الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

المدرسة الوطنية المتعددة التقنيات
قسم الآلية
École Nationale Polytechnique
Département d'Automatique

End-of-studies project dissertation

for obtaining the State Engineer's degree in Automation and Control

Contribution to fractional-order adaptive control for parallel kinematic manipulators: Application to VELOCE robot

Realized by:
Ms. KHOUMERI Bouchra

Publicly presented and defended on the 4^{th} of July 2024, in front of the jury composed of:

President	Pr. BERKOUK El Madjid	ENP
Promoter	Pr. LADACI Samir	ENP
Examiner	Dr. ACHOUR Hakim	ENP

ENP 2024

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

المدرسة الوطنية المتعددة التقنيات
قسم الآلية
École Nationale Polytechnique
Département d'Automatique

End-of-studies project dissertation

for obtaining the State Engineer's degree in Automation and Control

Contribution to fractional-order adaptive control for parallel kinematic manipulators: Application to VELOCE robot

Realized by:
Ms. KHOUMERI Bouchra

Publicly presented and defended on the 4^{th} of July 2024, in front of the jury composed of:

President	Pr. BERKOUK El Madjid	ENP
Promoter	Pr. LADACI Samir	ENP
Examiner	Dr. ACHOUR Hakim	ENP

ENP 2024

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

المدرسة الوطنية المتعددة التقنيات
قسم الآلية
École Nationale Polytechnique
Département d'Automatique

Mémoire de Projet de Fin d'Études

Pour l'Obtention du Diplôme d'Ingénieur d'État en Automatique

Contribution à la commande adaptative à ordre fractionnaire pour les manipulateurs parallèles : Application au robot VELOCE

Realisé par :
M^{me} KHOUMERI Bouchra

Présenté et soutenu publiquement le 4 Juillet 2024, devant le jury composé de :

Président	Pr. BERKOUK El Madjid	ENP
Promoteur	Pr. LADACI Samir	ENP
Examinateur	Dr. ACHOUR Hakim	ENP

ENP 2024

ملخص

تُركِّز هذه الدراسة على روبوت ECOLEV، وهو مناور متوازي بأربعة درجات حرية. يعتمد استراتيجيات التحكم التكيفية لمواجهة عدم اليقين في الأنظمة غير الخطية مثل المناورات الحركية المتوازية (sMKP). من خلال دمج منهجيات التحكم الكسرية، تقترح الأطروحة متحكمات تكيفية كسرية محسنة—متحكم كسرية مرجعي نموذجي تكيفي مع تغذية راجعة تناسبية-تفاضلية، ومتحكم كسرية $\mathcal{L}_1$ تكيفي مع مصطلحات تغذية أمامية. يهدف ذلك إلى تحسين الأداء وزيادة المتانة.

كلمات مفتاحية :

روبوتات المناولة، المناول السينماتيكي المتوازي PKMs ، روبوت VELOCE ، التحكم بالأوامر الكسرية، متحكم تكيفي بالمرجعية النموذجية ، متحكم $\mathcal{L}_1$ التكيفي، التحكم القائم على النموذج.

Résumé

Cette étude aborde l'intégration des robots manipulateurs dans les environnements industriels pour gérer des tâches répétitives et dangereuses, en se concentrant sur le robot VELOCE, un manipulateur parallèle à quatre degrés de liberté. En intégrant des méthodologies de contrôle fractionnaire, la thèse propose des contrôleurs adaptatifs améliorés—un contrôleur adaptatif de référence modélisé avec une rétroaction proportionnelle-dérivée et un contrôleur adaptatif fractionnaire $\mathcal{L}_1$ avec des termes d'alimentation directe. L'objectif est d'optimiser la performance et d'améliorer la robustesse.

Mots-clés :
Robots manipulateurs, Manipulateurs cinématiques parallèles (PKMs), robot VELOCE, contrôle à ordre fractionnaire, contrôleur adaptatif à référence de modèle (MRAC), contrôleur adaptatif $\mathcal{L}_1$.

Abstract

This study focuses on the VELOCE robot, a 4-degree-of-freedom parallel manipulator. By incorporating fractional order control methodologies, the thesis proposes enhanced fractional adaptive controllers—a fractional Model Reference Adaptive Controller with Proportional-Derivative feedback and a fractional $\mathcal{L}_1$ adaptive controller with feed-forward terms. The aim is to optimize performance and improve robustness.

Keywords :
Manipulator robots, Parallel Kinematic Manipulators (PKMs), VELOCE robot, Fractional order control, Model Reference Adaptive Controller (MRAC), $\mathcal{L}_1$ adaptive controller.

Dedication

“

To my source of joy and the balm to my wounds, my dear mother,

To my source of courage and determination, my dear father,

To my second mother and unwavering support, my Aunt Atika,

To the sister I never had and my confidante, my cousin Jouhaina,

To my best friends, Katia and Malak, whose friendship I cherish and hope will grow stronger and withstand all of life's challenges,

To Afaf and Isra, with whom I shared this journey and whose companionship has been invaluable,

To all my classmates, who every day inspire me and motivate me to strive for more.

”

- Bouchra

Acknowledgement

I am immensely thankful to everyone who contributed to the completion of this thesis. My heartfelt appreciation goes to my supervisor, Pr. LADACI, whose expertise, guidance, and unwavering support were indispensable throughout my research. His profound insights and constructive critiques greatly influenced the direction of this work.

I also extend my sincere thanks to the jury members, Pr. BERKOUK El Madjid and Dr. ACHOUR Hakim, for their meticulous evaluation and valuable feedback. Their insightful critiques and thoughtful suggestions played significantly enhanced the quality of this research.

Finally, my deepest gratitude goes to my family and friends for their unconditional love and encouragement throughout this journey.

Contents

List of Figures

List of Tables

List of Abbreviations

FK	Forward Kinematic
FO	Fractional Order
IAE	Integral Absolute Error
ITAE	Integral Time-weighted Absolute Error
IUE	Integral of Absolute Torques
IK	Inverse Kinematic
IO	Integer Order
L1-FF	L1 adaptive controller with feed-forward model-based term
MIT-rule	Massachusetts Institute of Technology rule
MPC	Model Predictive Control
MRAC-PD	Model Reference Adaptive Controller with PD feedback
PD-FF	Proportional Derivative controller with feed-forward term
PKM	Parallel Kinematic Manipulator
RMSE	Root Mean Squared Error

General Introduction

The emergence of manipulator robots in industry was driven by the need to handle repetitive and arduous tasks, as well as to operate in environments hostile to humans, such as high-pressure underwater settings, low-pressure space exploration, high-temperature areas, and high-radiation zones. A robot is an automatic device capable of manipulating objects or performing operations difficult for humans according to a fixed, modifiable, or adaptable program. Among these manipulators, Parallel Kinematic Manipulators (PKMs) are general-purpose industrial robots that meet all these requirements with high accuracy. PKMs are particularly suited for tasks requiring both heavy load carrying and precise positioning due to their distinctive structure, which allows the distribution of the load across multiple parts, enhancing the robot's rigidity.

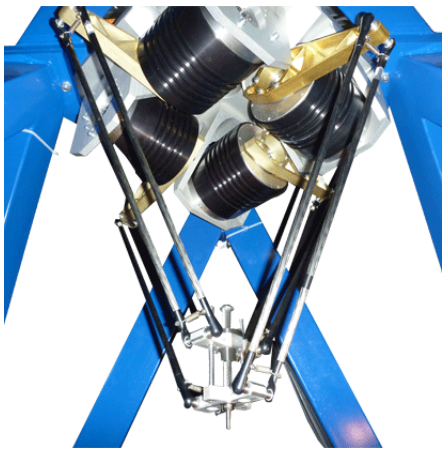


Figure 0.1: The VELOCE parallel kinematic manipulator, from [55]

In this work, we will consider the VELOCE robot, a 4-DOF parallel manipulator mainly designed for pick-and-place applications (see Figure 0.1). The VELOCE system comprises four kinematic chains enabling three independent translational degrees and one rotational degree around the vertical axis [1].

