric variation has been simulated. The proposed NLPID controller has shown the capacity to maintain fast performance with small variations in its performance measures. The proposed NLPID controller also manages to maintain stability under the parameter uncertainty showing its ability to mitigate the effects of inaccurate models in sensitive nonlinear systems.

#### FOPTD Case Study

The proposed NLPID controller is robust under large uncertainty in various forms. The proposed NLPID controller remains stable and maintains fast performance for $\pm 10\%$ parameter variations. The proposed NLPID controller has also been tested for performance and stability robustness under additive uncertainty for the FOPTD system. It has been shown that the proposed controller is robust and its performance is slightly deteriorated under higher-order dynamics in the additive uncertainty form. The proposed NLPID controller was also tested under multiplicative uncertainty showing a 20% uncertain delay dynamics of the FOPTD model. It has been shown that there is a slight performance deterioration with increased rise-time and settling time. However, the proposed controller remains stable under delay uncertainty in the multiplicative form.

#### Negative Gain SOPTD Case Study

The proposed NLPID controller has been shown to have robust performance and stability for the NG-SOPTD system under various forms of uncertainty. A parametric uncertainty of $\pm 10\%$ variations in the parameters has been conducted and the proposed NLPID controller has been shown to have robust performance and robustness with a shown increase in rise-time and settling-time. The proposed NLPID controller has also been tested for the NG_SOPTD system under additive uncertainty form with second-order dynamics in the additive form. The proposed NLPID controller has been shown to have robust stability, where the performance deteriorates and has shown oscillatory behaviour with a larger settling time. The controller was also simulated under multiplicative uncertainty depicting a 20% uncertainty in delay dynamics. The proposed NLPID controller has been shown to have both robust stability and robust performance under multiplicative uncertainty.

#### Non-Minimum Phase Case Study

The proposed NLPID controller has also been tested for its robustness for the NmP_SOPTD dynamical system. It has been shown that the proposed NLPID controller is robust in both the performance and stability under $\pm 10\%$ parametric variations of the system. The system has also been tested with additive uncertainty adding second-order dynamics to the plant system. The performance of the proposed NLPID controller shows significant deterioration in its performance, where oscillatory behaviour is observed with a significant increase in settling time. However, the proposed NLPID controller remains stable, showing robust stability. Finally, the proposed NLPID controller was also tested for the system under 20% multiplicative uncertainty in delay dynamics. It has been shown that the proposed NLPID controller has robust performance and robustness under multiplicative uncertainty.

## 5.7. Concluding Remarks

The robustness of the proposed NLPID controller has been shown for linear systems under a large variety of modelling uncertainties, which in most of the studied literature on NLPID control methods, has not been previously provided. This further contributes to the results of the suitability of the proposed NLPID controller to be applied in both industrial and linear systems. Finally, the robustness tests expand the indicative stability analysis, indicating further the stability margins of the proposed controller and providing further support on the design and tuning justification basis of the proposed controller's robustness against parametric uncertainty.

# Chapter 6

# Discussion, Conclusions, and Future Work

## 6.1. Discussion on the Results

The control of the NCSTR system with input saturation has been simulated to evaluate the performance of the proposed NLPID controller in nonlinear systems. The NCSTR system possesses three steady states that have different stability properties; two are stable, and one is unstable, which makes the control problem of moving from one to the next complicated for linear controllers and requires separate tuning for each steady state. The proposed NLPID controller has been tuned once and the controller can robustly control the system to all three states. The controller shows a fast transient response with an 18.9% overshoot when controlling the system from the stable equilibrium point $y_A$ to the unstable equilibrium point $y_B$. The proposed controller can also provide a fast transient response for the control of the NCSTR system from the unstable equilibrium point $y_B$ to the stable equilibrium point $y_C$. The controller provides a low rise time and settling time, with an overshoot of 15.4%. In both cases, the controller can reject input disturbances to the system. Extensive robustness tests have also been conducted for the NCSTR case with $\pm 5\%$ variations in all its parameters. The proposed controller shows robust performance and stability against the parametric uncertainty, with minimal deterioration in transient performance for the two control problems of $y_A \rightarrow y_B$ and $y_B \rightarrow y_C$. Moreover, the proposed controller has been for disturbance rejection performance to input disturbances, and it has been shown that for the case of $y_A \rightarrow y_B$ the controller can reject disturbance 55% the value of the set-point input. For the control case of $y_B \rightarrow y_C$, the disturbances that the controller can reject are up to 25% the set-point.

The proposed NLPID controller has also been benchmarked against the conventional control methods. In the case of the FOPTD system, the proposed controller shows stable control with fast performance. The proposed NLPID controller can minimise rise-time and settling time while maintaining minimal overshoot in the transient response. In contrast, the PID and 2D_PID controllers show lower rise-time and settling time of approximately one second when compared to the proposed NLPID controller. The proposed NLPID controller also outperforms the SP_PID controller with a similar rise time but a settling time of approximately 3 seconds lower. The proposed controller also shows similar system input energy efficiency when compared to the conventional methods, which is an advantage when considering the performance gains. The proposed NLPID controller has also been shown to be robust against ±10% parametric variations, showing minimal performance deterioration. The controller has also been tested against additive and multiplicative uncertainties. The controller managed to maintain stability against larger time constant dynamics in the additive uncertainty form, however, deterioration in performance is observed. Finally, the controller shows robust performance and robust stability against 20% time-delay uncertainty in the multiplicative form.

The proposed controller has been benchmarked against conventional methods in the NG-SOPTD system model. It has been shown that the proposed NLPID controller can control the system with no overshoot, however, it has a lower rise time and settling time of approximately 1 second against the PID, 2D_PID, and SP_PID controllers. It is also significantly slower in disturbance rejection when compared to the conventional methods. The proposed NLPID controller manages to control the system with no overshoot and the lowest system-input energy when compared to the conventional methods. The proposed NLPID controller has also been shown to be robust against ±10% variations in all the system parameters. However, the controller has a deterioration in performance with an increased overshoot, rise-time, and settling-time. For extensive robustness testing, the controller has also been simulated against additive and multiplicative uncertainties. The additive uncertainty considered the inclusion of second-order dynamics into the nominal plant. The proposed NLPID controller has been able to be robustly stable, however, there has been a deterioration in its performance, showing oscillatory response and a larger settling time of approximately 30 seconds. The proposed NLPID controller has also shown robustness in performance and stability to 20% uncertainty of the multiplicative form in the time delay.

