FEDERAL UNIVERSITY OF PARÁ
BELÉM CAMPUS
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

Daniel Abreu Macedo da Silva

DM 29/2023

Analysis of Classical and Advanced Control Techniques Tuned with Reinforcement Learning

Belém/PA
2023

Daniel Abreu Macedo da Silva

Analysis of Classical and Advanced Control Techniques Tuned with Reinforcement Learning

Master's Thesis submitted to the Graduate Program in Electrical Engineering from Federal University of Pará to achieve the title of Master Degree in Electrical Engineering.
Supervisor: Professor Antônio da Silva Silveira, Dr.

Belém/PA
2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Biblioteca do ITEC/UFPA-Belém-PA

S586e Silva, Daniel Abreu Macedo da, 1999-

Analysis of classical and advanced control techniques em tuned with reinforcement learning / Daniel Abreu Macedo da Silva.-2023.

Orientador: Antonio da Silva Silveira.

Dissertação (Mestrado) – Instituto de Tecnologia, Universidade Federal do Pará, Programa de Pós-Graduação em Engenharia Elétrica,

Belém, 2023.

1. Teoria do controle. 2. Inteligência artificial. 3. Controladores Programáveis. I. Título.

CDD 23. ed. – 6 2 9 . 8 3 1 2

Elaborado por Lucicléa Silva de Oliveira – CRB-2/648

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARÁ
POS-GRADUACAO EM ENGENHARIA ELETRICA

ATA DE DEFESA DE DISSERTAÇÃO Nº 29/2023 - PPGEE (11.41.18)

Nº do Protocolo: 23073.066293/2023-34

Belém-PA, 13 de setembro de 2023.

ATA DA APRESENTAÇÃO E DEFESA DE DISSERTAÇÃO DE MESTRADO

ATA DA 757ª SESSÃO DE APRESENTAÇÃO E DEFESA DE DISSERTAÇÃO DE MESTRADO PARA CONCESSÃO DE GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE **SISTEMAS DE ENERGIA ELÉTRICA**, REALIZADA ÀS NOVE HORAS DO DIA PRIMEIRO DE SETEBRO DE DOIS MIL E VINTE E TRÊS, VIA PLATAFORMA GOOGLE MEET, INTITULADA: **ANÁLISE DE TÉCNICAS DE CONTROLE CLÁSSICAS E AVANÇADAS SINTONIZADAS COM APRENDIZADO POR REFORÇO** APRESENTADA DURANTE **58 MINUTOS** PELO CANDIDATO **DANIEL ABREU MACEDO DA SILVA** DIANTE DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DO INSTITUTO DE TECNOLOGIA DA UNIVERSIDADE FEDERAL DO PARÁ, ASSIM CONSTITUÍDA: **PROF. DR. ANTÔNIO DA SILVA SILVEIRA** (ORIENTADOR - PPGEE/UFPA), **PROF. DR. WALTER BARRA JÚNIOR** (AVALIADOR INTERNO - PPGEE/UFPA), **PROF.ª DR.ª REJANE DE BARROS ARAÚJO** (AVALIADORA EXTERNA - IFPA) E **PROF. DR. TITO LUIS MAIA SANTOS** (AVALIADOR EXTERNO - UFBA). CONCLUÍDOS OS TRABALHOS DE APRESENTAÇÃO E ARGUIÇÃO, A BANCA EXAMINADORA DECIDIU PELA **APROVAÇÃO** DO CANDIDATO. FOI CONCEDIDO UM PRAZO DE TRINTA DIAS, PARA O CANDIDATO EFETUAR AS CORREÇÕES SUGERIDAS PELA COMISSÃO EXAMINADORA E APRESENTAR O TRABALHO EM SUA REDAÇÃO DEFINITIVA. ESTA ATA NÃO VALE COMO OUTORGA DE GRAU DE MESTRADO, DE ACORDO COM O DEFINIDO NA RESOLUÇÃO 072/2004-CONSEPE. E, PARA CONSTAR, FOI LAVRADA A PRESENTE ATA, QUE VAI ASSINADA PELOS MEMBROS DA COMISSÃO E PELO CANDIDATO.

(Assinado digitalmente em 13/09/2023 10:45)
ANTONIO DA SILVA SILVEIRA
PROFESSOR DO MAGISTERIO SUPERIOR
ITEC (11.41)
Matrícula: ###301#3

(Assinado digitalmente em 21/09/2023 15:20)
WALTER BARRA JUNIOR
PROFESSOR DO MAGISTERIO SUPERIOR
ITEC (11.41)
Matrícula: ###475#6

(Assinado digitalmente em 13/09/2023 10:49)
DANIEL ABREU MACEDO DA SILVA
DISCENTE
Matrícula: 2021########8

gov.br Documento assinado digitalmente
TITO LUIS MAIA SANTOS
Data: 29/09/2023 13:22:00-0300
Verifique em https://validar.iti.gov.br

Rejane de Barros Araújo
Assinado de forma digital por Rejane de Barros Araújo
Dados: 2023.09.29 13:32:02 -03'00'

Para verificar a autenticidade deste documento entre em https://sipac.ufpa.br/public/documentos/index.jsp informando seu número: **29**, ano: **2023**, tipo: **ATA DE DEFESA DE DISSERTAÇÃO**, data de emissão: **13/09/2023** e o código de verificação: **8e5923cb30**

I dedicate this master's thesis to my family.

ACKNOWLEDGEMENTS

I gratefully dedicate this master's thesis to the extraordinary individuals who have profoundly influenced my academic journey, enriched my knowledge, and provided unwavering support. To my beloved grandmother Raimunda, mother Creuza, brother Luiz, beloved Gabriela, aunt Benedita, aunt Ivonete, cousin Diogo, cousin Edmilson, my incredible professors André Nascimento, Rejane Araújo, and Antônio Silveira, and above all, to God, this thesis is a testament to the collective impact you have had on my growth and achievements.

To my dear grandmother Raimunda, your wisdom, strength, and unwavering love have been a constant source of inspiration. Your faith in my abilities has empowered me to pursue my academic goals with determination, and for that, I am deeply grateful. I offer my heartfelt appreciation for your enduring support and guidance. I know you're proud of me in heaven

Mother Creuza, your endless dedication and sacrifices have been instrumental in shaping my academic path. Your selflessness and unwavering belief in my potential have fueled my ambition to strive for excellence. This thesis is a heartfelt tribute to your immeasurable love and the imprints you have left on my life.

Brother Luiz, your unwavering support and encouragement have been a steady presence throughout my journey. Your belief in my abilities and the countless discussions we have shared have broadened my perspective and enriched my academic pursuits. I extend my deepest gratitude for your unwavering presence in my life.

My beloved Gabriela, your love, understanding, and unwavering faith in me have been my anchor. Your unwavering support and encouragement have provided me with the strength to overcome challenges and reach for the stars. This thesis is dedicated to you, my love, as an acknowledgment of the profound impact you have had on my academic and personal growth.

Aunt Benedita and Aunt Ivonete, your love, guidance, and unwavering support have played a significant role in my journey. Your encouragement and belief in my potential have been a source of motivation during times of doubt. I express my deepest gratitude for your unwavering presence and continuous support.

Cousin Diogo and Cousin Edmilson, our shared experiences and cherished moments have created lifelong memories. Your friendship, encouragement, and intellectual discussions have expanded my horizons and nurtured my passion for knowledge. This thesis is a testament to our enduring bond and the growth we have experienced together.

To my incredible professors André, Rejane, and Silveira, your exceptional guidance, expertise, and dedication have been invaluable in shaping my academic development. Your passion for teaching, mentorship, and the knowledge you have imparted

have left an indelible mark on my journey. I am deeply grateful for your unwavering support and the opportunity to learn from you.

Lastly, but most importantly, I offer my profound gratitude to God. It is through His grace, guidance, and blessings that I have been able to navigate the challenges of academia and reach this milestone. I acknowledge that this achievement would not have been possible without His divine presence and unwavering support.

To each and every one of you, I express my deepest appreciation for your unwavering belief in my abilities, constant encouragement, and continuous support. This thesis symbolizes not only my academic accomplishment but also serves as a tribute to the collective impact you have had on my life. I am forever grateful for your presence and the immeasurable contributions you have made to my journey.

“It’s not about whether it’s possible or not.
It’s about whether you can believe in yourself or not."
(Monkey D. Luffy, One Piece, 2004)

ABSTRACT

Control theory is used to stabilize systems and obtain specific responses for each type of process. Classic controllers, such as the PID used in this research, are spread globally in industries because they have well-studied topologies in the literature and are easily applied in microcontrollers or programmable lógic devices; advanced ones, such as GMV, GPC and LQR, also used in this work, have some resistance in common applications in base industries, but are widely used in energy, aerospace and robotic systems, since the complexity and structure of these methods generate robustness and reach satisfactory performances for processes that are difficult to control. In this work, these methods are studied and evaluated with a tuning approach that uses reinforcement learning. The tuning methods are used in two forms and are applied to the controllers, these are the Repeat and Improve method and the Differential Games method. The first works using offline iterations, where the process agent is the chosen control technique, which selects performance and robustness indexes as an environment (metric of how the process is evolving), being able to organize an adjustment policy for the controller, which is based on rewarding the weighting factor until reaching the process stopping criterion (desired response). The second method uses reinforcement strategies that reward the controller as the response changes, so the LQR learns the ideal control policies, adapting to changes in the environment, which allows for better performance by recalculating the traditional gains found. With the Ricatti equation for tuning the regulator; in this method, differential games are used as a framework to model and analyze dynamic systems with multiple agents. To validate what is presented, the Tachogenerator Motor and the Ar Drone have been chosen. The Tachogenerator Motor is modeled with least squares estimation in an ARX-SISO topology, in order to evaluate the first tuning method. The Ar Drone is modeled with a state space approach to evaluate the second tuning method.
Keywords: Control Theory. Reinforcement Learning. Repeat and Improve. Differential Games. Ricatti Equation.

RESUMO

A teoria de controle é utilizada para estabilizar sistemas e obter respostas específicas para cada tipo de processo. Controladores clássicos, como o PID utilizado nesta pesquisa, são difundidos globalmente nas indústrias, isto por possuírem topologias bem estudadas pela literatura e serem facilmente aplicados em microcontroladores ou controladores lógico programáveis; já os avançados, como GMV, GPC e LQR também utilizados neste trabalho, possuem certa resistência em aplicações comuns das indústrias de base, mas são muito utilizados em sistemas de energia, aerospaciais e robóticos, pois a complexidade e estrutura desses métodos gera robustez e alcança desempenhos satisfatórios para processos de difícil controle. Neste trabalho, esses métodos são estudados e avaliados com uma abordagem de sintonia que utiliza o aprendizado por reforço. São aplicadas duas formas de sintonia para os controladores, estas são o método da Repetição e Melhora e o método de Jogos Diferenciais. O primeiro utiliza iterações offline, onde o agente do processo é a técnica de controle escolhida, que trabalha com os índices de desempenho e robustez como ambiente (métrica de como o processo está evoluindo), sendo capaz de organizar uma política de ajuste para o controlador, que se baseia em recompensar o fator de ponderação até obter o critério de parada do processo (resposta desejada). O segundo método se baseia em utilizar estratégias de reforço que recompensam o controlador conforme a resposta se modifica, assim o LQR aprende as políticas de controle ideais, adaptando-se às mudanças do ambiente, o que permite obter melhor desempenho por recalcular os tradicionais ganhos encontrados com a equação de Ricatti para sintonia do regulador; neste método, os jogos diferenciais são utilizados como uma estrutura para modelar e analisar sistemas dinâmicos com múltiplos agentes. Para validar o que é apresentado, o Motor Tacogerador e o Ar Drone são escolhidos. O Motor Tacogerador é modelado com a estimação dos mínimos quadrados em uma estrutura ARX-SISO para avaliação do primeiro método de sintonia. O Ar Drone é modelado com uma abordagem em espaço de estados para avaliação do segundo método de sintonia.

Palavras-chave: Teoria de Controle. Aprendizado por Reforço. Repetição e Melhora. Jogos Diferenciais. Equação de Ricatti.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

UFPA	Universidade Federal do Pará
LACOS	Laboratory of Control and Systems
PID	Proportional Integral Derivative
PPID	Pseudo Proportional Integral Derivative
GMV	Generalized Minimum Variance
GPC	Generalized Predictive Control
GMVSS	Generalized Minimum Variance in States Space
LQR	Linear Quadratic Regulator
RST	Reference Sequence Tracking
TGM	Tacho Generator Motor
TGM	Motor Taco Gerador
LS	Least Squares
MQ	Mínimos Quadrados
SISO	Single Input, Single Output
MIMO	Multiple Input, Multiple Output
ARX	AutoRegressive with eXogenous input
AI	Artificial Inteligence
RL	Reinforcement Learning
MDP	Markov Decision Processes
PWM	Pulse Width Modulation
UAVs	Unmanned Aerial Vehicle
NRLS	Non Recursive Leasts Squares
SSLS	Space State Leasts Squares
OKID	Observer/Kalman filter Identification
ERA	Eigensystem Realization Algorithm

KF	Kalman Filter
ARMAX	AutoRegressive Moving Average with eXogeneus variable
MBC	Model Based Controller
MBPC	Model Based Predictive Controller
LTI	Linear Time-Invariant
ZOH	Zero Order Hold
DCAR	Deterministic Controlled Auto-Regressive
CARIMA	Controlled Auto-Regressive Integrated Moving Average
ISE	Integral Square Error
ISU	Integral Squared Control Signal
GM	Gain Margin
PM	Phase Margin
CLTF	Closed Loop Transfer Function
ZS	Zero-Sum
GARE	Game Algebraic Riccati Equation

CONTENTS

1 INTRODUCTION

1.1 OVERVIEW

The study of control theory is very important for obtaining automated systems in people's daily lives, in industry and commerce. Systems arising from these studies are implemented every day in various processes, such as manufacturing and production of beverages, vehicles, medicine, energy and even aerospace systems (STEVENS; LEWIS; JOHNSON, 2015). These processes are based on using the basic principle of measuring and acting, that is, the sensors are responsible for analyzing and verifying the variables to be controlled; the actuators are responsible for manipulating other variables in order to reach a reference with the controlled process variables; the controller analyzes the sensor data and calculates a correction factor to be applied to the process by the actuators. The way this controller works depends on the control technique to be applied. In the literature, more than a thousand forms and structures of controllers are presented, so, depending on the complexity and equipment available in the process, the designer can develop an appropriate control algorithm for a system (OGATA et al., 2010).

Controllers vary in their algebraic development structures; therefore, the way a controller deals with the error (difference between the measured signal and the reference signal) is different. Classic controllers tend to work with immediate responses and, normally, use few path information in their control law. On the other hand, advanced controllers, such as predictive ones, tend to use the stochasticity of the process, that is, past samples as a way of analyzing how the future will behave, thus, together with their parameters, predict possible disturbances and dynamics that occur in the systems.

A concern present in control systems is tuning. Classical controllers have tunings strongly documented in the literature, however, with the advancement of Artificial Intelligence (AI), new ways to parameterize such systems are discovered every year. This same principle is applied to advanced controllers, the discovery of new AI techniques always makes it possible to innovate in this bias; the implementation of neural networks, genetic algorithms, and reinforcement learning has received good reviews, since such methods improve the use of databases and allow performing several previous calculations that manually would be impossible or extremely exhausting (SUTTON; BARTO, 2018).

1.2 JUSTIFICATION

This master's thesis introduces an innovative approach to improve the tuning of various controllers in process control systems using advanced reinforcement learning techniques. The study combines the repeat and improve RL method with the differen-

tial games Q-learning approach to optimize the performance of Proportional-Integral-Derivative (PID), Generalized Minimum Variance (GMV), Generalized Predictive Control (GPC) and Linear Quadratic Regulator (LQR) controllers. The primary objective is to develop an autonomous and efficient tuning methodology that enhances process control and system performance, while reducing manual intervention and fine-tuning efforts. The potential advantages of employing reinforcement learning include adaptive control strategies, reduced tuning time, and improved stability and robustness of the overall control system.

The repeat and improve reinforcement learning method is applied to the PID, GMV and GPC controllers, enabling them to learn from their past actions and iteratively improve their control strategies. By adapting to changing process conditions and disturbances, these controllers can achieve superior performance compared to traditional manual tuning methods. Additionally, the differential games Q-learning approach is used to fine-tune the LQR controller. This method allows the LQR controller to interact with the process control environment and strategically adapt its parameters to achieve more efficient and collaborative control, particularly in multi-agent systems.

This research is expected to contribute significantly to the field of process control by demonstrating the efficacy of advanced reinforcement learning techniques for controller tuning. The proposed approach is an idea for process control practices that aims to help in control theory by providing an autonomous and efficient solution that optimizes the performance of diverse controllers. The study's outcomes may lead to improved control system stability, reduced control efforts, and enhanced disturbance rejection capabilities. Ultimately, the successful implementation of these advanced RL techniques could pave the way for more intelligent and autonomous control systems in various industrial applications, promoting efficiency and reliability in process control operations.

1.3 OBJECTIVES

The general objective of this master's thesis, based on what has already been exposed, is to analyze two forms of Reinforcement Learning tuning to parameterize classic and advanced controllers applied in processes with complex dynamics. In order to specify the stages of the work, the following specific objectives can be listed:

- Analyze controller structures and topologies.
- Introduce reinforcement learning as a tuning method.
- Use such tuning in controllers, these being SISO and MIMO.
- Expose performance and robustness indexes as a way of evaluating controllers and stopping criteria for tuning.
- Evaluate such methods applied in the processes.

1.4 STATE-OF-THE-ART

Control theory is a branch of engineering and mathematics that deals with the analysis and design of systems that can be controlled or regulated to behave in a desired manner. It involves the use of mathematical models to describe the behavior of systems and the design of controllers to regulate that behavior. This description was proposed by Ziegler-Nichols (1942) in one of the first papers proposed about feedback control. Such paper opened horizons for modern control that is applied in the 21st century, it laid out the foundational principles of feedback control theory, including the concepts of stability, transient response, and frequency response. Other papers as Flueggelotz (1961) try to introduce the concept of optimal control, which involves finding the control inputs that minimize a certain cost function. These papers and authors have had a significant impact on the development of control theory and have contributed to its widespread use in engineering and other fields.

Some literatures, such as Stevens, Lewis and Johnson (2015) and McRuer, Graham and Ashkenas (2014), proposes to study how to acquire desired outputs with aerospace systems, such as drones and aircraft; those that have multivariable dynamics and fast poles, which generate instabilities and difficulties for classical control techniques. Control theory is crucial for the design, analysis, and operation of aerospace systems because these systems often exhibit complex dynamic behavior and require a high degree of accuracy and reliability in their performance. Aerospace systems such as aircraft, spacecraft, and satellites must operate in harsh and unpredictable environments and respond quickly and accurately to changes in external conditions (HE; PACE, 2020). Control theory provides a set of tools and techniques for modeling the behavior of these systems, designing control algorithms to regulate their behavior, and analyzing their performance under different operating conditions, also being used to design control systems that ensure the stability and safety of aerospace systems. For example, flight control systems in aircraft are designed to maintain stable flight by adjusting the aircraft's control surfaces in response to external disturbances such as wind and turbulence.

Aerospace systems are controlled, most of the times, to achieve precision and accuracy, because control theory provides techniques for designing control systems that achieve precise control of the system's behavior, such as the attitude and orbit control systems used in spacecraft. Control techniques also provides fault tolerance and those systems must be able to operate even in the presence of faults or failures in their components,ensuring that the system continues to operate safely and reliably. Overall, control theory plays a critical role in the design, operation, and safety of aerospace systems, and it is essential for ensuring that these systems perform their mission objectives safely, accurately, and reliably.

In economics, control theory is used to model and analyze market behavior and

optimize economic systems, as proposed by Prescott (1977). It is also important for the economy because it allows decision-makers to make informed choices about future outcomes, and thus take actions to mitigate potential risks and maximize opportunities. In economic systems, modern control is used to forecast trends and patterns in market behavior, financial performance, and other economic indicators, and to guide decisions about investment, production, pricing, and resource allocation (ATHANS, 1974). For example, predictive control can be used to anticipate changes in consumer demand and adjust production levels accordingly, or to forecast market conditions and optimize investment strategies. By using predictive control, companies and policymakers can make more informed decisions that lead to greater efficiency, profitability, and growth. Overall, predictive control is important for the economy as it helps to reduce uncertainty, increase efficiency, and support decision-making in complex and dynamic economic environments.

In robotics, there are various variables that need to be controlled depending on the application and the type of robot. These variables are typically controlled using feedback control techniques such as PID control, model predictive control, and fuzzy logic control. Those variables, as said in Friedman (1959), Song, Yu and Zhang (2019) and Xiao *et al.* (2020), are position of a robot's end-effector or joints as one of the most basic variables that needs to be controlled. This involves specifying the desired position of the robot and ensuring that it moves to that position accurately. The velocity of a robot's end-effector or joints is another variable that needs to be controlled, which involves specifying the desired velocity of the robot and ensuring that it moves at that velocity accurately. Also, it is necessary to control the acceleration of the robot to ensure smooth and precise movement. In applications where the robot interacts with the environment, it may be necessary to control the force or torque exerted by the robot to ensure safety and accuracy. In applications where the robot is required to grip and manipulate objects, it may be necessary to control the gripping force to ensure that the objects are not damaged or dropped, those recently have been used also for biomedical applications, as proposed by Dutra, Silveira and Pereira (2021).

In general, control theory provides a framework for understanding and improving the behavior of complex systems. By designing and implementing control systems, it is possible to improve the efficiency, safety, and reliability of many different types of systems. Kaelbling, Littman and Moore (1996) and Sutton *et al.* (1999) innovated in the technical area of artificial intelligence and allowed new discoveries with the use of computing. Such works present sketches of what is possible to be analyzed nowadays. Kaelbling, Littman and Moore (1996) proposes a research that focuses on decision making, planning, and reinforcement learning in autonomous agents and robots. they have made significant contributions to the development of Markov decision processes (MDPs) and their applications in artificial intelligence. In addition, Kaelbling

has worked on natural language processing and human-robot interaction. This work is highly referenced when reviewing other literature on reinforcement learning, whether for problem solving or for tuning controllers.

Sutton and Barto (2018) has a research parallel to that of Kaelbling, this research is primarily focused on machine learning, particularly in the areas of reinforcement learning and artificial intelligence. Reinforcement learning is a type of machine learning in which an agent learns to make decisions by trial and error, receiving feedback in the form of rewards or punishments. Sutton's work has helped to develop the theory and algorithms that underlie reinforcement learning, and he has made significant contributions to the development of value function approximation, temporal difference learning, and policy gradient methods.

Sutton is also known for his contributions to the development of the popular reinforcement learning algorithm known as Q-learning. His book, co-authored with Andrew G. Barto, "Reinforcement Learning: An Introduction," is considered a seminal work in the field and is widely used as a textbook in courses on reinforcement learning. Sutton has received numerous awards and honors for his contributions to the field, including the AAAI Classic Paper Award, the IJCAI Computers and Thought Award, and the Royal Society of Canada's Rutherford Memorial Medal in Physics. This research has a lot of influence to this thesis, since this methods are used to tune the controllers presented.

Vrabie and Lewis (2013) also has used the propositions of Kaelbling, Littman and Moore (1996) and Sutton and Barto *et al.* (1999) to improve the Q learning method using the differential games with reinforcement method. It is also recognized as a big contribution to this area, since this book synthesizes and applies this theory in real models, with some criticisms, such as convergence time and computational effort; these are the biggest problems faced in the application of these algorithms.

The junction point between reinforcement learning and control theory is a quantitative analysis, that is, some value must tell intelligence that the control system is good. For this work, what was proposed in Postlewaite (1996), which provides a comprehensive introduction to the theory and practice of multivariable feedback control, which is the control of systems with multiple inputs and outputs. This reference has been really important for performance and robustness analysis, and is one of the biggest references in this area, being important for recent works, such as Araújo *et al.* (2017), Silva *et al.* (2021) and Yamaguti *et al.* (2022).

All these ideas are used to achieve the goals of this work, with some changes in terms, since computational intelligence and classical control theory have many similarities that are treated as different just because they have different names. This is treated with more details in a chapter of AI.

1.5 MASTER'S THESIS ORGANIZATION

This master's thesis is divided into six chapters, with the purpose of introducing the theme, discussing some of the main works found in the literature during the research and, finally, defining the objectives of the work. The first chapter has been presented as the introduction of the work, with an overview, the objectives and a state of art presentation.

Studying control is also a way of evaluating the application possibility, so it is presented in Chapter 2 which processes are used, in experimental ways or in simulated systems. In addition to proposing how these systems are modeled using identification tools.

Chapter 3 presents which controllers were tested for tuning with reinforcement learning methods. In this are shown the topologies, diagrams, and equations that represent such forms of control. As well as the challenges, advantages, and disadvantages of each one, based on other proposed works. In addition to presenting the performance and robustness indices, which are used in such techniques as a way to merge control with artificial intelligence

In Chapter 4, the computational intelligence tools, based on reinforcement learning, that were used are presented, exposing the main theorems that guided the analysis and the developed project.

The results of simulations and experimental tests performed are presented in Chapter 5. Tests were performed on different systems that are proposed in Chapter 2 and aim to assess whether such methods are viable or not.

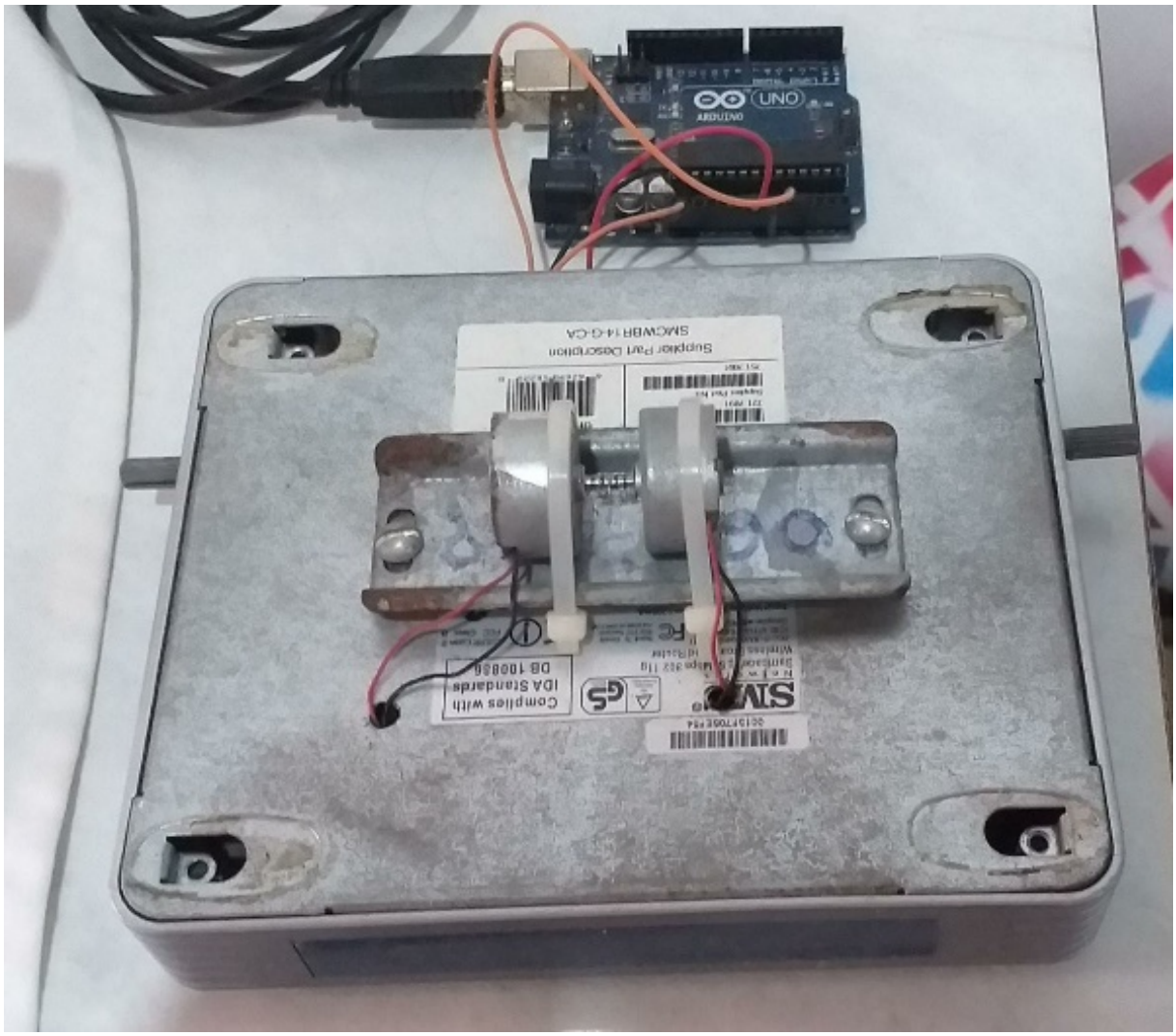
Finally, in Chapter 6, the final considerations on the study carried out are presented, in addition to the focus on suggestions for future work based on the results and on other questions raised during the reading of several works.

2 PROCESSES AND IDENTIFICATION

2.1 TACHO GENERATOR MOTOR

The Tachogenerator Motor (TGM) is a system used to convert the speed generated in a motor into voltage, from a coupling connected to a generator that produces electrical energy with the conversion of mechanical energy (SILVA; SILVEIRA; NASCIMENTO, 2022). In addition, this type of model is applied in industrial processes, such as centrifugal pumps, conveyors, and liquid flow meters, among others (BOLTON, 2021). Thus, the TGM process, Figure 1, presented in this work was designed for the application of process identification and control methods.

Figure 1 – TGM didactic system.

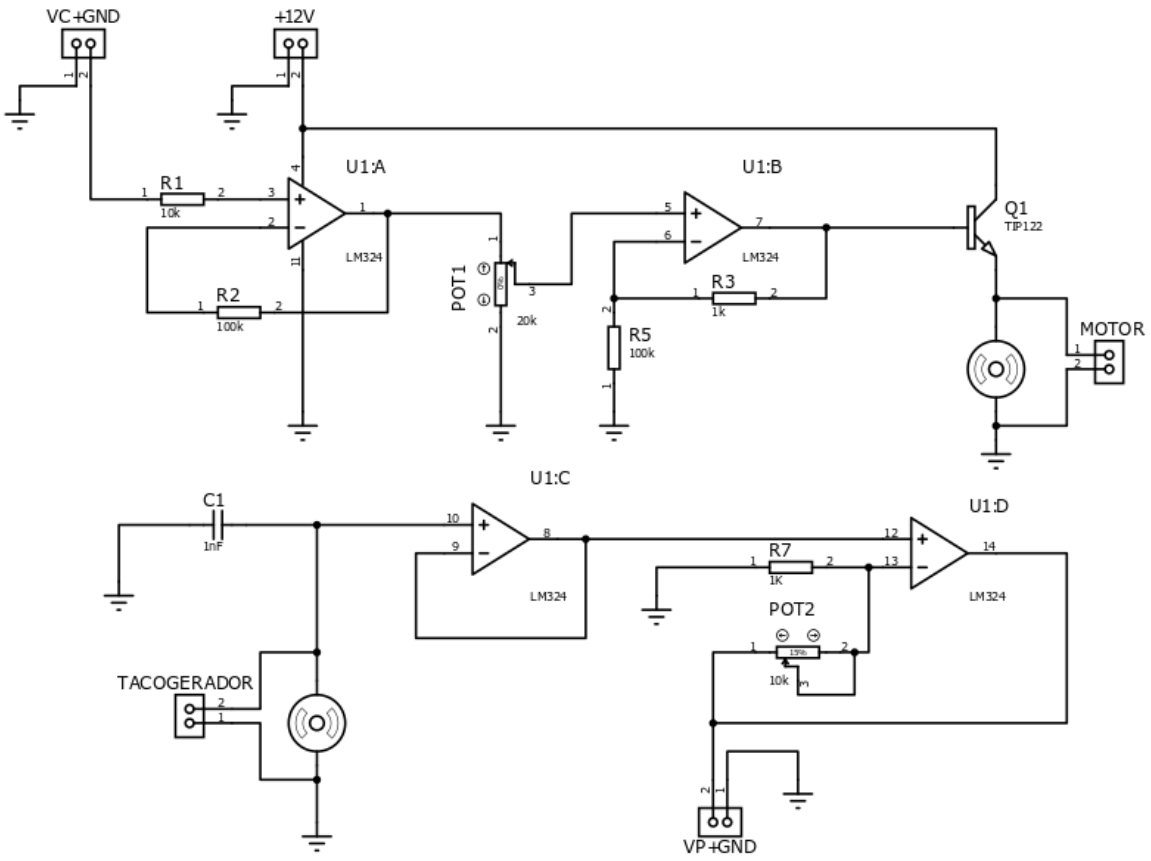


Source: Silva, Silveira and Nascimento (2022).

The designed TGM circuits allow the conversion of input signals from 0 V to 5 V (digital PWM signal) sent from an Arduino to the first motor, whose purpose is

to produce an analog signal by converting the speed in the first motor coupled to the second. Furthermore, the TGM can be regulated by means of potentiometers, to modify the static gain of the operational amplifiers, thus obtaining similar input and output values to apply the plant identification methods, based on the schematic circuit, presented in Figure 2.

Figure 2 – TGM schematic circuit.



Source: Silva, Silveira and Nascimento (2022).

Additionally, the TGM model's design employs two motors with identical features and specifications to operate the system at various operating points without substantially altering the conversion of mechanical and electrical energy. Thus, with the aim of allowing the change of motor model, potentiometers and the LM324 integrated circuit are used to regulate the set-point, by allowing the experiment with different specification motors and causing considerable changes in the response obtained at the system output.

2.2 AR DRONE

The AR Drone, also known as the Parrot AR Drone, is a line of unmanned aerial vehicles (UAVs) developed and produced by the French company Parrot and used in LACOS (Laboratory of Control and Systems) for practical tests in aerospace control, it is shown in Figure 3. It is equipped with multiple sensors, including an accelerometer, a gyroscope, and a magnetometer, which enable it to stabilize itself in mid-air and perform smooth and controlled flights.

Figure 3 – Quadrotor flying at a parking lot of the Federal University of Pará near the Guamá River.

Source: Silveira *et al.* (2020).

Beyond recreational applications, the AR Drone has also found applications in a variety of professional domains. Its mobility and versatility make it ideal for aerial photography and filming, as well as research and education. It is also utilized for custom apps and software that extend the capabilities of the drone, such as autonomous flight routes and computer vision-based tracking.

Over the years, the AR Drone has inspired a growing community of enthusiasts, hobbyists, and developers who have contributed to its continued development. Its open-source software and development kits have allowed users to tinker with its features, experiment with new applications, and customize its functionalities according to their

specific needs. For this purpose, it is used for identification and control theory tests, since its multivariable structure with complex dynamics is the perfect field for tests.

The lateral speed, longitudinal speed, and altitude are important parameters when it comes to operating and controlling the AR Drone. These measurements provide crucial information about the drone's flight characteristics and enable users to maintain control and navigate the drone effectively.

Lateral speed dictates the rate of sideward movement or horizontal displacement of the drone. Skillful management of lateral speed is crucial for maintaining precise positioning, executing dynamic maneuvers, and avoiding collisions during flight (STEVENS; LEWIS; JOHNSON, 2015). By adjusting lateral speed, pilots can navigate around obstacles, achieve smooth transitions between flight directions, and enhance overall flight control. Monitoring and adjusting lateral speed in real-time allow for safe and controlled navigation in various environments, ensuring both stable flight and successful mission execution.

The longitudinal speed refers to the drone's speed in the forward or backward direction, parallel to its longitudinal axis (STEVENS; LEWIS, Frank L; JOHNSON, 2015). It determines how quickly the drone moves along its flight path. Monitoring and controlling the longitudinal speed is crucial for various flight scenarios. For example, during aerial photography or videography, controlling the drone's speed ensures smooth and cinematic footage. In racing or agility-based activities, adjusting the longitudinal speed allows users to navigate obstacles and complete courses efficiently.

The final state of evaluation is the altitude, which refers to the height or distance above the ground at which the drone is flying. It is a critical parameter for safe and legal drone operation. Monitoring the altitude helps users comply with airspace regulations, avoid collisions with obstacles like buildings or trees, and maintain a clear line of sight with the drone. Altitude control is also crucial for capturing specific perspectives in aerial photography or maintaining a consistent height during surveying or mapping tasks.

These measurements assist users in making informed decisions, adjusting flight parameters, and maintaining safe and stable flight conditions. Whether for recreational or professional purposes, understanding and controlling these parameters contribute to the successful and enjoyable operation of the AR Drone. Controlling the lateral speed, longitudinal speed, and altitude of drones, including the AR Drone, involves the implementation of various control systems and techniques. The literature suggests several approaches to achieving precise and stable control over these parameters and that is important because the choice of control strategy depends on factors such as the drone's dynamics, available sensor information, computational resources, and the specific requirements of the application.

2.3 NON RECURSIVE LEASTS SQUARES ESTIMATION (NRLS)

2.3.1 Polynomial Approach

Using NRLS mode for the identification of a Single Input Single Output (SISO) model offers several advantages. This method is straightforward to implement and computationally efficient. It involves solving a set of linear equations, making it suitable for real-time or online system identification tasks where fast processing is essential (COELHO; COELHO, 2004). NRLS also allows batch processing of data, meaning that it can handle datasets collected over a period of time all at once. This approach is well-suited for offline identification scenarios where all the data is available in advance, and the identification can be done in a single step, as is used in this master's thesis and shown in the Results chapter. Another advantage is the robustness to noise in the data. By considering all the data points simultaneously, the impact of individual noisy measurements tends to average out, leading to more accurate parameter estimates, which is really important for the presented systems, since TGM has no capacitor filters and AR Drone is a complex model with many noisy sensors.