Although classical controllers such as Proportional Integral Derivative (PID) controllers and their variations, as well as Model Predictive Control (MPC), are widely used in industrial manipulators (robot arms), depending on the tasks to be performed, they do not always lead to optimized performance. This is because these types of controllers require precise knowledge of the complete model of the system, rendering them sensitive to parametric uncertainties and unmodeled dynamics. This limitation is particularly evident in highly nonlinear systems like PKMs, which are multi-articulated mechanical systems with dynamics that can be poorly defined, with variable parameters and subject to external disturbances. Consequently, there is a growing interest among the manufacturers and researchers in exploring alternative control strategies.

Adaptive control methods have emerged as promising control approaches for PKMs, capable of real-time uncertainty compensation and enhancing closed-loop performance and tracking accuracy [2]. Originating in aerospace applications in the 1950s, adaptive control has evolved to address uncertainties in various complex systems, including robotics, by adapting to dynamic environment changes and model inaccuracies [3].

Recently, fractional order control (FOC), meaning control of dynamic systems where

the system to be controlled and/or the regulator are governed by fractional differential equations, has peaked the interest of numerous researchers, mainly because of its ability to provide more accurate descriptions of complex dynamical systems and the improvements of robustness and delay margins. For instance in [4], an FO-PID controller was implemented on an industrial robot, achieving high accuracy and precision, as well as in [5], where FO control was applied to for frequency Regulation in Multi-Area Power Systems.

The combination of both adaptive and fractional theory for systems like robot manipulators has been proposed numerous times in multiple works, since the integration of FO theory on adaptive controllers has proven to enhance system robustness, improve the accuracy of adaptation, and decrease sensitivity to high-frequency noise as stated in [6] and [7].

This thesis proposes two adaptive control approaches for the VELOCE manipulator: a Model Reference Adaptive Controller with Proportional-Derivative (MRAC-PD) feedback and an $\mathcal{L}_1$ adaptive controller with feed-forward terms. Both methods will be enhanced by integrating fractional order adjustment laws into their adaptation mechanisms, aiming to improve performance and robustness in terms of reduced error, elimination of time lag, and decreased energy consumption.

This thesis will be structured according to the following outline:

In Chapter 1, parallel manipulators are introduced along with some key terminology and basic concepts. The historical evolution of PKMs and their integration across diverse applications are explored. Than a classification of PKMs will be provided, with emphasis on 4-degree-of-freedom (4-DOF) manipulators akin to the VELOCE robot studied in this thesis. Additionally, a state of the art of the key control strategies proposed in the literature for PKMs will be discussed and categorized into adaptive and non-adaptive approaches.

Chapter 2 will focus on modeling the dynamics and kinematics of the VELOCE manipulator. It will detail the derivation of model parameters based on the VELOCE prototype, discuss trajectory generation, and propose a reference trajectory for subsequent simulations. To validate the VELOCE robot model, a Proportional Derivative controller with a feed-forward term (PD-FF) is implemented, and the simulation results are presented and analyzed.

In Chapter 3, the theoretical framework and essential background on the two adaptive control methods proposed for controlling the VELOCE robot are provided. Beginning with an overview of MRAC methods, including their historical context and fundamental concepts, the development of an MRAC control law featuring MIT-based adaptation rules for second-order systems is detailed. This method will be utilized in Chapter 4 for adaptive control of the VELOCE robot. Subsequently, the $\mathcal{L}_1$ adaptive control method is introduced, outlining its architecture and explaining the design process of the control law. This approach will be implemented in Chapter 6 for controlling the VELOCE robot.

Chapter 4 introduces the basics of FO calculus theory, covering its origins, typical applications, and approximation methods such as the Singularity method (Charef's approximation). Numerical examples are provided to illustrate FO system characteristics and the operational principles of the Singularity technique.

Chapter 5 introduces the Fractional Order Model Reference Adaptive Controller (MRAC) with a Proportional-Derivative (PD) feedback for the VELOCE robot. It explains the rationale behind choosing this control scheme, outlines the control objectives, and details the theoretical foundation of the control law with integrated FO adjustment rules. The chapter also covers the simulation setup and presents results under both nominal conditions and varying payload scenarios.

In Chapter 6, the Fractional Order $\mathcal{L}_1$ adaptive controller is designed with model-based feedforward $\mathcal{L}_1$-FF for the VELOCE robot. It outlines the motivation behind opting for this approach and states the control objectives, then presents the theoretical basis of the control law with FO adaptation and FO filters, and details simulation setup and results under both nominal conditions and varying payloads.

This thesis is concluded with an overview of the key results obtained through this work. Some points for exploration and improvements in future research are also provided.

Chapter 1

Overview on Parallel Kinematic Manipulators (PKMs)

1 Introduction

In this chapter, parallel manipulators are introduced along with essential terminology and concepts to facilitate their study. We explore the historical evolution of parallel kinematic manipulators (PKMs), highlighting key milestones in their integration into diverse applications, from industrial manufacturing to medical and aerospace domains.

Many types of PKMs have been developed throughout history, aiming to exploit their advantages in terms of high speed and accuracy capabilities, we provide a comprehensive classification based on their architectural designs with a particular emphasis on 4-DOF manipulators that share similarities with the VELOCE robot

Given the significant research focus on parallel manipulators, this chapter provides a general overview of the most relevant control strategies proposed in the literature, categorizing them into adaptive and non-adaptive approaches.

2 Basic concepts and terminology

2.1 Definition of PKMs

Parallel robots, also called *parallel manipulators* or *parallel kinematic machines (PKM)*, are defined in Leinonen [8] as:

"*Robots that control the motion of their end-effectors by means of at least two kinematic chains going from the end-effector towards the fixed base.* "

Also according to Merlet [9], a generalized PKM can be defined as such:

" *A generalized parallel manipulator is a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains.*"

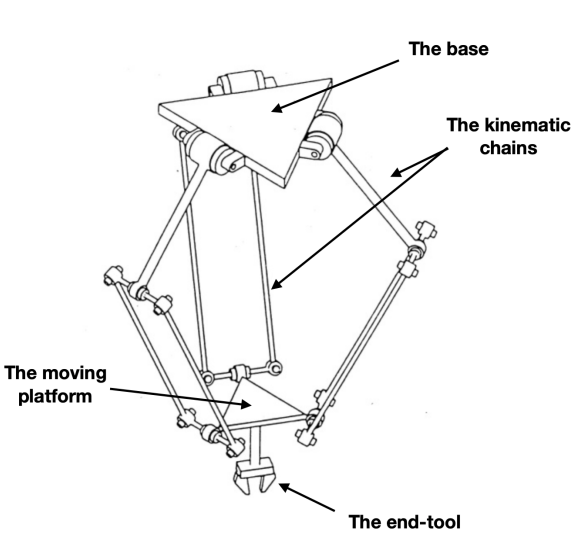


Figure 1.1: Key Components of Parallel Kinematic Manipulators, from Angeles et al

Based on these definitions, the main components of PKMs can be described as follow: The **base** (fixed part), which is the stationary component of the robot; the **platform** (moving part), on which the end-effector is typically mounted; the **end-tool** which is

responsible for performing the desired task; and the **kinematic chains**, also known as the robot legs, which link the base to the platform.
This particular architecture of PKMs allows for the distribution of forces and loads across multiple chains, thereby increasing their rigidity. Such characteristics can provide advantages in terms of speed, accuracy, and payload capacity [10].

2.2 Advantages and disadvantages of PKMs

Parallel Kinematic Machines (PKMs) are renowned for their high accuracy and precision. Firstly, the end-effector is connected to the base through multiple kinematic chains, each bearing a fraction of the total load. This distribution allows for the use of less powerful actuators, which enhances the payload-to-weight ratio and increases stiffness [9]. Additionally, the heavy actuators are typically installed on a single base platform, while the moving arms are kept lightweight. This concentration of mass reduces the overall moment of inertia, enabling faster movements. Furthermore, the closed-loop structure of PKMs prevents error accumulation along the kinematic chains, ensuring consistent precision and accuracy.