The final benchmarking of the proposed NLPID controller is the NmP-SOPTD system. The proposed NLPID controller has shown no significant improvements in performance when compared to the conventional methods. The proposed NLPID controller can have a similar transient response with minimal overshoot, rise-time, and settling time when compared to PID, 2D_PID, and SP_PID. However, the proposed NLPID controller has the largest undershoot. The proposed NLPID controller shows excellent balance in input and output disturbance rejection. The proposed NLPID controller outperforms the PID and has a similar performance to the 2D_PID controller. The SP_PID controller outperforms all the controllers to reject input disturbances. However, the proposed NLPID controller shows similar performance to the SP_PID controller in output disturbance rejection, and they both outperform PID and 2D_PID controllers. The proposed NLPID controller has similar system input energy to the conventional methods. The proposed NLPID controller has also been shown to be robust under ±10% parametric uncertainty in all its parameters, showing deterioration in performance with increased settling and rise time, but effectively maintains a low overshoot. Moreover, the additive and multiplicative uncertainties have been considered for the NmP-SOPTD system. The additive uncertainty also considered the inclusion of second-order dynamics. The proposed controller remains robustly stable at the cost of deteriorated performance, showing an increase in oscillations and settling time. The proposed NLPID controller shows robust performance and stability against the 20% delay-uncertainty dynamics expressed in the multiplicative form.

In summary of the discussion points, a novel nonlinear PID controller has been proposed for the improvement of performance and robustness in nonlinear systems. The proposed controller has a unique set of functions that describe the gains in terms of the feedback error for the proportional and integral actions, while the feedback error rate is used for the derivative action. The proposed controller can eliminate step set-point derivative kicks, improving the controller costs and performance. The controller has been benchmarked against the conventional control methods for a large variety of linear systems. The benchmarking of the proposed NLPID controller has been conducted against the conventional controllers for linear systems only because it is common practice within industry to linearise nonlinear models and apply a gain scheduled PID controller for the various linear plants. This approach can become quite cumbersome, and it is difficult to ensure that the controller maintains stability between two gains and that the controller can provide discontinuous plant inputs, which is an ineffective behaviour. Consequently, it is common practice to interpolate between different gain values of the scheduling procedure and define it as a continuous look-up table. This is similar to the benchmarking methodology considered in Chapter 4, however, the step of gain interpolation is avoided by simply considering individual linear plants near their region of linearisation. The proposed NLPID controller can extend the range of operating points of the NCSTR system when compared to conventional PID without changing the tuning of the NLPID controller, which is not possible with other linear control alternatives. It can provide good performance for various step set-point inputs, including inputs that the system's behaviour is divergent near the steady state. This indicates that the proposed NLPID control algorithm is robust, extends the operating region of the linear control alternatives, and provides good control performance. Moreover, although the proposed NLPID controller has not been compared against gain scheduling PID, it has been shown that it has the potential to be used in the NCSTR system as a replacement to gain scheduling, requiring less time for design and analysis.

## 6.2. Advantages and Potential of the Proposed Controller

The proposed NLPID controller improves on the fundamental limitations of the PID controller, such as:

1. The ability to eliminate derivative kicks from the step set-point action.
2. The ability to improve simultaneous performance and robustness in nonlinear systems.
3. It shows improved control performance of highly nonlinear systems, FOPTD systems,

and SOPTD systems containing reverse dynamics with minimal to no overshoots.

Furthermore, the proposed NLPID controller may have some additional advantages over other nonlinear PID controllers in the literature such as:

1. It has an easy to understand effect of the gain parameters to the performance and stability.
2. It has fewer tunable parameters when compared to other nonlinear PID controllers.
3. It is a model-free control methodology and works for various nonlinear systems with robustness and excellent performance.
4. It shows potential for a simple and easy-to-implement hardware control structure.
5. Maintains the same performance for any step set-point function.

## 6.3. Limitations of the Proposed Nonlinear Functions PID Controller

The proposed NLPID controller has a lot of advantages that have been shown, and the ability to improve performance, robustness, and operating range in the case of nonlinear systems. It has been tested for a large variety of different models, however, it has some limitations. The limitations that have been noticed are as follows:

1. The proposed NLPID controller is limited in its performance against the NG-SOPTD system model, and it shows a slow and oscillatory disturbance rejection.
2. The proposed NLPID controller does not have rigorous stability proof. Consequently, in this thesis, an extensive simulation approach has been utilised for the linearised dynamics.
3. The proposed NLPID controller has not been tested for the dynamic input tracking of systems; it has only been tested for various step set-point inputs. This is a limitation since most set-point inputs are dynamic and change across time.

## 6.4. Difficulties Faced in Nonlinear Control

Nonlinear systems possess multiple equilibrium points and have many subsystems, actuators, and multiple forms of nonlinearities that can restrict the control capabilities. In addition, linearization is commonly required for linear controllers to control nonlinear systems adequately, using multiple operating conditions and designing multiple controllers. This is a time-consuming process that requires continual re-design and analysis, where linear controllers are mostly focused on robustness rather than performance. Nonlinear control algorithms have consequently become more prominent because it is possible to use the process model to determine a nonlinear controller that encompasses the key nonlinearities, essentially shaping a linear controller. However, this method requires adequate process knowledge and accurate modelling with low robustness capabilities. Moreover, adaptive control algorithms can be computationally demanding and require the solution of two problems, optimization and model identification. This increases the complexity of the controller and requires larger hardware memory that may not be cost-effective to implement.

The most common problems that are faced in nonlinear control are as follows:

1. Nonlinear controllers usually rely heavily on the mathematical accuracy of the plant models. This reduces the robustness and mostly under-performs.
2. Nonlinear controllers are more complex and face the difficulty of hardware implementation. This is particularly problematic when there is a lack of expertise, a lack of funding, or a lack of intelligent hardware with enough memory space for the application.
3. The ability of model-based nonlinear controllers to explicitly deal with saturation is a complex problem, and many nonlinear controllers cannot resolve that issue.
4. Most nonlinear control algorithms also require a large number of parameters for their design and tuning, making the tuning problem more complex and time-consuming.
5. Nonlinear PID controllers mostly do not have a stability theory to understand the theoretical foundations of the control algorithm. This is particularly limiting since linear controllers have a large theoretical toolset to help with the understanding of the feedback system and its design.
6. The theoretical development of nonlinear PID controllers with a rigorous stability analysis to help understand the dynamics and apply analytical tools to design the nonlinear controllers.