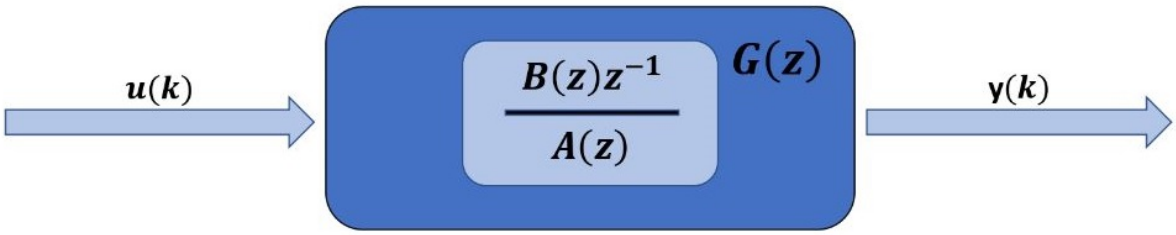
For SISO models, this method approach provides closed-form solutions for parameter estimation. This simplifies the identification process as there is no need for complex iterative algorithms, unlike recursive methods. Also allows for straightforward statistical analysis of the estimated model parameters. One can calculate confidence intervals, perform hypothesis testing, and assess the quality of the model fit, providing insights into the reliability of the identified model (AGUIRRE, 2004). It may suffer from sensitivity to the initial parameter estimates, leading to convergence issues or suboptimal results. In contrast, NRLS is less sensitive to initial guesses, which can save time and effort in obtaining satisfactory results. The final reason for use is the versatility since it is a method that can be easily extended to accommodate more complex model structures or system identification tasks, such as multiple input systems or time-varying models (YAMAGUTI; DUTRA; SILVEIRA, 2021).

For this master's thesis, the NRLS strategy is used in TGM and AR Drone, to identify a model to be used in model-based controllers. Figure 4 shows the SISO system block diagram, considering the Auto Regressive with inputs exogenous (ARX) model, as the process has one input ($u(k)$): the first motor voltage signal) and one output ($y(k)$: the generated voltage in the second motor), it is possible to write the output signal as (1). The same structure is used for AR Drone, since for lateral speed, longitudinal speed and altitude, the inputs are lateral, longitudinal and vertical thrusts, respectively.

$$y(k) = \frac{B(z)z^{-1}}{A(z)}u(k) \tag{1}$$

where the roots of $B(z)$ and $A(z)$ are, respectively, the z domain zeros and poles polynomials of the system.

Figure 4 – SISO system block diagram within ARX model.



Source: Author(2023).

Due to the systems under-damped dynamic, and considering that the estimated discrete model is second order, (1) can be represented as a difference equation, as in (2).

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u(k-1) + b_1 u(k-2) \tag{2}$$

Thus, as presented in Yamaguti, Dutra and Silveira (2021), using (2), the vector containing the read data (measurements vector – $\boldsymbol{y}$), presented in (3), the matrix encompassing inputs and output data of the system (matrix of regressors – Φ), presented in (4), and the vector of estimated parameters (θ), presented in (5), may be determined.

$$\boldsymbol{y^T} = \begin{bmatrix} y(1) & y(2) & \dots & y(N) \end{bmatrix} \tag{3}$$

$$\Phi = \begin{bmatrix} -y(1) & 0 & u(1) \\ -y(2) & -y(1) & u(2) \\ \vdots & \vdots & \vdots \\ -y(N-1) & -y(N-2) & u(N-1) \end{bmatrix} \tag{4}$$

$$\theta^T = \begin{bmatrix} a_1 & a_2 & b_0 & b_1 \end{bmatrix} \tag{5}$$

After defining (3), (4) and (5), the following algebraic equation appears:

$$\boldsymbol{y} = \Phi\theta \tag{6}$$

According to Coelho and Coelho (2004), to calculate θ using (6), it will be necessary that Φ is a square matrix, however Φ is a matrix of order $\Phi_{N\times 6}$. Thus, it is necessary to apply the pseudo-inverse matrix. As a result, the solution of non-recursive least squares estimator was determined by computing θ as shown in (7).

$$\theta = [\Phi^T \Phi]^{-1} \Phi^T \boldsymbol{y} \tag{7}$$

According to Coelho and Coelho (2004), NRLS is designed taking into account the knowledge of the dynamics of the process and the value of the squared Pearson correlation coefficient (R^2):

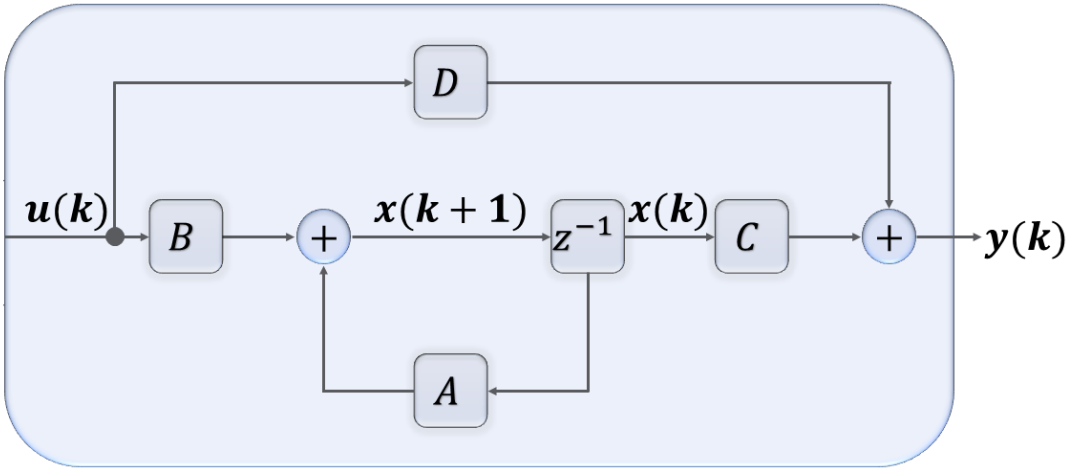
$$R^2 = 1 - \frac{\sum_{k=1}^{N} \left[y(k) - \hat{y}(k)\right]^2}{\sum_{k=1}^{N} \left[y(k) - \bar{y}\right]^2} \tag{8}$$

where $\hat{y}(k)$, $\bar{y}$ and N correspond to estimated output, average output, and number of samples, respectively. According to Coelho and Coelho (2004), for many practical applications, values of R^2 between 0.8 and 1.0 can be considered sufficient, but lower values than this range can model the system coherently as well and may be valid for testing the designed control techniques

2.3.2 State-space approach

The NRLS in a polynomial formulation was shown in the previous subsection to help create controllers that operate decentralized. The NRLS estimator will be presented in this subsection utilizing a state space approach (SSLS) to give a model where the controller interacts directly and centrally with the state variables. For this, it is considered the model represented in state space in Figure 5.

Figure 5 – Block diagram of system representation in state space



Source: Yamaguti, Dutra and Silveira (2021) adapted.

Considering a MIMO model, the state equation (9), and the output equation (10) can be represented as follows:

$$\begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} = \begin{bmatrix} \hat{a}_{11} & \cdots & \hat{a}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{a}_{n1} & \cdots & \hat{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ \vdots \\ x_n(k-1) \end{bmatrix} + \begin{bmatrix} \hat{b}_{11} & \cdots & \hat{b}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{b}_{n1} & \cdots & \hat{b}_{nn} \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ \vdots \\ u_n(k-1) \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{bmatrix} = \begin{bmatrix} I_{1\times n} & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} \tag{10}$$

Where $x_1(k)$ till $x_n(k)$ correspond to the estimated states, respectively. Thus, the SSLS solution lies in the determination of the estimated parameters vectors which is defined as

$$\begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_n^T \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} & \dots & b_{11} & \dots & b_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} & \dots & b_{n1} & \dots & b_{n2} \end{bmatrix} \tag{11}$$

For this, according to Silveira *et al.* (2020), it is necessary to use the calculation of future observations based on the vector of regressors, in order to achieve a better estimation of the system, where, according to Nogueira *et al.* (2019), the estimation of the state x_b can be performed by means of an approximation of type *backward* of the derivative of x_a. Thus, the state x_b estimates the output speed convergence x_a as follows

$$x_b(k) = \frac{x_a(k) - x_a(k-1)}{T_s} \tag{12}$$

Thus, the matrix of regressors, for the SSLS case, is organized as in (13), as an example for a one-state system with backward approximation as a second state.

$$\varphi = \begin{bmatrix} x_1(0) & x_2(0) & u_1(0) & u_2(0) \\ x_1(1) & x_2(1) & u_1(1) & u_2(1) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(N-1) & x_2(N-1) & u_1(N-1) & u_2(N-1) \end{bmatrix} \tag{13}$$

Thus, the calculation of the estimated parameters, for the SSLS case, can be performed using (14)

$$\theta = (\varphi^T \varphi)^{-1} \varphi^T \boldsymbol{y} \tag{14}$$

2.3.3 OKID

The Observer/Kalman filter Identification (OKID) objective is to enable the estimation of the system matrices A (States Matrix), B (Input Matrix), and Γ (Noise Identification Matrix), as well as the noise covariance matrices Q and R, using the state data estimation estimated by a Kalman filter state estimator (VICARIO, 2014).

The Eigensystem Realization Algorithm (ERA), which traditionally follows OKID methods, incorporates the OKID/ERA, which identifies the process model and the

optimal state observer (VICARIO, 2014). The matrices for the detected system are delivered by the ERA, however, they are organized as the Markov parameters of the system in (15).

$$H_0 = \begin{bmatrix} CB & CAB & \cdots & CA^{\frac{N}{2}-1}B \\ CAB & CA^2B & \cdots & CA^{\frac{N}{2}}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^{\frac{N}{2}-1}B & CA^{\frac{N}{2}}B & \cdots & CA^{N-2}B \end{bmatrix} \tag{15}$$

where N is the number of iterations.

Using the classical model in the form of a discrete-time stochastic dynamical state space system, the system can be represented by

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-d) + \Gamma w(k-1) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{16}$$

considering $x(k) \in \mathbb{R}^n$ as a vector of n states variables and $u(k) \in \mathbb{R}^{n_u}$ is a vector of n_u outputs $w(k) \in \mathbb{R}^n$ is a vector of n states disturbance inputs and $v(k) \in \mathbb{R}^{n_y}$ is a vector of n_y output disturbance inputs.

By adhering to the points made by Vicario (2014), the OKID provided in this work aims to achieve a direct technique to retrieve the system's matrices while keeping the fundamental components of OKID, as to use a state observer to implicitly estimate the state of the system to be identified, use the Least-Squares solution to ensure that the observer is the Kalman filter for the system in (16). The Least-Squares solution has the same properties as the Kalman filter in terms of the identified Markov parameters and its Gaussian residuals.

With the ARMAX (AutoRegressive Moving Average with eXogeneus inputs) system, presented in (17), the inputs, outputs and noises are used to achieve the output equation, that can also be represented as (18).

$$y(k) = \begin{bmatrix} -y(k-1) & \cdots & -y(k-n_a) \\ u(k-1) & \cdots & u(k-n_b) \\ w(k-1) & \cdots & w(k-n_c) \end{bmatrix}^T \begin{bmatrix} \hat{a_1}(k) \\ \vdots \\ \hat{a}_{n_a}(k) \\ \hat{b_1}(k) \\ \vdots \\ \hat{b}_{n_b}(k) \\ \hat{c_1}(k) \\ \vdots \\ \hat{c}_{n_c}(k) \end{bmatrix} + v_\theta(k) \tag{17}$$

$$y(k) = \hat{y}(k) + v_\theta(k) \tag{18}$$

where the hat symbol is used for estimated values representation.

The estimated output, $\hat{y}(k)$, and the estimated residuals, $v(k)$, are connected by this ARMAX model to reach the output $y(k)$. The equations (19) and (20) are the state-space realization suggested in order to estimate the parameters vector $\theta(k)$.

$$\theta(k+1) = I\theta(k) + W_\theta(k) \tag{19}$$

$$y(k) = I\theta(k) + W_\theta(k) \tag{20}$$

The true parameters vector represents the system's state $\theta(k)$. For that, $w(k)$ and $v(k)$ are zero mean Gaussian processes with covariance matrices $Q(k) \geq 0$ and $R(k) \geq 0$, respectively. As a result, the Kalman filter for parametric estimating shown below can be implemented as

$$\hat{\theta}(k+1) = I\hat{\theta}(k) + L_\theta(k)[y(k) - \hat{y}(k)] \tag{21}$$

$$\hat{y}(k) = \varphi_\theta^T(k)\hat{\theta}(k) \tag{22}$$

The estimator's gain is solved recursively in the same way as in the state estimation case, given by (23), achieved from the recursive solution of Ricatti's equation, presented in (24).

$$L = P_\theta(k)\varphi(k)[R_\theta(k) + \varphi^T(k)P_\theta(k)\varphi(k)]^{-1} \tag{23}$$

$$P_\theta(k+1) = P_\theta(k) - L_\theta(k)\varphi^T(k)P_\theta(k) + Q_\theta(k) \tag{24}$$

By calculating the variances of w and v, according to (25) and (26), respectively, one can automatically tune the Q and R matrices with the values found by OKID method, as shown in (27) and (28), respectively.

$$\sigma_{w_\theta}(k) = \frac{1}{k}\sum_{1}^{k}[w_\theta - \mu_{w\theta}]^2 \tag{25}$$

$$\sigma_{v_\theta}(k) = \frac{1}{k}\sum_{1}^{k}[v_\theta - \mu_{v\theta}]^2 \tag{26}$$

$$Q_\theta(k) = \begin{bmatrix} \sigma_{w_{\theta 1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{w_{\theta i}} \end{bmatrix} \tag{27}$$

$$R_\theta(k) = \sigma_{v_\theta} \tag{28}$$

In this way, using (21), (23) and (24), it is possible to solve the estimation problem with a Kalman Filter observer, using the vector $\hat{\theta}$ as in (29) to pass on the estimated matrices to the Kalman filter state estimator as in (30) and (31).

$$\begin{bmatrix} \hat{\theta}_1^T(k) \\ \vdots \\ \hat{\theta}_n^T(k) \end{bmatrix} = \begin{bmatrix} \hat{A}(k) & \hat{B}(k) & \hat{\Gamma}(k) \end{bmatrix} \tag{29}$$

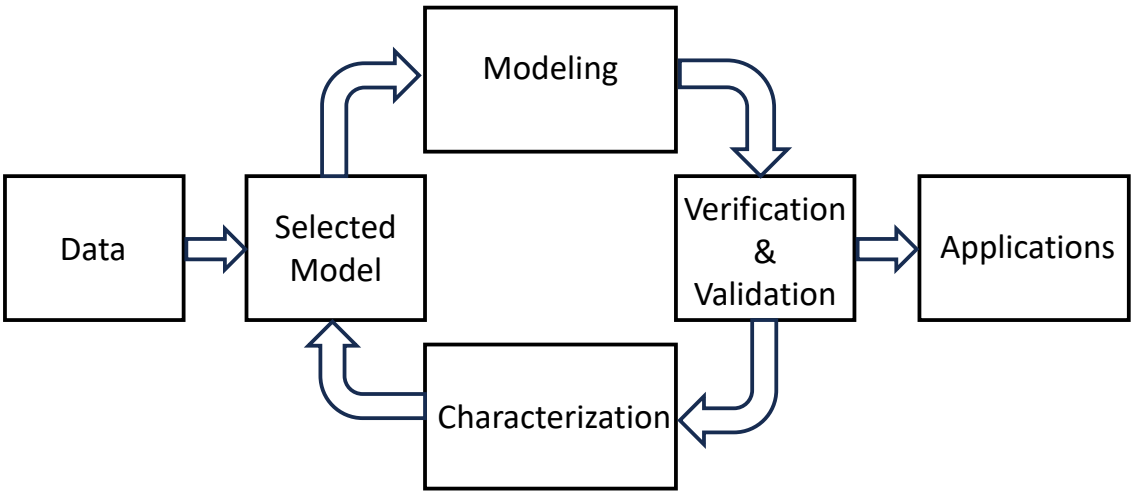
$$\bar{x}(k+1) = \hat{A}(k)\bar{x}(k) + \hat{B}(k)u(k) + L(k)[y(k) - \bar{y}(k)] \tag{30}$$

$$\bar{y}(k) = C\bar{x}(k) \tag{31}$$

3 CONTROL THEORY

Control theory for both nonlinear as well as linear systems is a part of the modern study in this field. Robust control, LQR design, and zero-pole assignment are examples of common design techniques for linear control systems (HOU; WANG, 2013). Lyapunov-based controller designs, back-stepping controller designs, feedback linearization. are examples of common controller design methodologies for nonlinear systems. They all qualify as typical Model-Based Control (MBC) system designs for controller design approaches. In MBC applications, modeling the plant or identifying the plant model comes first. The controller is then designed based on the plant model utilizing the certainty equivalence principle, with the belief that the plant model accurately represents the real system. As a result, the MBC hypothesis requires modeling and plant identification, as shown in Figure 6.

Figure 6 – Block diagram of a generic identification proccess.



Source: Sumathi, Surekha and Surekha (2007) adapted.

The parameters must be achieved either online or offline using measured data when modeling a plant from a real experiment. Using identification theory, it is possible to establish a plant model within a database that either represents the actual system or approximates it in terms of variance error on the identified model (COELHO; COELHO, 2004). Modeling is an approximation of the underlying system, and certain errors are inherent, whether done using fundamental principles or data identification. This is important because the investigation and development of methods that improve the re-

sponses of various systems depend on the understanding of the theoretical foundation of identification and control. There are numerous approaches to applying identification algorithms; the non-parametric method is based on applying a step at a plant's entrance and computing certain values that describe the system's dynamic behavior (AGUIRRE, 2004). A parametric method is a different approach that is described in the current work and involves knowing the system output when it is stimulated by a known input signal (COELHO; COELHO, 2004). Both methods determine how it is the difference between the model that determines the desired plant behavior.

After knowing the process and modeling it, it is possible to evaluate how to get the desired responses in the system, for this reason control strategies can be used if the designer wishes to get a different response from the open-loop system or notices that the plant is unstable (OGATA et al., 2010). These are designed to speed up or speed down the system's response and improve attributes that are beneficial to the entire process, such as reference tracking, a reduction in overshoots, or a shorter settling time. Since many of these controllers rely on a model to be created, if the system is not adequately modeled before it is used, the controllers may not track the reference when they are used in practice and may even damage the system's components.

Over the years, different strategies and topologies have been developed in the literature to obtain the aforementioned results. The difference in controller structures interferes with their ability to track adopted references. Such algorithms may have more or less complex control laws, be based on increasing model structures assuming noise and disturbances or even propose stochastic analysis to predict future outputs. This control study can be analyzed in different domains. The most popular are s-domain (continuous) and z-domain (discrete), but others types of transfer function modelling are being researched, as in (DASTYAR; MALEK; YOUSEFI, 2022).

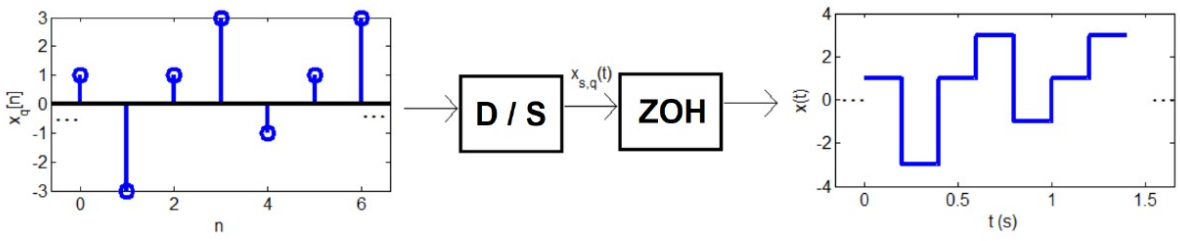
In control theory, the s-domain is a mathematical domain in which the Laplace transform is used to analyze the behavior of linear time-invariant (LTI) systems (OGATA et al., 2010). The Laplace transform is a mathematical tool that allows the designer to convert a time-domain signal or system function into the s-domain, where it can be represented as a transfer function. The s-domain is a complex frequency domain, with s representing the complex frequency variable (OPPENHEIM, 1999). In the s-domain, a transfer function of a system can be expressed as a rational function in the complex variable s, where the numerator and denominator polynomials represent the output and input, respectively. The transfer function describes the relationship between the system input and output in the frequency domain, and it is a useful tool for analyzing the stability, transient response, and steady-state response of a system (NISE, 2020).

In the s-domain, a transfer function of a system can be expressed as a rational function in the complex variable s, where the numerator and denominator polynomials represent the output and input, respectively. The transfer function describes the

relationship between the system input and output in the frequency domain, and it is a useful tool for analyzing the stability, transient response, and steady-state response of a system (OPPENHEIM, 1999). This domain is commonly used in control theory to design controllers, analyze closed-loop stability, and tune control parameters for optimal system performance. The system and controller transfer functions can be merged in the s-domain to generate the closed-loop transfer function, which characterizes the system's behavior under feedback control (ARAÚJO et al., 2017). This allows us to examine the closed-loop system's stability and tweak the controller parameters to achieve the required levels of performance.

The z-domain is another mathematical domain commonly used in digital signal processing and control theory to analyze the behavior of discrete-time systems. In this domain, the z-transform is used to represent discrete-time signals or systems as functions of a complex variable z, which is related to the sampling frequency (OPPENHEIM, 1999). Similar to the Laplace transform in the s-domain, the z-transform converts a discrete-time signal or system into a function in the z-domain and it is a powerful tool for analyzing discrete-time systems, it also allows us to derive transfer functions, analyze stability, and design controllers for digital systems. The Zero Order Hold (ZOH) method, used in this master's thesis for all control models, sustains the amplitude of $x[n_0]$, which is conveyed by the area of its corresponding impulse, during an interval of T_s, sampling time, seconds until the new sample $x[n_0 + 1]$ updates this amplitude and so on. Figure 7 assumes that the digital signal $x_q[n]$ has amplitudes from the set $M = [3,1,1,3]$ and $T_s = 0{,}2s$, as an example proposed by Klautau (2021).

Figure 7 – Signal reconstruction with a Zero Order Hold method.

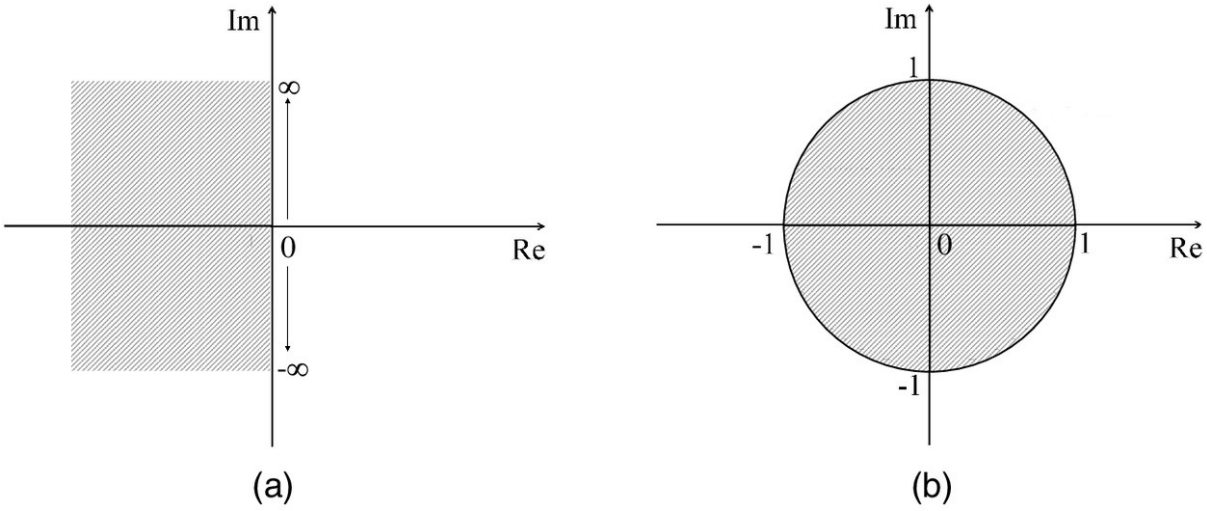


Source: Klautau (2021).

In control theory, the z-transform is often used to design digital controllers for discrete-time systems, such as digital filters or digital control systems, which can be implemented on digital computers or microcontrollers (COELHO; JERONYMO; ARAÚJO, 2019). The design of digital controllers in the z-domain is based on similar principles to those used in the s-domain for continuous-time systems but with modifications to account for the discrete-time nature of the system. Thus, the idea is to fit the zeros and poles of the controller, which were previously represented in an infinite plane, now in a

unitary plane, as represented in Figure 8.

Figure 8 – Schematic of mapping of stability regions from the continuous system to the discrete system: (a) s domain; and (b) z domain.



Source: Tang *et al.* (2021) adapted

Both domains are used in control theory, as demonstrated, whether for simulated or real-world applications. However, as contemporary systems have progressed toward applications employing microcontrollers or high-frequency switching circuits, analysis in the z domain has become more viable, as discrete controllers may be used and examined in this domain with more modern topologies. The algorithms studied throughout the work are presented in this chapter, as shown in the list below, finishing with the performance and robustness analysis that uses RST structures for SISO controllers.

- 2.1: Pseudo Proportional Integrative Derivative Control (PPID);
- 2.2: Generalized Minimum Variance Control (GMV);
- 2.3: Generalized Predictive Control (GPC);
- 2.4: Linear Quadratic Regulator (LQR);
- 2.5: Performance and Robustness analysis.

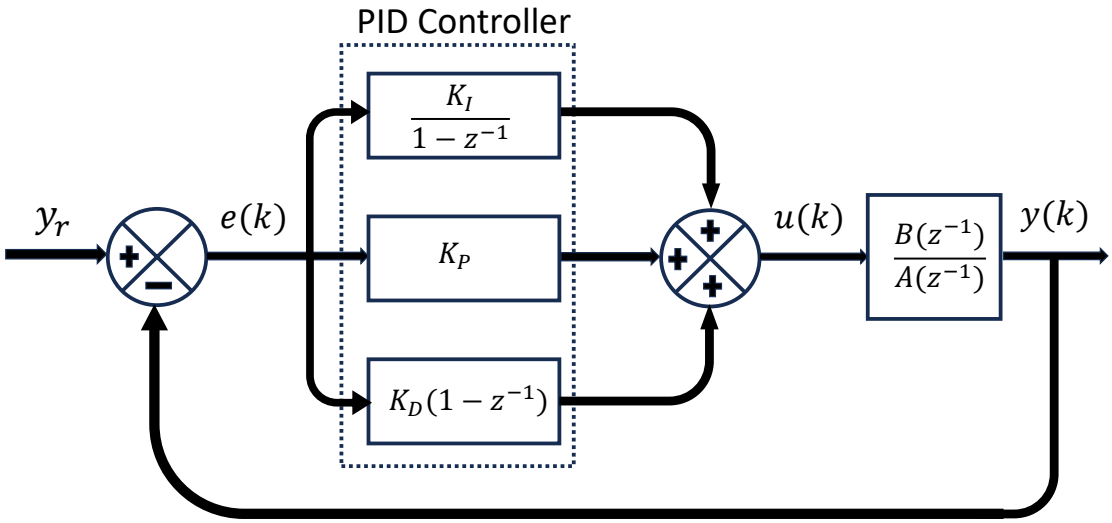
3.1 PSEUDO PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER

The proportional-integral-derivative (PID) controller is a widely used control algorithm in industrial control systems (OGATA et al., 2010). It is a feedback control system that continuously measures the error between the desired setpoint and the actual value of a process variable and uses that error to adjust the output of the controller to minimize the error and bring the process variable closer to the setpoint (OGATA et al., 2010).

The PID controller has three parameters, the proportional component of the controller, K_p, calculates an output that is proportional to the current error signal. This output is proportional to the difference between the setpoint and the current value of the process variable (NISE, 2020). The integral component of the controller, K_I, calculates an output that is proportional to the cumulative error signal. This output is proportional to the sum of the current and past error signals (NISE, 2020). The derivative component of the controller, K_D, calculates an output that is proportional to the rate of change of the error signal. This output is proportional to the difference between the current and past error signals (NISE, 2020).

The outputs of the proportional, integral, and derivative components are combined to produce the final output of the controller. The tuning of the PID controller involves adjusting the gains of the three components to achieve the desired control performance (NISE, 2020). The goal is to minimize the steady-state error, reduce the rise time, and minimize the overshoot and settling time of the controlled process. In Figure 9 is presented the parallel PID discrete mode structure with backward approximation.

Figure 9 – Discrete PID Block Diagram.



Source: Author (2023).

where y_r, $e(k)$, $B(z^{-1})$, $A(z^{-1})$, $u(k)$ and $y(k)$ are, respectively, reference, error, the numerator of the discrete transfer function, denominator of the discrete transfer function, control signal, and output signal.

The PID controller is also a popular choice for controlling discrete processes in industry because it is a simple and effective algorithm that can perform well in a wide

range of applications. Another reason why PID controllers are well-suited to discrete processes is that they can be easily implemented in digital control systems using microcontrollers or programmable logic devices. Those algorithms require only basic arithmetic operations and can be implemented using a few lines of code, making it a cost-effective solution for many industrial applications (OKUYAMA, 2014). As proposed by Ogata *et al.* (2010), the proportional term of the controller provides a quick response to changes in the process variable, while the integral term eliminates steady-state errors caused by disturbances or setpoint changes and the derivative term improves stability by damping out oscillations and reducing overshoot.

There are many topologies of PID controllers since the configurations depend on the application requirements. This choice depends on the specific requirements of the process being controlled, such as the desired response time, stability, and robustness to disturbances. Since the objective of the first Reinforcement Learning tuning method adopted in this work is to acquire one parameter to tune the PID controller, the Pseudo Proportional Integral Derivative (PPID), proposed by Silveira *et al.* (2012), was selected for this thesis.

Based on the diagram of Fig. 9, is possible to obtain the standard structure of the ideal discrete PID control law, as in (32).

$$u(k) = K_c \left\{ e(k) + \frac{T_s}{T_i} \sum_{i=1}^{k} e(i) + \frac{T_d}{T_s} \left[e(k) - e(k-1) \right] \right\} \tag{32}$$

This theoretical approach cannot be implemented in this form but is important to understand that the controller works based on the vector e, which is the difference between the reference set point vector, y_r, and the output vector, y, as shown in (33).

$$e(k) = y_r(k) - y(k) \tag{33}$$

It is important to point out that for the system to be implementable in microcontrollers, such as the Arduino, the output equation must be organized based on the susceptibility and linearity of time-invariant systems (OPPENHEIM, 1999). So it is necessary that the output is composed of variables that are already memorized. Therefore, if we have a sample k of a vector, this vector may contain other variables or a portion of itself at a previous sample, such as $k-1$, $k-2$, or $k-3$. In this way, the difference equations are formed in a way they can be implemented for the processes, as (32) can be transformed into (34)

$$u(k) = u(k-1) + K_c \left\{ e(k) - e(k-1) + \frac{T_s}{T_i} e(k) + \frac{T_d}{T_s} \left[e(k) - 2e(k-1) + e(k-2) \right] \right\} \tag{34}$$

Equation (34), as proposed in Visioli (2006), is appropriate to microcontrollers applications, because it works in single loops and is understandable for digital implementation from the viewpoints of operators and engineers. The proportional and

derivative bands are also multiplied by the system error. This has an effect on controller performance since sudden variations in the reference, as well as in the error, vary instantaneously, leading to control actions with excessive magnitudes. This circumstance may affect actuator implementation and process stability. To prevent practical issues such as loop saturation, the I+PD implementation can be used, assuming to keep the integral term with $e(k) = y_r(k) - y(k)$, as in (33), but with the substitution of the proportional and derivative terms by $e(k) = -y(k)$. As a result, the ideal digital PID control, (34), can be redefined as

$$u(k) = u(k-1) + K_c \left\{ -y(k) + y(k-1) + \frac{T_s}{T_i} e(k) + \frac{T_d}{T_s} \left[2y(k-1) - y(k) - y(k-2) \right] \right\} \quad (35)$$

A PPID controller with a single parameter is developed to have a simple practical calibration that not only maintains stability and closed-loop performance but also facilitates the operator's tuning duty. Silveira *et al.* (2012) proposed an analysis based on the relationship established by Visioli (2006), where is possible to set

$$\frac{T_s}{T_i} > \frac{1}{100} \quad ; \quad T_i = [2...5]\, T_d \quad (36)$$

After that, it is possible to achieve the following normalized expressions, (37), from (35) and (36).

$$\frac{T_d}{T_s} = 0.4 \quad ; \quad \frac{T_i}{T_d} = 4 \quad ; \quad \frac{T_s}{T_i} = 0.1 \quad (37)$$

Those numbers are chosen based on commercial microcontrollers and programmable logic controller tuning approaches in industries. A few critics gotta be made for Silveira *et al.* (2012) article, since those explanations are not clear and does not show how exactly those numbers appear. However, the structure of the controller works in many applications and using a simple control law can be implemented as shown below.

$$u(k) = u(k-1) + K_c \{0.1 y_r(k) - 3.6 y(k) + 6 y(k-1) - 2.5 y(k-2)\} \quad (38)$$

With (38) it is possible to analyze some characteristics of the pseudo-PID controller design, that justifies its application in the current master's thesis. The first one is that only the K_c parameter will be used to tune the control law, which makes it easier for the designer to find the optimal value for his design specification, which can be based on performance and robustness when evaluating the metrics proposed by the literature, which are explained in Chapter 3. The second one is this type of control law, which provides good performance in simple and complex plants (nonlinear). Finally, the structure of the PPID equation is appropriate from the viewpoint of implementation in digital technologies (hardware and software) and understanding by plant operators.

It is possible to examine the effect of the tuning parameter K_c in the frequency domain to guarantee stability for the closed-loop system. This makes it possible to

assess the robust stability, when there is a model plant mismatch, and the small gain theorem (BANERJEE; SHAH, 1992). The pseudo-PID control's digital equation can be reorganized into the RST canonic structure, as proposed in Silveira *et al.* (2012):

$$R(z^{-1})u(k) = T(z^{-1})y_r(k) - S(z^{-1})y(k) \tag{39}$$

$$R(z^{-1}) = \Delta = 1 - z^{-1} \tag{40}$$

$$S(z^{-1}) = K_c[3.6 - 6z^{-1} + 2.5z^{-2}] \tag{41}$$

$$T(z^{-1}) = 0.1K_c \tag{42}$$

3.2 GENERALIZED MINIMUM VARIANCE CONTROL

A Generalized Minimum Variance (GMV) controller is a robust control approach used in engineering to maximize dynamic system performance while minimizing output variance or risk. It is especially beneficial in scenarios including uncertainties, disruptions, and noise, which are ubiquitous in real-world applications (COELHO; JERONYMO; ARAÚJO, 2019).

One of the most well-known applications of GMV controllers is in industrial process control. Consider a chemical reactor that is programmed to produce a specific chemical compound. Temperature fluctuations, variations in reactant concentrations, and perturbations in the feed flow rate all impact the reactor's behavior. Using a GMV controller helps the system to respond to these uncertainties and disturbances, preserving stable operation and minimizing product output variances (CLARKE; MOHTADI, 1989).

The design of the GMV controller involves developing a dynamic model of the system that captures its behavior under diverse scenarios. This model is critical for forecasting how the system will respond to various control inputs and disturbances (COELHO; JERONYMO; ARAÚJO, 2019). The controller aims to calculate optimal control signals that steer the system toward desired setpoints while reducing output variation. This is frequently accomplished by creating an optimization problem and employing techniques such as quadratic programming (SILVEIRA *et al.*, 2020).

In addition to that, the control law is derived from the minimization of a cost function associated with the concept of generalized and stochastic systems proposed by Clarke and Gawthrope (1975). As a result, some performance parameters that apply to various applications can be included in the synthesis of this controller in order to give flexibility to the control structure. Because it is a SISO system described as an ARIX model, The goal of a good GMV controller is to determine control action $u(k)$ that minimizes the cost of the function $J = E[\varphi(k + d)]$, where $E[.]$, according to Coelho, Jeronymo and Araújo (2019), is the generalized system, which may be used to project the GMV controller implemented in this master's thesis, can be expressed as in (43).