From an economic perspective, PKMs are advantageous due to their use of simple, common components, which make them cost-effective to design and build. Furthermore, the use of identical kinematic chains reduces maintenance costs and the need for extensive spare parts storage [10]. This combination of performance and cost-efficiency makes PKMs a valuable choice in various industrial applications.

On the other hand, PKMs have some imitations such as smaller accessible space for the moving platform compared to the overall size of the machine [11]. Additionally, the higher number of passive joints in PKMs, relatively to the active joints linked to the driving actuators, increases the likelihood of collisions between the legs, especially in PKMs with higher degrees of freedom, such as hexapods. Another disadvantage of PKMs is the complexity of their singularity analysis [12]. At singular positions, PKMs completely lose stiffness, becoming unstable and potentially sustaining damage. This singularity issue restricts the range of movement for the robot's orientation angles, typically to less than $\pm 30°$. These limitations in workspace and movement range, along with the risk of instability at singularities, present significant challenges for the effective use of PKMs in certain applications.

2.3 Basic notions in robotics

In order to discuss the control and design of PKMs effectively, it is essential to establish a clear understanding of the fundamental concepts and terminology associated with these systems. This section defines the basic concepts related to the study of PKMs based on [13].

- **Degree of freedom:** The degrees of freedom of a manipulator are defined as the number of independent components of its twist i.e the independent movements the end-effector can perform.

- **Kinematic chains:** A kinematic chain is a series of rigid bodies (links) connected by joints. In PKMs, each chain connects the fixed base to the moving platform, forming a closed-loop structure. These chains, also called "legs," can be serial or tree-structured and composed of various joint types like revolute, prismatic, or spherical joints. The kinematic chains are critical in determining the motion capabilities and structural integrity of the manipulator.

- **Jacobian matrix:** Jacobian relates the linear and angular velocity of the end-effector to the vector of joint velocities. This is a time-varying, position dependent linear transform. It has a number of columns equal to the number of degrees of freedom in joint space, and a number of rows equal to the number of degrees of freedom in Cartesian space.

- **Singular configuration:** Singularities are configurations where the PKM loses stiffness and becomes unstable i.e the Jacobian matrix loses rank, leading to a loss of one or more degrees of freedom. Moreover, locally, the robot mechanical performance (stiffness, accuracy, etc.) are deteriorated. These configurations limit the movement of the robot.

- **Workspace:** The workspace of a PKM refers to the volume or area within which the end-effector can operate without entering a singularity configuration It is determined by the lengths and geometric configurations of the kinematic chains the robot is composed of.

- **Joints:** Joints are key components that connect the rigid links of a robotic arm, enabling it to move in different directions. There are two types of joints: active (actuated) and passive (non-actuated) joints. Depending on the movement they allow, they are classified into: prismatic (translational motion along a single axis), revolute (rotation around one axis) and universal (two independent rotations around two axes) joints.

- **Cartesian space:** It is the three-dimensional reference where the orientation and position of the end-effector are described.

- **Joint space:** It is the space where the robot's movements are described in terms of the joint angles and joint velocities. Its dimension depends on the DoF of the robot. The joint space is typically used for control and analysis of the robot's movements.

3 Historical background

Although the first parallel robot, designed by James E. Gwinnett in 1928 [14], was created for the entertainment industry as a motion platform for a movie theater, the development of PKMs is now often viewed as an alternative to serial manipulators. These systems address the shortcomings of serial manipulators and meet the high-speed and high-accuracy performance demands of industrial applications. In 1942, the next parallel manipulator was developed by Willard L.V. Pollard and aimed at industrial use [15]. This design is considered the first spatial industrial robot, intended for spray painting. It featured three chains driven by rotary motors, connected to the tool head using universal joints, allowing

for three degrees of freedom (3DoF). Unfortunately, receiving little interest from industry manufacturers at that time, this design was never turned into a prototype.

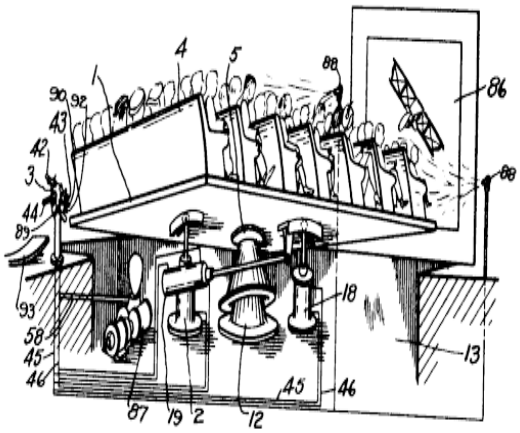


Figure 1.2: Amusement device based on a spherical PKM proposed by James E. Gwinnett, from [14]

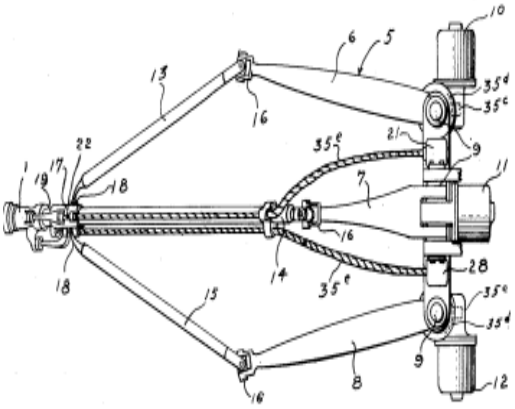


Figure 1.3: The spray painting 5-DOF parallel robot proposed in 1942 by Willard L.V. Pollard, from [15]

In 1954, Dr. Eric Gough introduced the "Universal Tyre Testing Machine," or simply the "Universal Rig," a mechanism with a closed-loop kinematic structure that allows the positioning and orientation of a moving platform [16]. A prototype of this machine was built for testing tire performance. Systems of this type are known under the acronym MAST, which stands for Multi-Axis Simulation (or Shake) Table, and are still manufactured by numerous companies nowadays. They have been used across multiple fields, including aerospace for aircraft and spacecraft simulation, surgical robots, rehabilitation devices, and as stable platforms in mobile robots.

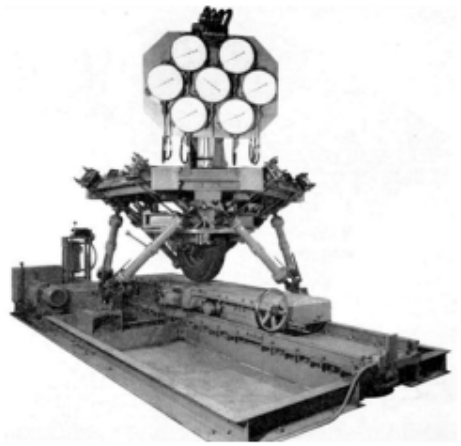


Figure 1.4: The original Gough platform in 1954, from Stewart et al

Figure 1.5: The first flight simulator based on a hexapod structure as in the mid of 1960s, from [16]

During the 1960s, the increase in the cost of pilot training, and the need to test new equipment without flying encouraged D. Stewart to develop the 'Stewart platform' or

'hexapod,' a 6-DOF robot designed for flight simulation [17].

The Stewart platform consists of a fixed base and a movable platform connected by six extendable legs, each with universal joints at both ends. This configuration allows the platform to move with six degrees of freedom: three translational and three rotational. Another device of this type for flight simulation applications was built by Klaus Cappel in the mid-sixties (Fig 1.5).

In the early 1980s, Reymond Clavel, a professor at the École Polytechnique Fédérale de Lausanne (EPFL), introduced the Delta robot, which revolutionized the parallel manipulators industry and broadened their applications [18]. The Delta robot features a stationary base connected to an end-effector platform by three lightweight arms with a shape of parallelograms, these arms ensures the end-effector remains parallel to the base at all times. This design enables high-speed and high-precision movements with minimal inertia, making Delta robots particularly well-suited for tasks such as pick-and-place operations, assembly, packaging, and sorting in manufacturing and pharmaceutical industries.