## 6.5. Key Outcomes of This Study

The key research aim was to develop a nonlinear PID controller with improved performance to nonlinear systems and robustness against uncertainty. This study has come to several conclusions regarding nonlinear PID control and the proposed NLPID control methodology. The study proceeded with simulating the proposed controller in various case scenarios, which included both linear and nonlinear systems. The proposed controller has undergone scrutiny to achieve its feasibility and comparison with contemporary methods. Firstly, the controller has been simulated against a highly nonlinear system, namely the NCSTR system. The nonlinear study showed in simulation that the proposed controller is tractable and can be easy to implement in a nonlinear system. In addition, it shows that the proposed controller is capable of regulating the NCSTR system in multiple steady states, including uncertainty in the dynamics. Secondly, the proposed controller was also benchmarked against a set of linearisation models of the NCSTR that are commonly seen in industry, namely the FOPTD, NG_SOPTD, and NmP_SOPTD systems. The benchmarking study gave a comparison of the proposed algorithm against alternative solutions of the common performance vs robustness trade-offs in PID and shows 8 the controller's capacity to improve upon the contemporary methods. The study showed that the proposed NLPID controller has shown its ability to improve the performance and robustness of nonlinear systems, and it was compared against the conventional methods such as PID, 2D_PID, and SP_PID. The comparative analysis was done for various linear systems and the proposed NLPID controller showed strength in the FOPTD system compared to the conventional methods. The weakness of the proposed controller was seen in the NG-SOPTD system, which indicated slow disturbance rejection. The proposed controller showed no significant improvements when compared to conventional methods in the NmP_SOPTD system. Thirdly, the design of the proposed controller has been achieved via a heuristic search method using the PSO algorithm and following with an indicative stability analysis to justify the design of the search algorithm constraints that formed the search space for the PSO. The indicative stability analysis determined the stable regions of gain parameters $k_0, k_1,$ and $k_2$ that generate a stabilisable NLPID controller for all three linearisations. Finally, the study also conducted an uncertainty analysis, and simulations were conducted to show the robustness of the proposed controller in various types of model uncertainties. It was shown that the proposed NLPID controller is robust against parametric, additive, and multiplicative uncertainties, showing the resilience and ability of the proposed controller to regulate systems under large variations. This also reflects the common approach of designing a linear controller for a linearised system, which then undergoes extensive robustness analysis under various uncertainties to ensure system stability.

## 6.6. Future Work

The proposed NLPID controller has shown in simulations its capacity to improve the performance and robustness of conventional PID controllers for nonlinear systems. In this project, there is still work to be done that is part of future work to showcase further the potential of the proposed controller. Consequently, some suggested future work of this thesis can be made in the following key uncovered areas.

1. Stability analysis of the feedback interconnection of the proposed NLPID controller with a nonlinear plant model.
2. The hardware application of the proposed NLPID controller to compare its effectiveness in practice against the simulation results.
3. Potential to combine the proposed NLPID controller with an observer design to improve its robustness and disturbance rejection.
4. Benchmark the proposed NLPID controller against other nonlinear algorithms.
5. The parameter value of 1 in the nonlinear functions can be replaced by a tunable parameter to give a different scaling effect to $r(t)$.

6. Provide weights to the objective function to prioritise the tuning for disturbance rejection rather than elimination of overshoot.

# Bibliography


Abushawish, A., Hamadeh, M. and Nassif, A. B. (2020). 'PID Controller Gains Tuning Using Metaheuristic Optimization Methods: A Survey'. In: *International Journal of Computers* 14, pp. 87–95. ISSN: 1998-4308. DOI: 10.46300/9108.2020.14.14. [Accessed: 4th Aug. 2023].

Alcántara, S., Vilanova, R. and Pedret, C. (2013). 'PID Control in Terms of Robustness/Performance and Servo/Regulator Trade-Offs: A Unifying Approach to Balanced Autotuning'. In: *Journal of Process Control* 23.4, pp. 527–542. ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2013.01.003. [Accessed: 16th Oct. 2023].

Alfaro, V. M., Vilanova, R. and Arrieta, O. (2008). 'Two-Degree-of-Freedom PI/PID Tuning Approach for Smooth Control on Cascade Control Systems'. In: *2008 47th IEEE Conference on Decision and Control*, pp. 5680–5685. DOI: 10.1109/CDC.2008.4738796.

Åström, K. J. (2000). 'Limitations on Control System Performance'. In: *European Journal of Control* 6.1, pp. 2–20. ISSN: 0947-3580. DOI: 10.1016/S0947-3580(00)70906-X. [Accessed: 5th Dec. 2022].

Åström, K. J. and Hägglund, T. (1995). *PID Controllers: Theory, Design and Tuning*. 2nd. Instrument Society of America. ISBN: 1-55617-516-7.

— (2001). 'The Future of PID Control'. In: *Control Engineering Practice*. PID Control 9.11, pp. 1163–1175. ISSN: 0967-0661. DOI: 10.1016/S0967-0661(01)00062-4. [Accessed: 22nd Feb. 2021].

— (2004). 'Revisiting the Ziegler–Nichols Step Response Method for PID Control'. In: *Journal of Process Control* 14.6, pp. 635–650. ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2004.01.002. [Accessed: 5th Dec. 2022].

Atherton, D. and Majhi, S. (1999). 'Limitations of PID Controllers'. In: *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*. Vol. 6, 3843–3847 vol.6. DOI: 10.1109/ACC.1999.786236.

Bernstein, D. S. (2022). 'Facing Future Challenges in Feedback Control of Aerospace Systems Through Scientific Experimentation'. In: *Journal of Guidance, Control, and Dynamics* 45.12, pp. 2202–2210. ISSN: 0731-5090. DOI: 10.2514/1.G006785. [Accessed: 4th Aug. 2023].

Boubaker, O. (2013). 'The Inverted Pendulum Benchmark in Nonlinear Control Theory: A Survey'. In: *International Journal of Advanced Robotic Systems* 10.5, p. 233. ISSN: 1729-8814. DOI: 10.5772/55058. [Accessed: 12th Feb. 2021].

Chaturvedi, S., Kumar, N. and Kumar, R. (2023). 'A PSO-optimized Novel PID Neural Network Model for Temperature Control of Jacketed CSTR: Design, Simulation, and a Comparative Study'. In: *Soft Comput.* ISSN: 1433-7479. DOI: 10.1007/s00500-023-09138-0. [Accessed: 6th Sept. 2023].

Chen, J., Fang, S. and Ishii, H. (2019). 'Fundamental Limitations and Intrinsic Limits of Feedback: An Overview in an Information Age'. In: *Annual Reviews in Control* 47, pp. 155–177. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2019.03.011. [Accessed: 4th Aug. 2023].

Cheng, L. (2021). 'Adaptive Neural Network Control of Nonlinear Systems with Unknown Dynamics'. In: *Advances in Space Research*, p. 10.