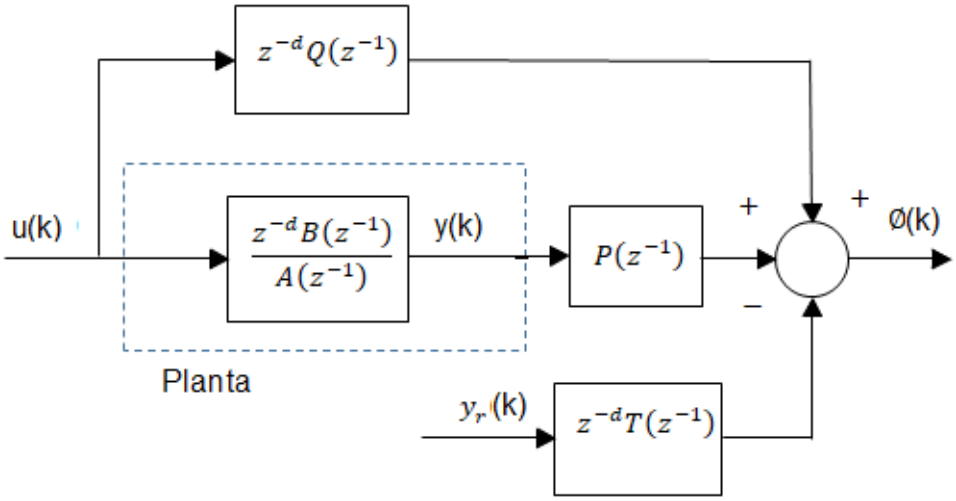
Being $P(z^{-1})$ (output filter), $Q(z^{-1})$ (control signal weighting factor) and $T(z^{-1})$ (Error correction in steady time) responsible for the system's respective signals, which act as GMV synthesis parameters. $y(k+d)$ is the d steps ahead output, as $u(k+d)$ is the control signal and $y_r(k)$ reference signal.

é o valor esperado (ou esperança matemática, ou simplesmente a média) e a saída generalizada (φ) é dada por:

$$J_{GMV} = [\Phi(k+d)]^2 = [P(z^{-1})y(k+d) - T(z^{-1})y_r(k) + Q(z^{-1})u(k)]^2 \tag{43}$$

The generalized output of the GMV controller is shown in Figure 10, whose representation leads to the admission that the discrete process is represented by the DCAR model (*Deterministic Controlled Auto-Regressivve*), as in (44).

Figure 10 – Generalized output block diagram of GMV controller.



Fonte: Coelho, Jeronymo and Araújo (2019).

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k) \tag{44}$$

The approach proposed in the dissertation thesis is an incremental indirect hybridization of GMV-RST, as presented in Coelho, Jeronymo and Araújo (2019). The polynom $P(z^{-1})$ is expressed in (45), and when is inserted in (43), results in the incremental GMV control law (46).

$$P(z^{-1}) = A(z^{-1})\Delta E(z^{-1}) + z^{-d}S(z^{-1}) \tag{45}$$

$$[B(z^{-1})E(z^{-1}) + Q(z^{-1})]\Delta u(k) = T(z^{-1})y_r(k) - S(z^{-1})y(k) \tag{46}$$

where the calculation of the plant's control signal is represented by (47).

$$u(k) = u(k-1) + \Delta u(k) \tag{47}$$

The GMVC function of transference in a closed loop is observed in (48). However, in order to ensure reference tracking e no error in steady time, it is necessary to consider $T(1) = t_0 = P(1) = S(1)$, as proposed by (COELHO; JERONYMO; ARAÚJO, 2019).

$$y(k) = \frac{B(z^{-1})T(z^{-1})}{B(z^{-1})P(z^{-1}) + A(z^{-1})\Delta Q(z^{-1})} y_r(k-d) \tag{48}$$

The incremental indirect GMV project also needs a determination of the order of the polynomials $\mathrm{E}(z^{-1})$ and $\mathrm{S}(z^{-1})$. The first, because it is a transportation-free process, has order zero; thus, the polynomial is just a constant e_0 and the second is as follows

$$N_s = N_a \tag{49}$$

For this reason $\mathrm{S}(z^{-1})$ is represented as

$$S(z^{-1}) = s_0 + s_1 z^{-1} + \ \ldots \ + s_n z^{-n_a} \tag{50}$$

$\mathrm{P}(z^{-1}) = 1$ is considered within this master's thesis since it simplifies the implementation and directs the focus on evaluating GMV tuning by means of RL instead of the influence of filters; nevertheless, in future works, it will be done as a means of comparison.

Based on the provided polynomials, it is possible to determine the values of s_0, s_1 and s_2 from (45), where due to order zero of the polynomials $\mathrm{P}(z^{-1})$ and $\mathrm{E}(z^{-1})$, $e_0 = 1$ and $p_0 = 1$. So this equation can be rewritten as

$$1 = (1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-na}) e_0 + z^{-1}(s_0 + s_1 z^{-1} + s_2 z^{-2} + ... + s_{n-1} z^{na}) \tag{51}$$

Using the polynomial identity shown above, s_0, s_1, and s_n assume the values presented below.

$$s_0 = 1 - a_1 \tag{52}$$

$$s_1 = a_1 - a_2 \tag{53}$$

$$s_n = -a_n \tag{54}$$

In order to achieve R polynomial values, Equation (55) is used, since it refers to a control law written in the traditional RST format. What takes the system to (56), because $\mathrm{Q}(z^{-1})$ is treated as a constant q_0 used to tune the controller, so is where the reinforcement learning will act.

$$R(z^{-1}) = B(z^{-1})E(z^{-1}) + Q(z^{-1}) \tag{55}$$

$$r_0 + r_1 z^{-1} + ... + r_n z^{-nb} = (b_0 + b_1 z^{-1} + b_n z^{-nb}) e_0 + q_0 \tag{56}$$

Based on (56), r_0, r_1 and r_n are calculated as below.

$$r_0 = b_0 + q_0 \tag{57}$$

$$r_1 = b_1 \tag{58}$$

$$r_n = b_n \tag{59}$$

As the general theory of this Chapter is gonna be implemented on second-order models, the GMVC control law is applied as below for the proposed systems.

$$\Delta u(k) = -\frac{r_1}{r_0}\Delta u(k-1) + \frac{t_0}{r_0}y_r(k) - \frac{s_0}{r_0}y(k) - \frac{s_1}{r_0}y(k-1) - \frac{s_2}{r_0}y(k-2) \tag{60}$$

3.3 GENERALIZED PREDICTIVE CONTROLLER

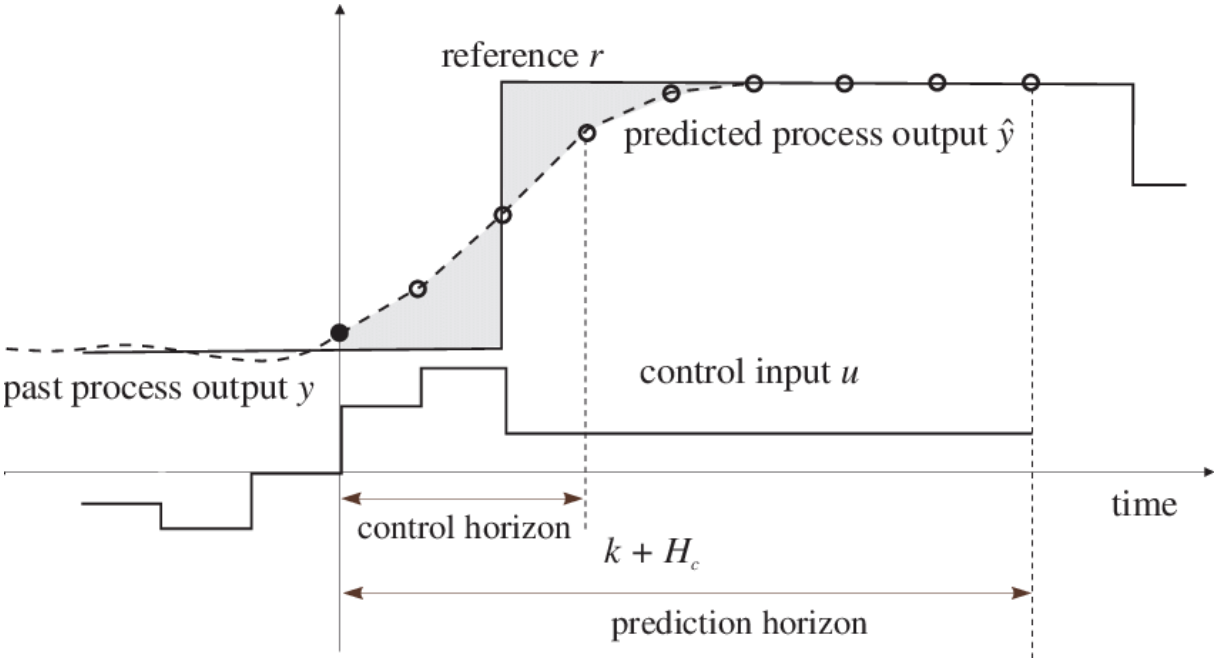
The Generalized Predictive Controller (GPC) is a popular model-based control technique applied in engineering and commercial applications. It is a kind of predictive control that attempts to maximize system performance by the prediction of system behavior in the future and the computation of control actions by that prediction. A mathematical model of the system is used by the GPC algorithm to characterize the system's dynamics and how it responds to control inputs. This model can be created using system identification approaches based on experimental data or from basic principles.

The fundamental concept of the GPC is to make predictions of the system's future behavior over a finite prediction horizon (BABUSKA et al., 2023). By predicting the future states of the system, the controller can optimize the control actions in advance to achieve desired setpoints or track reference trajectories while considering constraints on the control inputs and system outputs as represented in Figure 11. With this structure, GPC can manage complicated systems with various inputs and outputs and handle time delays, disturbances, and uncertainties, thanks to its predictive nature. Even for systems with time-varying features and nonlinearities, it can deliver good control performance.

The control actions that minimize a cost function across the prediction horizon are calculated by the GPC using an optimization technique, often based on quadratic programming (COELHO; JERONYMO; ARAÚJO, 2019). Finding the best control trajectory that satisfies the control objectives and limitations involves using the cost function, which takes into account both the control effort and the tracking error.

One of the advantages of the GPC is its ability to handle constraints on the control inputs and outputs, making it suitable for applications where certain variables need to stay within predefined limits. This makes it particularly useful in industrial processes where safety, stability, and efficiency are crucial (BITMEAD; GEVERS; WERTZ, 1989). It is important to inform that in this master's thesis, the linear programming methods normally adopted to include such restrictions will not be used, but based on the performance and robustness indices this will be done by Reinforcement Learning (RL). In

Figure 11 – Model Based predictive controllers and the horizons influence.



Source: Babuska *et al.* (2023).

this way, in order to achieve the desired response, this controller's parameters must be adjusted, those are N_y, N_u *and* λ, which are, respectively, the output prediction horizon, the control prediction horizon, and the control weight factor to be used for RL tuning in this work.

In the GPC project, the process is represented using a CARIMA (*Controlled Auto-Regressive Integrated Moving Average*) discrete linear parametric model, as in (61).

$$A(z^{-1})\Delta y(k) = z^{-1}B(z^{-1})\Delta u(k) + C(z^{-1})v(k) \tag{61}$$

Δ is the difference discrete operator $1 - z^{-1}$, $v(k)$ is a Gaussian disturbance, and $C(z^{-1})$ is a monic polynomial which can characterize the influence of the colored noise case in GPC (BITMEAD; GEVERS; WERTZ, 1989; YOON; CLARKE, 1994). A cost function in a predictive controller quantifies the discrepancy between desired and predicted outcomes, guiding control decisions. It is designed to be minimized, ensuring optimal system behavior while accounting for control objectives and constraints (CAMACHO; BORDONS, 2007). In GPC it can be explained as the error between the prediction of the future and the reference value as well as the increase in future control, as in (62), where N_1 e j are, respectively, the minimum horizon and the interval between the control and output horizons.

$$J_{GPC} = \sum_{j=N_1}^{N_y} [y(k+j) - y_r(k+j)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u(k+j-1)]^2 \tag{62}$$

To solve this minimal order problem it is necessary to do a minimization process in (62), a predicted output ($\hat{y}$) achieved using a function of the instantly known signals values at time k (instant sample) and also the future control entries that must be calculated. With (63), where $E_j(z^{-1})$ and $F_j(z^{-1})$ are determined by the plant model and the variable j in conjunction with (62) is possible to achieve the predicted output in (64).

$$C(z^{-1}) = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \tag{63}$$

$$\hat{y}(k+j) = \frac{F_j(z^{-1})}{C(z^{-1})}y(k) + \frac{E_j(z^{-1})B(z^{-1})}{C(z^{-1})}\Delta u(k+j-1) \tag{64}$$

Equation (65) is used to separate the control's past and future values in order to generate (66), where $u_f(k)$ and $y_f(k)$ are the filtred values of $\Delta u(k)$ and $y(k)$, by $C(z^{-1})$.

$$E_j(z^{-1})B(z^{-1}) = G_j(z^{-1})C(z^{-1}) + z^{-j}\overline{G_j}(z^{-1}) \tag{65}$$

$$\hat{y}(k+j|k) = \overline{G_j}(z^{-1})u_f(k-1) + F_j(z^{-1})y_f(k) \tag{66}$$

Vector φ can be created with the free response predictions and the incremental future control mechanism, as, respectively, in (67) and (68).

$$\varphi = \begin{bmatrix} \hat{y}(k+1|k) & \hat{y}(k+2|k) & \dots & \hat{y}(k+N_y|k) \end{bmatrix}^T \tag{67}$$

$$\widetilde{U} = \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N_u-1) \end{bmatrix}^T \tag{68}$$

Based on those vectors, (66) can be rewritten in a vectorial form as in (69), where $\hat{Y}$ is composed by the output responses.

$$\hat{Y} = G\widetilde{U} + \varphi \tag{69}$$

A step response matrix G, presented in (70) with ($N_y x N_u$) dimension and lower triangular Toeplitz matrix structure is calculated, admitting $\Delta u(k+j) = 0$ when $j \geq N_u$. With this, the cost function can be minimized, in order to obtain the control vector and the control law in (71) and (72), respectively, where K_{GPC} is this controller's gain and

the first line of g_t, shown in (73).

$$G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ g_{Nu-1} & g_{Nu-2} & \cdots & g_0 \\ \vdots & \vdots & \cdots & \vdots \\ g_{Ny-1} & g_{Ny-2} & \cdots & g_{N_y-N_u} \end{bmatrix} \tag{70}$$

$$\widetilde{U} = (G^T G + \lambda I)^{-1} G^T (y_r - \varphi) \tag{71}$$

$$u(k) = u(k-1) + K_{GPC}(y_r - \varphi) \tag{72}$$

$$g_t = (G^T G + \lambda I)^{-1} G^T \tag{73}$$

Coelho, Jeronymo and Araújo (2019) propose the use of the diophantine equation in (63) to project the system that can be effectively implemented, using the multiplication of polynomials A(z^{-1}) and Δ, achieving $\widetilde{A}(z^{-1})$, as presented in (74). This approach has been chosen for the particular implementation of second-order models implemented in this master's thesis.

$$\widetilde{A}(z^{-1}) = A(z^{-1})\Delta = 1 + a_1 z^{-1} + a_2 z^{-2}(1 - z^{-1}) = 1 + (a_1 - 1)z^{-1} + (a_2 - a_1)z^{-2} - a_2 z^{-3} \tag{74}$$

As proposed in GMVC design, $C(z^{-1})$ has a unitary value in order to evaluate the system without filtering. The ratio of this unitary polynomial and $\widetilde{A}(z^{-1})$, results in $E_j(z^{-1})$ and $F_j(z^{-1})$. They are removed from the division as indices of the quotient and rest of the ratio equation, respectively. $E(z^{-1})$ order is achieved with (75) and $F(z^{-1})$ with (76). This fact brings up the assessment of gaps in the GPC since it is a controller that presents this tuning paradigm that can be evaluated in future works, but which briefly occurs when the controller works for a range of horizons and within this range fails for some value without explanation in the literature, being one of the motivations for Silveira *et al.* (2012). The idea of using reinforcement learning in different ranges would also be to analyze whether these gaps disappear.

$$N_{e_j} = j - 1 \tag{75}$$

$$N_{f_j} = N_a \tag{76}$$

It is important to analyze that the GPC uses the predicted tracking error multiplied by the gain K_{GPC}, which is important for the RST hybridization, since the equality in

(72) and (55), can be organized in R(z^{-1}), S(z^{-1}) and T(z^{-1}) polynomials as shown below.

$$R(z^{-1}) = \left[1 + z^{-1} \sum_{j=1}^{Ny} K_{GPC_j} \bar{G}_j\right] \tag{77}$$

$$S(z^{-1}) = \sum_{j=1}^{Ny} K_{GPC_j} F_{GPC_j} \tag{78}$$

$$T(z^{-1}) = \sum_{j=1}^{Ny} K_{GPC_j} z^j \tag{79}$$

where R(z^{-1}) is monic, being $r_0 = 1$.

3.4 LINEAR QUADRATIC REGULATOR

The Linear Quadratic Regulator (LQR) is a popular control algorithm widely used in control theory for designing optimal control systems. LQR is a type of state feedback control algorithm that computes a set of feedback gains that minimize a performance criterion. The criterion is typically a quadratic function of the state variables and the control inputs. The LQR controller is designed based on a mathematical model of the system to be controlled, which describes the dynamics of the system. It uses this model to compute the optimal feedback gains.

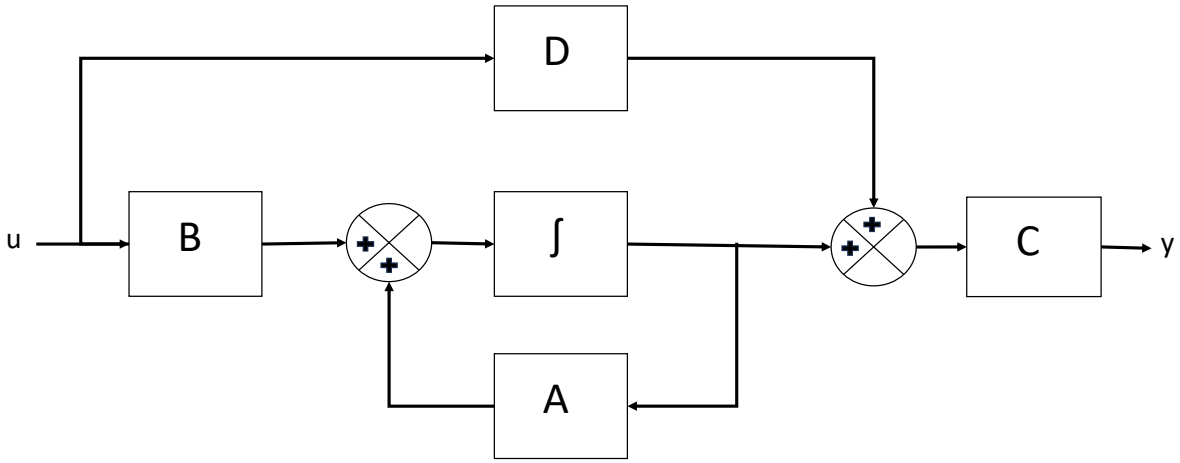
The LQR algorithm is used in control theory to design control systems for a wide range of applications, including aerospace, automotive, and industrial control systems. This regulator is used to control systems with a continuous and discrete time, linear time-invariant (LTI) dynamical system model. This includes systems with multiple inputs and outputs and systems with unstable dynamics. This algorithm is widely used because it is simple, robust, and easy to implement. Its implementation is done in state space, based on the model as shown in (86). The block diagram of the state space model representation is presented in Figure 12.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{80}$$

In Figure 12, x is the state vector, with $x(t) \in \mathbb{R}^n$; y is the output vector, with $y(t) \in \mathbb{R}^q$; u is the control signal vector, with $u(t) \in \mathbb{R}^p$; A is the state matrix, with $dim[A] = n \times n$; B is the input matrix, with $dim[B] = n \times p$; C is the output matrix, with $dim[C] = q \times n$; D is the feedforward matrix, with $dim[D] = q \times p$, where $dim[.]$ refers to the dimension of these matrices.

The LQR controller is based on the principle of feedback control, which involves measuring the output of the system and using this information to adjust the inputs to the

Figure 12 – Space State Representation.



Source: Author (2023)

system. This algorithm computes the feedback gains that minimize the error between the actual and desired state of the system, and the control inputs that minimize the quadratic performance criterion. In summary, the LQR controller is a state feedback control algorithm widely used in control theory for designing optimal control systems. To achieve that, the proposed system in (86) is associated to the infinite-horizon quadratic cost function or performance index, as in (81).

$$V(x(t_0),t_0) = \int_{t_0}^{\infty} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau))d\tau \tag{81}$$

According to Lewis and Vrabie (2012), if the weight matrices $Q \geq 0$, $R > 0$, and (A,B) is stabilizable, then there is a possibility that LQR can stabilize the system. It can be noticed, that if $(A,\sqrt{Q})$ is detectable, the system's unstable modes can be seen in the output $y = \sqrt{Q}x$, therefore the system can be stabilized.

Finding a control policy that minimizes costs is necessary for solving the LQR optimum control problem, as proposed in (82).

$$u^*(t) = arg \min_{\substack{u(t) \\ t_0 \leq t \leq \infty}} V(t_0,x(t_0),u(t)) \tag{82}$$

The proposed solution for (82) is equated by (83), where the matrix gain K is expressed by (84).

$$u(t) = -Kx(t) \tag{83}$$

$$K = R^{-1}B^{T}P \tag{84}$$

A positive definite solution to the algebraic Riccati equation (ARE), presented in (85), is matrix P.

$$A^{T}P + PA + Q - PBR^{-1}B^{T}P = 0 \tag{85}$$

There is a singular positive semidefinite solution to the ARE that produces a stabilizing closed-loop controller, given by (83), under the stabilizability and detectability requirements. This solution is where the closed-loop system $A - BK$ is asymptotically stable (LEWIS; VRABIE, 2012). Equation (85) is, also, an offline solution method that must be used to solve the ARE with complete information of the system dynamics matrices (A,B). Therefore, a new optimal control solution must also be computed whenever the system dynamics change or the performance index changes while the system is in operation.

Since the LQR is implemented in a digital structure, the theory has a few changes for discrete mode. (86) has a few changes that are assumed for the algorithms, as the presence of Γ (noise vector) and ε as an identification estimation error, which brings a stochasticity analysis of the disturbance for the regulator. The equation can be rewritten as

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-d) + \Gamma\varepsilon \\ y(k) &= Cx(k) \end{aligned} \tag{86}$$

This master's thesis uses the LQR with zero amplitude reference; for this reason, the control law is written as (87) as a minimization of the cost function presented in (88).

$$u(k) = -Kx(k) \tag{87}$$

$$J = \sum_{0}^{\infty} \left[x^{T}(k)Qx(k) + u^{T}(k)Ru(k) \right] \tag{88}$$

3.5 PERFORMANCE AND ROBUSTNESS ANALYSIS

3.5.1 Performance analysis

The trade-off between performance and robustness is a key issue in control design (ÅSTRÖM; WITTENMARK, 2013). Since performance indicators are chosen with an emphasis on the specifications considered important to the system, their use as a quantitative measure to assess the implemented controller is interesting because when they are minimized, the control system is considered effective or performing within the desired standards (ARAÚJO et al., 2017).

Two examples of performance indexes that measure a controller's performance are the mean integral of the squared error (ISE) and the mean integral of the squared

control signal (ISU), both of which are applied in this master's thesis, the first for evaluating the reference tracking efficiency and the second for control effort. ISE and ISU can be calculated with (89) and (90), respectively.

$$ISE = \frac{1}{N}\sum_{k=1}^{N}[e(k)]^2 \tag{89}$$

$$ISU = \frac{1}{N}\sum_{k=1}^{N}[u(k)]^2 \tag{90}$$

When evaluating the performance of the control systems, the stochastic and probabilistic analysis of the signals can reveal additional intriguing factors (MARTINS; GONTIJO; GONÇALVES, 2019). In this way, evaluating the variance of the error (σ_e^2) and control (σ_u^2) signals is interesting to obtain a probabilistic and stochastic evaluation of the processing of digital signals. Those can be calculated, respectively, as (91) and (92).

$$\sigma_e^2 = \frac{1}{N}\sum_{k=1}^{N}\left[e(k)-\mu_e\right]^2 \tag{91}$$

$$\sigma_u^2 = \frac{1}{N}\sum_{k=1}^{N}\left[u(k)-\mu_u\right]^2 \tag{92}$$

where μ_e and μ_u are, respectively, the mean value of error and control signals.

3.5.2 Robustness analysis

Robustness indexes are required to qualify the implemented controller as optimal in the sense of being robust to external disturbances, such as noise, modeling uncertainties, and load. Among those indexes are Gain Margin (GM) and Phase Margin (PM), which are directly related to the robust stability of the process. The higher the values of these indexes, more robust (less sensitive to unwanted disturbances) the system is on the other hand, the response velocity becomes slower (ARAÚJO et al., 2017; SILVA et al., 2021).

According to Coelho, Jeronymo and Araújo (2019), the GM is defined as the required variation in the open-loop gain, necessary to make the system unstable, and the PM also provides a measure of the relative stability, indicating how much transport delay can be included in the feedback loop before instability to occurs.

Other two variables are interesting to achieve the values of GM and PM on the controlled system, the Sensitivity function (S_{sen}) and the Complementary Sensitivity (T_{com}), presented in (93) and (94), respectively.

$$S_{sen}(z) = \frac{1}{1 + G_c(z)G_p(z)} \tag{93}$$

$$T_{com}(z) = \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)} \tag{94}$$

where $G_c(z)$ and $G_p(z)$ are, respectively, the controller and the process discrete transfer functions.

S_{sen} characterizes the effect of an external disturbance acting on the output of the control loop, therefore indicates how the closed-loop system is sensitive to process changes, while T_{com} is the Closed Loop Transfer Function (CLTF) for set-point changes (ARAÚJO et al., 2017; SEBORG et al., 2016; SKOGESTAD; POSTLETHWAITE, 2007).

The maximum values of the amplitude ratio of $S_{sen}(z)$ and $T_{com}(z)$ for all frequencies, respectively, M_S and M_T (known as resonant peak), provide useful robustness measures and also shows a control system design criterion. These functions can be described by (95) and (96).

$$M_S \stackrel{D}{=} max||S_{sen}(e^{j\omega T_s})|| \tag{95}$$

$$M_T \stackrel{D}{=} max||T_{com}(e^{j\omega T_s})|| \tag{96}$$

The value of M_S can also be calculated after a polynomial equality between the control law of the controller and the canonical form of the controller RST because, in this way, it is possible to incorporate the advantageous characteristics of the proposed predictive controllers into the controller (ARAÚJO et al., 2017). The outcome of this equality is (97) after the proposed consideration using (94).

$$M_s \stackrel{D}{=} max_{(0 \leq \omega_n < \pi)} \frac{A(e^{j\omega T_s})\Delta R(e^{j\omega T_s})}{A(e^{j\omega T_s})\Delta R(e^{j\omega T_s}) + B(e^{j\omega T_s})S(e^{j\omega T_s})} \tag{97}$$

According to Postlethwaite (1996), with M_S and M_T is possible to obtain GM and PM, as in (98) and (99), respectively. This mathematical relation is valid for all implemented controllers of the paper.

$$GM \geq min\left[\frac{M_S}{M_S - 1}, \frac{M_T + 1}{M_T}\right] \tag{98}$$

$$PM \geq min\left[2sin^{-1}\left(\frac{1}{2M_S}\right), 2sin^{-1}\left(\frac{1}{2M_T}\right)\right] \tag{99}$$

4 REINFORCEMENT LEARNING TUNING METHODS

Reinforcement Learning (RL) is a type of machine learning in which an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties based on its actions, enabling it to learn optimal strategies over time. Through trial and error, the agent aims to maximize cumulative rewards, effectively solving complex problems in dynamic and uncertain environments.

RL has been applied to a wide range of applications, including robotics, gaming, recommendation systems, and autonomous vehicles. In robotics, it has been used to train them to perform complex tasks such as grasping objects, navigation, and manipulation. RL algorithms can enable robots to learn from experience and adapt to changing environments, making them more versatile and capable. On Gaming has been used to train agents to play games such as chess, Go, and Atari games. In some cases, RL algorithms have been able to achieve superhuman performance, surpassing human players. RL has been used to train autonomous vehicles to make decisions in complex driving scenarios, such as avoiding obstacles, navigating intersections, and merging into traffic. RL algorithms can enable vehicles to learn from real-world experience and improve their driving skills over time.

Overall, RL has shown promising results in a wide range of applications, and its potential for solving complex problems continues to be explored by researchers and practitioners in various fields. It has emerged, also, as a powerful tool for solving control problems in a wide range of applications. Classical control theory, which encompasses proportional-integral-derivative (PID) controllers and other linear control techniques, has long been the dominant paradigm for controlling systems (DOGRU et al., 2022). However, RL offers a new approach to control that can learn from experience and adapt to changing environments, making it well-suited to many real-world control problems.

Classical control techniques, such as PID controllers, have been used for decades to control systems in a wide range of applications. These techniques are based on linear models of the system being controlled and rely on the assumption that the system's behavior is predictable and stable, as explained in Chapter 2. However, many real-world systems are complex, nonlinear, and subject to disturbances that can cause unpredictable behavior. As a result, classical control techniques can be limited in their ability to control these systems effectively. Furthermore, classical control techniques require expert knowledge and tuning to achieve optimal performance. This process can be time-consuming and labor-intensive, and may not be suitable for systems that are subject to changes in their operating conditions or parameters.

Reinforcement Learning offers several advantages over classical control techniques. First, RL can learn from experience, allowing it to adapt to changing environments and handle nonlinear dynamics and disturbances. RL can also learn optimal

control policies without the need for expert knowledge or tuning, making it well-suited to systems that are difficult to model or control.

Another advantage of RL is that it can handle multiple objectives or constraints simultaneously, allowing it to optimize system performance under a range of conditions. RL can also handle systems with large state and action spaces, which may be intractable for classical control techniques.

Since RL is a type of machine learning where an agent learns to make decisions by interacting with an environment and receiving feedback in the form of rewards. The goal of the agent is to learn a policy, or a mapping from states to actions, that maximizes the cumulative reward over time.

The RL agent interacts with the environment by taking actions and observing the resulting state and reward. The agent then updates its policy based on the observed state and reward, using techniques such as value iteration, policy iteration, or Q-learning.

RL can be applied to a wide range of control problems, including continuous control, discrete control, and partially observable control. RL can also handle stochastic environments and nonlinear dynamics, making it well-suited to many real-world control problems.

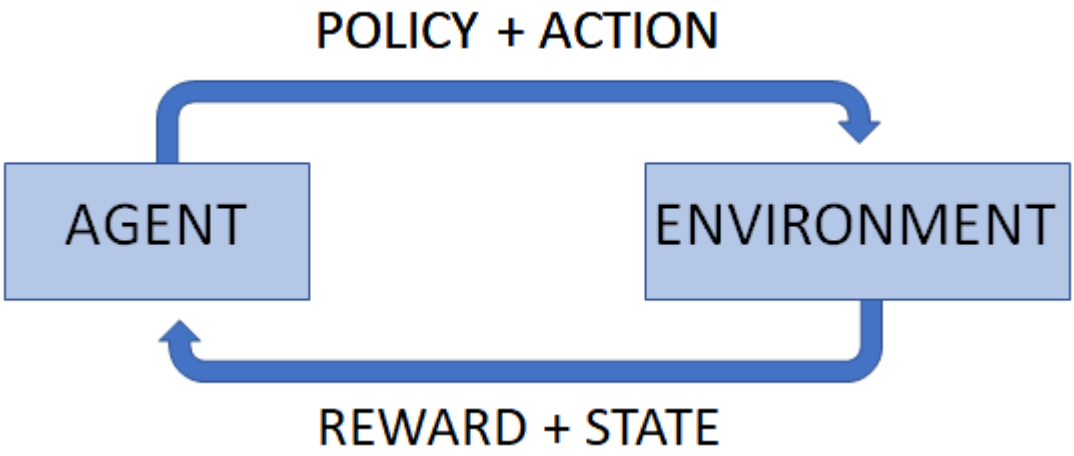
The algorithms of RL can be summarized in three main steps, and represented as in Fig. 13:

- Action selection: The RL agent selects an action based on the current state of the environment and the agent's policy. The policy is a mapping from states to actions that the agent has learned through experience.
- Reward observation: The environment returns a reward signal to the agent based on the action taken. The reward signal indicates the desirability of the agent's action and is used by the agent to update its policy.
- Policy update: The agent updates its policy based on the observed reward signal and the current state of the environment. There are several methods for updating the policy, including value iteration, policy iteration, and Q-learning.

As proposed by Sutton and Barto (2018), The agent updates its policy based on the observed reward signal and the current state of the environment. The policy update is based on the principle of reinforcement, which is to increase the likelihood of actions that lead to positive rewards and decrease the likelihood of actions that lead to negative rewards.

There are several methods for updating the policy, including value iteration, policy iteration, and Q-learning. Value iteration involves estimating the value function, which is the expected cumulative reward starting from a given state and following a given policy. Policy iteration involves optimizing the policy directly by iteratively improving the value function. Q-learning involves learning the optimal action-value function, which is the

Figure 13 – Reinforcement Learning Diagram.



Source: Silva, Silveira and Nascimento (2022)

expected cumulative reward starting from a given state and taking a given action, and then updating the policy based on the learned Q-values (VRABIE; LEWIS, 2013).

RL algorithms can be further categorized based on the type of learning being used, such as on-policy or off-policy learning, model-based or model-free learning, and deep reinforcement learning. On-policy learning involves updating the policy based on experience generated by the current policy, while off-policy learning involves updating the policy based on experience generated by a different policy. Model-based learning involves using a model of the environment to learn the optimal policy, while model-free learning involves directly learning the optimal policy without a model of the environment. Deep reinforcement learning involves using deep neural networks to approximate the value function or policy.

The performance and robustness indexes of classical control theory can be used in RL for process control for several reasons, as familiarity, since control engineers are often more familiar with the performance and robustness indexes of classical control theory, such as settling time, overshoot, and stability margins, than with RL-specific measures such as cumulative reward or convergence rate. By using familiar indexes, control engineers can better understand and interpret the performance of RL-based control systems. Other reason is to compare with existing controllers that control engineers can use the same performance and robustness indexes to compare the performance of RL-based controllers with existing classical control techniques. This can help control engineers to evaluate the potential benefits of using RL-based controllers in comparison to existing control techniques. Furthermore, optimization of RL-based controllers, since control engineers can use the performance and robustness indexes to optimize the RL-based controller's parameters, such as the learning rate or exploration rate, to

achieve better performance or robustness.

By setting the optimization objective to be the same as the classical control indexes, the control engineer can ensure that the RL-based controller performs at least as well as existing control techniques. Finally, the integration with existing control systems, since RL-based controllers can be integrated with existing classical control systems, and the performance and robustness indexes can be used to evaluate the overall performance of the integrated system. By comparing the performance of the integrated system with the performance of the classical control system alone, control engineers can evaluate the benefits of integrating RL-based controllers into existing systems.

Overall, using performance and robustness indexes of classical control theory in RL-based process control can help control engineers to better understand and interpret the performance of RL-based controllers, the use of this computational tool is made in this master's thesis based on Repeat and Improve and Differential Games methods, the first using the base of reinforcement learning proposed by Sutton and Barto (2018) with a high computational effort; and the second method using optimal tuning strategies to achieve the parameters of LQR.

4.1 REPEAT AND IMPROVE METHOD

Reinforcement Learning (RL) is a type of machine learning developed in the Computational Intelligence Community (VRABIE; LEWIS, 2013), the way it works can be represented by the diagram in Figure 13, where the agent is the control method used that has as environment the responses based on performance indexes from a control system loop. With those, the controller assumes a policy of adjusting the tuning parameter or not, and as reward, the indexes achieve the desired status.

Considering that the agent is the controller and the environment is the reference and disturbances received by the control system, RL with repeat and improve is used in this master's thesis based on the *ISE* and *PM*, in order to achieve a better performance and the desired robustness. The idea is to set a target value (Reward) for robustness and performance, and, by offline simulations, reach these parameters with the action addition, calculated based on the step response of the process, as in (100).

$$\left\{ \begin{array}{ll} if(y_{reward\ check}) < 100, & action = 0.001 \\ if(100 < y_{reward\ check}) < 1000, & action = 0.01 \\ if(y_{reward\ check}) > 1000, & action = 0.1 \end{array} \right\} \tag{100}$$

In (100), $y_{reward\ check}$ is the number of terms in the step response of the process. This has been achieved based on the complexity of the systems and using an empirical analysis of how the control actions act in control theory.

The reward ISE also has an analysis based on tests and using some control guidance, as in Äström and Winttenmark (2013). Where it is achieved with (101), when Δy is an output excited by an unitary step u signal and 10 was a constant that empirically brought the desired results. The reward PM is only based on the desired phase margin of the system, which, in many tests, has also adopted the GM value for an acceptable range, and has to be chosen based on reasonable values.

$$Reward \;\; ISE = \frac{\sum_{0}^{y_{reward\ check}} \frac{\Delta y}{\Delta u}}{10\ y_{reward\ check}} \tag{101}$$

The action value is summed to K_{tct} (gain of the tunned control technique) till the stop criteria, rewards, are achieved. K_{tct} is for all the applied techniques, the control weighting factor, which for *PPID* is K_c, for *GMV* is q_0 and for *GPC* is λ. So, when the system obtains the desired result, the parameter is chosen and automatically used in the real-time implementation of the process as a control policy. In (102) the pseudo-code is presented based on Matlab's programming language.

$$\begin{aligned} &While \;\; (ISE > reward\ ISE \;\; || \;\; PM < reward\ PM) \left\{ \;\; K_{dct} = K_{tct} + action \right. \\ &While \;\; (ISE \leq reward\ ISE \;\; \&\& \;\; PM > reward\ PM) \left\{ \;\; K_{tct} = K_{tct} \right. \end{aligned} \tag{102}$$

It is important to mention that this technique is based on mathematical machine effort and has it rewards based on the system model. It can be implemented in an adaptative structure, using each iteration to achieve a better policy, but as presented in Results Chapter, this method uses a lot of machine processing to reach the rewards, so for online applications high-speed microprocessors must be used, as in Lewis and Vrabie (2009) and Vrabie and Lewis (2010) papers.