In addition to these parallel robots, there has been a recent surge in the development of parallel mechanisms that use rotary joints instead of prismatic joints. Some examples include the "Hexa" robot developed by Uchiyama in 1994, the Adept Quattro, a 4-DOF parallel robot, and the ABB Flexpicker, another 4-DOF parallel robot developed by ABB.

Figure 1.6: IRB 360 FlexPicker (ABB technology)

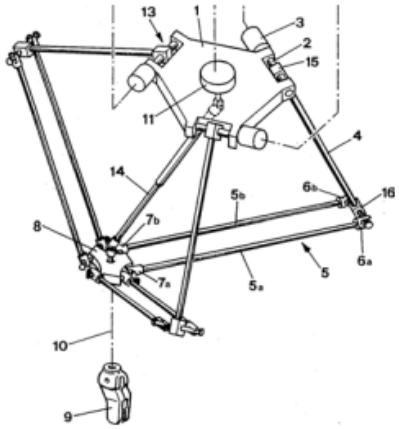


Figure 1.7: Technical drawing of the original Delta robot, from US patent No. 4,976,582

Currently, Japan and Europe are at the forefront of the parallel robot production market, while China is the largest consumer of parallel robots. The majority of parallel robots are produced by companies such as ABB, Fanuc, and Yaskawa [19].

4 Typical applications of PKMs

- **Industrial Applications**
  Parallel robots have rapidly found widespread applications in various industrial domains due to their high performance in precise positioning, accuracy, inherent rigidity, and high load-to-mass ratio. Some prominent applications of parallel kinematic machines (PKMs) in the industrial field are outlined below:

– **Machine Tools:** A machine tool is a power-driven device, such as a lathe, mill, or grinder, used for cutting, shaping, and altering rigid materials, primarily metals. As the industry trends towards high-speed operations, parallel manipulators are increasingly being integrated into these machines. There are three primary types of PKM-based machine tools, categorized by their actuation types including those with prismatic joints and variable leg lengths (as in Fig 1.8), revolute joints and fixed leg lengths, and linear actuators with fixed leg lengths [20].

Figure 1.8: Cosmo Center PM-600: PKM-machine based tool by Okuma corp

Figure 1.9: Pick and place robot arm by HNC-RDT

– **Laser cutting:** Laser cutting is a technology widely used in various industries to cut through materials using a high-powered laser beam. This process requires high flexibility to perform complex patterns and high accelerations around the edges. Consequently, PKM (Parallel Kinematic Machine) tools are well-suited for such applications due to their precision and agility. Several studies have aimed to improve these machines and address related issues [21], [22].

– **Pick and place:** In the industrial context, the pick-and-place process refers to the automation of tasks that involve picking up objects from one location and placing them in another predetermined location [23]. Some of the most popular parallel manipulators for pick-and-place applications include the Delta robot, Par4, and Heli4. These robots are favored for their high-speed performance, precision, and efficiency, making them ideal for tasks such as assembling electronic components, packaging, and sorting (Fig 1.9).

- **Medical Applications**
  Even though PKMs are commonly used in the industry field, there have been some research efforts to integrate them in the medical field. One notable application is the use of Delta-like robots for cardiopulmonary resuscitation (CPR) on patients undergoing cardiac arrest [24]. PKMs are also employed in robot-assisted surgery (RAS). One commonly known example is the PI M-850 Hexapod (Fig 1.10), a 6-DOF parallel-kinematics micro-positioning system, which is used to accurately position an instrument tip with a specific orientation for spine surgery [25].

  The application of PKMs in rehabilitation has also been explored. For example, a spherical parallel manipulator with three degrees of freedom has been developed

Figure 1.10: M-850 Hexapod: a PKM for spine surgery by PI corp

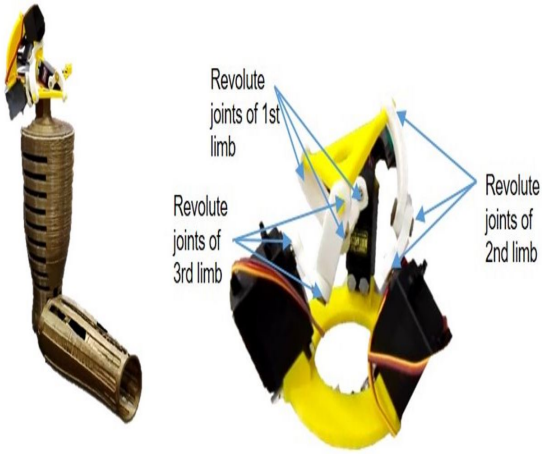


Figure 1.11: 4B-SPM shoulder exoskeleton prototype by Hunt et al

to replace the shoulder mechanism of a prosthetic human arm [26], enhancing the functionality and mobility of the prosthetic (Fig 1.11).

- **Space and Aerospace Applications**
  Parallel manipulators have been used in flight simulations since their inception, driving significant advancements in the field. Most flight simulators are improved versions of the Gough platform, and they are utilized in training centers provided by companies like Airbus, FlightSafety International, CAE (Fig 1.13), and Boeing Training.

  Today, many companies are developing virtual reality motion simulators for various applications beyond aircraft, including ships, trains, and truck driving. Space simulators are also a significant application, used both as terrestrial devices to simulate zero gravity and as onboard devices [27].

  In satellite instrumentation, parallel structures are also employed. For example, a hexapod mounted on the International Space Station (ISS) was developed by ADS International for the European Space Agency (ESA). In the aerospace industry, PKMs have proven to be highly efficient for tasks such as aircraft wing assembly. They enable the production of high-performance, lightweight, and complex components while reducing costs and environmental impact (as Exechon robot).

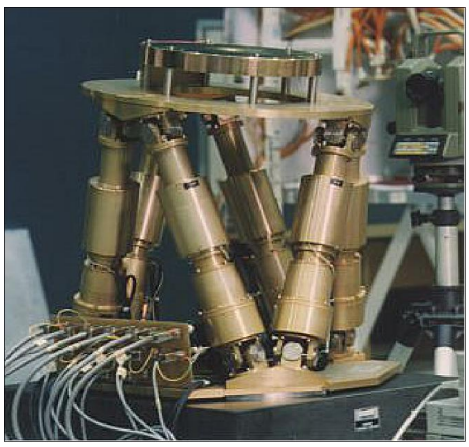


Figure 1.12: Hexapod PKM for satellite instrumentation by ADS International

Figure 1.13: CAE flight simulator based on PKM by CAE Inc

5 Classification of PKMs

Parallel manipulators can be classified according to their nature of motion in planar (PPMs), spatial (SPMs) or Spherical parallel manipulators [10].

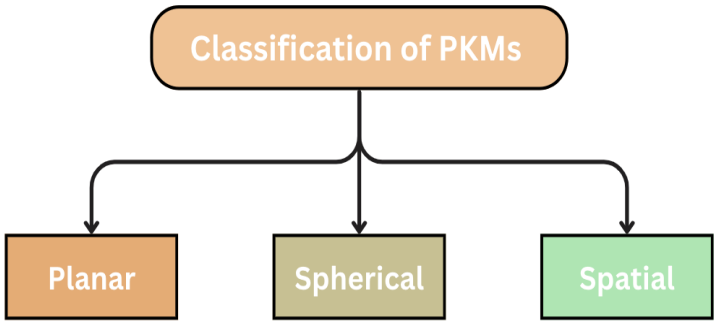


Figure 1.14: Classification of PKMs according to nature of their motion

- **Planar parallel manipulator (PPM):** They consist of two or more planar kinematic chains working together on a common rigid platform. Different types of manipulators can be created depending on the nature of these chains. The motion of all elements within a PPM occurs in the same plane or in parallel planes [28]. The Dextar is an example of a planar five-bar mechanism designed at ETS Montrèal as shown in Fig 1.15.