Chong, S.-H. et al. (2021). 'Enhanced Nonlinear PID Controller for Positioning Control of Maglev System'. In: *Control Based on PID Framework - The Mutual Promotion of Control and Identification for Complex Systems*. IntechOpen. ISBN: 978-1-83968-367-1. DOI: 10.5772/intechopen.96769. [Accessed: 4th Aug. 2023].

Clerc, M. and Kennedy, J. (2002). 'The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space'. In: *IEEE Transactions on Evolutionary Computation* 6.1, pp. 58–73. ISSN: 1941-0026. DOI: 10.1109/4235.985692.

Colantonio, M. C., Desages, A. C., Romagnoli, J. A. and Palazoglu, A. (1995). 'Nonlinear Control of a CSTR: Disturbance Rejection Using Sliding Mode Control'. In: *Ind. Eng. Chem. Res.* 34.7, pp. 2383–2392. ISSN: 0888-5885. DOI: 10.1021/ie00046a022. [Accessed: 20th July 2023].

Crespi, G. P., La Torre, D. and Rocca, M. (2003). 'Second-Order Mollified Derivatives and Optimization'. In: *Rend. Circ. Mat. Palermo* 52.2, pp. 251–262. ISSN: 0009-725X, 1973-4409. DOI: 10.1007/BF02872232. [Accessed: 31st July 2023].

Dangor, M., Dahunsi, O. A., Pedro, J. O. and Ali, M. M. (2014). 'Evolutionary Algorithm-Based PID Controller Tuning for Nonlinear Quarter-Car Electrohydraulic Vehicle Suspensions'. In: *Nonlinear Dyn* 78.4, pp. 2795–2810. ISSN: 1573-269X. DOI: 10.1007/s11071-014-1626-4. [Accessed: 24th Feb. 2021].

Faccin, F. and Trierweiler, J. O. (n.d.). 'Performance Limitations of Some Industrial PID Controllers'. In: (), p. 6.

Fraga-Gonzalez, L. F., Fuentes-Aguilar, R. Q., Garcia-Gonzalez, A. and Sanchez-Ante, G. (2017). 'Adaptive Simulated Annealing for Tuning PID Controllers'. In: *AI Communications* 30.5, pp. 347–362. ISSN: 09217126. DOI: 10.3233/AIC-170741. [Accessed: 26th Feb. 2021].

Frank, S. A. (2018). *Control Theory Tutorial Basic Concepts Illustrated by Examples.* SpringerBriefs in Applied Sciences and Technology. Springer Cham. ISBN: 978-3-319-91706-1.

Gahinet, P., Chen, R. and Eryilmaz, B. (2013). 'Automated PID Controller Design'. US8467888B2. [Accessed: 5th Dec. 2022].

Garpinger, O., Hägglund, T. and Åström, K. J. (2014). 'Performance and Robustness Trade-Offs in PID Control'. In: *Journal of Process Control* 24.5, pp. 568–577. ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2014.02.020. [Accessed: 31st Mar. 2022].

Gnanamurugan, M. and Senthilkumar, M. (2018). 'Smith Predictor for Control of the Process with Long Dead Time'. In: *IJERT* 2.6. ISSN: (Online) 2278-0181.

Gomes, M., Bássora, L., Morandin, O. and Vivaldini, K. (2016). 'PID Control Applied on a Line-Follower AGV Using a RGB Camera'. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 194–198. DOI: 10.1109/ITSC.2016.7795553.

Gomez, V. et al. (2020). 'Pareto Optimal PID Tuning for Px4-Based Unmanned Aerial Vehicles by Using a Multi-Objective Particle Swarm Optimization Algorithm'. In: *Aerospace* 7.6, p. 71. DOI: http://dx.doi.org/10.3390/aerospace7060071. [Accessed: 23rd Feb. 2021].

Hale, J. K. and LaSalle, J. P. (1963). 'Differential Equations: Linearity vs. Nonlinearity'. In: SIAM Review 5.3, pp. 249–272.

Han, J. (1994). 'A New Type of Controller: NLPID Control and Decision (Chinese)'. In: *Control and Decision* 9.6, pp. 401–407.

Harmon Ray, W. (1981). *Advanced Process Control.* Butterworths. ISBN: 0-409-90231-4.

Hirahara, H., Tanaka, A., Yamamoto, S. and Ara, T. (2017). 'Systematic Design of Two-Degree-of-Freedom PID Controller for Positioning and Tracking Drives of Linear Servo Motors'. In: *IEEJ Journal IA* 6.6, pp. 443–448. ISSN: 2187-1094, 2187-1108. DOI: 10.1541/ieejjia.6.443. [Accessed: 24th Feb. 2021].

Hua, H., Fang, Y., Zhang, X. and Qian, C. (2020). 'Auto-Tuning Nonlinear PID-type Controller for Rotorcraft-Based Aggressive Transportation'. In: *Mechanical Systems and Signal Processing* 145, p. 106858. ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2020.106858. [Accessed: 24th Feb. 2021].

Hung, S.-B., Yu, C.-C. and Cheng, Y.-C. (2004). 'A Smith Predictor Enhanced PID Controller'. In: *IFAC Proceedings Volumes* 37.9, pp. 661–666. ISSN: 14746670. DOI: 10.1016/S1474-6670(17)31885-2. [Accessed: 1st Feb. 2023].

Iqbal, J., Ullah, M., Khan, S. G., Khelifa, B. and Ćuković, S. (2017). 'Nonlinear Control Systems - A Brief Overview of Historical and Recent Advances'. In: *Nonlinear Engineering* 6.4, pp. 301–312. ISSN: 21928010. DOI: http://dx.doi.org/10.1515/nleng-2016-0077. [Accessed: 16th Feb. 2021].

Isidori, A. (1995). *Nonlinear Control Systems*. 3rd. Springer Berlin Heidelberg. ISBN: 3-540-19916-0.

Jin, G.-G. and Son, Y.-D. (2019). 'Design of a Nonlinear PID Controller and Tuning Rules for First-Order Plus Time Delay Models'. In: *STUD INFORM CONTROL* 28.2. ISSN: 12201766, 1841429X. DOI: 10.24846/v28i2y201904. [Accessed: 5th May 2023].

Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O. and Khammas, B. M. (2022). 'Metaheuristic Algorithms for PID Controller Parameters Tuning: Review, Approaches and Open Problems'. In: *Heliyon* 8.5, e09399. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2022.e09399. [Accessed: 4th Aug. 2023].