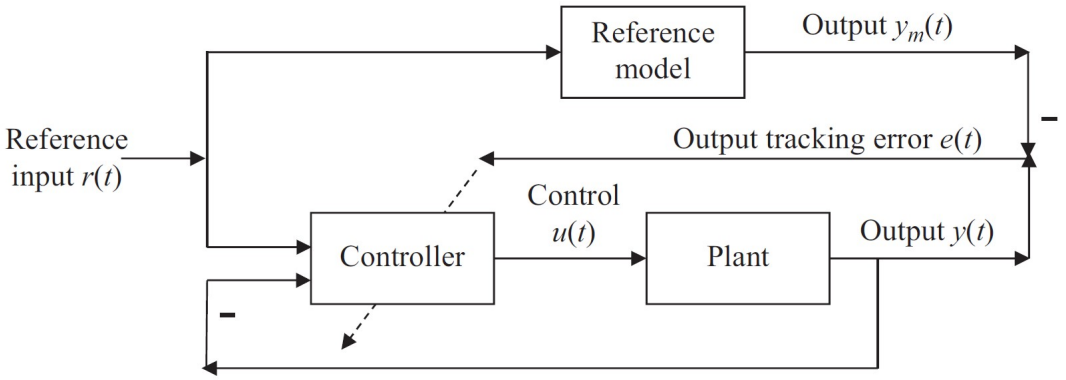
4.2 DIFERENTIAL GAMES METHOD

Vrabie and Lewis (2013) described the integration of optimal control and RL techniques to design adaptive control systems. It is known that this technique explains how optimal control is used to design controllers that provide the best possible control actions for a given system. It is known that optimal control is a mathematical technique that involves finding the control actions that minimize a given cost function while satisfying the system's constraints and is useful when the system's dynamics are well-known and can be modeled accurately (KIRK, 2004).

Furthermore, RL is used when the system's dynamics are uncertain or difficult to model accurately, so basically it is a machine learning technique that involves analysis from trial-and-error interactions with the environment. In RL, an agent interacts with the environment, receiving feedback in the form of rewards or penalties, and learns to take actions that maximize the cumulative reward over time, as presented in the last subsection for the offline loop method. Figure 14 shows how a traditional adaptative

controller works, using a desired output as a reference for the controller, which sends a control signal for the plant and for the identification algorithm. With both outputs, a new estimation of the system is done and a new control signal is calculated based on these new parameters of the process.

Figure 14 – AdaptativeControl Relation Diagram.



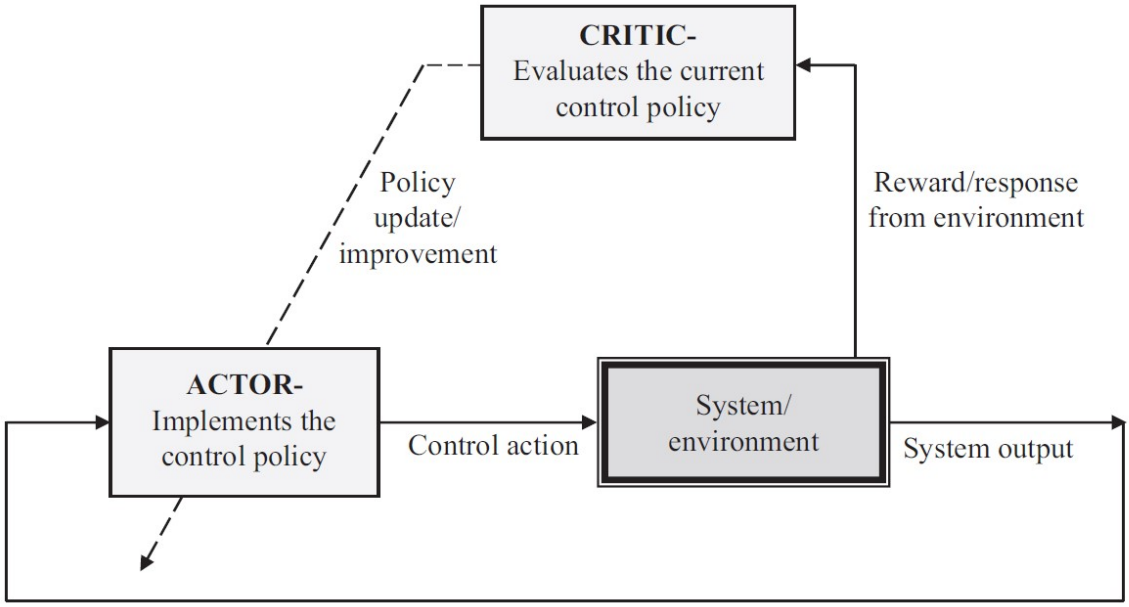
Source: Vrabie and Lewis (2013).

Figure 15 is an illustration of the adaptive control diagram in which the structures provide real-time algorithms, where an actor component applies a control policy or action to the environment and a critic component examines the advantages and disadvantages of that action. The actor-critic structure's learning mechanism consists in a policy evaluation, performed by the critic, and a policy improvement, carried out by the actor (VRABIE; LEWIS, 2013). The policy evaluation step takes place by simply consuming the results of current activities in the environment and determining how similarly beneficial the current action is (SUTTON *et al.*, 1999). Based on the performance evaluation, one of the many methods can be used to change or enhance the control policy and it provides a new value that is better than the old one.

The integration of these two techniques by using RL to adapt the optimal control policies to changing environmental conditions or to unknown system dynamics is proposed in this work to tune the LQR controller. Specifically, the RL will help the controller to learn the optimal control policies and adapt the control policies to changes in the environment. This controller is a modern optimal control technique used to design a linear feedback control law that minimizes a quadratic cost function (BEMPORAD et al., 2002), as shown in Chapter 3. The LQR controller is used as a baseline control strategy because it is a well-known and widely used control technique that provides good performance when the system's dynamics are well-known and can be modeled accurately.

It is important the performance analysis of the RL-based adaptive control strategies with the LQR controller in different scenarios, including uncertain and unknown

Figure 15 – Reinforcement Learning Relation Diagram.



Source: Vrabie and Lewis (2013).

system dynamics, and changing environmental conditions, since the strategies are designed to learn the optimal control policies online and adapt them to the changing environment, while the LQR controller is designed offline based on the known system dynamics and is not adaptive.

It is important to understand that differential games are used as a framework for modeling and analyzing dynamic systems with multiple agents. A differential game is a mathematical framework for studying the interactions between multiple agents or players, where each player tries to optimize its own objective function while taking into account the actions of the other players (VRABIE; LEWIS, 2013). The goal of each player is to find a strategy that maximizes its own objective function while taking into account the strategies of the other players. The RL-based adaptive control strategies learn the optimal control policies online and adapt them to the changing environment, which allows the agents to achieve better performance in the differential games. Therefore, it is possible to understand that differential games are used as a framework for modeling and analyzing dynamic systems with multiple agents and uses RL-based adaptive control strategies to design controllers for the agents in the differential games (VRABIE; LEWIS, 2010).

To equate this proposed system, it must be evaluated that in the linear quadratic (LQ) zero-sum (ZS) game, the algorithm has linear dynamics as proposed in (103).

$$\dot{x} = Ax + Bu + Dw \tag{103}$$

The state vector is $x(t) \in \mathbb{R}^n$, control input $u(t) \in \mathbb{R}^m$ and disturbance $w(t) \in \mathbb{R}^m$.

This system is associated with the infinite-horizon quadratic cost function or performance index, as shown in (104).

$$V(x(t),u,d) = \frac{1}{2}\int_t^\infty (x^T Qx + u^T Ru - \gamma^2 \|d\|^2)d\tau \equiv \int_t^\infty r(x,u,d)d\tau \tag{104}$$

According to Vrabie and Lewis (2013), using the control weighting matrix $R = R^T > 0$ and $\gamma > 0$, it is possible to achieve the control strategy in the LQ ZS game that maximizes the cost associated with the disturbance while minimizing the cost associated with the control, as in (105).

$$V^*(x(0)) = \min_u \max_d J(x(0),u,d) = \min_u \max_d \int_0^\infty (Q(x) + u^T Ru - \gamma^2 \|d\|^2)dt \tag{105}$$

Equation (105) expresses the intent of the control efforts to bring the states to zero while using the least amount of energy, whereas the disturbance aims to move the states away from zero while using the least amount of energy. The state-feedback policies provide the solution to this optimal control problem, as written in (106) and (107), with d being an observer variable used to achieve information for the game (or generalized) algebraic Riccati equation (GARE) equation.

$$u(x) = -R^{-1}B^T Px = -Kx \tag{106}$$

$$d(x) = \frac{1}{\gamma^2}D^T Px = Lx \tag{107}$$

The intermediate matrix P is the solution to the GARE, as shown in (108)

$$0 = A^T P + PA + Q - PBR^{-1}B^T P + \frac{1}{\gamma^2}PDD^T P \tag{108}$$

As proposed by Basar and Olsder (1998), a solution $P > 0$ can be obtained when the rules, proposed as a pseudocode in (109), are followed.

$$P > 0 \;\; \textit{Solution is possible} \left\{ \begin{array}{ll} \textit{if } (A,B) & \textit{is stabilizable} \\ \textit{if } (A,\sqrt{Q}) & \textit{is observable} \\ \textit{if } \gamma > \gamma^* & \end{array} \right\} \tag{109}$$

The GARE equation for the nonnegative definite optimal value kernel $P > 0$ has been solved as part of the ZS game problem. The optimal control is subsequently provided as state variable feedback in terms of the ARE solution (106) and the worst-case disturbance (107). In order to solve the GARE, it is an offline solution approach that requires a thorough understanding of the system dynamics matrices (A, B, D). Also, if the system dynamics (A, B, D) or the performance index (Q, R, G) change while the system is operating, a new optimal control solution must be determined.

For understanding the next steps, it is important to review Markov Decision Processes (MDP), because they provide a framework for studying the Q-Learning RL

method. With the consideration of MDP (X, U, P, R), where X is a set of states and U is a set of actions or controls. The transition probabilities, presented in (110), mathematize for each state of $x \in X$ and action $u \in U$, the conditional probability in (111), in order to transition the states given the MDP is in state x and takes action u (VRABIE; LEWIS, 2013).

$$\rho : X \times U \times X \mapsto [0,1] \tag{110}$$

$$\rho^{u}_{x,x'} = Pr\left\{x'|x,u\right\} \tag{111}$$

With this analysis, the cost function $R : X \times U \times X \mapsto R$ is the expected immediate cost for $R^{u}_{x,x'}$ paid after transition to state $x' \in X$ given that the MDP starts in state $x \in X$ and takes action $u \in U$. Markov property refers to the fact that transition probabilities $P^{u}_{x,x'}$ depend only on the current state x and not on the history of how the MDP attained that state (VRABIE; LEWIS, 2013). For this reason, this method is different from Repeat and Improve previously presented. Convergence and Performance show that, over time, the RL agent may converge to a stable policy that represents an effective controller for the given system (SUTTON *et al.*, 1999), so the performance of the tuned controller is evaluated based on its ability to achieve the desired control objectives and adapt to changes in the environment.

The basic problem for MDP is to find a mapping, that is a deterministic policy to solve the problem, where $\pi : X \times U \mapsto [0,1]$ that gives, for each state x and action u, the conditional probability $\pi(x,u) = Pr\{u|x\}$ of taking action u given that the MDP is in state x. Such a mapping is referred to as a closed-loop control or action strategy or policy, which is stochastic if there is a nonzero probability of selecting more than one control when in state x (VRABIE; LEWIS, 2010). MDPs that have finite state and action spaces are termed finite MDPs, for this reason, there is a convergence for the covariance, Q and R matrices.

Because the systems employed in this master's thesis change in a causal way throughout time, in order to force the MDP to act and change state at nonnegative integer stage values k, we take into account sequential decision issues and impose a discrete stage index k. The phases could be related to time or, more generally, to a series of happenings. The stage value is referred to as the time. Thus, the notion of optimality should be captured in selecting control policies for MDPs. Define a stage cost at iteration k by $r(k) = r(k)[x(k)u(k)x(k+1)]$, then achieving (112) that is rewarded based on (113), where E is the expected value operator and $0 \leq \gamma < 1$ is a discount factor that reduces the weight of costs incurred further in the future (LEWIS; VRABIE, 2009). Furthermore, the discrete matrixes are presented as A_d, B_d and C_d for traditional state space matrixes.

$$R^{u}_{x,x^T} = E\left\{r(k) = x,\, u(k) = u,\; x(k+1) = x^T\right\} \tag{112}$$

$$J = \sum_{k=0}^{k=nit} \gamma^{nit} r(k) \tag{113}$$

Based on (112) and (113), the value of a policy can be defined as the conditional expected value of future cost when starting in state x at k and following policy $\pi(x,u)$, creating (114), where $V(k)$ is known as the value function for policy $\pi(x,u)$, which is the value of being in state x given that the policy is $\pi(x,u)$, where *nit* is the total number of iterations of the simulation, based on the expectation operator.

$$V(k) = E_{\pi}\{J(k)|x(k) = x\} = E_{\pi}\left\{\sum_{k=0}^{k=nit} \gamma^{nit} r(k)|x(k) = x\right\} \tag{114}$$

Considering (113), it is necessary to minimize the expected future costs, and MDP does it with the policy of $\pi(x,u)$ as in (115), what leads $V(k)$ to an optimal policy that corresponds to the value given in (116).

$$\pi^*(x,u) = \underset{\pi}{arg\,min}\ \ V(k) = \underset{\pi}{arg\,min}\ \ E_{\pi}\left\{\sum_{k=0}^{k=nit} \gamma^{nit} r(k)|x(k) = x\right\} \tag{115}$$

$$V_x^*(k) = \underset{\pi}{min}\ V(k) = \underset{\pi}{min}\ E_{\pi}\left\{\sum_{k=0}^{k=nit} \gamma^{nit} r(k)|x(k) = x\right\} \tag{116}$$

Vrabie and Lewis (2010) proposed that an optimal policy has the property that no matter what the previous control actions have been, the remaining controls constitute an optimal policy with regard to the state resulting from those previous controls, based on Bellman's optimality principle. For this reason (116) can be rewritten as (117)

$$V_x^*(k) = \underset{\pi}{min} \sum_{u} \pi(x,u) \sum_{x^T} P_{x,x^T}^{u}[R_{x,x^T}^{u} + \gamma V_x^*(k+1)] \tag{117}$$

Using the MDP presented equations in (105) and assuming Bellman's equation for the Discrete-Time LQR, it leads to a theory to tune LQR in discrete time, where Bellman's equation becomes a Lyapunov equation and (104) changes to (118) (VRABIE; LEWIS, 2013).

$$\begin{aligned} V(k) &= \frac{1}{2}(x^T(k)Q(k)x(k) + u^T(k)Ru(k)) + \frac{1}{2}\sum_{i=k+1}^{nit}(x_i^T Q(k)x_i + u_i^T Ru_i) \\ &= \frac{1}{2}(x^T(k)Q(k)x(k) + u^T(k)Ru(k)) + V(k+1) \end{aligned} \tag{118}$$

Inserting (103), it is possible to rewrite it in terms of the system state equation, as in (119) and assuming the constant, that is, stationary, state feedback policy in (106), for some stabilizing gain K (120) can be achieved.

$$2V(k) = x^T(k)Q(k)x(k)+u^T(k)Ru(k)+[A_dx(k)+B_du(k)]^TP(k)[A_dx(k)+B_du(k)] \quad (119)$$

$$\begin{aligned} 2V(k) &= x^T(k)P(k)x(k) \\ &= x^T(k)Q(k)x(k) + x^T(k)K^TRKx(k) + x^T(k)[A_d - B_dK]^TP(k)[A_d - B_dK]x(k) \end{aligned} \quad (120)$$

Since the performance index is undiscounted, that is, $\gamma = 1$, a stabilizing gain K, that is, a stabilizing policy, must be selected, using Ricatti's equation in (121), presented in discrete mode.

$$[A_d - B_dK]^TP(k)[A_d - B_dK] - P(k) + Q(k) + K^TR(k)K = 0 \quad (121)$$

According to Watkins (1989) and Watkins and Dayan (1992), the conditional value for (117) is (122).

$$Q'[x,u] = \sum_{x^T} P^u_{xx^T}[R^u_{xx^T} + \gamma V^*(k+1)] \quad (122)$$

The Q function is equal to the expected return for taking an arbitrary action u at sample k in state x and thereafter following an optimal policy (WATKINS; DAYAN, 1992). The Q function is a function of the current state x and the action u. In terms of the Q' function, the Bellman optimality equation has a particularly simple form, represented by (123), when the system is represented in Bellman's equation.

$$Q' = \frac{1}{2}(x^T(k)Qx(k) + u^T(k)Ru(k)) + V(k+1) \quad (123)$$

with P being the Riccati solution, yields the Q function for the discrete-time LQR in (124), defined as (126) for kernel matrix S.

$$Q' = \frac{1}{2}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}\begin{bmatrix} A_d^TP(k)A_d + Q(k) & B_d^TP(k)A_d \\ A_d^TP(k)B_d & B^TP(k)B_d + R \end{bmatrix}\begin{bmatrix} x(K) \\ u(K) \end{bmatrix} \quad (124)$$

$$Q'[x(k),u(k)] = \frac{1}{2}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}' S \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}'\begin{bmatrix} S_{xx} & S_{xu} \\ S_{ux} & S_{uu} \end{bmatrix}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \quad (125)$$

Finishing, the control law uses (126) in (124) to reach (127).

$$\frac{\partial Q[x(k),u(k)]}{\partial u(k)} = 0 \quad (126)$$

$$u(k) = -S^{-1}_{u^Tu}S_{ux}x(k) = -(B_d^TP(k)B_d + R)^{-1}B_d^TP(k)A_dx(k) \quad (127)$$

For either the policy iteration or value iteration steps of the latter equation, knowledge of the system dynamics (A, B) is required. On the other hand, (127) requires knowledge only of the Q function matrix kernel (VRABIE; LEWIS, 2010).

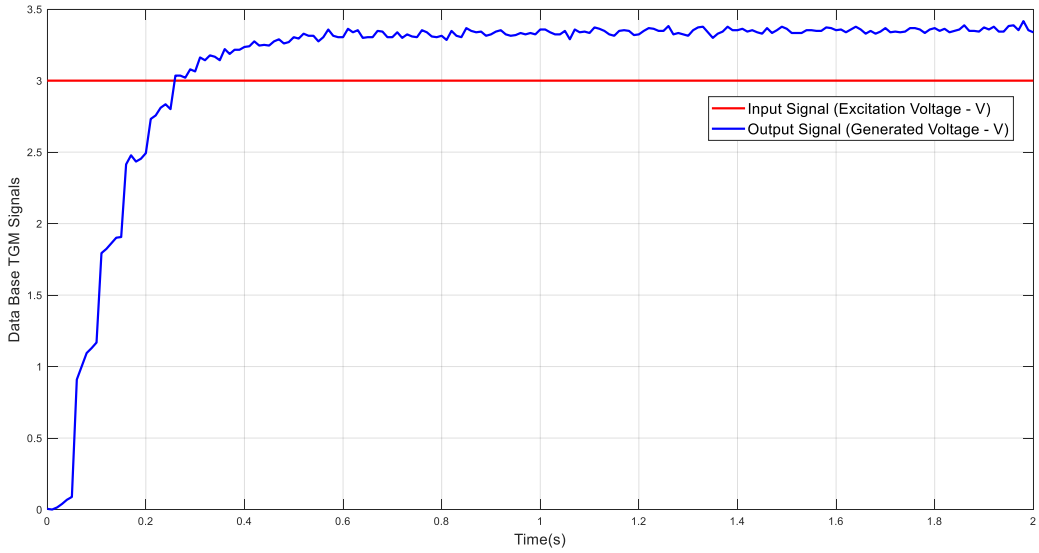
5 RESULTS

5.1 MODELLING AND IDENTIFICATION

5.1.1 Data Aquisition

The systems were modeled based on data files of real tests. TGM and AR drone data have been registered at LACOS and were used for LS estimation as follows in the next subsection. For the TGM process the input signal is the excitation voltage in the first motor and the output signal is the generated voltage by the second motor. In terms of identification, it was used a 3 V input step and achieved a linear output that stabilizes at approximately 3,4 V, with T_S = 0.01 s and nit = 200. TGM output signal when excited with this input is presented in Figure 16.

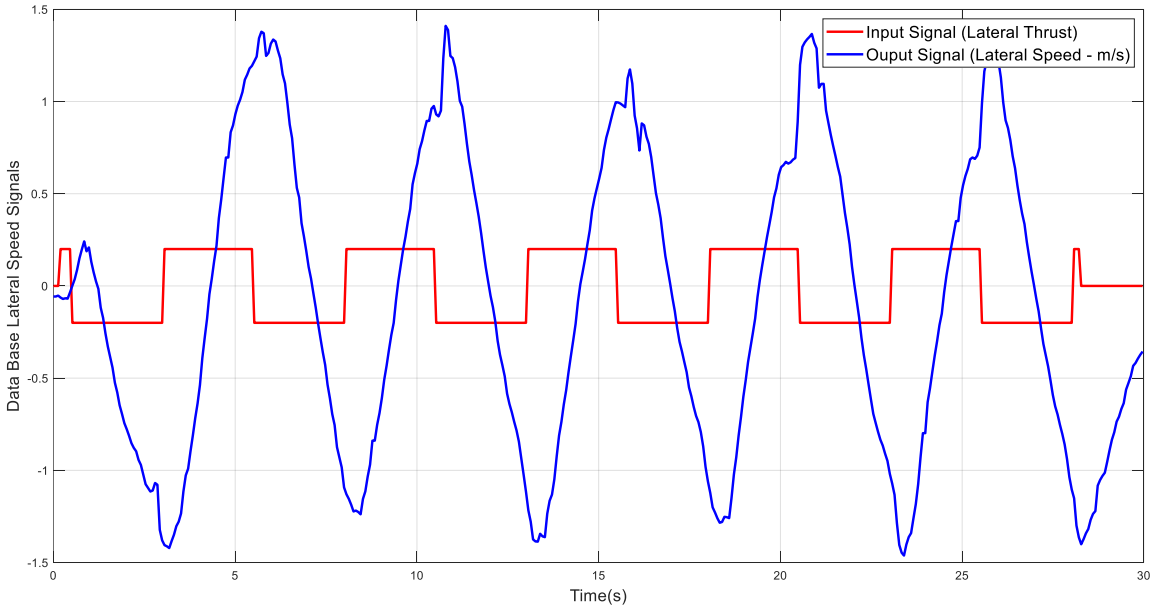
Figure 16 – TGM data for identification.



Source: Author (2023).

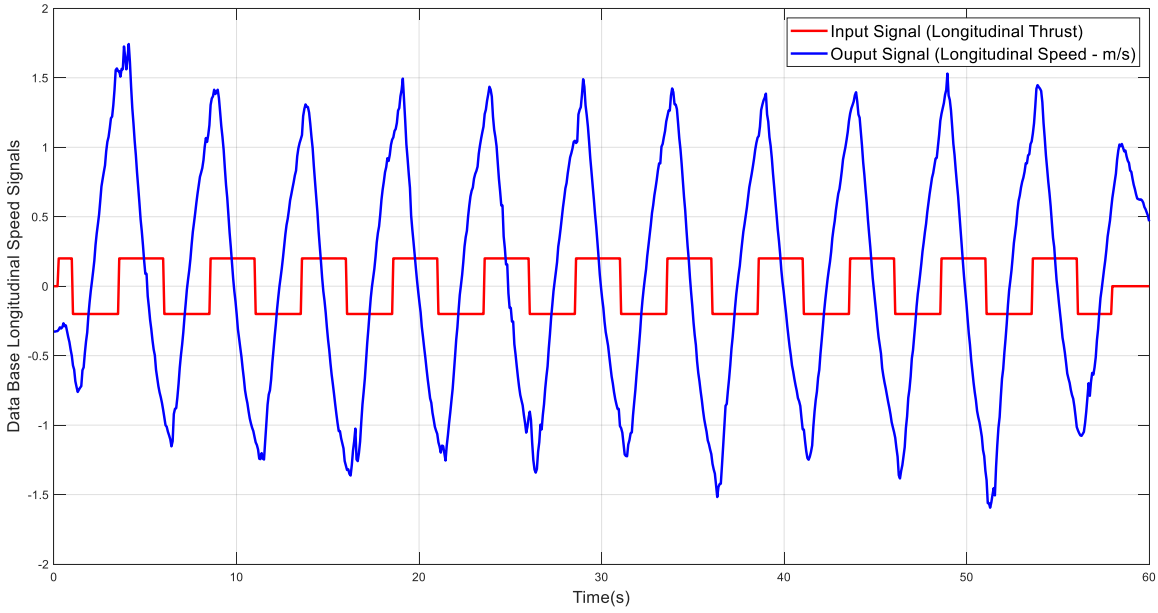
In the Ar Drone process, there are 3 state variables. They were analyzed in SISO and MIMO topologies. The outputs are the lateral speed (m/s), longitudinal speed (m/s) and altitude (m), excited, respectively, with lateral thrust, longitudinal thrust and vertical thrust, with all thrusters being dimensionless and working in the range of $[-1;1] \in \mathbb{R}$. As it is a complex model, the data uses a secure range of outputs, used by LACOS researchers in previous publications, with T_S = 0,065 s and nit = 462, nit = 924 and nit = 2308 for the respectives variables. Those responses are, respectively shown in Figures 17, 18 and 19. The whole dataset is used with a batch approach, seeking the best parameters to evaluate the future control design.

Figure 17 – Lateral Speed data for identification.



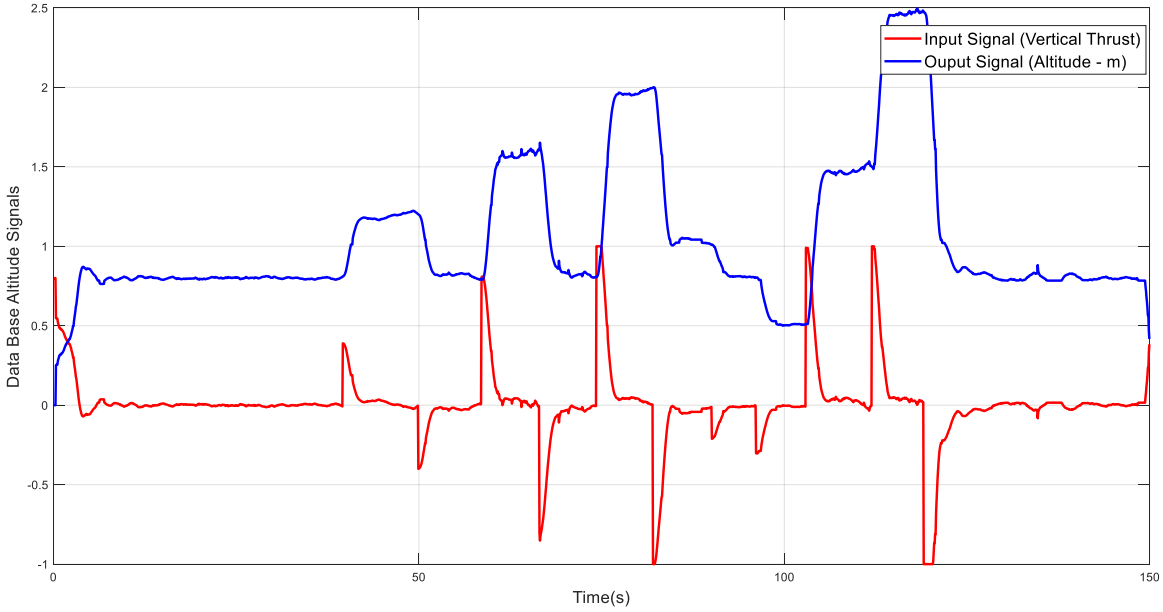
Source: Author (2023).

Figure 18 – Longitudinal Speed data for identification.



Source: Author (2023).

Figure 19 – Altitude data for identification.



Source: Author (2023).

5.1.2 NRLS Polinomial Estimation

In order to design the controllers, the systems are modeled. Using NRLS polynomial estimation theory presented in Chapter 2, TGM, lateral speed, longitudinal speed and altitude models are presented as ARX structure in equations (128), (129), (130) and (131), respectively.

$$y_{TGM}(k) = \frac{(-0.2175 + 0.2964z^{-1})z^{-1}}{1 - 0.8076z^{-1} - 0.1224z^{-2}}u(k) \tag{128}$$

$$y_{LaS}(k) = \frac{(0.0411 + 0.1126z^{-1})z^{-1}}{1 - 1.4557z^{-1} + 0.4626z^{-2}}u(k) \tag{129}$$

$$y_{LoS}(k) = \frac{(0.0025 - 0.1408z^{-1})z^{-1}}{1 - 1.5649z^{-1} + 0.5682z^{-2}}u(k) \tag{130}$$

$$y_{Alt}(k) = \frac{(-0.0053 + 0.0308z^{-1})z^{-1}}{1 - 1.3568z^{-1} + 0.3567z^{-2}}u(k) \tag{131}$$

As a discrete system, the tests are done with the same input vectors used in the database in the difference equation structure presented, respectively, in equations (132), (133), (134) and (135) for validation of the identification stage. Figures 20, 21, 22 and 23 show the respective graphical results of the identification stage with NRLS with a polynomial approach and Table 1 shows the identification indexes for identification evaluation.

$$y_{TGM}(k) = 0.8076y_{TGM}(k-1) + 0.1224y_{TGM}(k-2) - 0.2175u(k-1) + 0.2964u(k-2) \quad (132)$$

$$y_{LaS}(k) = 1.4557y_{LaS}(k-1) - 0.4626y_{LaS}(k-2) + 0.0411u(k-1) + 0.1126u(k-2) \quad (133)$$

$$y_{LoS}(k) = 1.5649y_{LoS}(k-1) - 0.5682y_{LoS}(k-2) + 0.0025u(k-1) - 0.1408u(k-2) \quad (134)$$

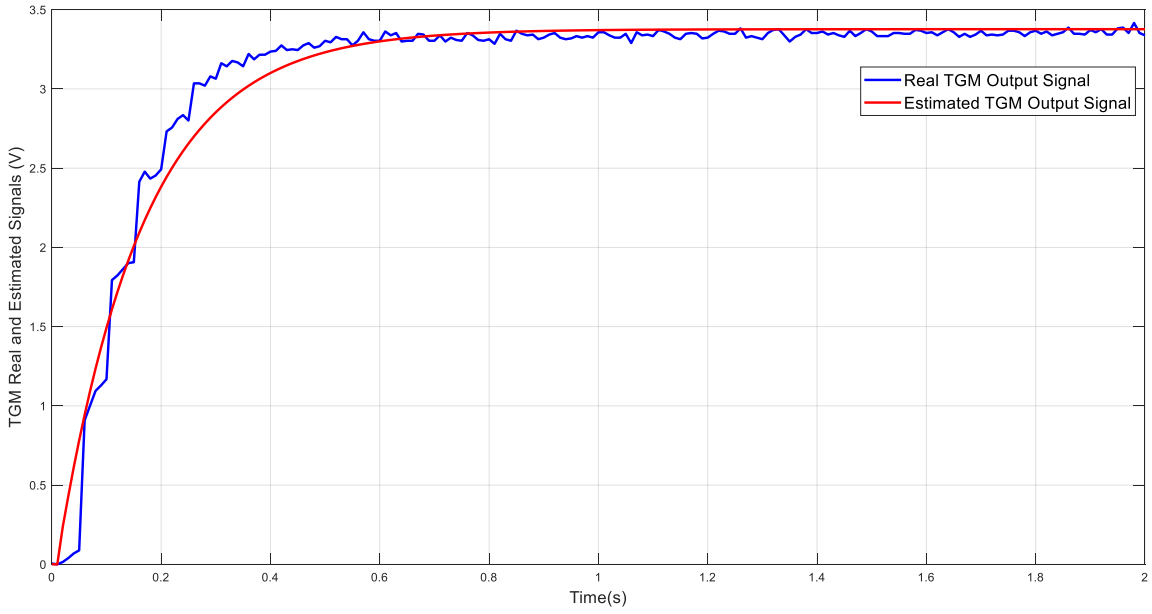
$$y_{Alt}(k) = 1.3568y_{Alt}(k-1) - 0.3567y_{Alt}(k-2) - 0.0053u(k-1) + 0.0308u(k-2) \quad (135)$$

Table 1 – NRLS identification indexes

Identification Indexes	TGM	Lateral Speed	Longitudinal Speed	Altitude
J_{NRMSE}	74,1314%	63,4499%	76,1247%	62,1472%
R^2	0,8333	0,7521	0,8412	0,7314

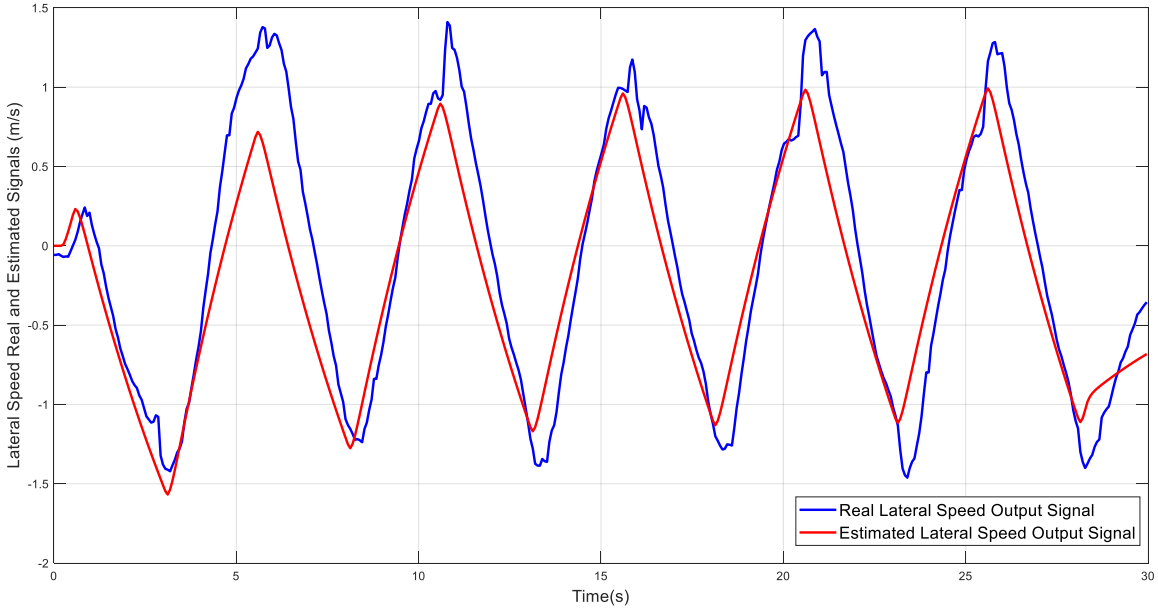
Source: Author (2023).

Figure 20 – TGM NRLS identification output signals.



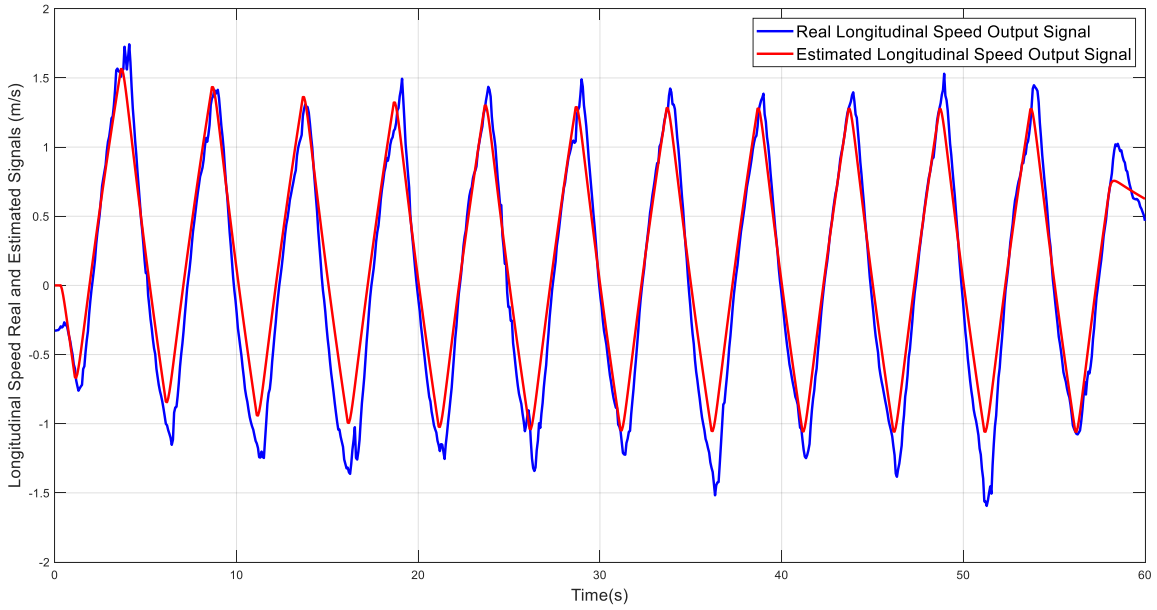
Source: Author (2023).

Figure 21 – Lateral Speed NRLS identification output signals.