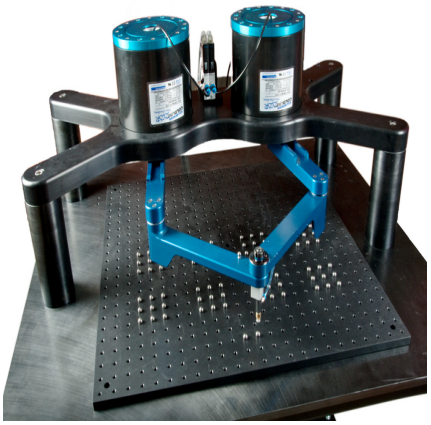


Figure 1.15: Dextar robot: 5-links parallel manipulator developed by Gosselin et al

- **Spatial parallel manipulator (SPM):** The large majority of PKM have been designed to allow mobility of their platforms in 3D space; i.e the end effector can perform both translational and rotational motions. Many SPMs are designed with fewer than 6-DOF to reduce complexity and cost while still meeting application requirements [29]. Some examples include robots with three translational DOF such as the Delta robot and the Tripteron developed by Gosselin (Fig 1.16), and also robots with some DOF of rotation which are constrained with the DOF of translation (Fig 1.17).
- **Spherical parallel manipulator:** Spherical manipulators are able to generate movement of their end effector through controlled motion of the spherical joints.

Figure 1.16: The Tripteron robot: spatial manipulator with 3 translational DOF developed by C.M Gosselin et al

Figure 1.17: The Exechon robot: spatial manipulator with 3 exotic DOF by Exechon Enterprises LLC company

These robots possess three rotational DOF, most of them allow the platform to rotate around one given fixed point [30]. The most known is the Agile Eye developed by Gosselin in 1996 (Fig 1.18). Another way to classify parallel manipulators is based on the degrees of freedom (DOF) of their end-effector, such as 2DOF, 3DOF, 4DOF, and so on.

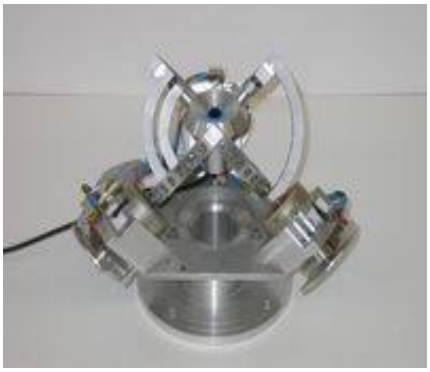


Figure 1.18: The Agile Eye: spherical parallel manipulator with 3DOF, developed by C.M Gosselin et al

Since in this work, we will focus on the VELOCE robot, a 4DOF spatial parallel manipulator with three translations and one rotation around a vertical axis, the major families of 4DOF parallel manipulators will be presented to provide a comprehensive understanding of their characteristics.

- **The H4 manipulator:**
  The H4 manipulator is a fully parallel mechanism (Fig 1.19). The primary distinction between the H4 and most other parallel manipulators lies in the design of the traveling plate. Typically, parallel manipulators consist of three main components: a base, kinematic chains, and a rigid traveling plate. In contrast, the H4 features a traveling plate that is a three-body system connected by two revolute joints. This new design integrates joints into the traveling plate, allowing an increase in the maximum range of rotational motion [31].

Figure 1.19: The H4 manipulator

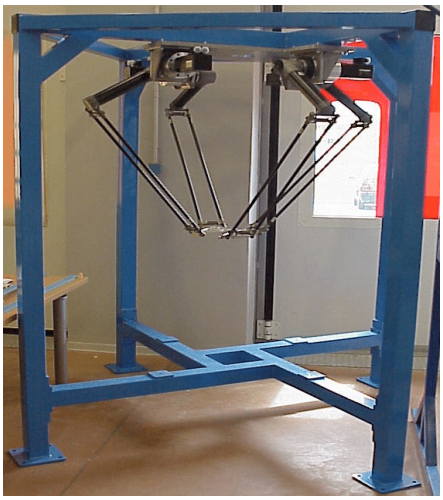


Figure 1.20: The I4 manipulator

- **The I4 manipulator:**
  This family of 4DOF parallel manipulators shares architectural and operational advantages with the H4 family (Fig 1.20). However, it introduces an additional feature: an articulated traveling plate equipped with a gear-based amplification system [32]. This system allows for a larger and adjustable range of orientation for the end-effector, enhancing its flexibility and precision.

- **The Par4 manipulator:**
  Similar to previous parallel robots, the Par4 robot features an articulated traveling plate that eliminates the need for a central telescopic leg. This design consists of four main components: two primary parts connected by two rods through revolute joints, as shown in Fig 1.21. A key modification in the Par4 robot is its moving plate, which takes the form of a planar parallelogram. This structural modification allows for symmetrical positioning of the actuators with the absence of singular configurations [23].

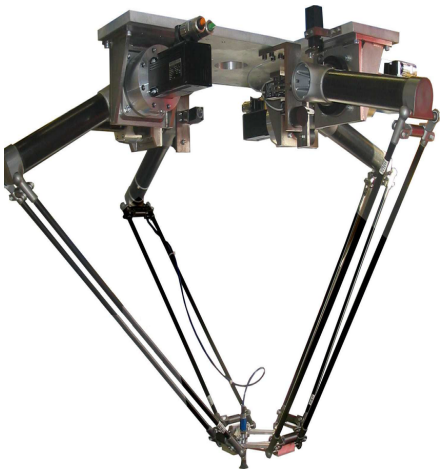


Figure 1.21: The Par4 manipulator

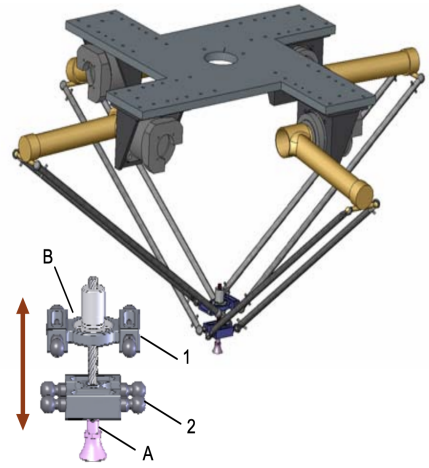


Figure 1.22: The Heli4 manipulator

- **The Heli4 manipulator:**
  On the contrary of the other mentioned manipulators, which possess either an amplification system (H4, Par4) or a motion transformation system (as in I4), the

distinct form of the articulated platform of Heli4 allows for directly producing the the desired rotation amplitude [33]. The platform of the Heli4 robot consists of three parts: an upper(1) and lower (2) platforms, a screw (3), which are connected by a pivot joint (A) on one side and a helical joint (B) on the other side to the two main parts of the platform as presented in Fig 1.22.

6 State of the art

Parallel Kinematic Manipulators (PKMs) offer numerous advantages, motivating researchers to improve their performance aiming to meet specific application requirements. These efforts span across various areas, including design and architecture optimization, development of more accurate and moderately complex kinematic and dynamic models, exploration of motion planning techniques to avoid workspace singularities, and using more advanced control schemes for PKMs.