Kapsalis, D., Sename, O., Milanés, V. and Martinez, J. J. (2020). 'Gain-Scheduled Steering Control for Autonomous Vehicles'. In: *IET Control Theory & Applications* 14.20, pp. 3451–3460. ISSN: 1751-8652. DOI: 10.1049/iet-cta.2020.0698. [Accessed: 24th Apr. 2021].

Kasim, I. and Riyadh, W. (2016). 'On the Improved Nonlinear Tracking Differentiator Based Nonlinear PID Controller Design'. In: *ijacsa* 7.10. ISSN: 21565570, 2158107X. DOI: 10.14569/IJACSA.2016.071032. [Accessed: 12th Feb. 2021].

Khalil, H. K. (2002). *Nonlinear Systems*. third. United States of America: Prentice Hall. ISBN: 0-13-067389-7.

Konstantopoulos, G. C. and Baldivieso-Monasterios, P. R. (2020). 'State-Limiting PID Controller for a Class of Nonlinear Systems with Constant Uncertainties'. In: *International Journal of Robust and Nonlinear Control* 30.5, pp. 1770–1787. ISSN: 1099-1239. DOI: 10.1002/rnc.4853. [Accessed: 24th Feb. 2021].

Korkmaz, M., Aydoğdu, Ö. and Doğan, H. (2012). 'Design and Performance Comparison of Variable Parameter Nonlinear PID Controller and Genetic Algorithm Based PID Controller'. In: *2012 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 1–5. DOI: 10.1109/INISTA.2012.6246935.

Krishna, D., Suryanarayana, K., Aparna, G. and Padma Sree, R. (2012). 'Tuning of PID Controllers for Continuous Stirred Tank Reactors'. In: *Taylor & Francis* 54.3, pp. 157–179. ISSN: 0019-4506. DOI: 10.1080/00194506.2012.766005.

Lawrence, C. E. (1998). *Partial Differential Equations*. Vol. 19. Graduate Studies in Mathematics. American Mathematical Society. ISBN: 0-8218-0772-2.

Liu, T. (2022). 'Research on Stability of Hydraulic System Based on Nonlinear PID Control'. In: *Nonlinear Engineering* 11, pp. 494–499. DOI: 10.1515/nleng-2022-0222. [Accessed: 4th Aug. 2023].

Maddi, A., Guessoum, A. and Berkani, D. (2014). 'Design of Nonlinear PID Controllers Based on Hyper-Stability Criteria'. In: *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 736–741. DOI: 10.1109/STA.2014.7086743.

Mangera, M., Pedro, J. O. and Panday, A. (2022). 'GA-optimised Nonlinear Pseudo-Derivative Feedback Control of a Sustainable, High-Speed, Ultratall Building Elevator'. In: *Int. J. Dynam. Control* 10.6, pp. 1903–1921. ISSN: 2195-268X, 2195-2698. DOI: 10.1007/s40435-022-00929-1. [Accessed: 5th Jan. 2023].

Mo, H. and Farid, G. (2019). 'Nonlinear and Adaptive Intelligent Control Techniques for Quadrotor UAV – A Survey'. In: *Asian Journal of Control* 21.2, pp. 989–1008. ISSN: 1934-6093. DOI: 10.1002/asjc.1758. [Accessed: 15th Feb. 2021].

Mohan, V., Chhabra, H., Rani, A. and Singh, V. (2019). 'An Expert 2DOF Fractional Order Fuzzy PID Controller for Nonlinear Systems'. In: *Neural Computing & Applications* 31.8, pp. 4253–4270. ISSN: 09410643. DOI: http://dx.doi.org/10.1007/s00521-017-3330-z. [Accessed: 17th Feb. 2021].

Normey-Rico, J. E. and Camacho, E. F. (2007). *Control of Dead-Time Processes*. Advanced Textbooks in Control and Signal Processing. Springer London. ISBN: 978-1-84628-828-9.

Normey-Rico, J. E. and Camacho, E. F. (2008). 'Dead-Time Compensators: A Survey'. In: *Control Engineering Practice*. Special Section on Manoeuvering and Control of Marine Craft 16.4, pp. 407–428. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2007.05.006. [Accessed: 14th Dec. 2023].

Normey-Rico, J. E., Santos, T. L. M., Flesch, R. C. C. and Torrico, B. C. (2022). 'Control of Dead-Time Process: From the Smith Predictor to General Multi-Input Multi-Output Dead-Time Compensators'. In: *Frontiers in Control Engineering* 3. ISSN: 2673-6268. [Accessed: 14th Dec. 2023].

O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*. 3rd ed. London : Hackensack, NJ: Imperial College Press ; Distributed by World Scientific Pub. ISBN: 978-1-84816-242-6.

Parnianifard, A. and Azfanizam, A. S. (2020). 'Metamodel-Based Robust Simulation-Optimization Assisted Optimal Design of Multiloop Integer and Fractional-Order PID Controller'. In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 33.1. e2679 JNM-18-0321.R2, e2679. ISSN: 1099-1204. DOI: 10.1002/jnm.2679. [Accessed: 3rd Mar. 2021].

Parnianifard, A., Fakhfakh, M., Kotti, M., Zemouche, A. and Wuttisittikulkij, L. (2021). 'Robust Tuning and Sensitivity Analysis of Stochastic Integer and Fractional-Order PID Control Systems: Application of Surrogate-Based Robust Simulation-Optimization'. In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 34.2, e2835. ISSN: 1099-1204. DOI: 10.1002/jnm.2835. [Accessed: 3rd Mar. 2021].

Parnianifard, A., Zemouche, A., Chancharoen, R., Imran, M. A. and Wuttisittikulkij, L. (2020). 'Robust Optimal Design of FOPID Controller for Five Bar Linkage Robot in a Cyber-Physical System: A New Simulation-Optimization Approach'. In: *PLoS One* 15.11, e0242613. DOI: http://dx.doi.org/10.1371/journal.pone.0242613. [Accessed: 24th Feb. 2021].

Parsopoulos, K. E. and N.Varsatis, M. (2002). 'Particle Swarm Optimization Method for Constrained Optimization Problem'. In: *Intelligent Technologies-Theory and Applications: New Trends in Intelligent Technologies*. Vol. 76. Frontiers in Artificial Intelligence and Applications. Fairfax, VA, U.S.A.: IOS Press, pp. 214–220.

Pathak, D., Bhati, S. and Gaur, P. (2020). 'Fractional-Order Nonlinear PID Controller Based Maximum Power Extraction Method for a Direct-Driven Wind Energy System'. In: *International Transactions on Electrical Energy Systems* 30.12, e12641. ISSN: 2050-7038. DOI: 10.1002/2050-7038.12641. [Accessed: 17th Mar. 2021].