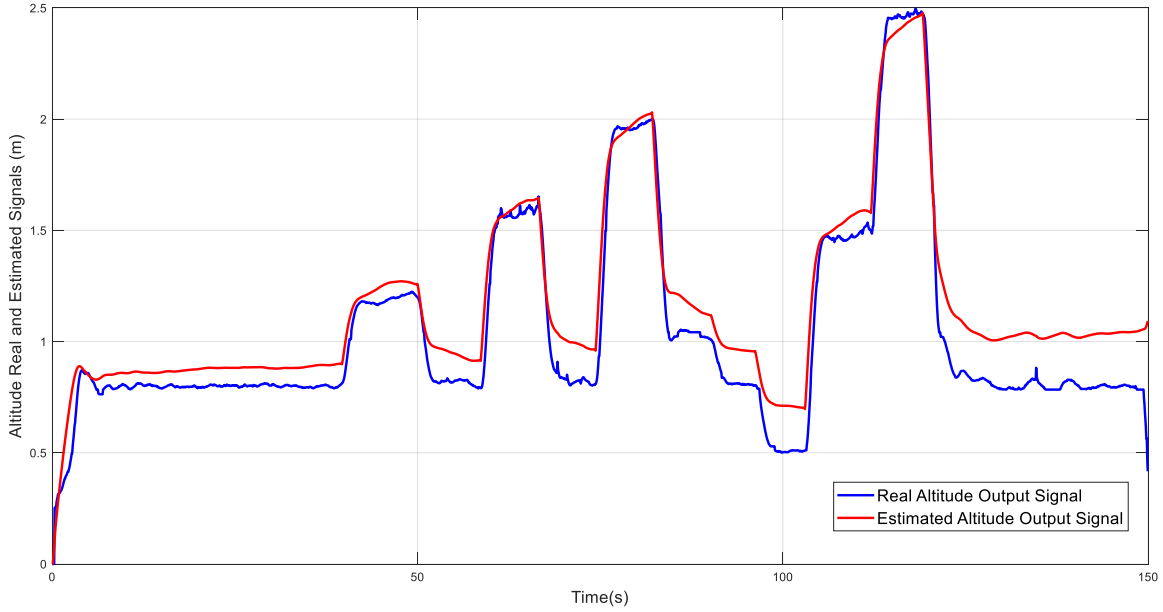
Source: Author (2023).

Figure 22 – Longitudinal Speed NRLS identification output signals.



Source: Author (2023).

Figure 23 – Altitude NRLS identification output signals.



Source: Author (2023).

5.1.3 NRLS Space State Estimation

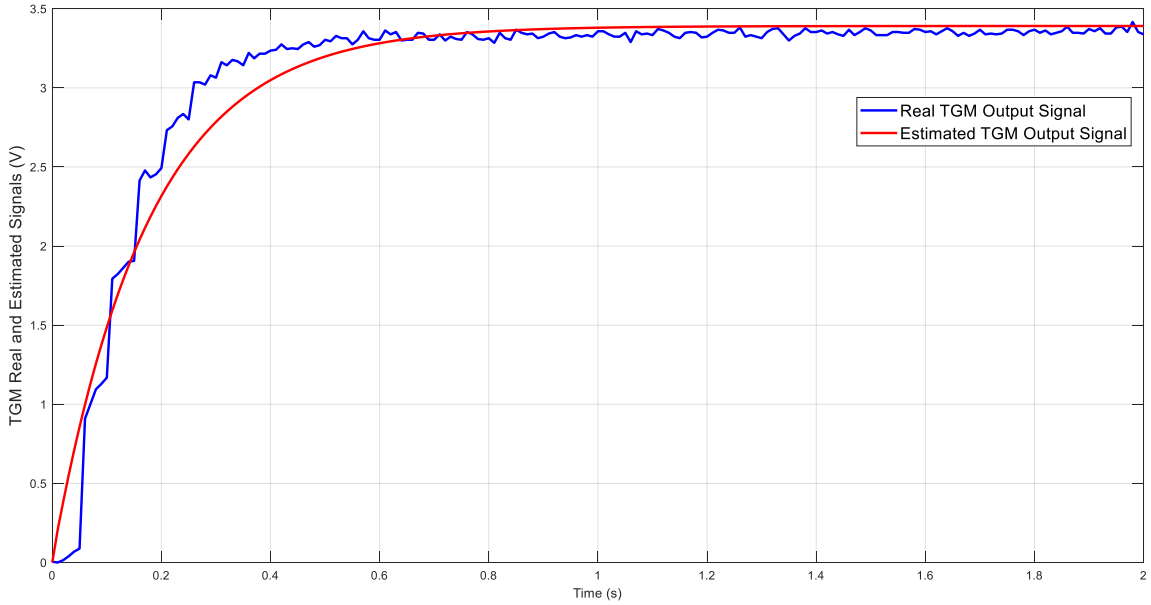
In this approach, both systems are modeled. TGM is the first, with the first state (x_1) being it voltage output and the second state (x_2) obeying the relation in equation (12). The estimation is similar as the polynomial approach, but achieving a model as a state space representation similar to Figure 5 and equationaly represented as (10). The estimated state equation is presented in (136) and the output state is x_1, in order to achieve that (137) has been created. Figures 24 and 25 show x_1 and x_2 estimations, respectively.

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} 0.9387 & -0.0010 \\ -6.1335 & -0.0977 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.0693 \\ 6.9331 \end{bmatrix} u(k-1) \tag{136}$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) + 0 \tag{137}$$

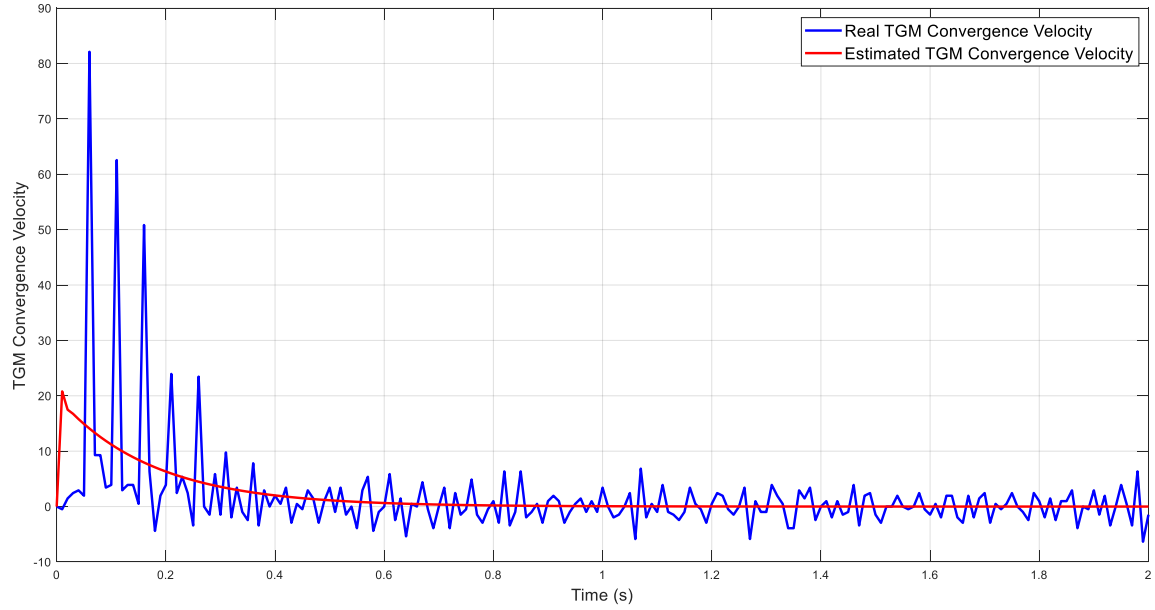
In the AR drone model estimation, x_1, x_2, x_3, x_4 and x_5 states are, respectively, lateral speed (m/s), longitudinal speed (m/s), altitude (m), longitudinal acceleration (m/s²) and lateral acceleration (m/s²), because x_4 and x_5 states obey the relation in (12) for x_1 and x_2, respectively. The estimated state equation is presented in (138), in order to achieve that (139) has been created. Figures 26 to 30, shows x_1 to x_5 estimations, respectively, using the 30 initial seconds of each database response.

Figure 24 – TGM NRLS output signal estimation in state space.



Source: Author (2023).

Figure 25 – TGM NRLS convergence velocity signals estimation in state space.

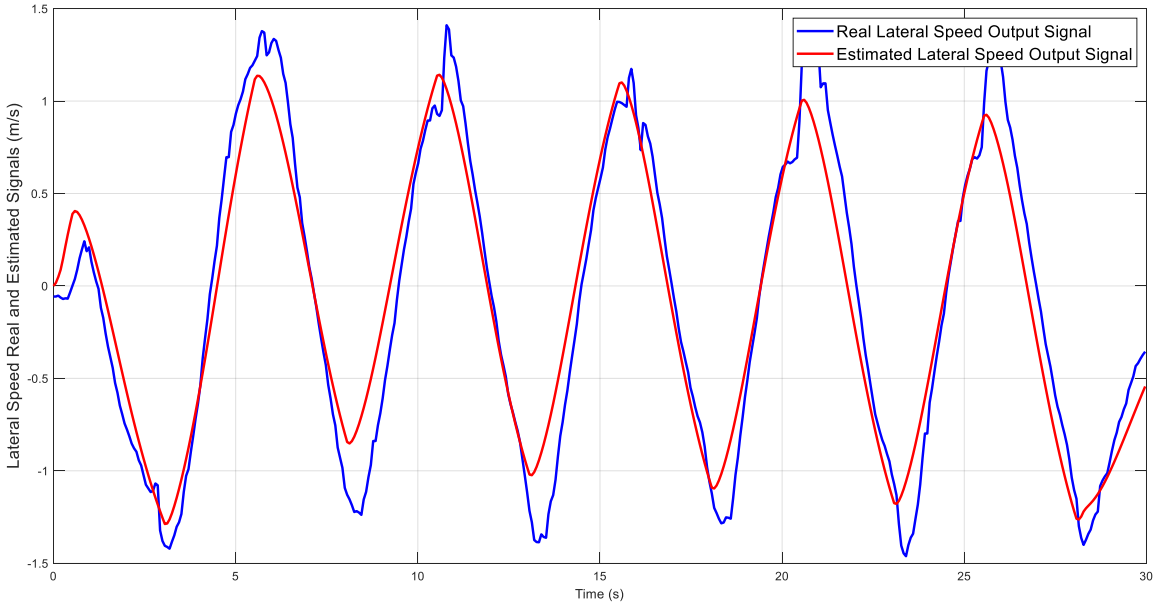


Source: Author (2023).

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} = \begin{bmatrix} 1.0142 & 0.0245 & -0.0047 & 0.0298 & -0.0080 \\ -0.0855 & 0.9272 & 0.0286 & 0.0027 & 0.0288 \\ 0.0019 & 0.0023 & 1.0003 & -0.0005 & 0.0011 \\ 0.2189 & 0.3765 & -0.0723 & 0.4581 & -0.1238 \\ -1.3148 & -1.1194 & 0.4399 & 0.0412 & 0.4429 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_4(k-1) \\ x_5(k-1) \end{bmatrix} + \begin{bmatrix} 0.0882 & 0.0002 & 0.0214 \\ -0.0336 & -0.0013 & -0.0779 \\ 0.0073 & -0.0000 & 0.0352 \\ 1.3565 & 0.0030 & 0.3299 \\ -0.5172 & -0.0204 & -1.1985 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \\ u_3(k-1) \end{bmatrix} \tag{138}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x(k) \tag{139}$$

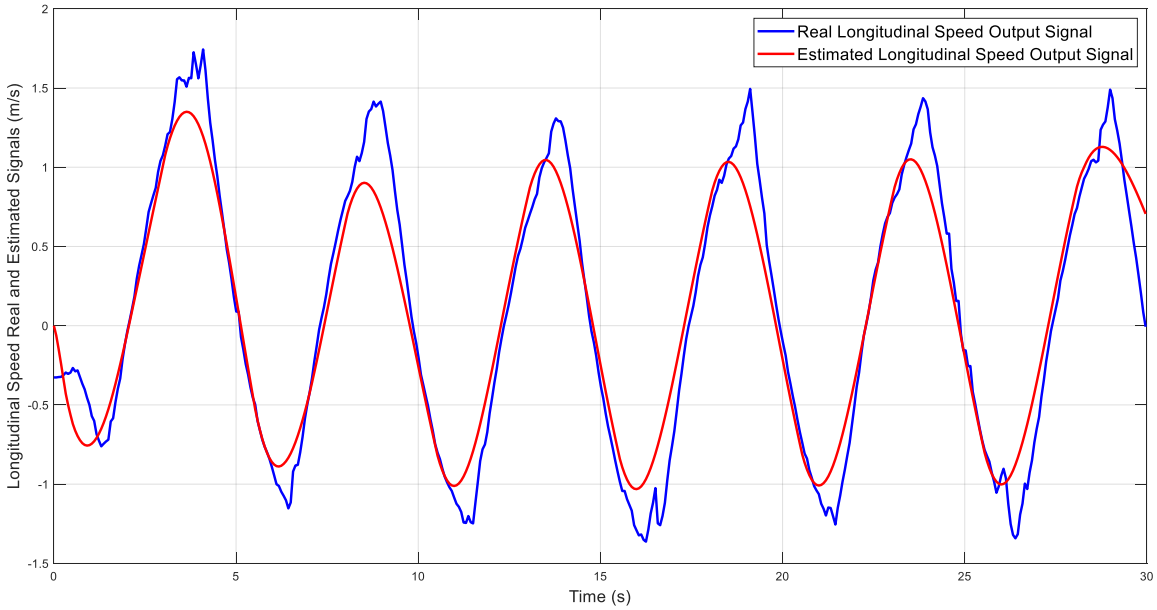
Figure 26 – Lateral Speed NRLS output signal estimation in state space.



Source: Author (2023).

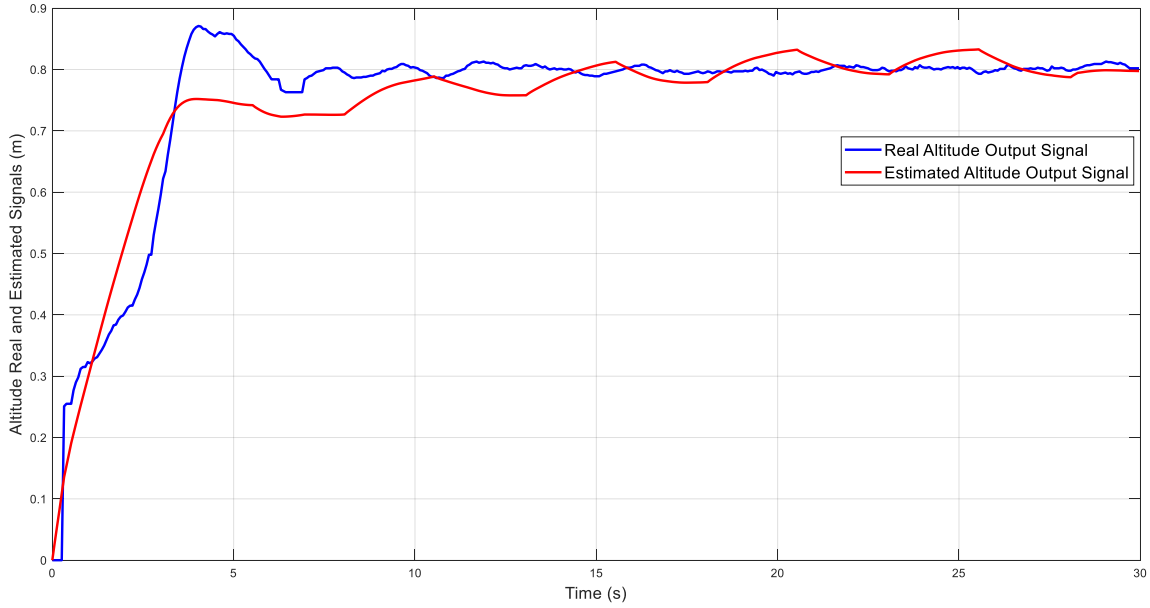
Table 2 present the estimation indexes for SS approach, the indexes achieved quantify a closer model to the real signals, less in altitude system. However, the intention is not to compare the methods but to identify different ways for use in control algorithms.

Figure 27 – Longitudinal Speed NRLS output signal estimation in state space.



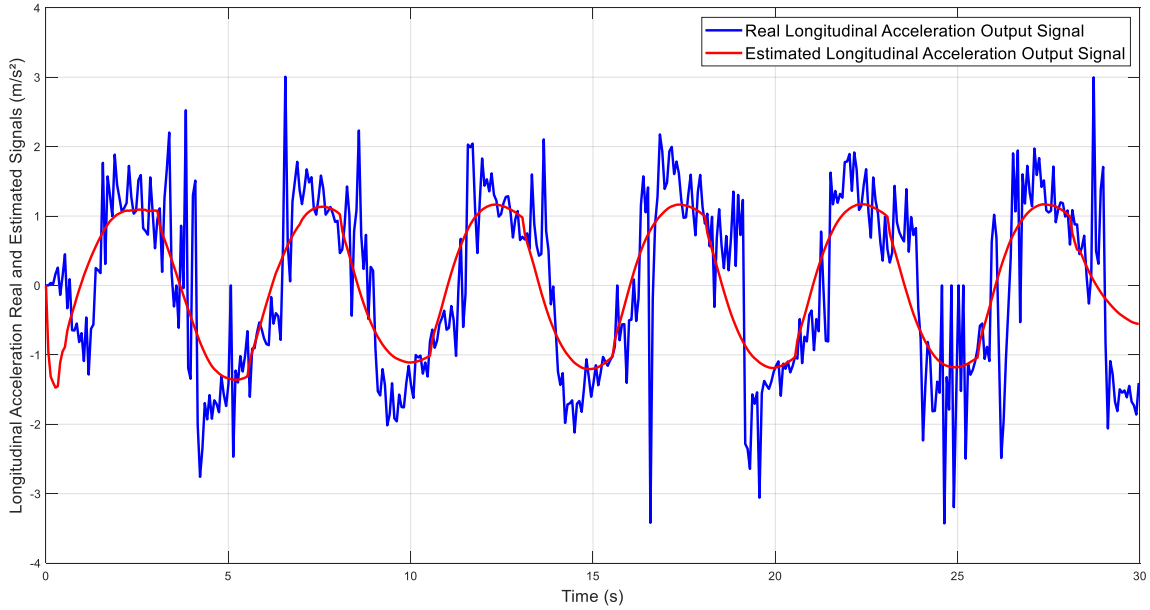
Source: Author (2023).

Figure 28 – Altitude NRLS output signal estimation in state space.



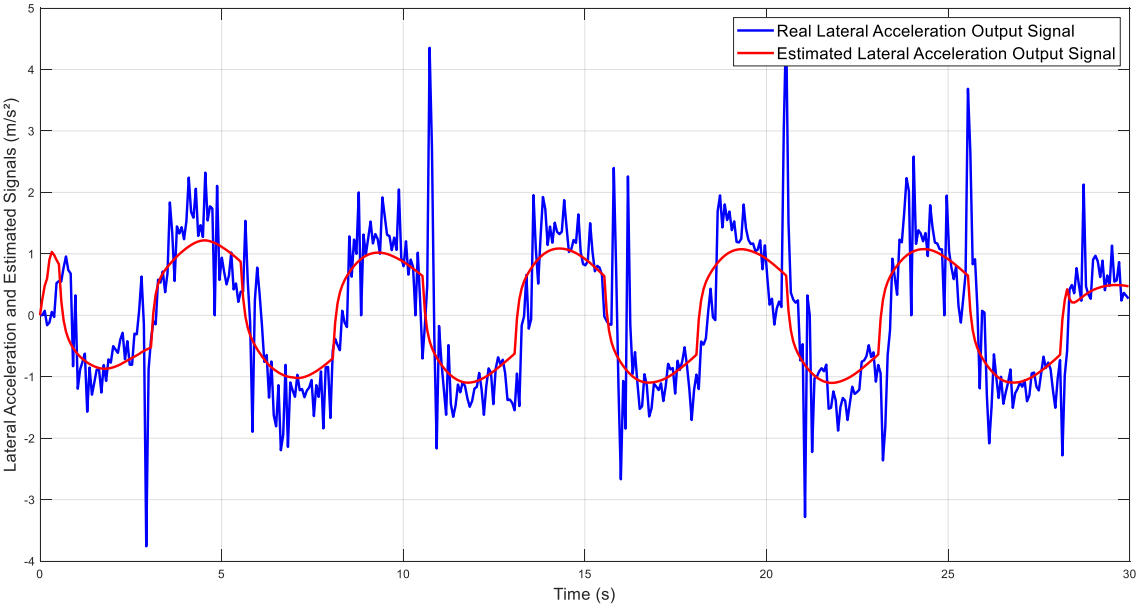
Source: Author (2023).

Figure 29 – Lateral Acceleration NRLS output signal estimation in state space.



Source: Author (2023).

Figure 30 – Longitudinal Acceleration NRLS output signal estimation in state space.



Source: Author (2023).

Table 2 – NRLS estimation in state space indexes.

Identification Indexes	TGM	Lateral Speed	Longitudinal Speed	Altitude	Lateral Acceleration	Longitudinal Acceleration
J_{NRMSE}	75.1723%	68.1742%	78.8471%	60.3214%	57.1471	57.2641
R^2	0.8417	0.7987	0.8614	0.7314	0.6941	0.6957

Source: Author (2023).

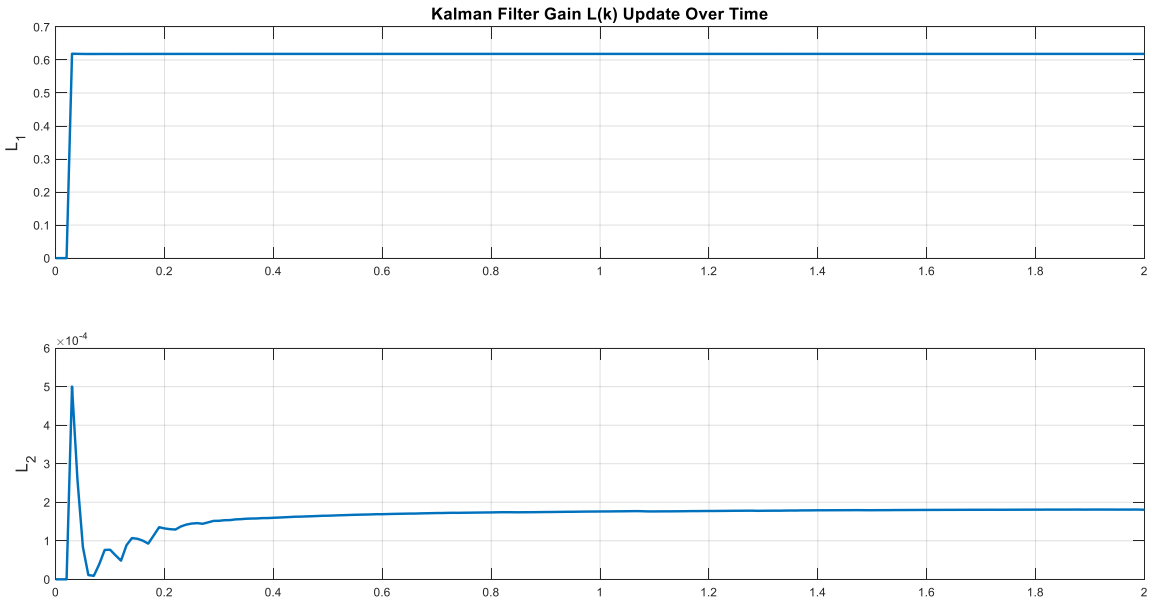
5.1.4 OKID Estimation

OKID method achieves a model that takes into consideration the noises of the system data. With it, the TGM state equation identified, based on the presented equations of Chapter 2, is Figure (140). In order to achieve better results, the gains are improved for the first state, but the observer gains are calculated for both, as represented in Figure 31. Figure 32 shows the OKID response as an identification method. As the second state has such a small weight factor for the observer, its identification will not be shown.

$$\begin{aligned}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = &\begin{bmatrix} 0.9321 & -0.0011 \\ -3.2101 & -0.0095 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.0765 \\ 3.6042 \end{bmatrix} u(k-1) \\ &+ \begin{bmatrix} 0.0039 & -0.0000 \\ -0.8031 & 0.0088 \end{bmatrix} \begin{bmatrix} w_1(k-1) \\ w_2(k-1) \end{bmatrix}\end{aligned} \tag{140}$$

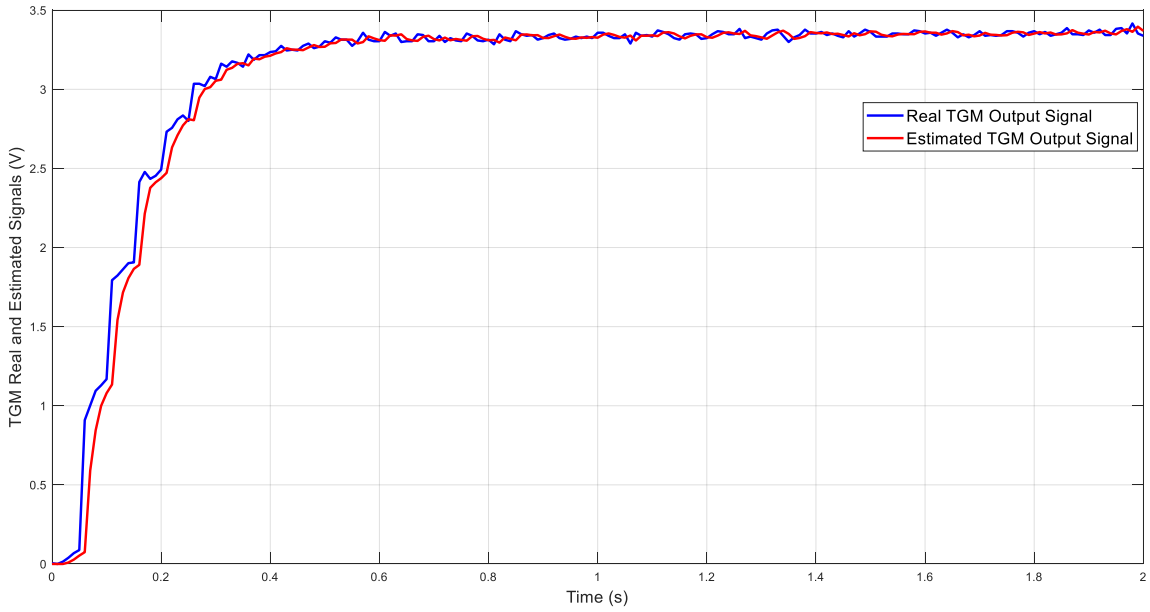
For the AR drone estimation, x_1, x_2, x_3, x_4 and x_5 states are estimated with the same structure of SSNRLS. The achieved state equation is presented in (141), the output is similar to NRSL in (139). The observer gains are presented in 33. Figures 34 to 38, shows x_1 to x_5 estimations, respectively, using the 30 initial seconds of each database response.

Figure 31 – Observer gains adaptation for TGM OKID.



Source: Author (2023).

Figure 32 – TGM OKID output signal estimation.



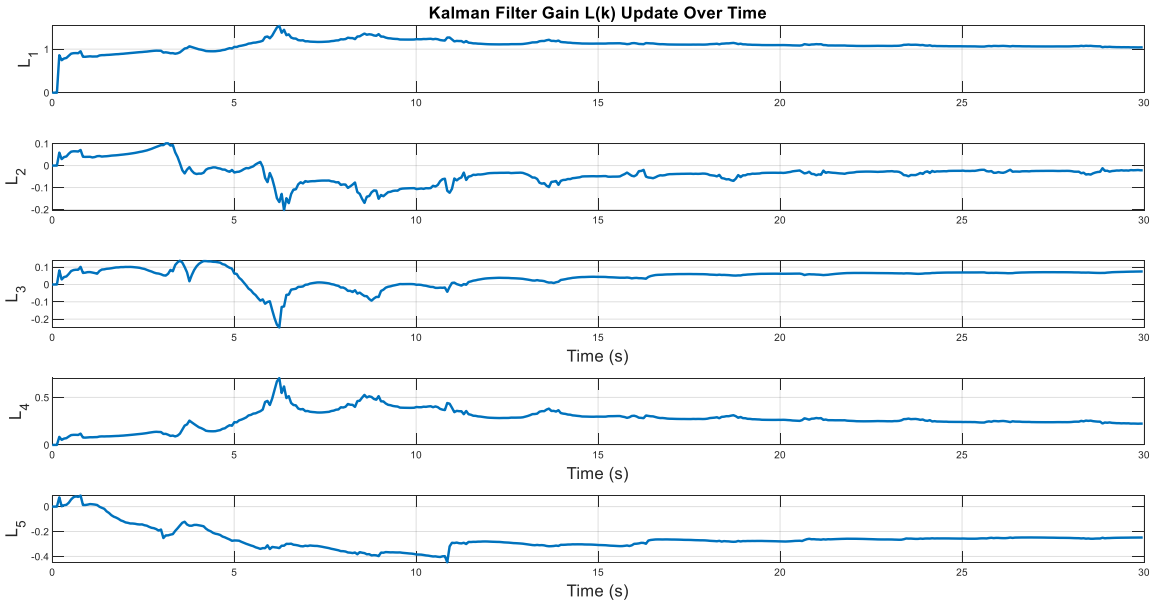
Source: Author (2023).

$$
\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} = \begin{bmatrix} 0.9805 & -0.0047 & 0.0005 & 0.0303 & -0.0049 \\ -0.0283 & 0.9702 & -0.0004 & 0.0012 & 0.0408 \\ 0.0061 & 0.0066 & 0.9999 & -0.0017 & 0.0007 \\ -0.2901 & -0.0646 & 0.0072 & 0.4650 & -0.0734 \\ -0.4270 & -0.4516 & -0.0062 & 0.0168 & 0.6285 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_4(k-1) \\ x_5(k-1) \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0.1097 & 0.0817 & -0.0058 \\ -0.0037 & -0.0736 & -0.0193 \\ 0.0051 & -0.0054 & 0.0475 \\ 1.6873 & 1.2512 & -0.0827 \\ -0.0575 & -1.1356 & -0.2904 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \\ u_3(k-1) \end{bmatrix} \tag{141}
$$

$$
+ \begin{bmatrix} 0.0414 & 0.1135 & 0.0553 & -0.0112 & -0.0065 \\ 0.0737 & 0.0096 & 0.0061 & -0.0032 & -0.0179 \\ -0.0053 & 0.0713 & -0.0319 & 0.0020 & -0.0052 \\ 0.6214 & 1.7402 & 0.8505 & -0.1712 & -0.1019 \\ 1.1258 & 0.1435 & 0.0924 & -0.0481 & -0.2763 \end{bmatrix} \begin{bmatrix} w_1(k-1) \\ w_2(k-1) \\ w_3(k-1) \\ w_4(k-1) \\ w_5(k-1) \end{bmatrix}
$$

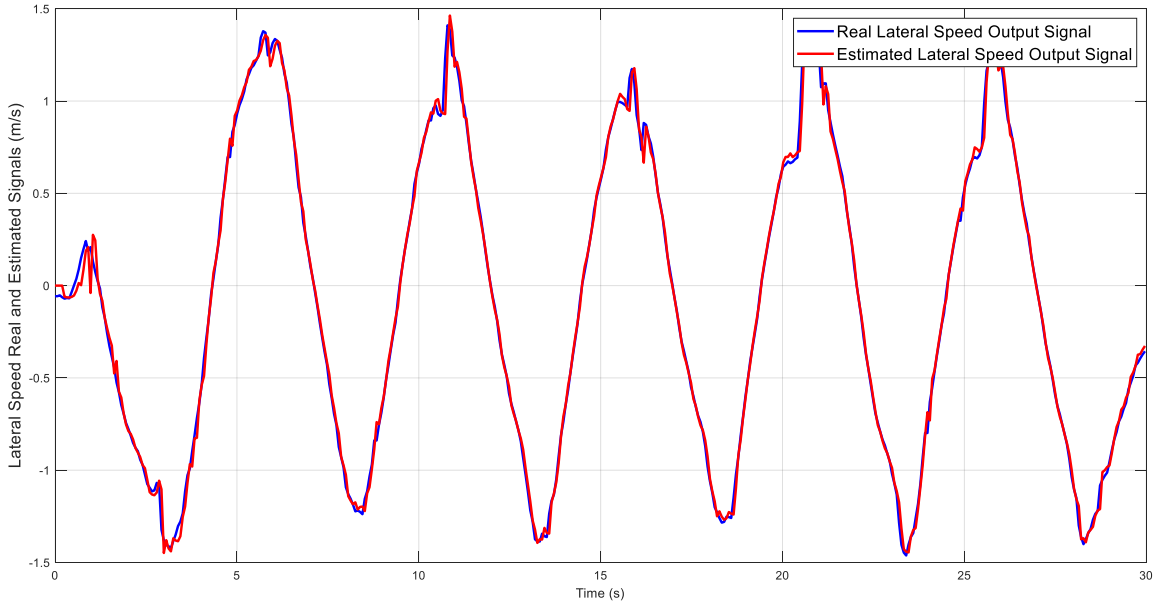
Figure 33 – Observer gains adaptation for Ar Drone OKID.



Source: Author (2023).

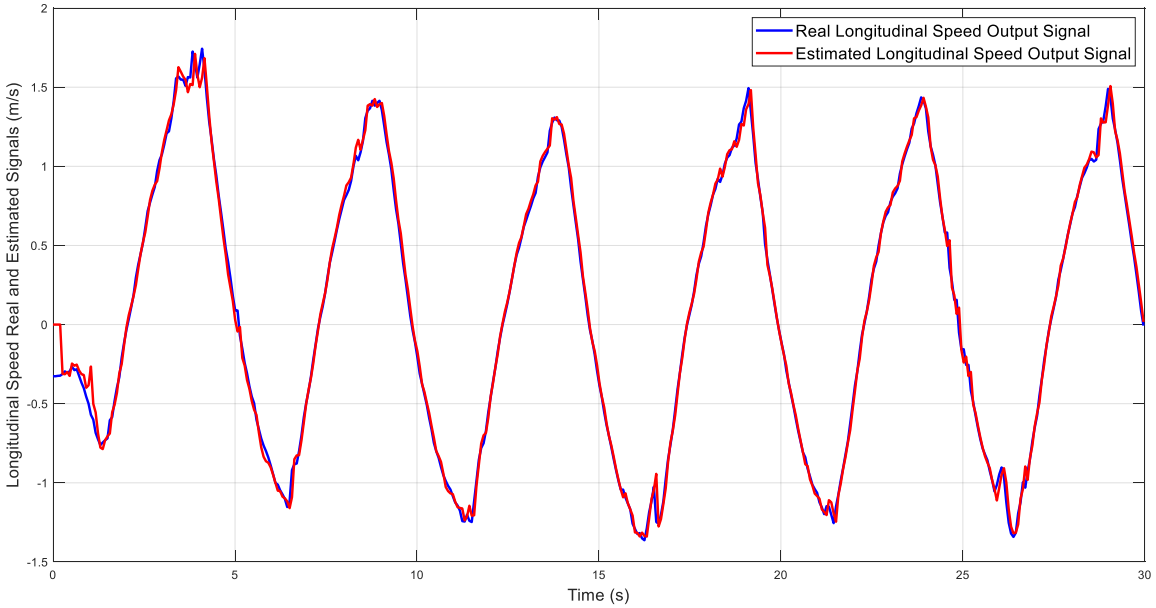
Table 3 presents the estimation indexes for SS OKID approach, confirming a closer match to the registered data when compared to Table 2 results.

Figure 34 – Lateral Speed OKID output signal estimation.



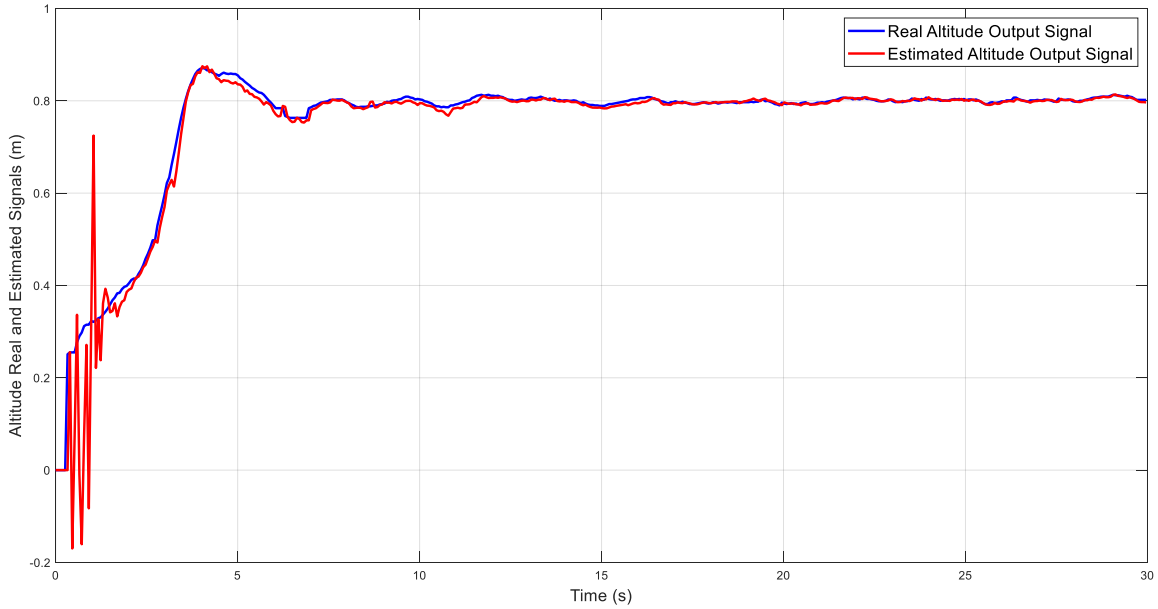
Source: Author (2023).