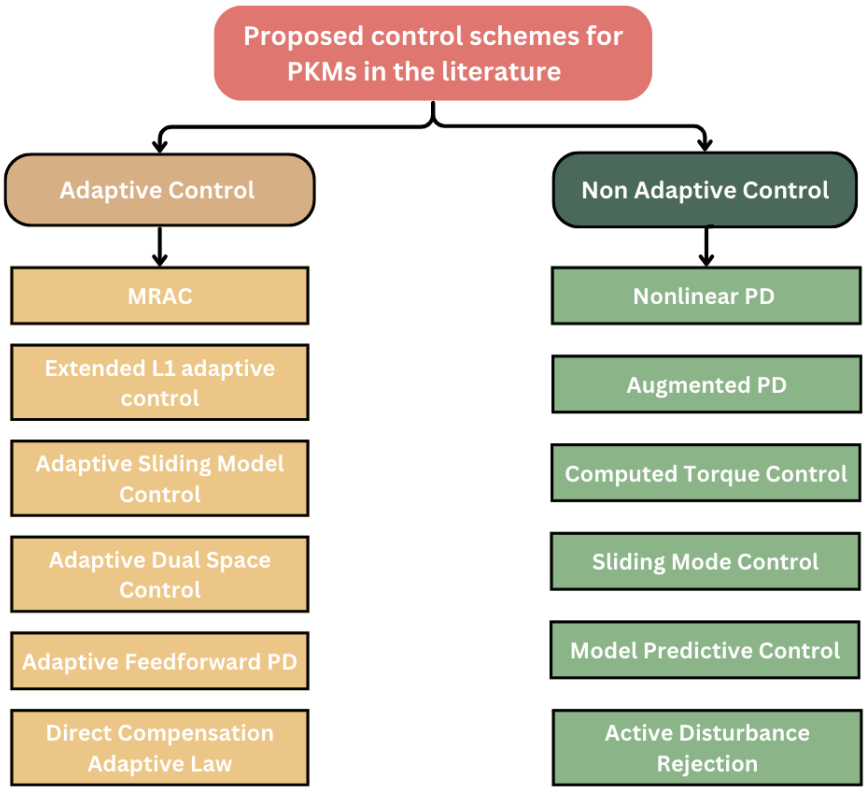


Figure 1.23: Classification of proposed control strategies for PKMs in the literature

Designing an optimal controller for PKMs that achieves both high accuracy and high speed simultaneously necessitates consideration of all aspects of the manipulator, including its nonlinear dynamics and operating environment. This complexity makes the control task for PKMs particularly challenging. Initially, PID controllers, known for their simplicity and widespread use in industrial applications, were applied to PKMs. However, these controllers exhibited limited performance, especially in high-speed tasks, and could not adequately address the advanced challenges presented by PKMs. Consequently, more sophisticated, nonlinear controllers were developed.

Among the challenges faced by PKMs are coupled dynamics due to their closed-loop kinematic structure, which requires precise synchronization between the actuators of each robotic arm. Any failure or discrepancy in one actuator can propagate and destabilize the entire system. Additionally, PKMs exhibit significant nonlinearities under high acceleration conditions, leading to mechanical vibration issues [34]. Furthermore, internal and external uncertainties, such as unmodeled phenomena, parametric variations due to changing operating environments, sensor noise, and component wear, affect controller performance.

Several classifications of control strategies for parallel manipulators exist in the literature, including kinematic versus dynamic controllers, model-based versus non-model-based controllers, and centralized versus decentralized control strategies. For the purpose of this work, we adopt the classification used in [35], which distinguishes between adaptive and non-adaptive control for PKMs (Fig 1.14).

6.1 Non-Adaptive control methods

PKMs have attracted significant interest due to their advantages and suitability for high-speed and precision applications, leading researchers to explore various control methods from the literature. This has resulted in a diverse and rich body of research aimed at improving PKMs performance. Consequently, a wide array of control schemes for parallel manipulators has emerged, each with its own improvements and drawbacks.

The first control methods applied to PKMs were based on classical linear PID controllers due to their practicality and extensive use in industry. Subsequent modifications and improvements to PID-based methods aimed to address the nonlinear and complex dynamics of PKMs. This led to the development of various nonlinear controllers, such as Nonlinear PD (NPD) controllers [36].

Integrating knowledge about the dynamics of manipulators into control schemes resulted in model-based controllers, including augmented PD (APD) [37], adaptive feedforward with PD (AFFPD) [38], fractional-order PID (FOPID) [39], and PD gravity compensation controllers [40]. One significant drawback of these methods is their lack of robustness. To overcome this, other model-based control methods not based on PID controllers were proposed, such as Sliding Mode Control (SMC) [41], which offers fast transient response and robustness against system disturbances and nonparametric uncertainties, Adaptive Disturbance Rejection Control (ADRC) [42], which handles large model uncertainty and unknown nonlinearities, and Computed Torque Control, which linearizes the closed-loop equation in terms of tracking errors [43].

6.2 Adaptive control methods

Adaptive strategies have been developed to address the numerous uncertainties in the controlled system and its environment that may degrade control performance. Adaptive control schemes feature additional parameters estimation loop, based on the dynamics of the system and the tracking errors, allowing real-time parameter updates to find ideal values. These schemes provide precise tracking and maintain desired closed-loop performance despite uncertainties and variations in the PKM's dynamics and environment.

For instance, Model Reference Adaptive Control (MRAC) has been used to control

a 6-DOF parallel manipulator based on the Stewart platform [44]. This control scheme comprises a PD feedback controller combined with an adaptation mechanism to adjust feedback gains. L1 control, proposed as an improvement over MRAC, resolves the coupling problem between adaptation and the control loop, enabling high adaptation rates [45]. Additionally, adaptive control schemes based on the computed torque method, with an adaptation loop to estimate system parameters in real-time, have been tested on PKMs [46].

Recent advancements have focused on hybrid control schemes that blend the strengths of nonlinear adaptive control with nonlinear robust techniques. For instance, the Direct Compensation Law has been modified to incorporate nonlinear time-varying feedback-loop gains instead of constant ones [47]. Furthermore, the addition of adaptive mechanisms for parametric uncertainties has helped reduce high-frequency chattering in classical terminal sliding mode control (TSM) [48]. Another example is the use of adaptive feedback gains in a robust integral sign of the error (RISE) control scheme, which produces corrective actions based on the values of the joint tracking errors [49].

There are also instances where adaptive controls were combined with advanced state estimation techniques – such as Dual-space Control aiming at dampening mechanical vibrations at very high accelerations [50]– or using a linear–quadratic integral regulator with a time-delay estimator with adaptive gains adjusted through artificial neural networks [51].

7 Conclusion

This chapter has provided a comprehensive overview of the diverse modern and typical applications of Parallel Kinematic Machines (PKMs), highlighting their significant advantages, particularly in terms of high stiffness and payload-to-weight ratios. We have traced the evolution of PKMs from simplistic prototypes to more sophisticated architectures that meet the contemporary demands for high-speed and high-accuracy performance.

The nonlinear nature of PKM dynamics, in addition to uncertainties arising from unmodeled dynamics and parametric variations due to changing operating environments, has necessitated the exploration and design of various control strategies. Some of these strategies were discussed in this chapter and categorized into adaptive and non-adaptive methods. This classification provides a clearer understanding of the position of the work presented in this thesis within the broader research landscape on PKMs.

In the next chapter, we will describe the VELOCE robot, highlighting its features and characteristics along with an analysis of its workspace and singularities. Following this, we will provide the kinematic and dynamic models of this robot as developed in prior works. Finally, numerical simulations of a PD with feed-forward term controller (PD-FF) for the VELOCE robot are also included.

Chapter 2

Modeling of VELOCE manipulator

1 Introduction

Developing an adequate model for complex systems is as important as developing the adequate control scheme. However, finding a model that accurately captures the dynamics of PKMs while maintaining simplicity is challenging in part due to their complex geometry and closed-loop structure, as well as the challenge of identifying accurate values for the model parameters [52].
Another key challenge with PKMs is the prevalence of singularities within their workspace, which requires careful consideration when designing and generating trajectories.

This chapter will focus on modeling the dynamics and kinematics of the VELOCE manipulator. It will include the provision of model parameters and a brief description of the VELOCE prototype from which they were identified. Additionally, the trajectory generation problem will be discussed, and a reference trajectory for the VELOCE robot will be proposed for the simulations in the following chapters. Finally, to validate the presented VELOCE robot model, a Proportional Derivative controller with a feed-forward term (PD-FF) will be implemented, and the simulation results will be presented and discussed.

2 Description of VELOCE manipulator

VELOCE is a non-redundant, fully actuated parallel manipulator with 4 DOF (3-Translations, 1-Rotation) i.e the number of its kinematic chains equals the number of its degrees of freedom. It was developed at LIRMM (Laboratory of Informatics, Robotics and Microelectronics of Montpellier) inspired by the architecture of Heli4 manipulators. The goal was to create a more compact and simpler design compared to H4 and I4 manipulators [33].