Pesce, V., Colagrossi, A. and Silvestrini, S. (2023). *Modern Spacecraft Guidance, Navigation, and Control*. Elsevier. ISBN: 978-0-323-90916-7.

Pugazhenthi P, N., Selvaperumal, S. and Vijayakumar, K. (2021). 'Nonlinear PID Controller Parameter Optimization Using Modified Hybrid Artificial Bee Colony Algorithm for Continuous Stirred Tank Reactor'. In: *Polska Akademia Nauk. Bulletin of the Polish Academy of Sciences: Technical Sciences* 69.3. ISSN: 02397528. DOI: 10.24425/bpasts.2021.137348. [Accessed: 4th Aug. 2023].

Qin, Y., Zhao, G., Hua, Q., Sun, L. and Nag, S. (2019). 'Multiobjective Genetic Algorithm-Based Optimization of PID Controller Parameters for Fuel Cell Voltage and Fuel Utilization'. In: *Sustainability* 11.12, p. 20. DOI: 10.3390/su11123290.

Rakesh, M. K., Satheesh, b. I. and Thirunavukkarasu (2014). 'Design and Implementation of Two-Degree-of-Freedom Nonlinear PID Controller for a Nonlinear Process'. In: *IOSR-JEEE* 9.3, pp. 59–64. ISSN: 2278-1676.

Reddy, H., Immanuel, J., Parvathi, C., Bhaskar, P. and Sudheer, L. (2011). 'Implementation of PID Controller in MATLAB for Real Time DC Motor Speed Control System'. In: *Sensors & Transducers* 126, pp. 110–118.

Rehman, O. U., Petersen, I. R. and Pota, H. (2017). 'Nonlinear Robust State Feedback Control System Design for Nonlinear Uncertain Systems'. In: *International Journal of Robust and Nonlinear Control* 27.13, pp. 2234–2252. ISSN: 1099-1239. DOI: 10.1002/rnc.3680. [Accessed: 17th Feb. 2021].

Rezaei, A. and Hashemzade, F. (2016). 'Absolute Stability of Nonlinear Systems with Piecewise Linear Sector Condition'. In: *IJCA* 134.9, pp. 15–18. ISSN: 09758887. DOI: 10.5120/ijca2016908000. [Accessed: 1st Dec. 2022].

Saxena, A. and Dubey, Y. (2019). 'ITEE Journal A Comparative Analysis of Optimization Algorithms for Self Tuning PID Controllers'. In: 8, pp. 1–6.

Scherlozer, A., Orsini, M. and Patole, S. (2016). 'Simulation and Numerical Analysis and Comparative Study of a PID Controller Based on Ziegler-Nichols and Auto Turning Method'. In: *12th IEEE International Conference on Control and Automation*. Kathmandu, Nepal. [Accessed: 9th Jan. 2023].

Schröders, S., Maier, C. C., Ebner, W., Fidlin, A. and Hametner, C. (2020). 'Two-Degree-of-Freedom MIMO Control for Hydraulic Servo-Systems with Switching Properties'. In: *Control Engineering Practice* 95, p. 104246. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2019.104246. [Accessed: 24th Feb. 2021].

Shaikh, M. and Yadav, D. (2022). 'A Review of Particle Swarm Optimization ( PSO) Algorithm'. In: DOI: 10.17605/OSF.IO/KQ34H. [Accessed: 4th Aug. 2023].

Shamseldin, M. A. (2023). 'Design of Auto-Tuning Nonlinear PID Tracking Speed Control for Electric Vehicle with Uncertainty Consideration'. In: *World Electric Vehicle Journal* 14.4, p. 78. ISSN: 2032-6653. DOI: 10.3390/wevj14040078. [Accessed: 4th Aug. 2023].

Sharma, R., Gaur, P. and Mittal, A. P. (2015). 'Performance Analysis of Two-Degree of Freedom Fractional Order PID Controllers for Robotic Manipulator with Payload'. In: *ISA*

*Transactions* 58, pp. 279–291. ISSN: 0019-0578. DOI: 10.1016/j.isatra.2015.03.013. [Accessed: 25th Feb. 2021].

Showalter, R. E. (1994). 'Hilbert Space Methods for Partial Differential Equations'. In: *ejde* Mon. 01, pp. 1–214. ISSN: 1072-6691. DOI: 10.58997/ejde.mon.01. [Accessed: 31st July 2023].

Sigurd Skogestad, C. G. (2018). 'Should We Forget the Smith Predictor?' In: *IFAC-PapersOnLine*. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018 51.4, pp. 769–774. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2018.06.203. [Accessed: 14th Dec. 2023].

Sinha, A. and Mishra, R. K. (2018). 'Control of a Nonlinear Continuous Stirred Tank Reactor via Event Triggered Sliding Modes'. In: *Chemical Engineering Science* 187, pp. 52–59. ISSN: 0009-2509. DOI: 10.1016/j.ces.2018.04.057. [Accessed: 20th July 2023].

Sivadasan, J., Iruthayarajan, M. W., Stonier, A. A. and Raymon, A. (2023). 'Design of Cross-Coupled Nonlinear PID Controller Using Single-Objective Evolutionary Algorithms'. In: *Mathematical Problems in Engineering* 2023, e7820047. ISSN: 1024-123X. DOI: 10.1155/2023/7820047. [Accessed: 4th Aug. 2023].

Skogestad, S. and Postlethwaite, I. (2001). *Multivariable Feedback Control: Analysis and Design*. Second. Chichester, New York, Brisbane, Toronto, Singapore: John Wiley & Sons, Incorporated. ISBN: 978-0-470-01168-3.

Slama, S., Errachdi, A. and Benrejeb, M. (2019). 'Neural Adaptive PID and Neural Indirect Adaptive Control Switch Controller for Nonlinear MIMO Systems'. In: *Mathematical Problems in Engineering* 2019, pp. 1–11. ISSN: 1024-123X, 1563-5147. DOI: 10.1155/2019/7340392. [Accessed: 24th Feb. 2021].

Slotine, J.-J. E. and Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, N.J: Prentice Hall. ISBN: 978-0-13-040890-7.

So, G.-B. (2019). 'EA-Based Design of a Nonlinear PID Controller Using an Error Scaling Technique'. In: *SIC* 28.3, pp. 279–288. ISSN: 12201766, 1841429X. DOI: 10.24846/v28i3y201904. [Accessed: 8th Mar. 2021].

— (2021). 'A Modified 2-DOF Control Framework and GA Based Intelligent Tuning of PID Controllers'. In: *Processes* 9.3, p. 423. ISSN: 2227-9717. DOI: 10.3390/pr9030423. [Accessed: 25th Mar. 2022].