Figure 35 – Longitudinal Speed OKID output signal estimation.



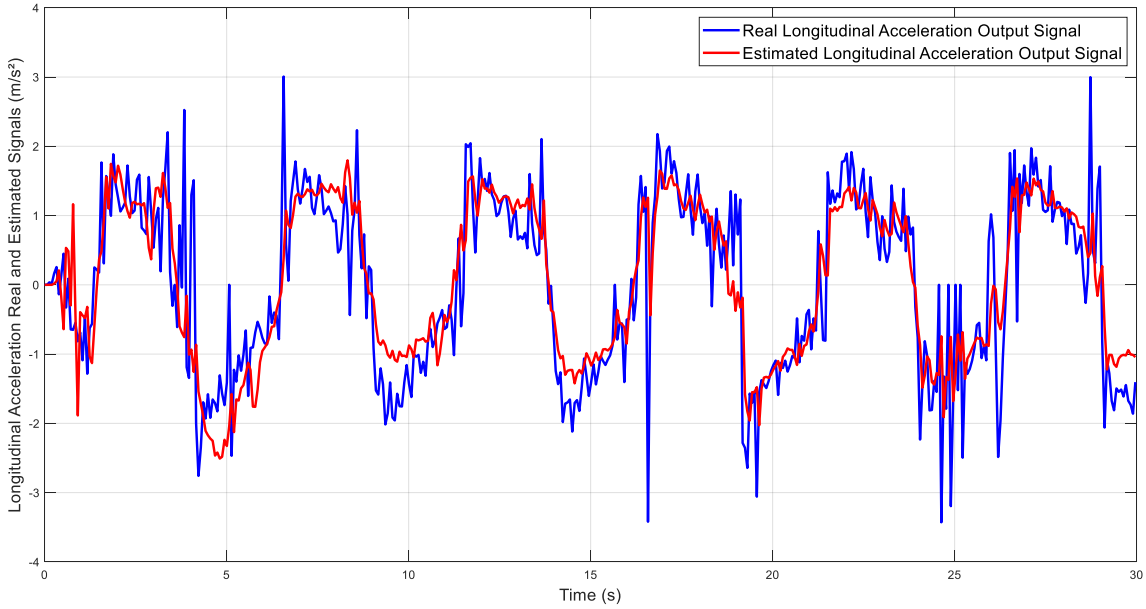
Source: Author (2023).

Figure 36 – Altitude OKID output signal estimation.



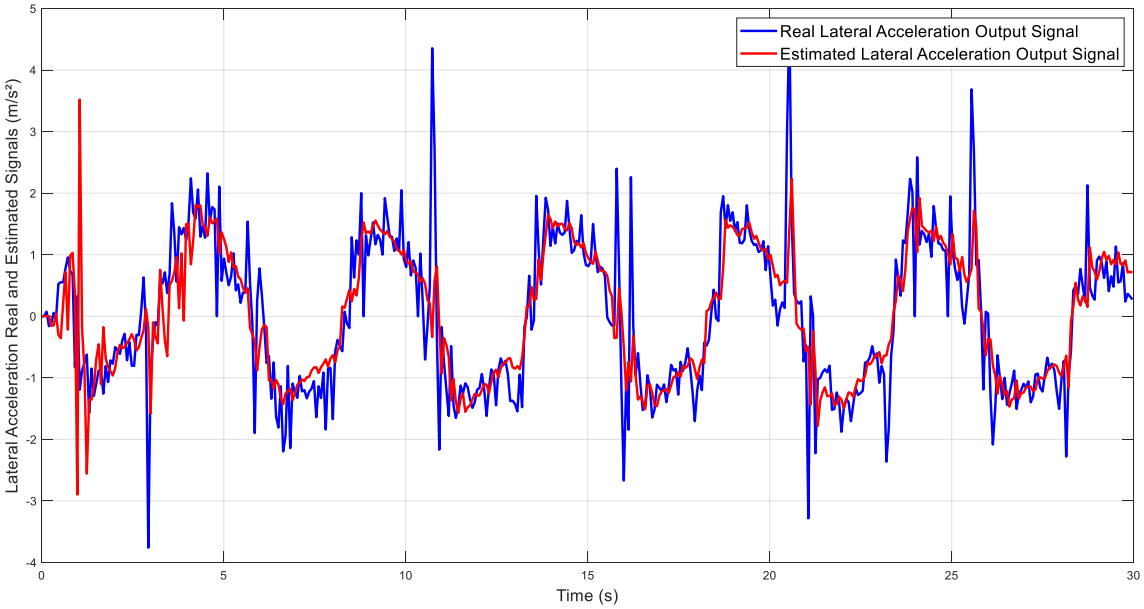
Source: Author (2023).

Figure 37 – Lateral acceleration OKID output signal estimation.



Source: Author (2023).

Figure 38 – Longitudinal Acceleration OKID output signal estimation.



Source: Author (2023).

Table 3 – OKID estimation in state space indexes.

Identification Indexes	TGM	Lateral Speed	Longitudinal Speed	Altitude	Lateral Acceleration	Longitudinal Acceleration
J_{NRMSE}	85.2347%	95.2471%	89.1471%	82.2493%	77.5547	76.1723
R^2	0.8417	0.9214	0.8921	0.8213	0.7914	0.7992

Source: Author (2023).

5.2 CONTROL

The control results are presented with simulated and real tests results. For TGM, all tests have been implemented using daqduino for serial communication between Arduino and Matlab softwares, with 9600 bits/s Baud Rate and Arduino UNO as controller. The idea of using RL is for achieving the desired parameter for each control with only mathematical effort, after this machine learning, the achieved parameter is used as the chosen parameter to be implemented in the control algorithm that is used in the system.

As a method that uses much computer processing, it implementation with non-adaptative control is reasonable since the tests achieve the parameter and are used for all reference tracking signals based on a model previously identified. However, for adaptative control, the improve and repeat method needs to be implemented in high-speed microprocessors, since the RL algorithm achieves in each sample time one control action based on the identified model. For simulated tests, as the AR Drone

results, the idea is to keep the responses as close as the real tests. To achieve this, white Gaussian noises were used based on the estimation error of NRLS, using this difference as a vector amplitude in all tests for respective variables.

PPID, GMV and GPC controllers are used in TGM, lateral speed, longitudinal speed and altitude SISO polynomial NRLS identified transfer functions. For initial conditions, TGM real implementations tests have 3 amplitude reference signals with 3 seconds of duration each, those are 0.5 V, 2.5 V and 1.5 V; the control signal has a minimum value of 0 V and a maximum value of 5 V for Arduino limitations; the PM goal is 60º. Have been assumed the following conditions for testing the AR drone: The reference vector has 1 m/s as amplitude for lateral and longitudinal speeds, and for altitude, it has 0.5 m, the control signal has a minimum value of -1 and a maximum value of 1, with the respective direction thrusts for each system, the PM goal is 45º for all drone variables, with a 60 seconds test and 0.065 s as sample time.

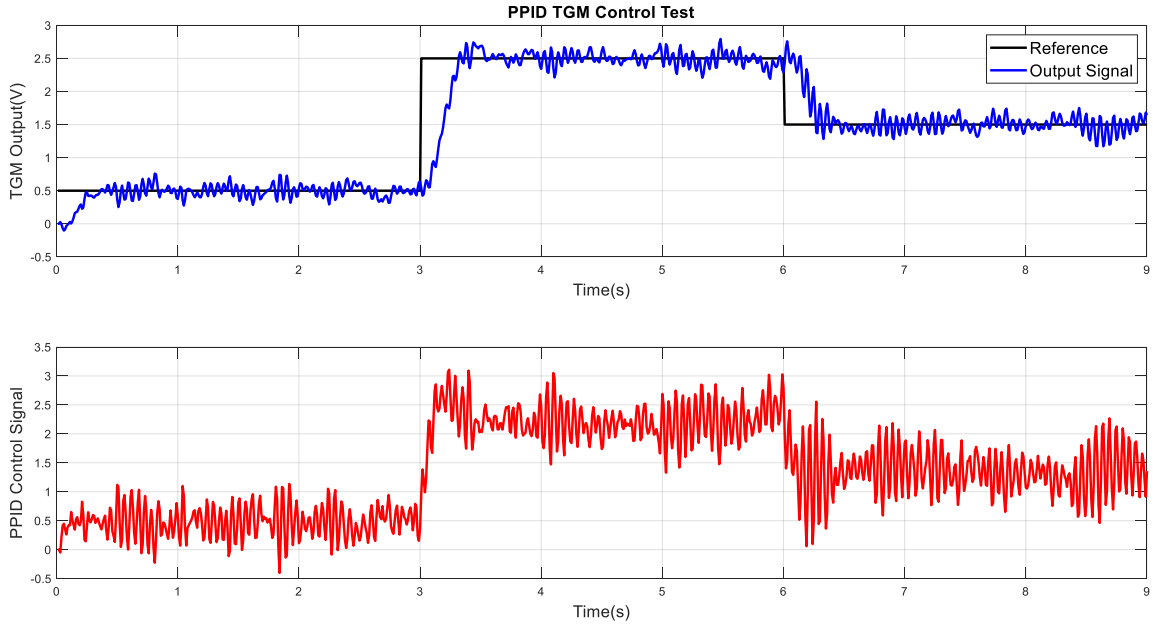
For the LQR, the vector of references is null, since the implementation obtained was for the first case proposed in Vrabie and Lewis (2010), however, the results were obtained in the final periods of writing with non-null references and using the Kalman filter, making an LQG, but this will be used in future works and publications. LQR has an appliance into drone landing and in future real tests will also be implemented, as initial conditions, the system starts with lateral and longitudinal speed of 1 m/s and altitude of 0.5 m, and has to reach the ground with null values. Traditional LQR and RL LQR have 12 seconds time duration and 0.065 s as sample time.

It is important to mention that the tests have been done with an Intel i5 processor, with Nvidia RTX 3050 and 16 GB Ram, with 46 iterations per second for the PPID controller, 27 iterations per second for the GMV controller and 11 iterations per second for the GPC controller, what already leads an evaluation that the more complex is the control structure algorithm more time it takes to add a RL control action.

5.2.1 PPID Control

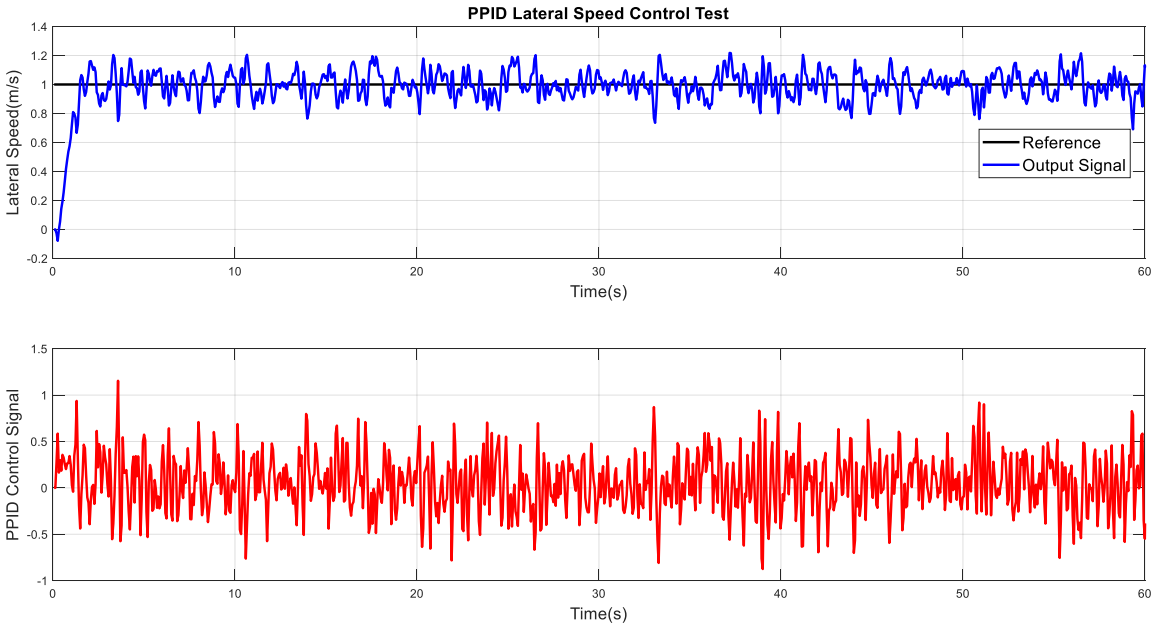
PPID control technique obtained the results (reference tracking and control signal) presented in Figures 39 to 42, where the outputs followed the references, even with the noises of the process. Figure 43 shows the four sensitivity functions plots, where it can be proved the target PM of the controlled systems. Figures 44 to 47 show the convergence of performance and robustness indexes among the iterations. Table 4 shows the final iteration PPID indexes. This method presented fast tuning when implemented with RL, given that it is a method with less mathematical structure, it makes resolving calculations for software to respond to implemented actions faster. It is worth noting that the tests show that, for the same model, the PPID uses lower gains to synchronize the controller, which is also one of the reasons for the speed of convergence of the indices evaluated.

Figure 39 – TGM PPID experimental control responses.



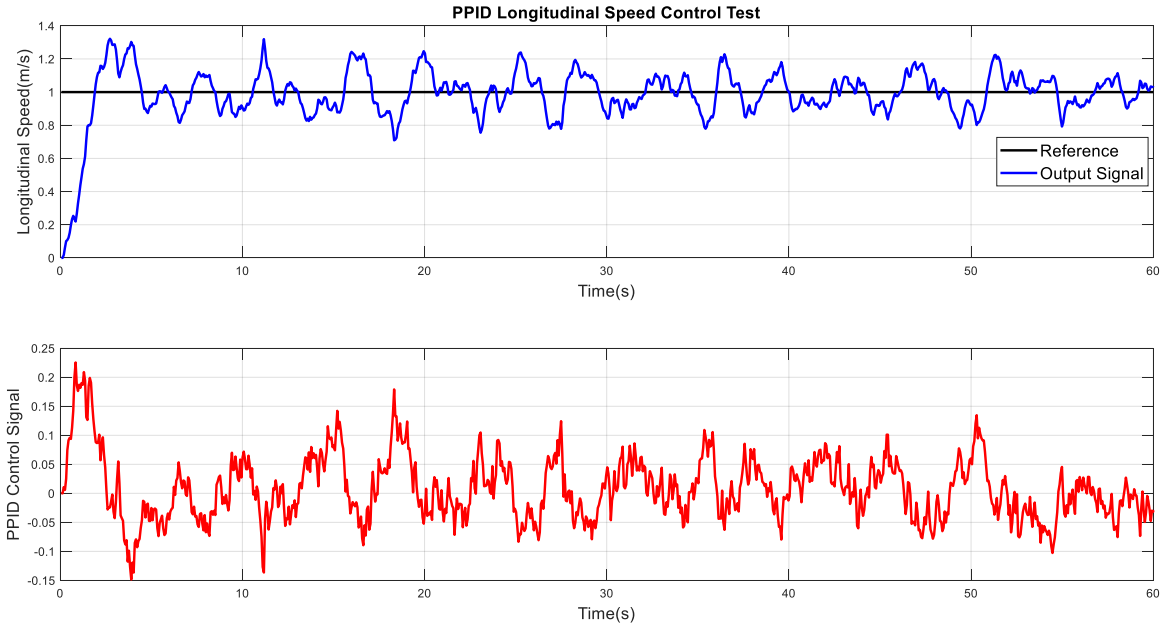
Source: Author (2023).

Figure 40 – Lateral Speed PPID control responses.



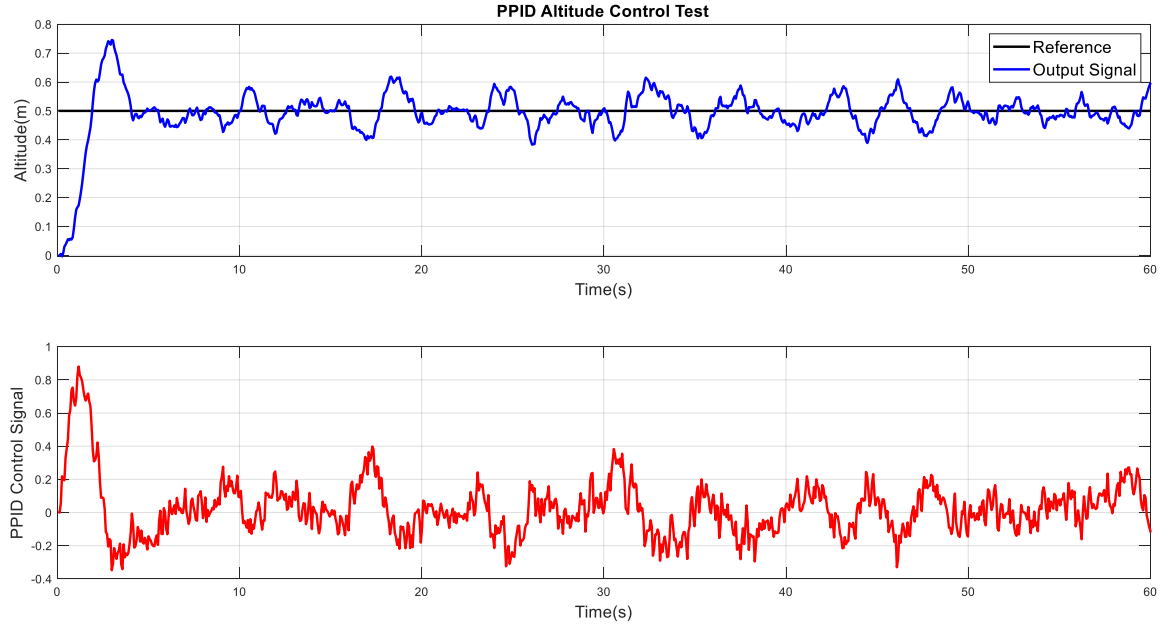
Source: Author (2023).

Figure 41 – Longitudinal Speed PPID control responses.



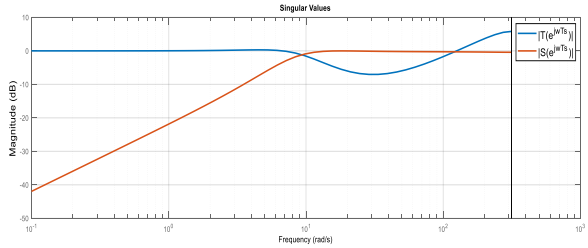
Source: Author (2023).

Figure 42 – Altitude PPID control responses.

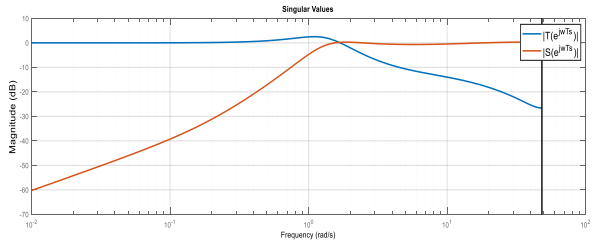


Source: Author (2023).

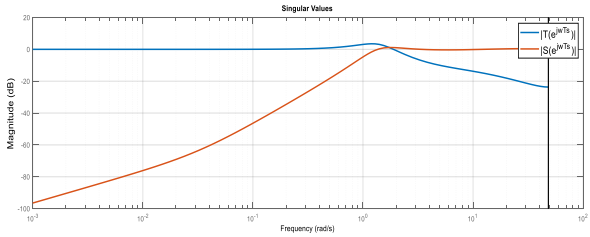
Figure 43 – Final sensitivity function plots for PPID robustness validation.



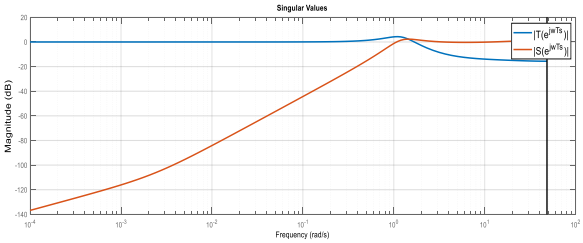
a. PPID sensitivity function decomposition for TGM control.



b. PPID sensitivity function decomposition for lateral speed control.



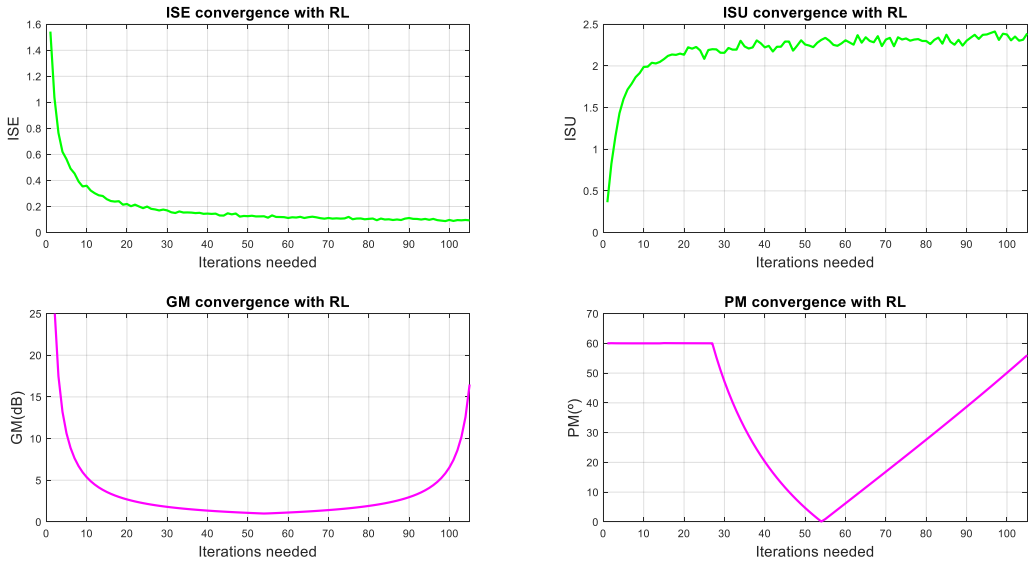
c. PPID sensitivity function decomposition for longitudinal speed control.



d. PPID sensitivity function decomposition for altitude control.

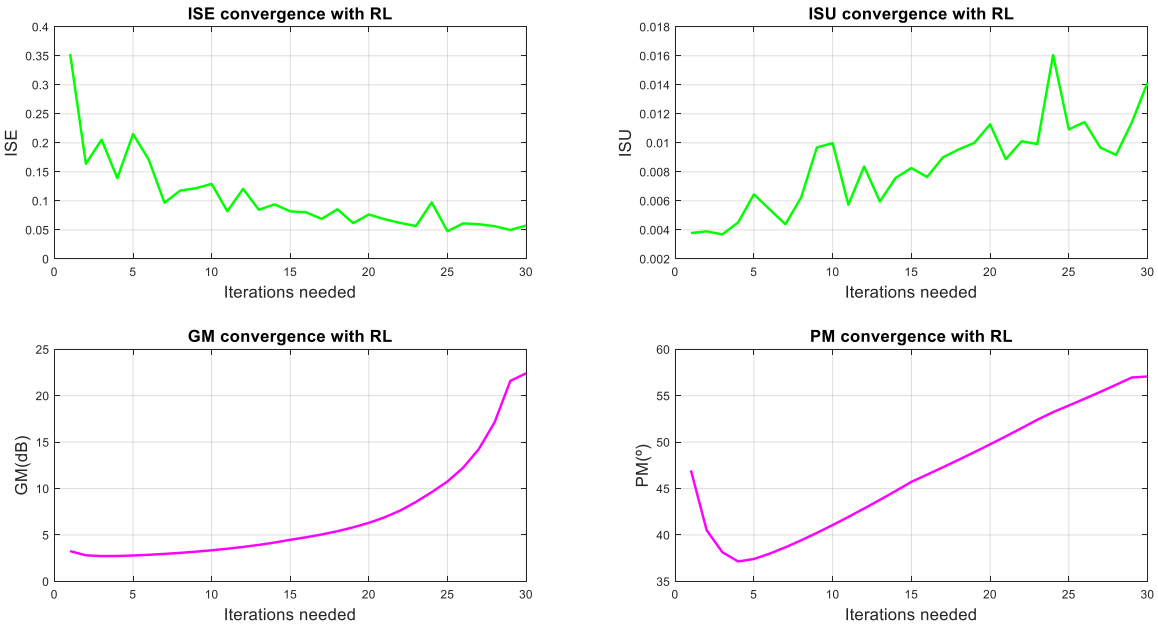
Source: Author (2023).

Figure 44 – TGM PPID indexes convergence through iterations.



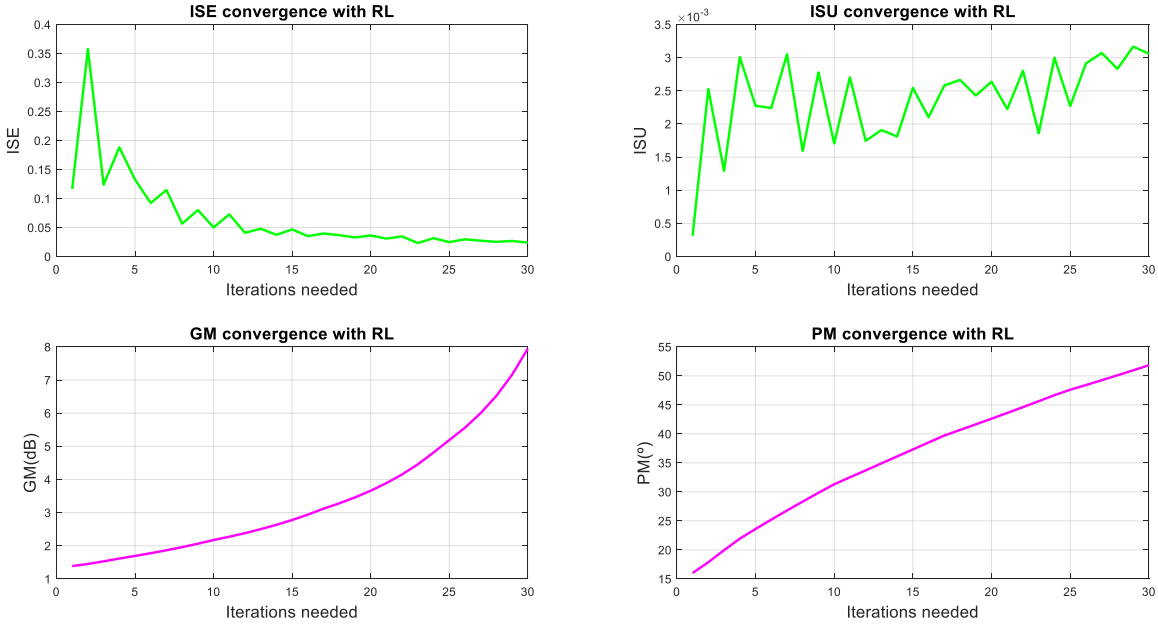
Source: Author (2023).

Figure 45 – Lateral Speed PPID indexes convergence through iterations.



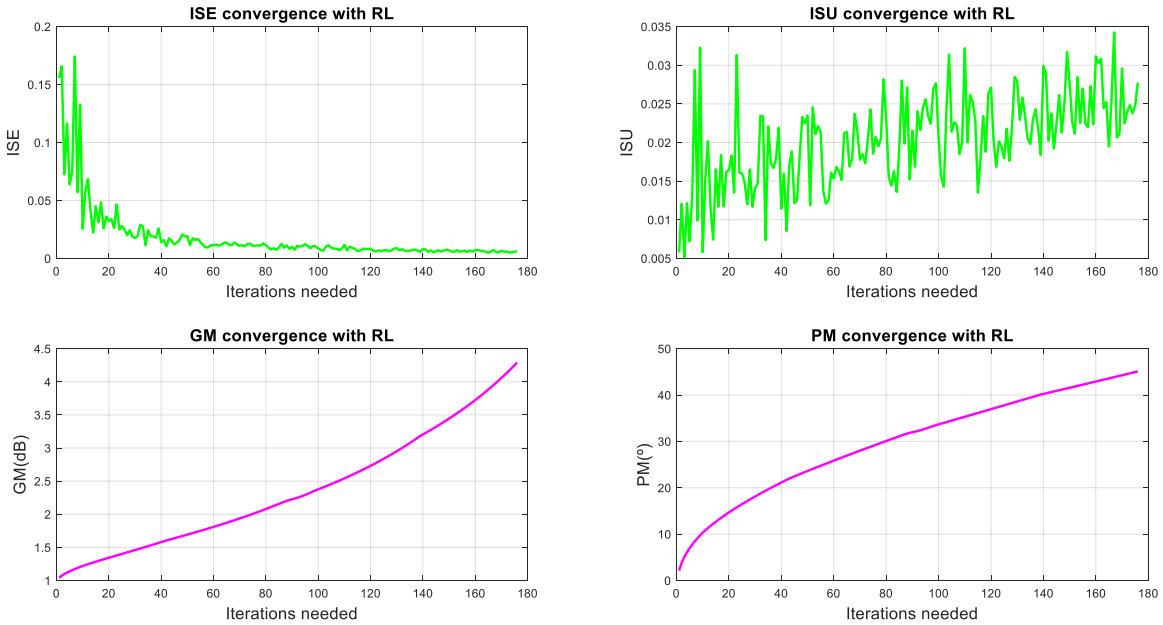
Source: Author (2023).

Figure 46 – Longitudinal Speed PPID indexes convergence through iterations.



Source: Author (2023).

Figure 47 – Altitude PPID indexes convergence through iterations.



Source: Author (2023).

Table 4 – Final iteration PPID indexes.

PPID Indexes	TGM	Lateral Speed	Longitudinal Speed	Altitude
K_c	1.1110	0.3290	0.3010	1.7610
ISE	0.0982	0.0482	0.0266	0.0063
ISU	2.4552	0.0134	0.0033	0.0284
σ_e^2	0.0972	0.0480	0.0264	0.0063
σ_u^2	0.6371	0.0083	0.0031	0.0283
GM	3.5918	22.4131	7.9471	4.2909
PM	60.0505 °	57.0695°	51.8360 °	45.0982 °

Source: Author (2023).

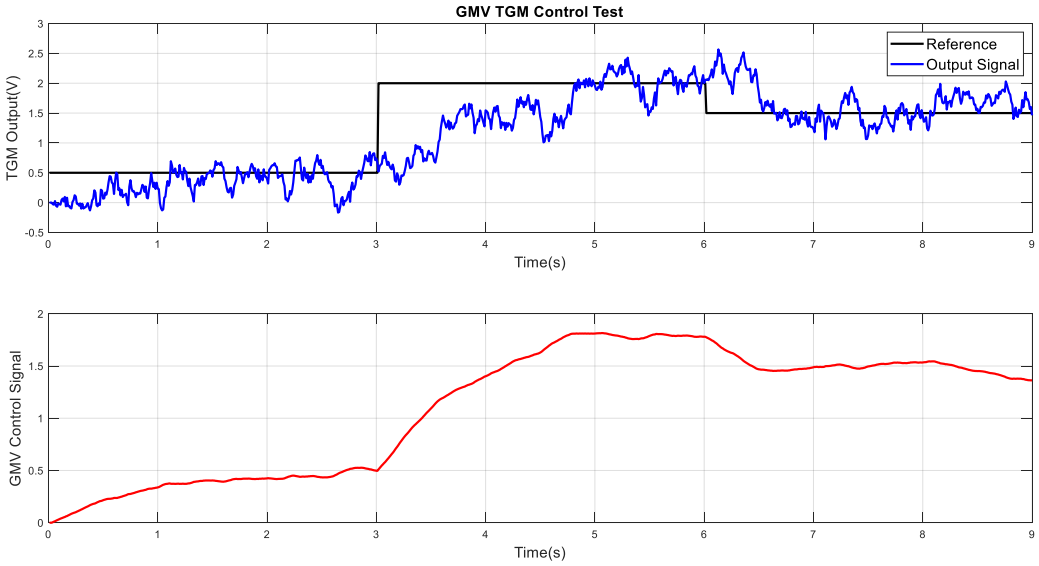
The convergence plot presented above brings an interesting analysis normally explored by computational intelligence articles and theses. The idea is to evaluate how many iterations are taken to achieve a result, which depends on the hardware configuration and the software used. In PPID it has a noisy alteration of ISE and ISU, which proportionally changes through time. GM and PM change with the system dynamics, and they have to be a study topic in future works.

5.2.2 GMV Control

Reference tracking and control signal are presented in Figures 48 to 51, where the outputs followed the references, even with the noises of the process for GMV. Figure

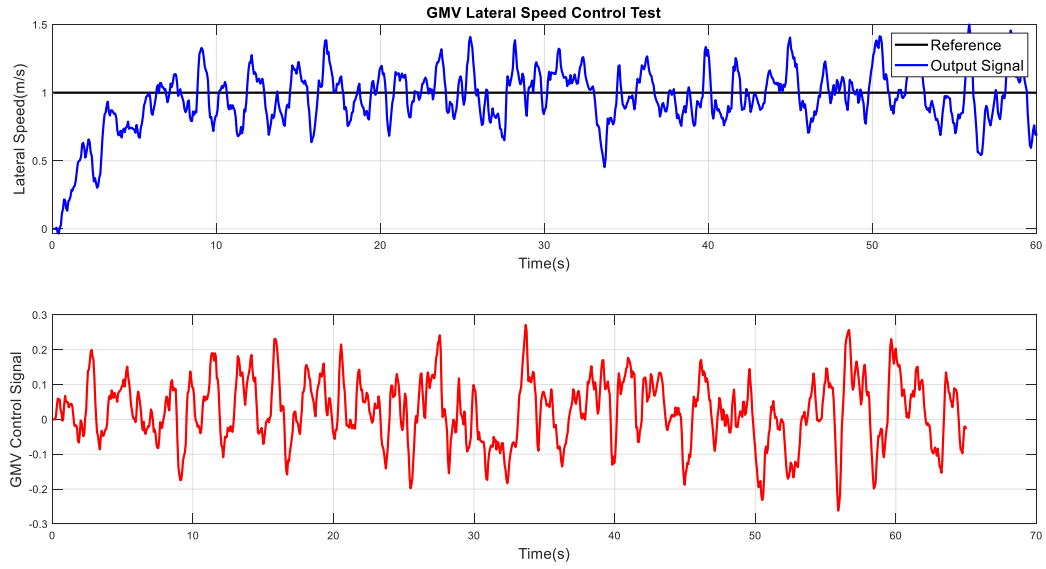
52 shows the four sensitivity functions plots, where it can be proved the target PM of the controlled systems. Figures 53 to 56 show the convergence of performance and robustness indexes among the iterations. Table 5 shows the final iteration GMV indexes.

Figure 48 – TGM GMV experimental control responses.



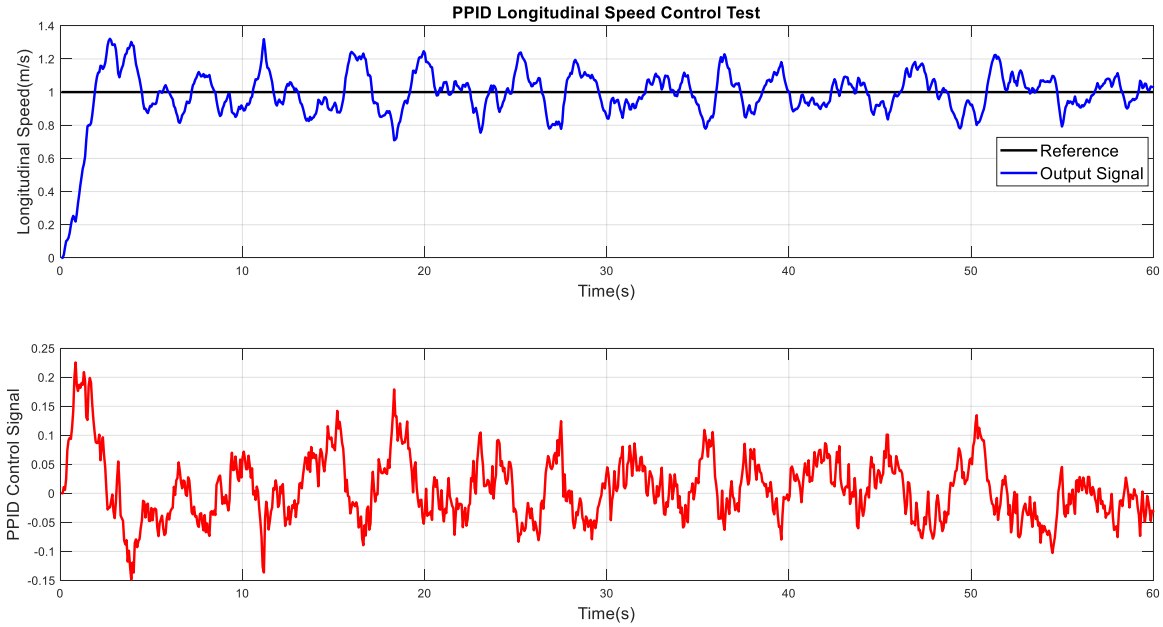
Source: Author (2023).

Figure 49 – Lateral Speed GMV control responses.