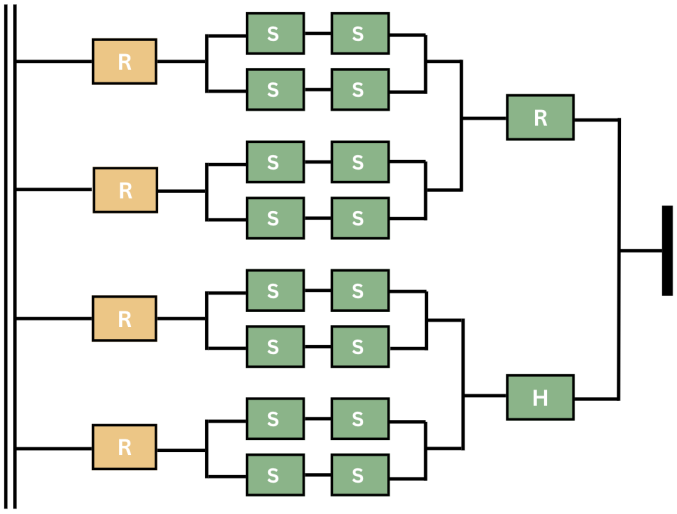


Figure 2.1: Layout graph of VELOCE robot: (S) spherical, (R) revolute, (H) helical joints, from [23]

The VELOCE robot consists of four kinematic chains connected through actuators to a fixed base. Each chain comprises a rear arm and a two-link forearm forming a

parallelogram attached via spherical joints to both the rear arm and the traveling plate (Fig 2.1). Its moving platform composed of two main parts – an upper part and lower part – mounted on one screw (Fig 2.3). As such, we can infer that this moving platform has three independent translational degrees along x, y, and z axes as well as one rotational degree around the z-axis.

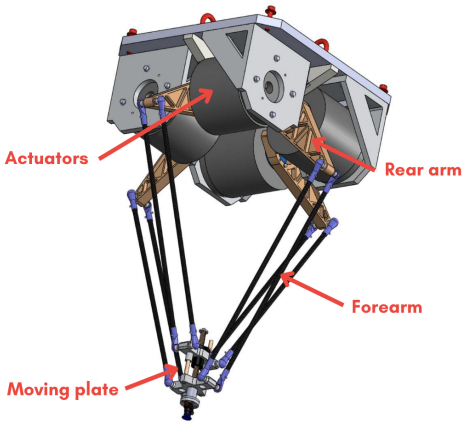


Figure 2.2: CAD view of Veloce manipulator, from [55]

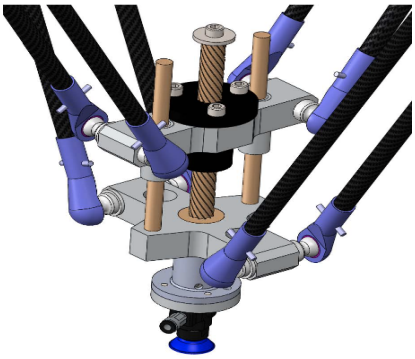


Figure 2.3: Moving platform of Veloce from [53]

The VELOCE manipulator developed at LIRMM and whose model is used in this study has the following characteristics: four direct-drive motors (TMB0140-100-3RBS ETEL) with a maximum torque of 127 Nm and a top speed of 550 rpm. All actuators are equipped with non-contact incremental optical encoders, each providing 5000 pulses per revolution. The overall structure can accommodate a maximum payload of 10 kg, achieve a peak velocity of 10 m/s, and reach a maximum acceleration of 200 m/s2. These capabilities must be considered during the numerical simulations.

3 Previous works on VELOCE

Multiple control strategies have been designed and implemented on the VELOCE manipulator, including linear and non-linear control, robust control, as well as adaptive control both model-based and non-model based. In [53], an $\mathcal{L}_1$ adaptive control scheme was experimentally implemented on the VELOCE robot, demonstrating improved tracking performance compared to a PD controller, due to the compensation of nonlinearities in the manipulator's dynamic model.

Building on this work, subsequent research incorporated an additional dynamics-based term into the L1 control scheme, computed based on the desired robot trajectory. This modification reduced the impact of uncertainties on the closed-loop system [53]. The same principle of integrating a feedforward term based on the system's dynamics was tested with a PD controller in [38], which also involved identifying actuator and friction dynamics parameters using Least Square Estimation.

Further advancements combined an adaptive feedforward term, capable of adjusting its parameters in real-time to variations in the manipulator's dynamics, with a terminal

sliding mode controller [48]. This approach leveraged the linear-in-the-parameters property of the inverse dynamics of the manipulator.

An additional improvement was achieved in [54] by introducing a time-varying adaptation gainto the previous control scheme. This adjustment aimed to overcome the limitations of a constant adaptation gain and enhance parameter estimation. The benefits of the adaptive feedforward approach were also demonstrated in a comparative analysis between non-model-based controllers, as PD, PID, (Nonlinear PD) NPD, and model-based controllers, as (Augmented PD) APD, (Adaptive feed0forward)AFFPD [1]. Additionally, in [55] a fast model predictive controller (NMPC) was proposed with the objective of reducing computation time and enabling high processing speeds.

4 Kinematic Model of VELOCE

The study of parallel robot kinematics involves a comprehensive analysis of the movement and positioning of the end-effector relative to its base. This analysis is fundamental for achieving precise control over the robot's motion and planning its trajectory within a specified workspace. In this section, we present both the forward and inverse kinematics models of the VELOCE robot.

4.1 Inverse Kinematics

The inverse kinematics problem consists of determining the required joint angles for achieving a desired end-effector pose (Fig 2.4).

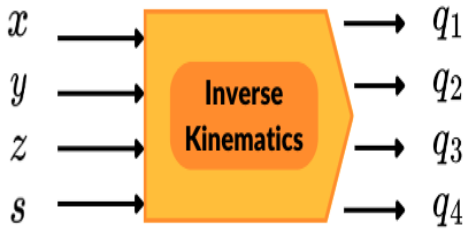


Figure 2.4: Inverse Kinematics Scheme

Since the VELOCE robot has 4 identical mechanical chains, the kinematic model can be derived for one chain (as illustrated in Fig 2.5) and then extended to the others.

The Cartesian coordinates of the moving platform of the manipulator can be expressed with respect to the fixed-base frame in a four-dimensional vector $X = \begin{bmatrix} x & y & z & s \end{bmatrix}^T$, with x, y and z being the translational coordinates and s the rotational amplitude, defined as:

$$s = \frac{p}{\pi}\alpha \tag{2.1}$$

where p is the pitch of the helical joint of the traveling plate and α the rotation angle around the z-axis.

The vector X effectively represents the position and orientation of the end-effector. In the joint space, the angular position of each actuator is represented by an element of a four-dimensional vector $q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$.