So, G.-B. and Jin, G.-G. (2018). 'Fuzzy-Based Nonlinear PID Controller and Its Application to CSTR'. In: *Korean J. Chem. Eng.* 35.4, pp. 819–825. ISSN: 1975-7220. DOI: 10.1007/s11814-017-0329-1. [Accessed: 20th July 2023].

Somefun, O. A., Akingbade, K. and Dahunsi, F. (2021). 'The Dilemma of PID Tuning'. In: *Annual Reviews in Control* 52, pp. 65–74. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2021.05.002. [Accessed: 5th Dec. 2022].

Son, Y.-D., Bin, S.-D. and Jin, G.-G. (2021). 'Stability Analysis of a Nonlinear PID Controller'. In: *Int. J. Control Autom. Syst.* 19.10, pp. 3400–3408. ISSN: 2005-4092. DOI: 10.1007/s12555-020-0599-y. [Accessed: 30th Nov. 2022].

Su, Y. X., Sun, D. and Duan, B. Y. (2005). 'Design of an Enhanced Nonlinear PID Controller'. In: *Mechatronics* 15.8, pp. 1005–1024. ISSN: 0957-4158. DOI: 10.1016/j.mechatronics.2005.03.003. [Accessed: 24th Feb. 2021].

Sung, S. W. and Lee, I.-B. (1996). 'Limitations and Countermeasures of PID Controllers'. In: *Ind. Eng. Chem. Res.* 35.8, pp. 2596–2610. ISSN: 0888-5885, 1520-5045. DOI: 10.1021/ie960090+. [Accessed: 1st Dec. 2022].

Suthar, H. A. (2015). 'Two-Degree-of-Freedom PID Controller, Its Equivalent Forms and Special Cases'. In: *IAES International Journal of Robotics and Automation* 4.4, pp. 269–283. [Accessed: 18th Feb. 2021].

Taguchi, H. and Araki, M. (2000). 'Two-Degree-of-Freedom PID Controllers — Their Functions and Optimal Tuning'. In: *IFAC Proceedings Volumes*. IFAC Workshop on Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, 5-7 April 2000 33.4, pp. 91–96. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)38226-5. [Accessed: 25th Feb. 2021].

Valluru, S. K. (2017). 'Stabilization of Nonlinear Inverted Pendulum System Using MOGA and APSO Tuned Nonlinear PID Controll'. In: *Cogent Engineering*, p. 16.

Valluru, S. K. and Singh, M. (2018). 'Performance Investigations of APSO Tuned Linear and Nonlinear PID Controllers for a Nonlinear Dynamical System'. In: *Journal of Electrical Systems and Information Technology* 5.3, pp. 442–452. ISSN: 2314-7172. DOI: 10.1016/j.jesit.2018.02.001. [Accessed: 12th Feb. 2021].

Wang, D., Tan, D. and Liu, L. (2018). 'Particle Swarm Optimization Algorithm: An Overview'. In: *Soft Comput* 22.2, pp. 387–408. ISSN: 1432-7643, 1433-7479. DOI: 10.1007/s00500-016-2474-6. [Accessed: 4th July 2021].

Wang, L. (2020). *PID Control System Design and Autotuning Using MATLAB/Simulink*. Wiley-IEEE Press. ISBN: 978-1-119-46934-6.

Wang, X., Yan, X., Li, D. and Sun, L. (2018). 'An Approach for Setting Parameters for Two-Degree-of-Freedom PID Controllers'. In: *Algorithms* 11.4, p. 48. DOI: http://dx.doi.org/10.3390/a11040048. [Accessed: 24th Feb. 2021].

Wu, Y., Liu, L., Yang, Y. and Dai, S. (2019). 'Optimal Control Method for Robot-Tracking Based on Control-Lyapunov-Function'. In: *IEEE Access* PP, pp. 1–1. DOI: 10.1109/ACCESS.2019.2923658.

Zaidner, G. et al. (2010). 'Non Linear PID and Its Application in Process Control'. In: *2010 IEEE 26-Th Convention of Electrical and Electronics Engineers in Israel*, pp. 000574–000577. DOI: 10.1109/EEEI.2010.5662155.

Zerong, Y. and Zhigang, W. (2017). 'Smith Predictive PID Control in Vapor Temperature Control of Ground Source Heat Pump System'. In: *GRC Transactions* 41.

Zhang, B. and Mao, Z. (2017). 'A Robust Adaptive Control Method for Wiener Nonlinear Systems'. In: *International Journal of Robust and Nonlinear Control* 27.3, pp. 434–460. ISSN: 1099-1239. DOI: 10.1002/rnc.3580. [Accessed: 16th Feb. 2021].

Zhou, K. and Doyle, J. C. (1998). *Essentials of Robust Control*. United States of America: Prentice Hall. ISBN: 0-13-525833-2.