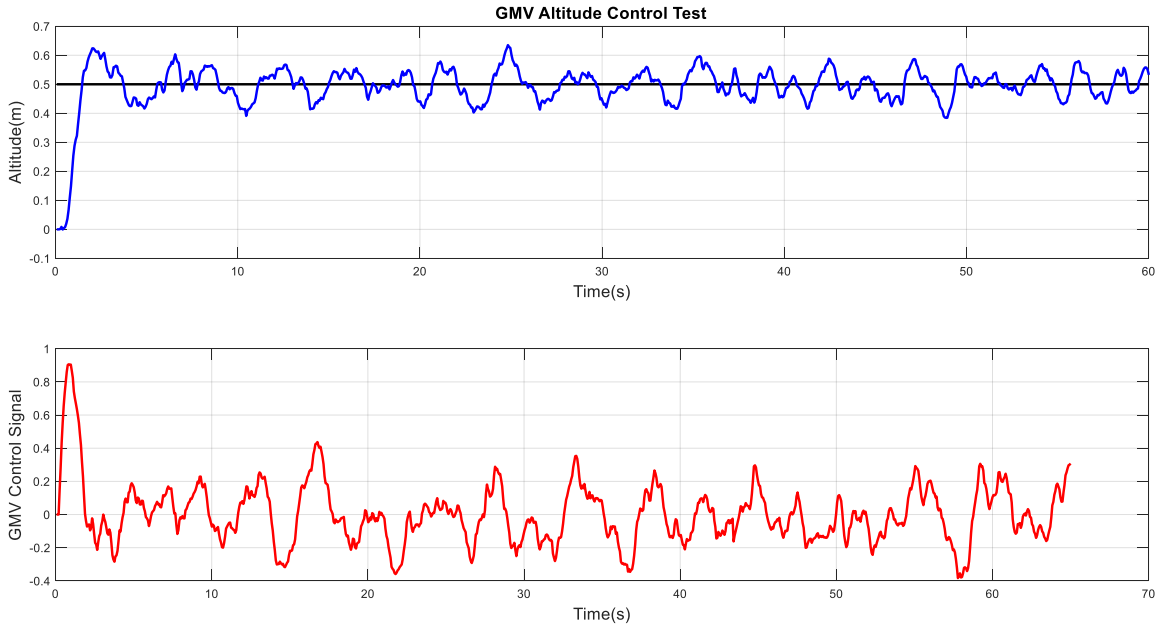
Source: Author (2023).

Figure 50 – Longitudinal Speed GMV control responses.



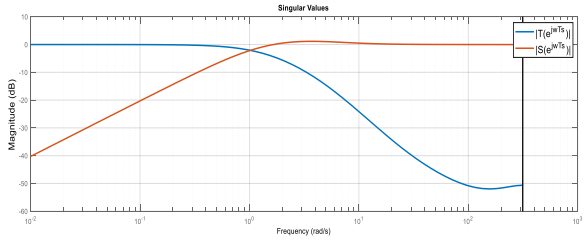
Source: Author (2023).

Figure 51 – Altitude GMV control responses.

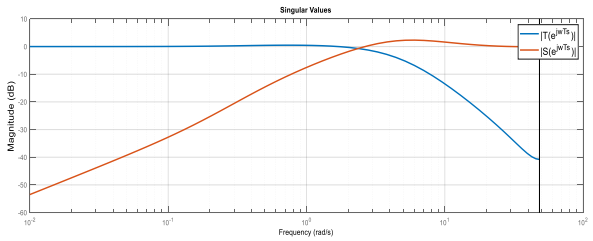


Source: Author (2023).

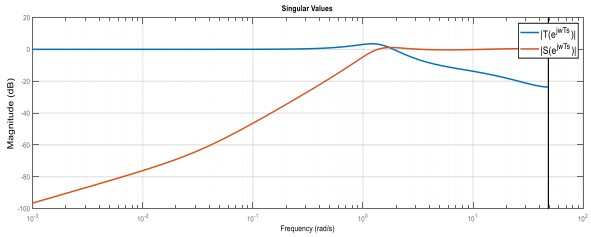
Figure 52 – Final sensitivity function plots for GMV robustness validation.



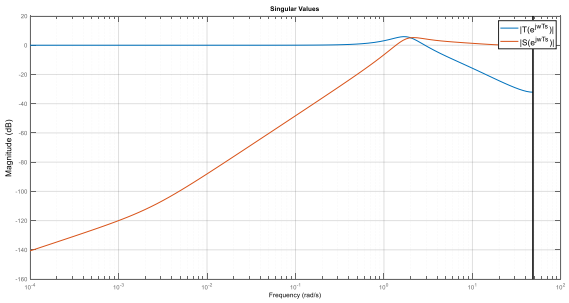
a. GMV sensitivity function decomposition for TGM control.



b. GMV sensitivity function decomposition for lateral speed control.



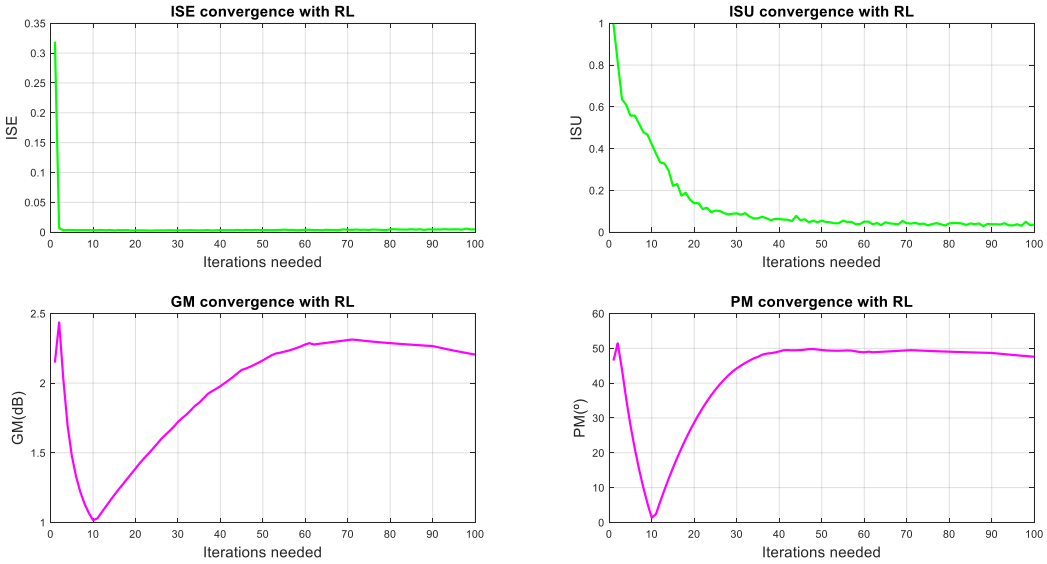
c. GMV sensitivity function decomposition for longitudinal speed control.



d. GMV sensitivity function decomposition for altitude control.

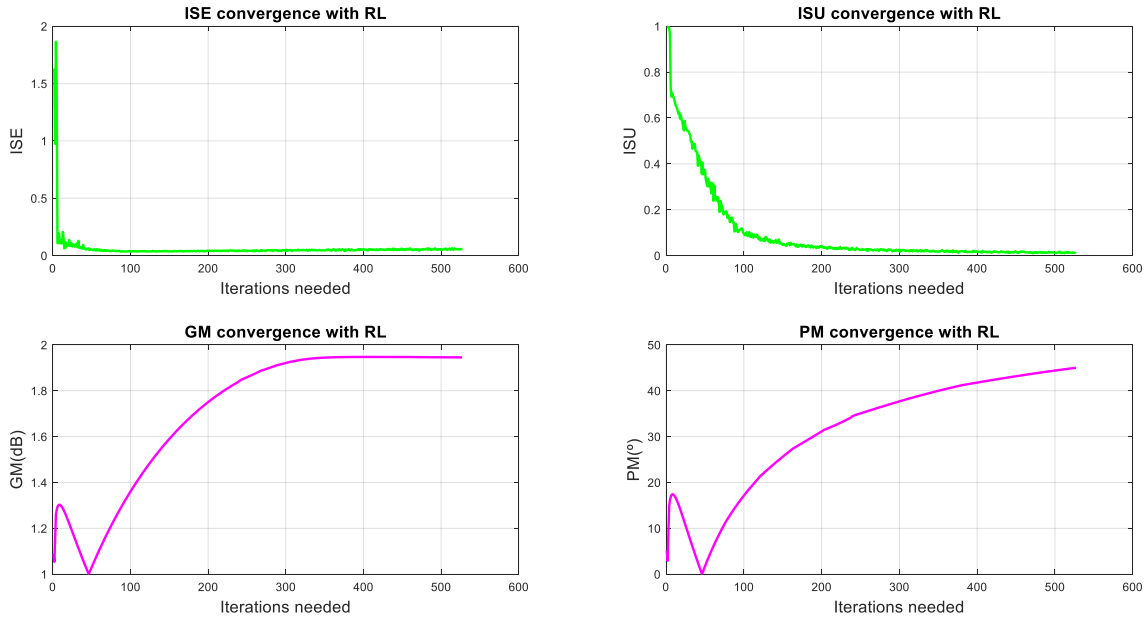
Source: Author (2023).

Figure 53 – TGM GMV indexes convergence through iterations.



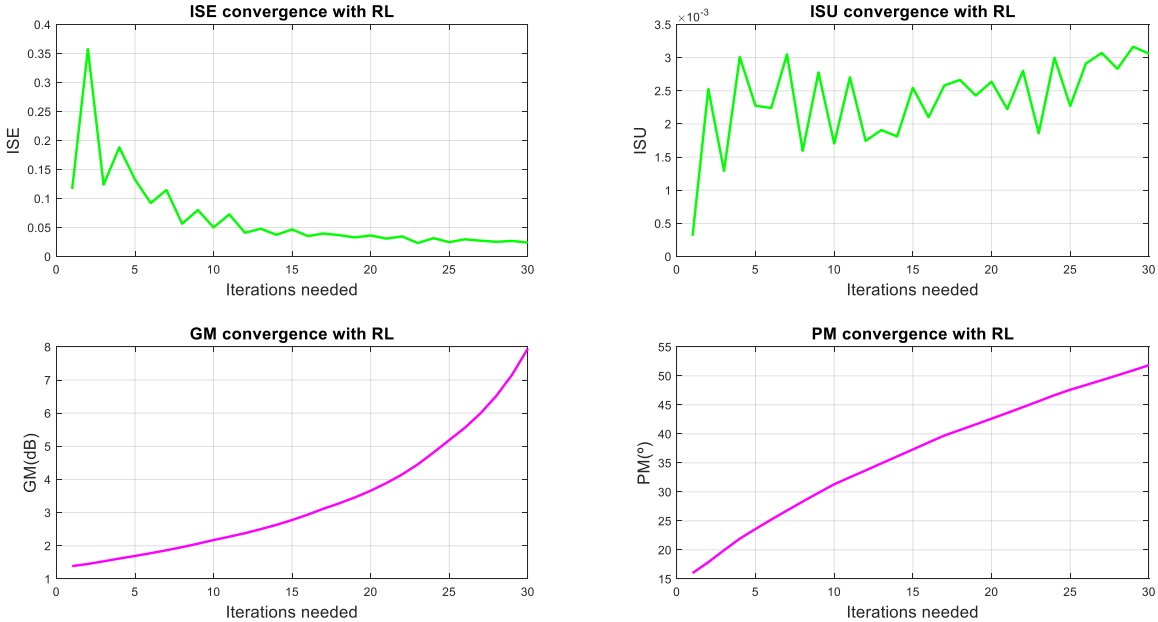
Source: Author (2023).

Figure 54 – Lateral Speed GMV indexes convergence through iterations.



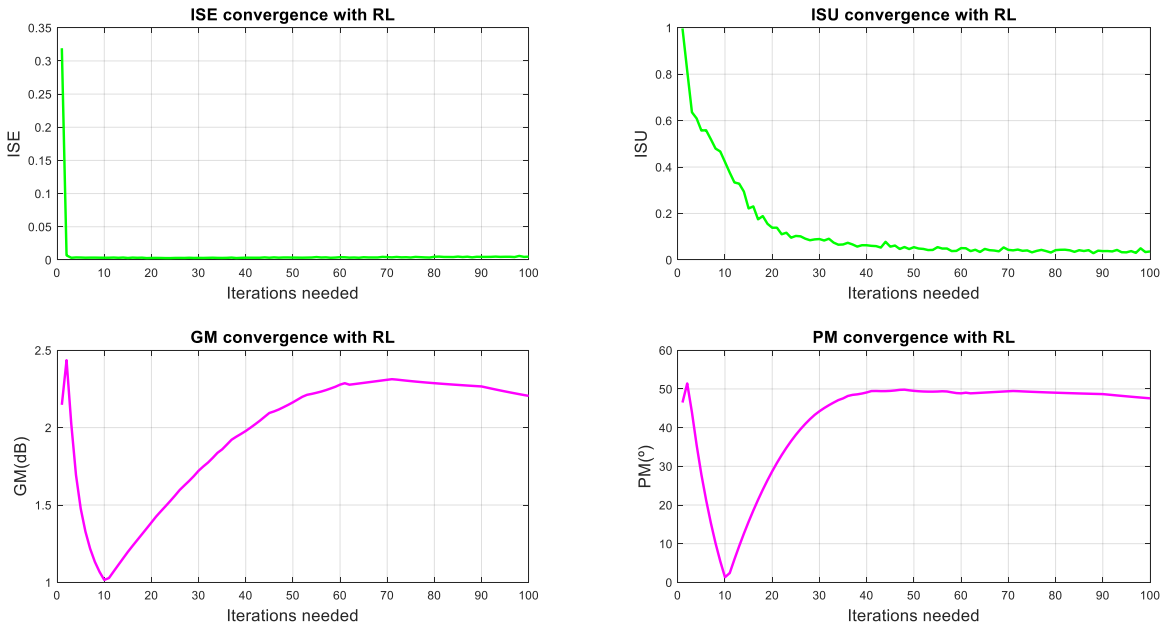
Source: Author (2023).

Figure 55 – Longitudinal Speed GMV indexes convergence through iterations.



Source: Author (2023).

Figure 56 – Altitude GMV indexes convergence through iterations.



Source: Author (2023).

Table 5 – Final iteration GMV indexes.

GMV Indexes	TGM	Lateral Speed	Longitudinal Speed	Altitude
q_0	137.5610	5.2710	2134.1470	0.9300
ISE	0.2175	0.0558	0.1558	0.0048
ISU	1.3927	0.0142	0.1317	0.0404
σ_e^2	0.2077	0.0536	0.1417	0.0048
σ_u^2	0.2937	0.0109	0.0914	0.0404
GM	2.0011	1.9454	7.8231	2.2529
PM	60.1003°	45.0044°	45.1472°	48.4333°

Source: Author (2023).

It is interesting to evaluate that this control technique achieved a lower performance when compared to PPID, using RL tuning method. The noises also were a problem, since the ARIX structure was used for the project not using $C(z^{-1})$ as a filter. A future paper will be written, that has been implemented in TGM is use the ARMAX identification to identify a model for control design. In the convergence graphs, it is possible to evaluate that this method takes more iterations than the orders to reach the stop criteria.

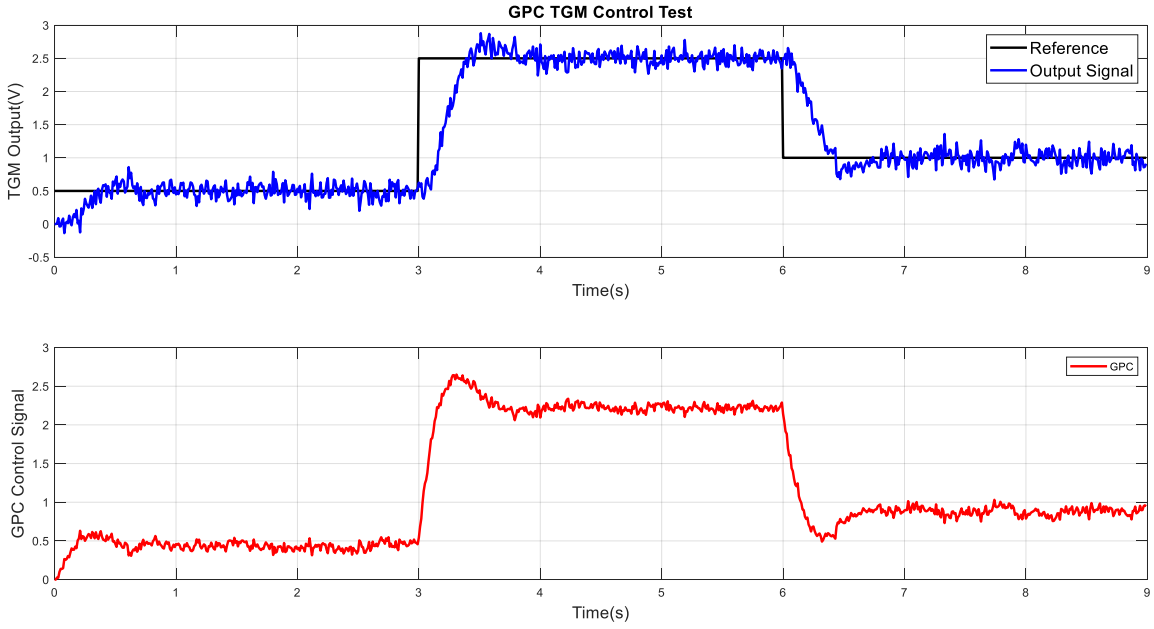
5.2.3 GPC Control

The last controller implemented with repeat and improve method is the GPC, which obtains the reference tracking and control signal presented in Figures 57 to 60, where the outputs followed the references, even with the noises of the process. Figure 65 shows the four sensitivity function plots, where it can be proved the target PM of the controlled systems. Figures 61 to 64 show the convergence of performance and robustness indexes among the iterations. Table 6 shows the final iteration GPC indexes.

As its structure is the most complex with the ARX structure used, it has been the algorithm that takes more time between iterations to achieve a new control action. The results with this controller have achieved excellent performance indexes, with lower values than the other ones for ISE and for the variances. For TGM experimental implementation, to achieve the λ implemented, the algorithm took more iterations and more time than the other tested algorithms. It is also interesting to evaluate, also, that the dynamics inferred directly into this controller, as for Altitude, the control action was summed with a 0.001 value in each iteration, which explains the x-axis on Figure 64.

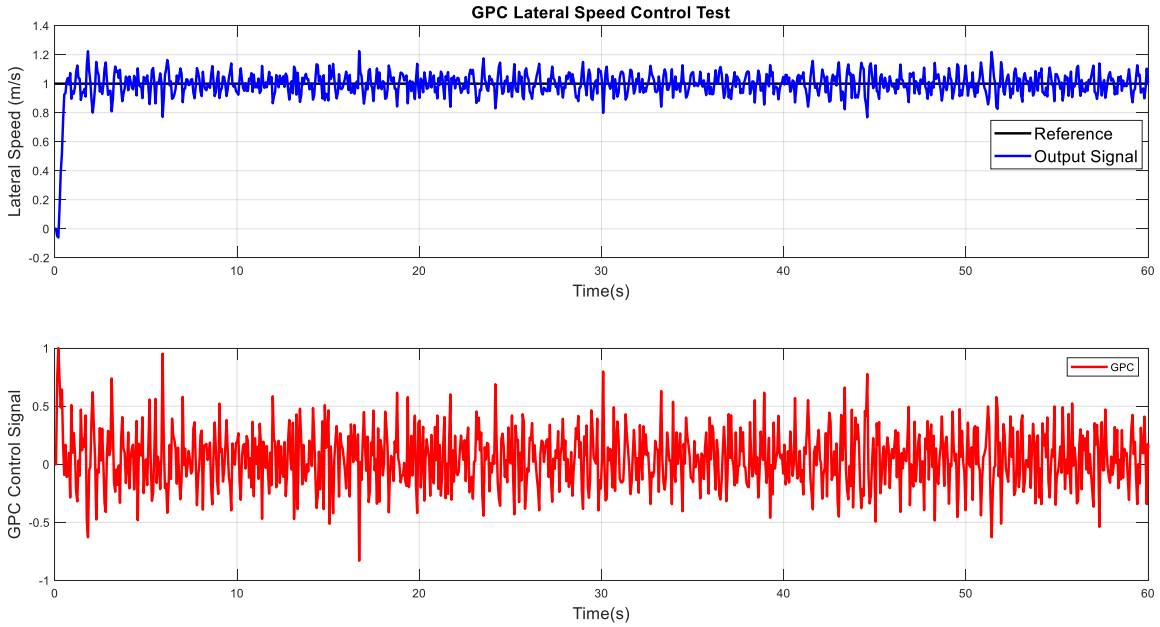
As in GMV test, this algorithm also has been tested with $C(Z^{-1})$ polynomial, using ARMAX structure as model, but the results have not been considerably different, with a performance index difference lower then 5%.

Figure 57 – TGM GPC experimental control responses.



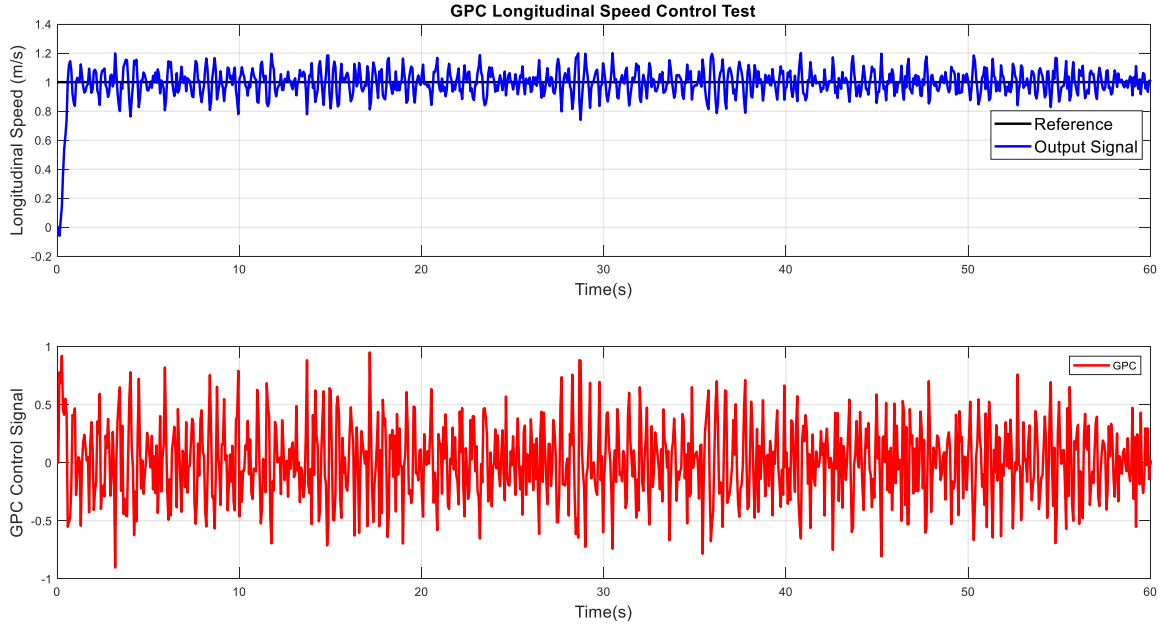
Source: Author (2023).

Figure 58 – Lateral Speed GPC control responses.



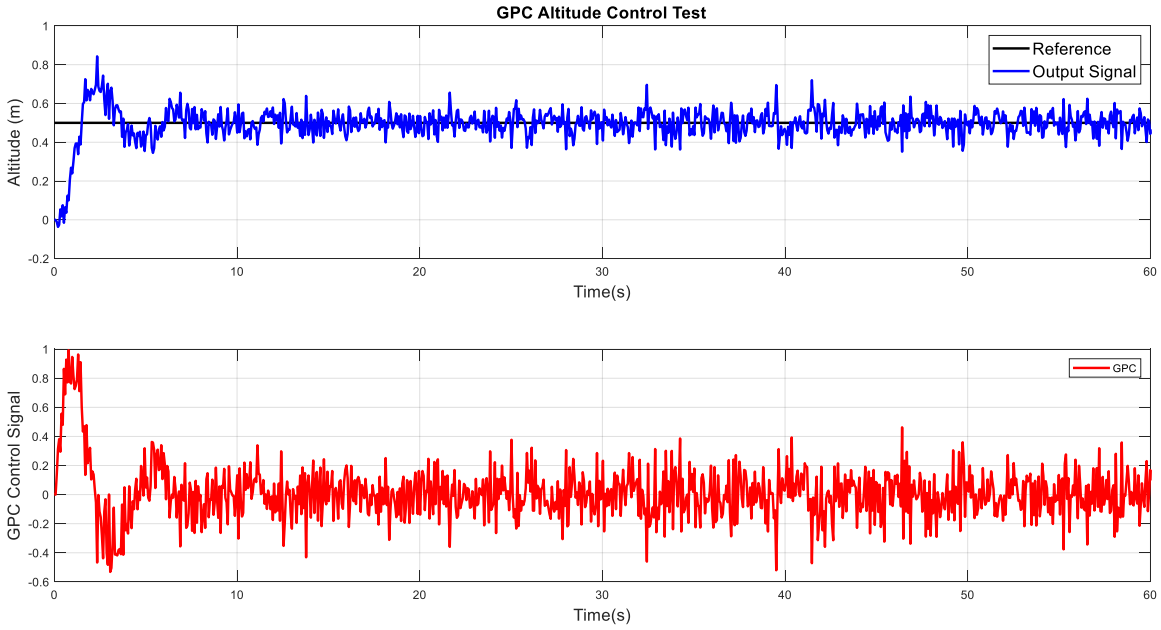
Source: Author (2023).

Figure 59 – Longitudinal Speed GPC control responses.



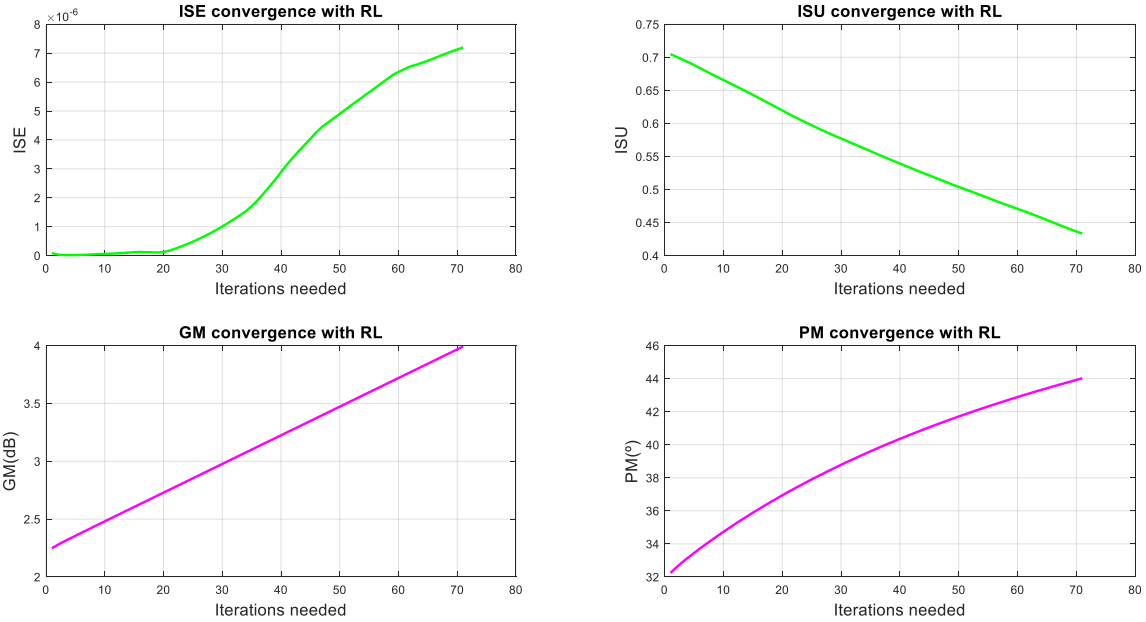
Source: Author (2023).

Figure 60 – Altitude GPC control responses.



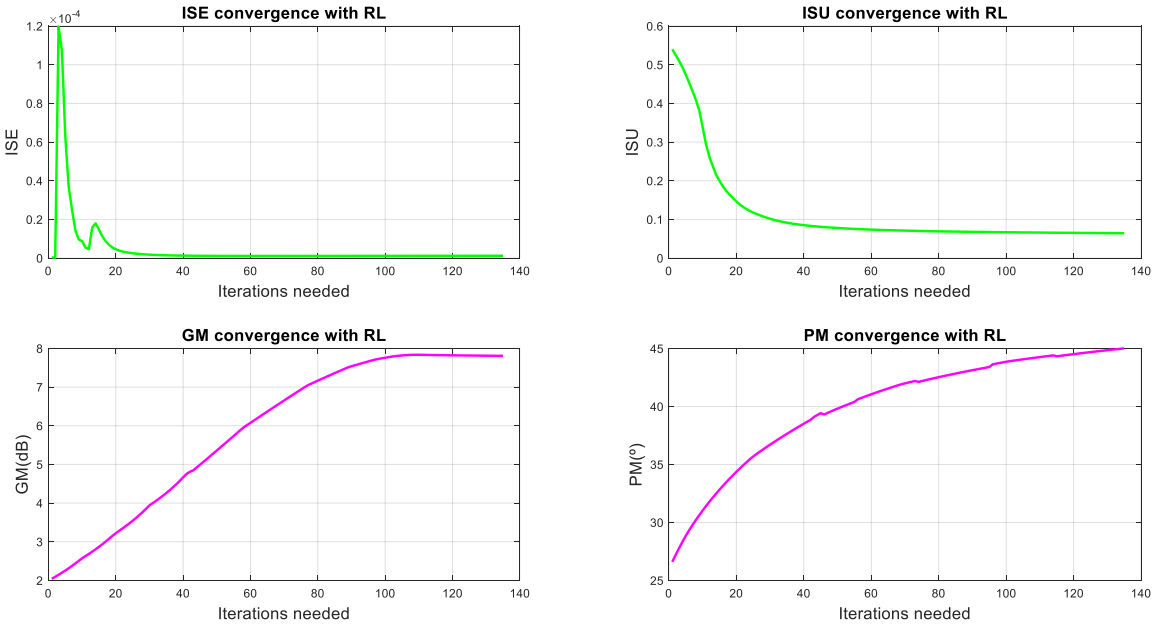
Source: Author (2023).

Figure 61 – TGM GPC indexes convergence through iterations.



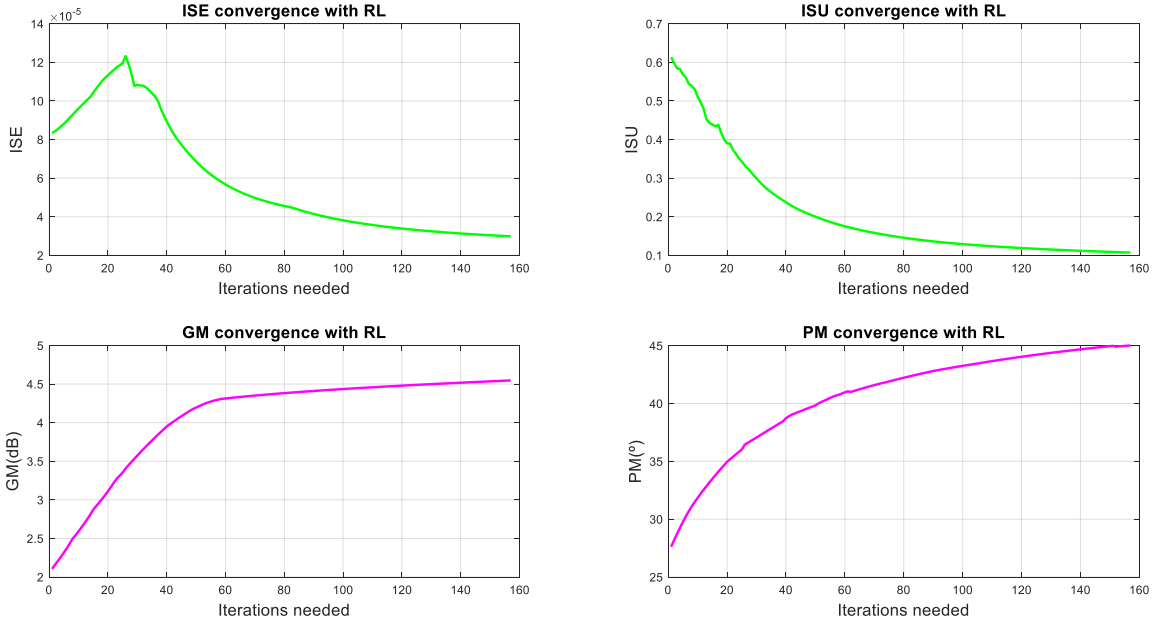
Source: Author (2023).

Figure 62 – Lateral Speed GPC indexes convergence through iterations.



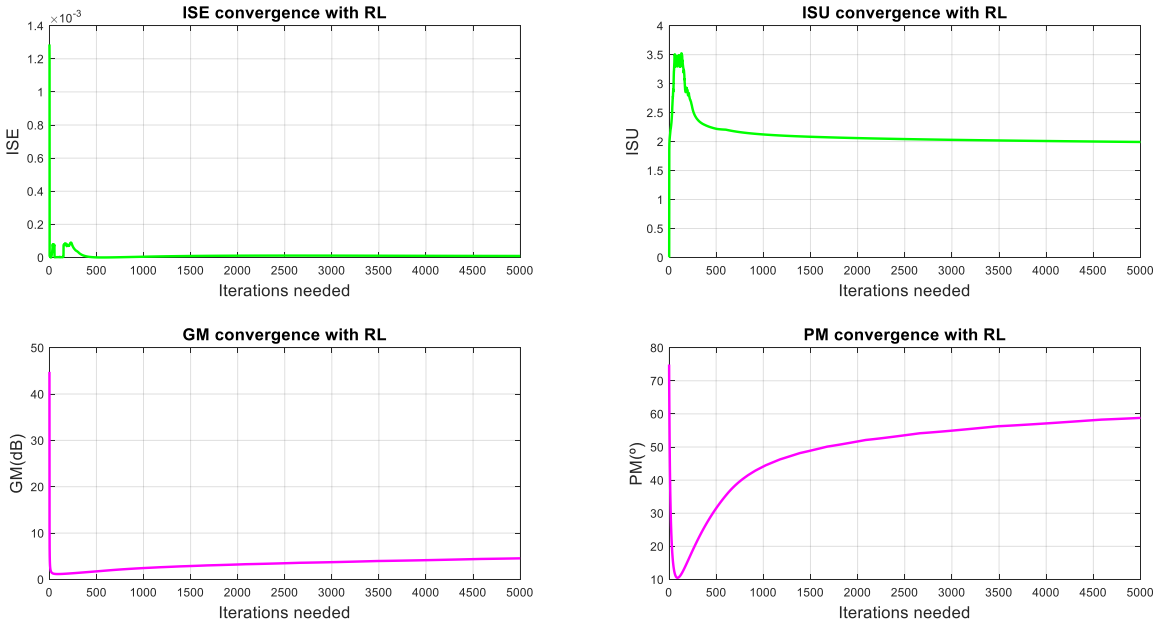
Source: Author (2023).

Figure 63 – Longitudinal Speed GPC indexes convergence through iterations.



Source: Author (2023).

Figure 64 – Altitude GPC indexes convergence through iterations.



Source: Author (2023).

Table 6 – Final iteration GPC indexes.

GPC Indexes	TGM	Lateral Speed	Longitudinal Speed	Altitude
λ	224.4210	1.3510	1.5710	3.2710
ISE	1.8107	0.5140	0.7601	1.3103
ISU	0.9168	0.0701	0.1051	0.1028
σ_e^2	0.0814	0.0823	0.0917	0.1217
σ_u^2	0.5628	0.0678	0.1050	0.1022
GM	7.1319	7.8069	4.5469	4.0374
PM	60.0014°	45.0100°	45.0144°	45.1918°

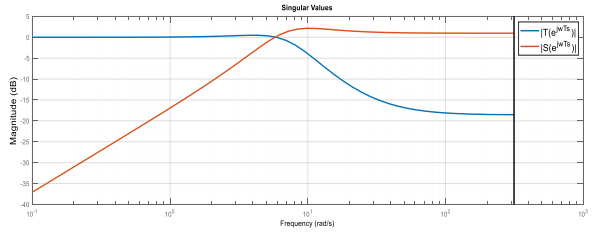
Source: Author (2023).

5.2.4 LQR Control

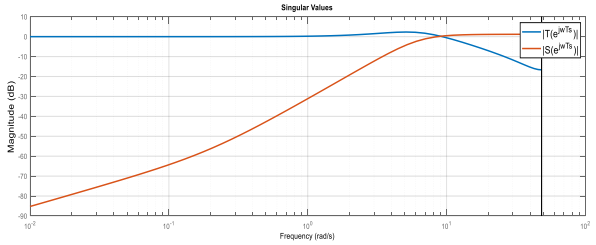
The idea of using LQR is to show how Q learning changes its Q matrix, rewarding the system along the test and, as in Vrabie and Lewis (2013), has normally better results when compared to traditional LQR with static Q and R matrix. For both simulations, Q and R have been initialized with unitary values, and using the SS NRLS identified system, with the $x_1 = y_1 -> \ Lateral\ Speed$,$x_2 = y_2 -> \ Longitudinal\ Speed$ and $x_3 = y_3 -> \ Altitude$.