# Appendices

# Appendix A

# MATLAB PSO Algorithm

```
%%      PARTICLE SWARM OPTIMIZATION ALGORITHM FOR SIMULINK SIMULATIONS
    %%
clc;
clear;
clear all;
%                   Initialize Simulation Parameters
r = 1; % Set-Point
kp = 1; % Gain System Parameter
tp = 1; % Lag System Parameter
l = 0; % Delay Parametric Uncertainty
k3 = 0.5;
%             Initialize Optimization procedure PSO Algorithm
tic; % Begin timer
nvars = 4; % Number of variables to be optimized
PN = 30; % Number of Particles
MaxIteration = 90; % Maximum number of iterations
x = 2*abs(rand(PN,nvars))+0*(1-abs(rand(PN,nvars))); % Begin random
    particles
X = struct; % Build structure
X.Particle = x; % Embed random particles that represent PID Gains in
    structure
% Initialization
% Literature suggested parameters:%Parameters that are changeable.
c1 = 1.3;
% iteration step 1:MaxIteration
V = struct; % Define Velocity parameter Matrix
V.Particle = zeros(PN,nvars);  % Assign initial velocity matrix to zero
Pbest = zeros(PN, nvars);
Gbest = zeros(MaxIteration, nvars+1);
% Maximum/Minimum values of the particles
k0max = 2;
k1max = 2;
k2max = 2;
amax = 2;
amin = 0.5;
%                 Initialization of iterations 0, 1.
```

```
                        %
for j = 1:PN
    if abs(X(1).Particle(j,1))>=k0max
        X(1).Particle(j,1)=k0max*abs(rand(1))+0*(1-abs(rand(1)));
        k0 = X(1).Particle(j,1);
    else
        k0 = X(1).Particle(j,1); % Proportional Gain
    end
    if abs(X(1).Particle(j,2))>=k1max
        X(1).Particle(j,2)= k1max*abs(rand(1))+0*(1-abs(rand(1)));
        k1 = X(1).Particle(j,2);
    else
        k1 = X(1).Particle(j,2); % Integral Gain
    end
    if abs(X(1).Particle(j,3))>=amax
        X(1).Particle(j,3)= amax*abs(rand(1))+0*(1-abs(rand(1)));
        a = X(1).Particle(j,3);
    else
        a = X(1).Particle(j,3); % Proportional Gain
    end
    if abs(X(1).Particle(j,4))>=k2max
        X(1).Particle(j,4) = k2max*abs(rand(1))+0*(1-abs(rand(1)));
        k2 = X(1).Particle(j,4);
    else
        k2 = X(1).Particle(j,4); % Proportional Gain
    end
    % Simulate with Simulink
    try
        s = sim('PerformanceTest.slx','TimeOut',10);
    catch
    end
    % Extract error from simulink simulation (this uses the "To
        Worskspace" structure in Simulink.
    X(j).Error = s.err.Data;
    stepresp = stepinfo(s.output.Data,s.tout);
    settime = stepresp.SettlingTime;
    Over = stepresp.Overshoot;
    % Evaluate objective function (Any objective function can be
        defined)
    X(1).Objective(j) = ise(X(j).Error,s.t.Data,settime);
    [B, index] = min(nonzeros(X(1).Objective));
    globe(1,1) = index;
```

```
    Gbest(1,1:nvars) = X(1).Particle(index,:);
    Gbest(1,end) = B;
    for k = 1:nvars
        r1 = rand(1);
        phi1 = r1*c1;
        V(2).Particle(j,k) = abs((V(1).Particle(j,k)+phi1*(Gbest(1,k)-X
            (1).Particle(j,k))));
        X(2).Particle(j,k) = abs(X(1).Particle(j,k)+V(2).Particle(j,k))
            ;
    end
end
%                       Algorithm Iteration Procedure
                         %
for i = 2:MaxIteration
    for j = 1:PN
        % Simulate for 1 up to Maximum iterations
        if abs(X(i).Particle(j,1))>=k0max
            X(i).Particle(j,1)=k0max*abs(rand(1))+0*(1-abs(rand(1)));
            k0 = X(i).Particle(j,1);
        else
            k0 = X(i).Particle(j,1); % Proportional Gain
        end
        if abs(X(i).Particle(j,2))>=k1max
            X(i).Particle(j,2)=k1max*abs(rand(1))+0*(1-abs(rand(1)));
            k1 = X(i).Particle(j,2);
        else
            k1 = X(i).Particle(j,2); % Integral Gain
        end
        if abs(X(i).Particle(j,3))>=amax
            X(i).Particle(j,3)=amax*abs(rand(1))+amin*(1-abs(rand(1)));
            a = X(i).Particle(j,3);
        elseif abs(X(i).Particle(j,3))<=amin
            X(i).Particle(j,3)=amax*abs(rand(1))+amin*(1-abs(rand(1)));
            a = X(i).Particle(j,3);
        else
            a = X(i).Particle(j,3); % Proportional Gain
        end
        if abs(X(i).Particle(j,4))>=k2max
            X(i).Particle(j,4) = k2max*abs(rand(1))+0*(1-abs(rand(1)));
            k2 = X(i).Particle(j,4);
        else
            k2 = X(i).Particle(j,4); % Proportional Gain
```

```
        end
        % Simulate with Simulink
        try
            s = sim('PerformanceTest.slx','TimeOut',10);
        catch
        end
        % Extract error from simulink simulation
        X(j).Error = s.err.Data;
        stepresp = stepinfo(s.output.Data,s.tout);
        settime = stepresp.SettlingTime;
        Over = stepresp.Overshoot;
        % Evaluate objective function
        X(i).Objective(j) =  ise(X(j).Error,s.t.Data,settime);
        for p = 1:i
            L(p) = X(p).Objective(j);
        end
    end
    % Find global best
    [B, index] = min(nonzeros(X(i).Objective));
    globe(i,1) = index;
    % Take all objectives and find the smallest
    for p = 1:i
        CompareObjectives(p,1) = X(i).Objective(globe(p,1));
        [SmallestObjective, index] = min(CompareObjectives);
    end
    if X(i).Objective(globe(index,1)) < SmallestObjective
        Gbest(i,1:nvars) = X(i).Particle(globe(index,1),:);
        Gbest(i,nvars+1) = X(i).Objective(globe(index,1));
    elseif X(i).Objective(globe(index,1)) >= SmallestObjective
        Gbest(i,1:nvars) = X(i-1).Particle(globe(i-1),:);
        Gbest(i,nvars+1) = X(i).Objective(globe(index,1));
    end
    for j = 1:PN
        for k = 1:nvars
            r1 = rand(1);
            phi1 = c1*r1;
            V(i+1).Particle(j,k) = abs((V(i).Particle(j,k)+phi1*(Gbest(
                i,k)-X(i).Particle(j,k))));
            X(i+1).Particle(j,k) = abs(X(i).Particle(j,k)+V(i+1).
                Particle(j,k));
        end
    end
```

```
end
%                    Extract the best values into the simulation
                 %
[Val, m1] = min(nonzeros(Gbest(:,end)));
k0 = Gbest(m1,1); % Proportional Gain
k1 = Gbest(m1,2); % Integral Gain
k2 = Gbest(m1,4); % Proportional Gain
a = Gbest(m1,3); % Proportional Gain

s = sim('PerformanceTest.slx','TimeOut',10);

t = toc; % End timer
```

# Appendix B

# Optimization Objective Function

The optimization problem was established to compute the ITAE performance measure as the objective function as shown below.

$$
\begin{aligned}
\underset{k_0, k_1, k_2, a}{\text{minimize}} \quad & f(t, \epsilon(t), t_s) = \int_0^{t_f} t|\epsilon(t)| \, \mathrm{d}t + t_s \\
\text{subject to} \quad & k_{min} \leqslant k_0, k_1, k_2 \leqslant k_{max}, \\
& a_{min} \leqslant a \leqslant a_{max}
\end{aligned}
\tag{B.1}
$$

where $t_s$ is the settling time, $\epsilon(t)$ is the feedback error, and $t_f$ is the final time.

The following MATLAB code represents the above objective function evaluation, where the constraints are implemented in the MATLAB PSO Code shown in Appendix A.

```
function [Ob] = ise(err,t,st)

Ob = sum(t.*abs(err))+st;

end
```