Figures 66 and 67 show, respectively, the achieved responses using the traditional LQR based on SSNRLS and OKID models tunned using Ricatti's equation with Q and R being initialized, respectively, as diagonal matrixes with 100 and 1 values. Figure

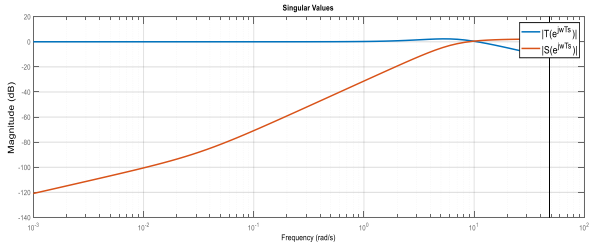
Figure 65 – Final sensitivity function plots for robustness GPC validation.



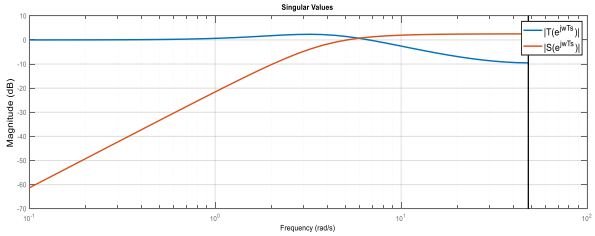
a. GPC sensitivity function decomposition for TGM control.



b. GPC sensitivity function decomposition for lateral speed control.



c. GPC sensitivity function decomposition for longitudinal speed control.



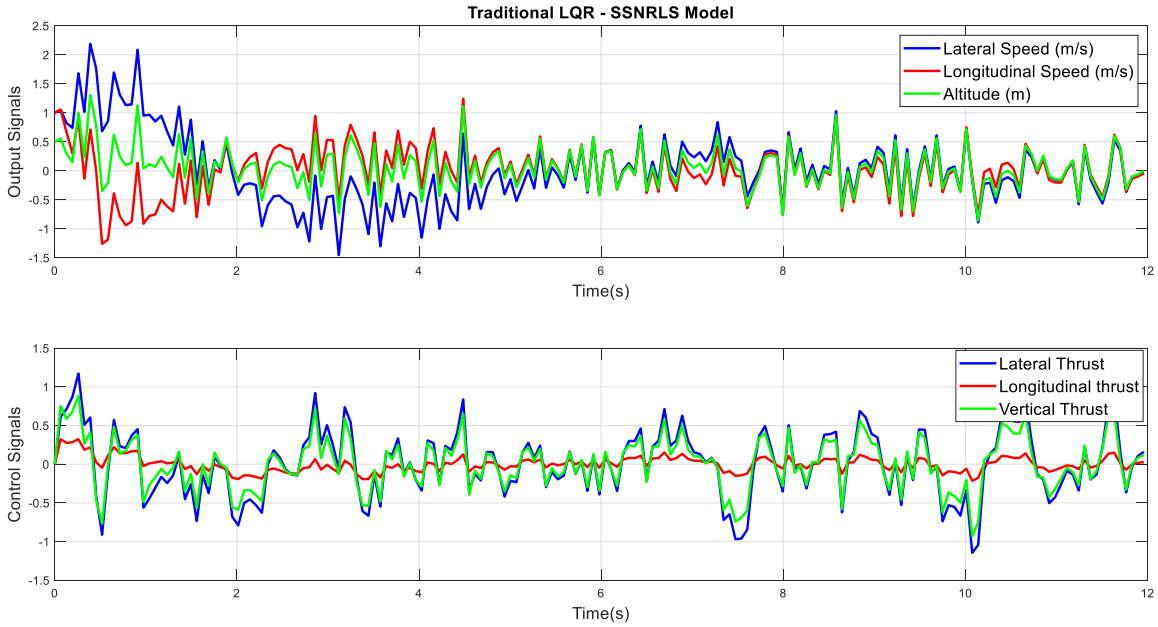
d. GPC sensitivity function decomposition for altitude control.

Source: Author (2023).

68 and 69 show, respectively, the obtained results using the RL LQR proposed based on SSNRLS and OKID models. This second method uses Q-learning to change the Q-matrix, as presented in Figure 70, using a ponderation from which state needs a higher control action, then the LQR policy uses higher weighs for the Ricatti's equation gain, then reaching the desired output faster, so reward got smaller when the action does not have to be applied, as presented in Figure 71. It is important to mention that this method does not change R matrix, using it as a diagonal matrix with value 1 for all terms, as in traditional LQR.

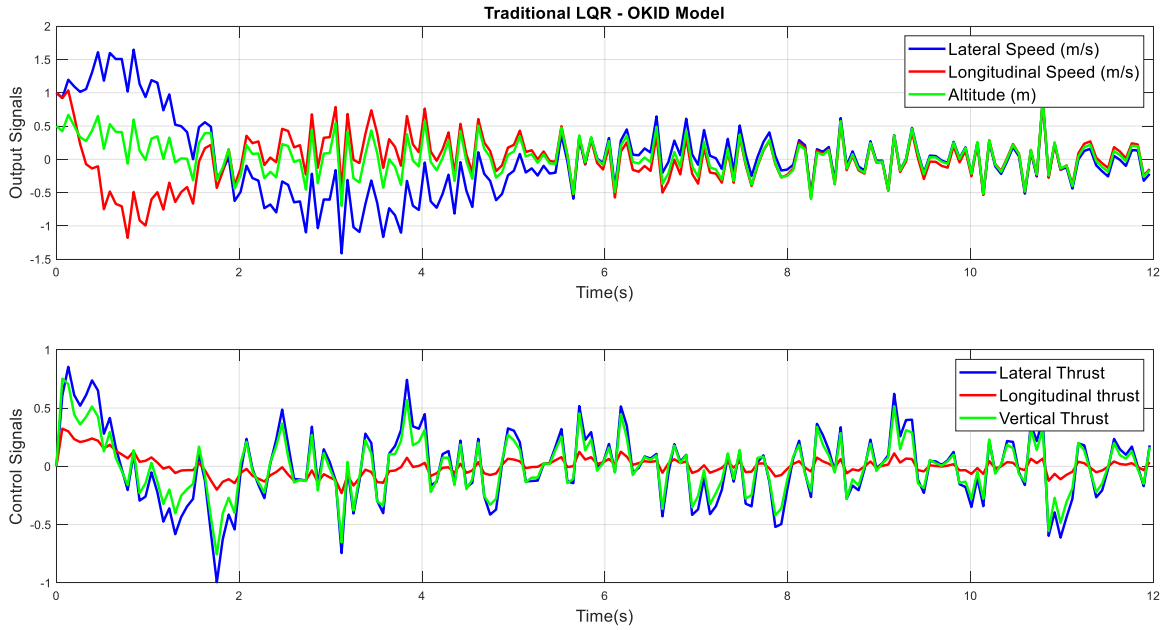
It is presented in Table 7 the final indexes of LQR, splitting for each output the achieved values of the proposed performance and robustness indexes. It is possible to evaluate that OKID model using Γ matrix helps both tuning methods to stabilize the systems. For the traditional tuning method, is possible to check that the algorithm contours the noises, but presents many peaks that in a real implementation may be dangerous for the AR Drone. When RL is evaluated is possible to check noisy signals for SSNRLS model and lower for OKID model, but with a small variance from the previous sample, which is mathematically proved with the σ^2 indexes. The robustness evaluation has presented similar results for all methods, with a small vantage for RL.

Figure 66 – Traditional LQR signals using SSNRLS estimated model.



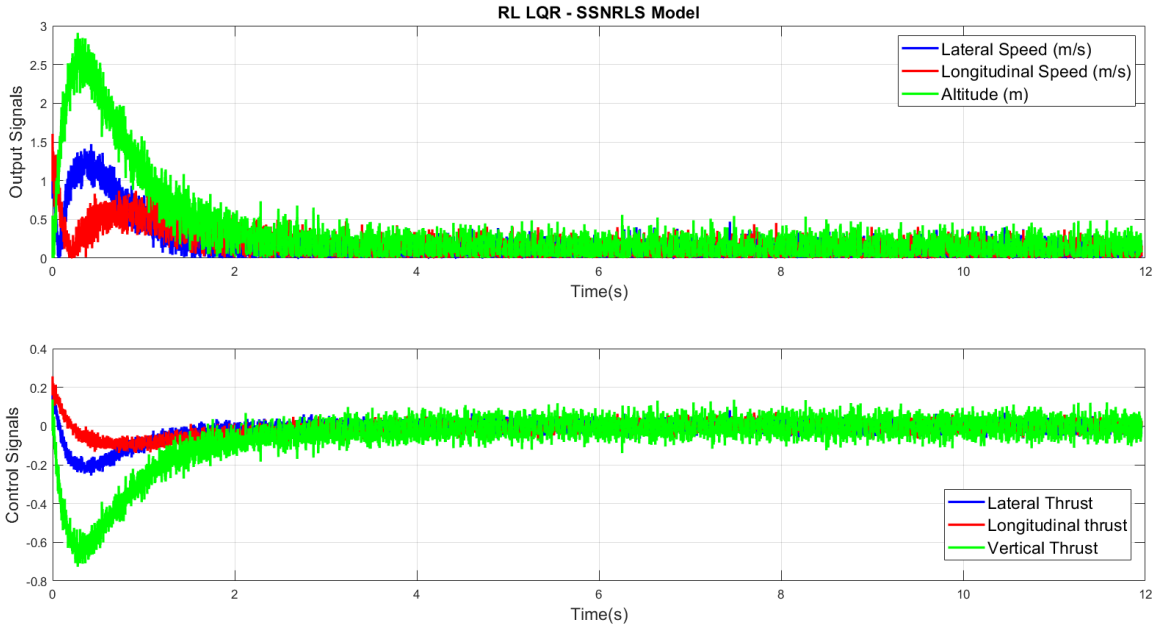
Source: Author (2023).

Figure 67 – Traditional LQR signals using OKID estimated model.



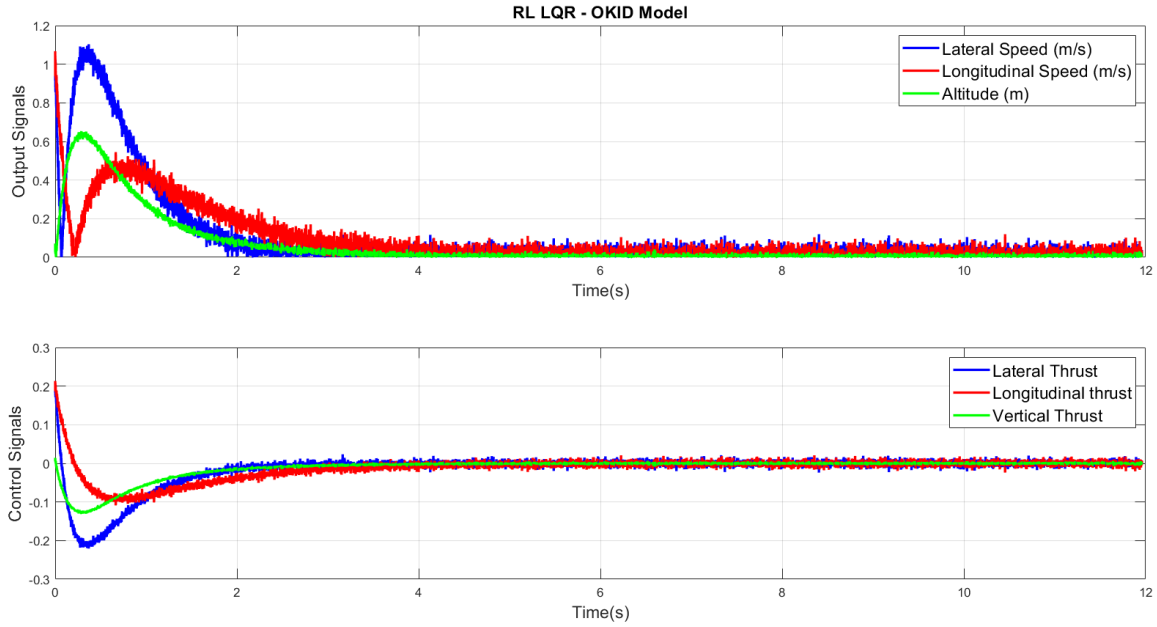
Source: Author (2023).

Figure 68 – RL tunned LQR signals using SSNRLS estimated model.



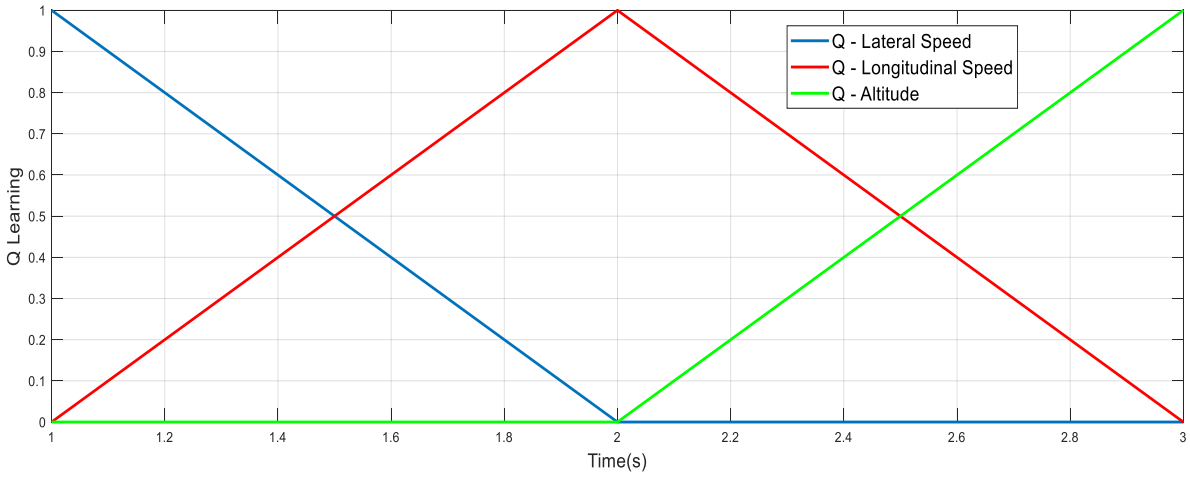
Source: Author (2023).

Figure 69 – RL tunned LQR signals using OKID estimated model.



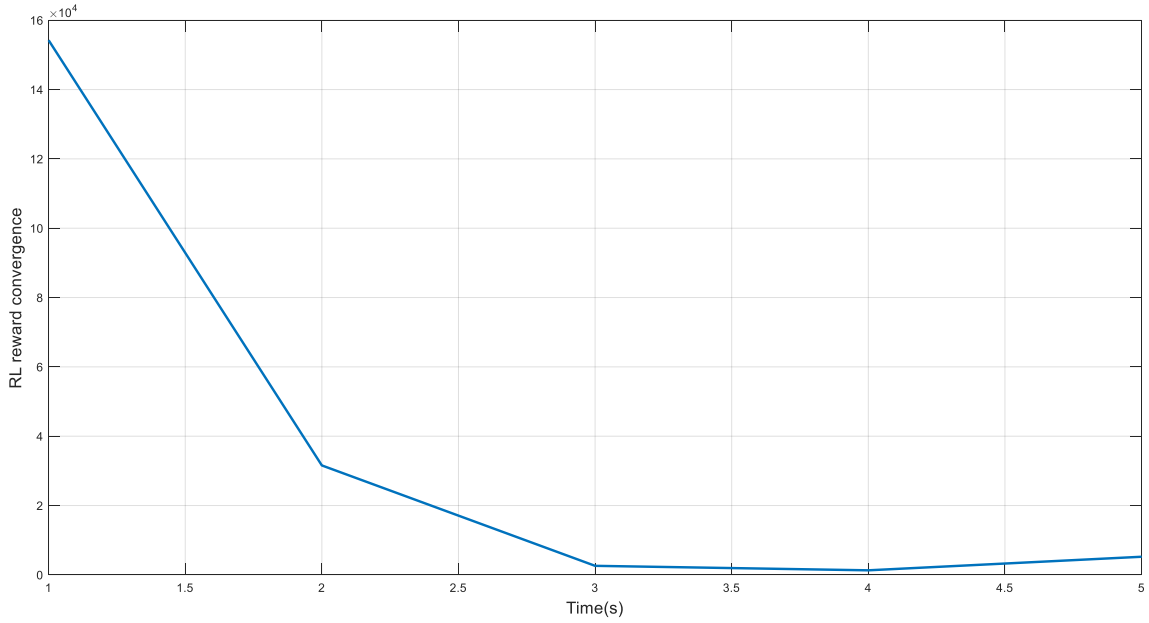
Source: Author (2023).

Figure 70 – Q Learning matrix for RL LQR.



Source: Author (2023).

Figure 71 – Reward convergence for RL LQR.



Source: Author (2023).

Table 7 – LQR indexes.

LQR Indexes	NRLS Trad. LQR	OKID Trad. LQR	NRLS RL LQR	OKID RL LQR
$ISE_{Lateral\ Speed}$	4.1754	2.4152	2.3147	1.9746
$ISU_{Lateral\ Speed}$	0.3147	0.2017	0.2019	0.2005
$\sigma^2_{e\ Lateral\ Speed}$	0.2014	0.1823	0.1817	0.1813
$\sigma^2_{u\ Lateral\ Speed}$	0.6628	0.6278	0.6250	0.6122
$ISE_{Longitudinal\ Speed}$	3.2740	1.9314	1.9311	1.7146
$ISU_{Longitudinal\ Speed}$	0.1785	0.1584	0.1564	0.1021
$\sigma^2_{e\ Longitudinal\ Speed}$	0.1914	0.1827	0.1827	0.1723
$\sigma^2_{u\ Longitudinal\ Speed}$	0.4639	0.3772	0.3750	0.3742
$ISE_{Altitude}$	0.9541	0.7412	1.7314	0.8614
$ISU_{Altitude}$	0.9168	0.0701	0.1051	0.1028
$\sigma^2_{e\ Altitude}$	0.3044	0.2923	0.3917	0.3019
$\sigma^2_{u\ Altitude}$	0.2928	0.2818	0.2754	0.2222
GM	6.0214	6.0066	6.0861	6.1384
PM	60.0014°	60.0100°	60.0144°	60.1918°

Source: Author (2023).

6 CONCLUSION AND FUTURE WORK PROPOSALS

6.1 CONCLUSIONS

In this master's thesis, have been delved into the realm of control theory research with a specific focus on reinforcement learning tuning methods. Throughout this study, various control techniques, algorithms and methodologies have been explored, such as: PPID, GMV, GPC and LQR, aiming to provide novel insights and contributions to the field of control theory.

The first phase of the research involved a review of the existing literature, where we analyzed and synthesized the current state-of-the-art in control theory. This comprehensive survey not only established the foundation for our investigation but also highlighted the gaps and opportunities for further advancement in the field. That was important for analyzing the different structures of the proposed controllers, and how the methods work in different topologies.

Subsequently, it is proposed a novel for control model tuning, with repeat and improve and sum differential games with Q learning to address the challenges posed by control theory. The model leveraged uses RL to reach the target values of PM and ISE for the SISO controllers, in order to optimize control inputs while considering the model-based structure of those systems. For the Q-learning approach, the study uses MDPs to optimize Ricatti's equation, changing the Q matrix along iterations for LQR control.

To validate the efficacy of the proposed control method, have been conducted extensive simulations and, where applicable, implemented in real-world experiments. The results obtained from these analyses showcased the method's robustness, efficiency, and performance in contribution to existing control strategies. The achieved results showed the efficiency of the tuning method used. PPID, GMV and GPC achieved the intended results and obtained the targets, using only machine math effort, with less possible human interference, tuning the applied controllers for TGM and Ar drone. On LQR the method also achieved the intended results, with faster response speed when compared to the traditional LQR.

Furthermore, it was shown the versatility and applicability of the model through its successful integration into the identified models, using TGM through real experiments and on Ar Drone only by simulations with white Gaussian noises. The positive outcomes from these practical implementations indicate the potential impact of this research on various real-world control systems and how this method can be an alternative to traditional tuning methods.

Despite the achievements made, our study also uncovers certain areas that warrant further investigation. For instance, regular PCs cannot support the math effort of those methods and cheap microcontrollers either. Addressing these limitations can

lead to the model's broader applicability and increased effectiveness in diverse control scenarios, since the evolution of technology tends to increase the implementation capacity of those algorithms, which are already used in high-level aerospace control in North America and Europe (HODGE; HAWKINS; ALEXANDER, 2021; LEWIS, Frank L.; VRABIE, 2009).

Overall, the findings from this research contribute to the growing body of knowledge in control theory, enriching the domain with a novel approach that can potentially enhance the efficiency and stability of various control systems. Tuning in PPID presents viability for classic methods of simpler structures, which are more accepted in the industry, and for presenting performance and robustness indexes considered good for complex applications. Furthermore, the feasibility of using this same method of improve and repeat in predictive SISO controllers, such as GMV and GPC, also brings an advantage by using a structure more accepted by academia to control highly complex systems and that will converge to the stop criteria in less time with the evolution of machine learning embedded in computational systems or high-speed microprocessors.

6.2 FUTURE WORK PROPOSALS

Future research in control theory, particularly involving RL in advanced control techniques, holds tremendous promise to help the field and address complex control challenges in diverse applications. The integration of RL techniques into traditional control paradigms offers an exciting avenue for enhancing control strategies, adapting to uncertain environments, and achieving optimal performance in dynamic systems.

Future studies could explore stochastic predictive control addresses systems with inherent uncertainty, it may be done with the incorporation of RL techniques, developing adaptive controllers that learn to cope with stochasticity and make informed decisions under uncertainty. These RL-based stochastic predictive controllers could find applications in areas such as autonomous vehicles, UAVs and robotics, where environmental uncertainties play a crucial role.

Deep Reinforcement Learning (DRL) methods, such as Deep Q-Networks and Proximal Policy Optimization, have demonstrated impressive capabilities in learning complex tasks. Combining DRL with model predictive control can enable control systems to learn online models of the environment and optimize control actions based on these learned models. This could lead to improved performance in control tasks and adaptability to previously unseen scenarios.

Hybrid Predictive Control and RL in Multi-Agent Systems, can be also a research area with the appliance of predictive control techniques in multi-agent systems, such as swarms of drones or autonomous vehicles, which presents unique challenges. Future research can explore integrating RL into hybrid predictive control strategies to enable agents to learn from each other's experiences and adapt to the dynamic interactions

between them. This could lead to more efficient and coordinated behavior in multi-agent environments.

LQR is a well-established control technique, but it assumes a known linear system model, which may not always hold in practical scenarios. Integrating RL into LQR can create adaptive control systems that learn and update system models online, thus accommodating system changes and uncertainties. These RL-enhanced adaptive LQR controllers could be particularly valuable in applications where accurate modeling is challenging, using Kalman filter estimation in adaptative control, what is the next step of the presented research.

Reinforcement learning (RL) tuning for power electronics microgrids presents a compelling and promising avenue for future research. Power electronics microgrids are complex systems with dynamic energy sources, loads, and storage elements, requiring sophisticated control strategies to optimize performance and stability. By integrating RL into the tuning process, control parameters of power electronics devices can be dynamically adjusted based on the system's performance and environmental conditions, leading to improved control accuracy, robustness, and adaptability. RL's ability to learn from data and adapt to changing microgrid conditions can overcome challenges related to uncertainties and nonlinearities, providing an efficient and scalable solution for control optimization in microgrid environments. Moreover, RL-tuned power electronics microgrids could enhance energy efficiency, grid integration, and grid resiliency, contributing to the sustainable development and integration of renewable energy sources in future smart grids. This is also being researched in LACOS laboratory in partnership with IFPA power electronics laboratory and UFPB institution.

In conclusion, the integration of reinforcement learning with predictive control, stochastic predictive control, and linear quadratic control offers an exciting and promising direction for future research in control theory. These hybrid control paradigms have the potential to address complex and uncertain control problems, leading to more efficient, adaptive, and robust control systems across various domains. As researchers delve deeper into this multidisciplinary intersection, the field of control theory is poised to witness groundbreaking advancements that will shape the future of intelligent control systems.

BIBLIOGRAPHY

AGUIRRE, Luis Antonio. **Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais**. [S.l.]: Editora UFMG, 2004.

ARAÚJO, Rejane de Barros et al. Controladores preditivos filtrados utilizando otimização multiobjetivo para garantir offset-free e robustez, 2017.

ÅSTRÖM, Karl J; WITTENMARK, Björn. **Computer-controlled systems: theory and design**. [S.l.]: Courier Corporation, 2013.

ATHANS, Michael. The importance of Kalman filtering methods for economic systems. In: ANNALS of Economic and Social Measurement, Volume 3, number 1. [S.l.]: NBER, 1974. P. 49–64.

BABUSKA, Robert; VERBRUGGEN, H; HELLENDOORN, H; SIEMENS, Þ. Promising fuzzy modeling and control methodologies for industrial applications, July 2023.

BANERJEE, Pranob; SHAH, Sirish L. Tuning guidelines for robust generalized predictive control. In: IEEE. [1992] Proceedings of the 31st IEEE Conference on Decision and Control. [S.l.: s.n.], 1992. P. 3233–3234.

BAŞAR, Tamer; OLSDER, Geert Jan. **Dynamic noncooperative game theory**. [S.l.]: SIAM, 1998.

BEMPORAD, Alberto; MORARI, Manfred; DUA, Vivek; PISTIKOPOULOS, Efstratios N. The explicit linear quadratic regulator for constrained systems. **Automatica**, Elsevier, v. 38, n. 1, p. 3–20, 2002.

BEQUETTE, B Wayne. **Process control: modeling, design, and simulation**. [S.l.]: Prentice Hall Professional, 2003.

BITMEAD, RR; GEVERS, MR; WERTZ, V. **The thinking man's GPC**. [S.l.]: Snofart Press, 1989.

BOLTON, William. **Instrumentation and control systems**. [S.l.]: Newnes, 2021.

CAMACHO, Eduardo F; BORDONS, C. Robust Model Predictive Control. In: MODEL Predictive control. [S.l.]: Springer, 2007. P. 217–248.

CHESTNUT, Harold. Objectives and trends in feedback control systems progress. **Electrical Engineering**, IEEE, v. 77, n. 1, p. 58–63, 1958.

CLARKE, David W; GAWTHROP, Peter J. Self-tuning controller. In: IET, 9. PROCEEDINGS of the Institution of Electrical Engineers. [S.l.: s.n.], 1975. v. 122, p. 929–934.

CLARKE, David W; MOHTADI, C. Properties of generalized predictive control. **Automatica**, Elsevier, v. 25, n. 6, p. 859–875, 1989.

COELHO, AAR; JERONYMO, DC; ARAUJO, RB. Sistemas Dinâmicos Controle Classico e Preditivo Discreto. **Editora da UFSC**, 2019.

COELHO, Antonio Augusto Rodrigues; SANTOS COELHO, Leandro dos. **Identificação de sistemas dinâmicos lineares**. [S.l.: s.n.], 2004.

D. VRABIE, K. Vamvoudakis; LEWIS, F. L. **Optimal adaptive control and differential games by reinforcement leanring principles**. [S.l.: s.n.], 2013.

DA SILVA, Daniel Abreu Macedo; SILVEIRA, Antonio Da Silva; DO NASCIMENTO, André Cavalcante. State Space Predictive Minimum Variance Controller Applied to a Tacho Generator Motor. In: IEEE. 2022 14th Seminar on Power Electronics and Control (SEPOC). [S.l.: s.n.], 2022. P. 1–6.

DASTYAR, Mohammad Ebrahim; MALEK, Alaeddin; YOUSEFI, Sohrabali. Solving two-dimensional bioheat optimal control problem in solid and vessel domain by pseudospectral discretization. **Transactions of the Institute of Measurement and Control**, SAGE Publications Sage UK: London, England, v. 44, n. 12, p. 2358–2368, 2022.

DOGRU, Oguzhan; VELSWAMY, Kirubakaran; IBRAHIM, Fadi; WU, Yuqi; SUNDARAMOORTHY, Arun Senthil; HUANG, Biao; XU, Shu; NIXON, Mark; BELL, Noel. Reinforcement learning approach to autonomous PID tuning. **Computers & Chemical Engineering**, Elsevier, v. 161, p. 107760, 2022.

DUTRA, Bruno; SILVEIRA, Antonio; PEREIRA, Antonio. Grasping force estimation using state-space model and Kalman Filter. **Biomedical Signal Processing and Control**, Elsevier, v. 70, p. 103036, 2021.

EDGAR, Thomas; HAHN, Juergen. Process Automation, Jan. 2009.

FLUEGGE-LOTZ, Irmgard; HALKIN, Hubert. **Pontryagin's maximum principle and optimal control**. [S.l.], 1961.

FRIEDMAN, Leonard. Robot Control Strategy. In: IJCAI. [S.l.: s.n.], 1969. P. 527–540.

GARCIA, Carlos E; MORARI, Manfred. Internal model control. A unifying review and some new results. **Industrial & Engineering Chemistry Process Design and Development**, ACS Publications, v. 21, n. 2, p. 308–323, 1982.

HE, Liang; PACE, John. Estimating Altitude of Drones Using Batteries. In: IEEE. 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI). [S.l.: s.n.], 2020. P. 264–265.

HODGE, Victoria J; HAWKINS, Richard; ALEXANDER, Rob. Deep reinforcement learning for drone navigation using sensor data. **Neural Computing and Applications**, Springer, v. 33, p. 2015–2033, 2021.

HOU, Zhong-Sheng; WANG, Zhuo. From model-based control to data-driven control: Survey, classification and perspective. **Information Sciences**, Elsevier, v. 235, p. 3–35, 2013.

KAELBLING, Leslie Pack; LITTMAN, Michael L; MOORE, Andrew W. Reinforcement learning: A survey. **Journal of artificial intelligence research**, v. 4, p. 237–285, 1996.

KIRK, Donald E. **Optimal control theory: an introduction**. [S.l.]: Courier Corporation, 2004.

KLAUTAU, Aldebaro. Digital Communications and Signal Processing: An Introduction using Octave or Matlab - Unpublished. **Federal University of Pará: Electrical Engineering Programm**, 2020.

LEWIS, Frank L; VRABIE, Draguna; SYRMOS, Vassilis L. **Optimal control**. [S.l.]: John Wiley & Sons, 2012.

LEWIS, Frank L.; VRABIE, Draguna. Reinforcement learning and adaptive dynamic programming for feedback control. **IEEE Circuits and Systems Magazine**, v. 9, n. 3, p. 32–50, 2009.

MARTINS, Flávia Cordeiro; GONTIJO, Danielle Silva; GONÇALVES, Eduardo Nunes. Sıntese de observadores PI baseada em otimizaçao evolutiva multiobjetivo H/H2. **Simpósio Brasileiro de Automação Inteligente-SBAI**, 2019.

MCRUER, Duane T; GRAHAM, Dunstan; ASHKENAS, Irving. **Aircraft dynamics and automatic control**. [S.l.]: Princeton University Press, 2014. v. 740.

NISE, Norman S. **Control systems engineering**. [S.l.]: John Wiley & Sons, 2020.

NOGUEIRA, Carlos Eduardo Durans; SILVEIRA, Antônio da Silva; YAMAGUTI, Nelson Nayoshi Nakamoto; SODRÉ, Lucas de Carvalho. Desenvolvimento e identificação de uma planta MIMO de duas entradas e duas saídas para ensino de sistemas de controle. **cobenge**, 2019.

OGATA, Katsuhiko et al. **Modern control engineering**. [S.l.]: Prentice hall Upper Saddle River, NJ, 2010. v. 5.

OKUYAMA, Yoshifumi. **Discrete control systems**. [S.l.]: Springer, 2014.

OPPENHEIM, Alan V. **Discrete-time signal processing**. [S.l.]: Pearson Education India, 1999.

POSTLETHWAITE, Ian. **Multivariable feedback control: analysis and design**. [S.l.]: John Wiley & Sons, Inc., 1996.

PRESCOTT, Edward C. Should control theory be used for economic stabilization? In: ELSEVIER. CARNEGIE-ROCHESTER Conference Series on Public Policy. [S.l.: s.n.], 1977. v. 7, p. 13–38.

RIVERA, Daniel E; MORARI, Manfred; SKOGESTAD, Sigurd. Internal model control: PID controller design. **Industrial & engineering chemistry process design and development**, ACS Publications, v. 25, n. 1, p. 252–265, 1986.

SEBORG, Dale E; EDGAR, Thomas F; MELLICHAMP, Duncan A; DOYLE III, Francis J. **Process dynamics and control**. [S.l.]: John Wiley & Sons, 2016.

SHABAN, EM; SAYED, H; ABDELHAMID, A. A novel discrete PID+ controller applied to higher order/time delayed nonlinear systems with practical implementation. **International Journal of Dynamics and Control**, Springer, v. 7, p. 888–900, 2019.

SILVA, Daniel Abreu Macedo da; NASCIMENTO, André Cavalcante do; BARROS ARAÚJO, Rejane de; SOUZA MELO, Rogério José de. Generalized Predictive Controller Applied in a Bidirectional DC-DC Converter. In: IEEE. 2021 Brazilian Power Electronics Conference (COBEP). [S.l.: s.n.], 2021. P. 1–6.

SILVEIRA, Antonio; SILVA, Anderson; COELHO, Antonio; REAL, José; SILVA, Orlando. Design and real-time implementation of a wireless autopilot using multivariable predictive generalized minimum variance control in the state-space. **Aerospace Science and Technology**, Elsevier, v. 105, p. 106053, 2020.

SILVEIRA, Antonio da Silva et al. Contribuições ao controle de variância mınima generalizado: abordagem de projeto no espaço de estados. Florianópolis, 2012.

SILVEIRA, ANTONIO S; COELHO, ANTONIO AR. Generalised minimum variance control state-space design. **IET control theory & applications**, IET, v. 5, n. 15, p. 1709–1715, 2011.

SILVEIRA, Antonio S; COELHO, Antonio AR; FRANCA, Aline A; KNIHS, Valter L. Pseudo-PID controller: design, tuning and applications. **IFAC Proceedings Volumes**, Elsevier, v. 45, n. 3, p. 542–547, 2012.

SILVEIRA, Antonio S; RODRIGUEZ, Jaime EN; COELHO, Antonio AR. Robust design of a 2-DOF GMV controller: A direct self-tuning and fuzzy scheduling approach. **ISA transactions**, Elsevier, v. 51, n. 1, p. 13–21, 2012.

SKOGESTAD, Sigurd; POSTLETHWAITE, Ian. **Multivariable feedback control: analysis and design**. [S.l.]: Citeseer, 2007. v. 2.

SONG, Peng; YU, Yueqing; ZHANG, Xuping. A tutorial survey and comparison of impedance control on robotic manipulation. **Robotica**, Cambridge University Press, v. 37, n. 5, p. 801–836, 2019.

STEVENS, Brian L; LEWIS, Frank L; JOHNSON, Eric N. **Aircraft control and simulation: dynamics, controls design, and autonomous systems**. [S.l.]: John Wiley & Sons, 2015.

SUMATHI, Sai; SUREKHA, P; SUREKHA, P. **LabVIEW based advanced instrumentation systems**. [S.l.]: Springer Berlin, 2007. v. 488.

SUTTON, Richard S; BARTO, Andrew G, et al. Reinforcement learning. **Journal of Cognitive Neuroscience**, v. 11, n. 1, p. 126–134, 1999.

SUTTON, Richard S; BARTO, Andrew G. **Reinforcement learning: An introduction**. [S.l.]: MIT press, 2018.

TANG, Yu; REN, Dawei; QIN, Hui; LUO, Chao. New family of explicit structure-dependent integration algorithms with controllable numerical dispersion. **Journal of Engineering Mechanics**, American Society of Civil Engineers, v. 147, n. 3, p. 04021001, 2021.

VICARIO, Francesco. **OKID as a general approach to linear and bilinear system identification**. [S.l.]: Columbia University, 2014.

VISIOLI, Antonio. **Practical PID control**. [S.l.]: Springer Science & Business Media, 2006.

VRABIE, Draguna; LEWIS, Frank. Integral reinforcement learning for online computation of feedback Nash strategies of nonzero-sum differential games. In: IEEE. 49TH IEEE Conference on Decision and Control (CDC). [S.l.: s.n.], 2010. P. 3066–3071.

WATKINS, Christopher JCH; DAYAN, Peter. Q-learning. **Machine learning**, Springer, v. 8, p. 279–292, 1992.

WATKINS, Christopher John Cornish Hellaby. Learning from delayed rewards. King's College, Cambridge United Kingdom, 1989.

XIAO, Bing; CAO, Lu; XU, Shengyuan; LIU, Liang. Robust tracking control of robot manipulators with actuator faults and joint velocity measurement uncertainty. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 25, n. 3, p. 1354–1365, 2020.

YAMAGUTI, Nelson NN; DUTRA, Bruno G; SILVEIRA, Antonio S. Development of a didactic plant and a human-machine interface to compare different digital controllers. In: SIMPÓSIO Brasileiro de Automação Inteligente-SBAI. [S.l.: s.n.], 2021. v. 1.

YAMAGUTI, Nelson NN; SILVA, Daniel AM da; DUTRA, Bruno G; SILVEIRA, Antonio S. Performance and robustness analysis in adaptive and non-adaptive GMVC applied to a MISO process.

YOON, Tae-Woong; CLARKE, David W. Adaptive predictive control of the benchmark plant. **Automatica**, Elsevier, v. 30, n. 4, p. 621–628, 1994.

ZIEGLER, John G; NICHOLS, Nathaniel B. Optimum settings for automatic controllers. **Transactions of the American society of mechanical engineers**, American Society of Mechanical Engineers, v. 64, n. 8, p. 759–765, 1942.