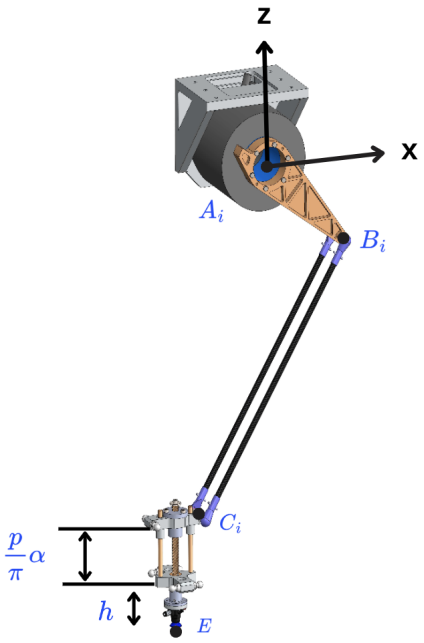


Figure 2.5: VELOCE robot kinematic chain

The angular position of the *i-th* actuator with respect to the fixed frame is given by:

$$\beta_i = (i-1)\frac{\pi}{2} \tag{2.2}$$

To develop the inverse kinematic model the coordinates of the points A_i, B_i and C_i shown in Fig 2.5 are given as follow:

$$A_i = R \begin{bmatrix} cos(\beta_i) & sin(\beta_i) & 0 \end{bmatrix}^T \tag{2.3}$$

$$B_{i_{/O}} = A_i + L \begin{bmatrix} cos(\beta_i)cos(q_i) & cos(\beta_i)sin(q_i) & sin(q_i) \end{bmatrix}^T \tag{2.4}$$

$$C_i = \begin{bmatrix} x + r\,cos(\beta_i) & y + r\,sin(\beta_i) & z + h + \frac{p_i}{\pi}\alpha \end{bmatrix}^T \tag{2.5}$$

with p_i defined as follow:

$$p_i = \begin{cases} 0 & i = 1, 3 \\ p & i = 2, 4 \end{cases}$$

According to [56], finding the inverse kinematics equations of the VELOCE robot involves solving for the intersection of a circle with A_i as its center and L as its radius and a sphere with its center at C_i and l, the length of the forearms, as its radius. This will allow for finding the coordinates of the points B_i from which the values of the actuators joints can be computed using the following equation:

$$q_i = \text{atan2}\left(z_{B_i}, x_{B_i}\right) \tag{2.6}$$

A detailed explanation of the development of the Inverse Kinematic model of the VELOCE robot is provided in Appendix 1.

4.2 Forward Kinematics

The forward kinematic model (FKM) allows us to determine the position and orientation of the end-effector given a specific set of actuated joint values. It usually involves solving a system of nonlinear equations that relate the joint variables to the end-effector pose.

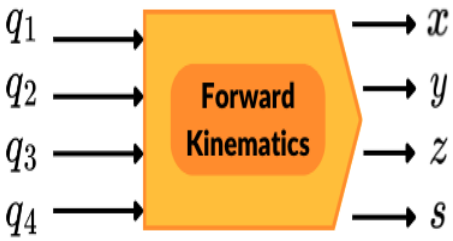


Figure 2.6: Forward Kinematics Scheme for VELOCE Robot

Several methods have been suggested for calculating the FKM of the VELOCE robot, most of which are based on the geometric properties of the system. The method applied here is straightforward to understand and implement in numerical simulations.

Finding the Forward Kinematic model of the VELOCE robot comes to solving the following system of nonlinear equations:

$$\begin{cases} (x+r-x_{B_1})^2+(y-y_{B_1})^2+(z+h-z_{B_1})^2=l_1^2 \\ (x-x_{B_2})^2+(y+r-y_{B_2})^2+\left(z+h+\frac{p}{\pi}\alpha-z_{B_2}\right)^2=l_2^2 \\ (x-r-x_{B_3})^2+(y-y_{B_3})^2+(z+h-z_{B_3})^2=l_3^2 \\ (x-x_{B_4})^2+(y-r-y_{B_4})^2+\left(z+h+\frac{p}{\pi}\alpha-z_{B_4}\right)^2=l_4^2 \end{cases} \tag{2.7}$$

where $l_i \quad \forall i = 1, \cdots, 4$ are the lengths of the forearms.

The unknowns in this system are the coordinates of the end-effector since the coordinates of the points A_i, B_i, and C_i can be directly calculated from the joint angles. By solving the system (2.7), which can be interpreted as the intersection of four spheres, we obtain the coordinates of the end-effector x, y, z, and α.

5 Differential Kinematics of VELOCE

Finding the Jacobian matrix, which represents the relationship between the Cartesian velocity vector $\dot{X}$ and the joint velocity vector $\dot{q}$, is referred to as the differential kinematics of the manipulator:

$$\dot{X} = J\dot{q} = J_x^{-1} J_q \, \dot{q} \tag{2.8}$$

with J_x and J_q calculated using Equations (2.9) and (2.10), respectively:

$$J_x = [J_{x_1} \; J_{x_2} \; J_{x_3} \; J_{x_4}]^T \tag{2.9}$$

where $J_{x_i} = \begin{bmatrix} B_i C_i^\top & (z_{C_i} - z_{B_i})\frac{p_i}{\pi} \end{bmatrix}^T \quad \forall i = 1, \cdots, 4$

$$J_q = \text{diag}\left\{ t_1^\top B_1 C_1, t_2^\top B_2 C_2, t_3^\top B_3 C_3, t_4^\top B_4 C_4 \right\} \tag{2.10}$$

Based on Equation (2.8) we can obtain an expression that relates $\ddot{X}$, the Cartesian accelerations, to $\dot{q}$ and $\ddot{q}$, the joint velocities and accelerations of the actuators, respectively:

$$\ddot{X} = J\ddot{q} + \dot{J}\dot{q} \tag{2.11}$$

6 Dynamic Model of VELOCE

Due to the VELOCE manipulator's many similarities with the Delta robot, similar assumptions are used to develop a simplified dynamics model [57].

- ***Assumption 1:*** *Dry and viscous frictions in passive and active joints are ignored because the joints are specifically engineered to minimize friction effects. This neglects any potential impact from these forms of friction.*
- ***Assumption 2:*** *The rotational inertia of the forearms is ignored, and their weight is divided into two equal parts. One part is included in the arm's mass, while the other part is accounted for with the moving platform. This assumption is supported by the relatively small weight of the forearms compared to other body components.*

The dynamic model can be obtained by analyzing the dynamics in the joint space and the traveling-plate space (Cartesian space) separately, then summing up the two equations of motion [58]. When considering the moving platform, there are two types of forces that affect it: the gravitational forces $G_p \in \mathbb{R}^4$ and the inertial forces $F_p \in \mathbb{R}^4$. These forces can be expressed as:

$$G_p = -M_p \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix}^T \tag{2.12}$$

$$F_p = M_p \ddot{X} \tag{2.13}$$

where $M_p \in \mathbb{R}^{4\times4}$ is the mass matrix of the moving platform, including the half-masses of the forearms, $g = 9.81$ m/s^2 is the acceleration due to gravity, and $\ddot{X} \in \mathbb{R}^4$ is the platform acceleration vector.

The contributions of G_p and F_p to each actuator can be computed using the Jacobian matrix $J(q, X) \in \mathbb{R}^{4\times4}$ as follows:

$$\Gamma_{G_p} = -J^T M_p \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix}^T \tag{2.14}$$

$$\Gamma_p = J^T M_p \ddot{X} \tag{2.15}$$

On the joints side, the dynamics of the actuators are influenced by the torques produced by the movement of the rear-arms in addition to the half-masses of the forearms. The torque contribution resulting from the gravitational forces of the rear-arms is given by Equation (2.16):

$$\Gamma_{G_a} = -Ml_{G_a} g \begin{bmatrix} \cos(q_1) & \cos(q_2) & \cos(q_3) & \cos(q_4) \end{bmatrix}^T \tag{2.16}$$

where $Ml_{G_a} = m_{ra} r_g + m_{fa} l$. Here, m_{ra} represents the rear-arms mass and m_{fa} the forearms mass. Additionally, the inertial forces caused by the acceleration of the rear-arms are expressed in Equation (2.17):

$$\Gamma_{arm} = I_a \ddot{q} \tag{2.17}$$

where $I_a \in \mathbb{R}^{4\times4}$ is a diagonal inertia matrix accounting for the inertia of the rear-arms and the half-masses of the forearms. By applying the virtual work principle, which states that the sum of non-inertial forces is equal to that of the inertial ones, and after rearranging the terms, we get Equation (2.18):

$$\Gamma_p + \Gamma_{G_p} + \Gamma_{arm} + \Gamma_{G_a} = \Gamma \tag{2.18}$$

Using the differential kinematic relationship given in Equation (2.11), the dynamics of the manipulator can be expressed as follows:

$$(I_a + J^T M_p J)\ddot{q} + (J^T M_p \dot{J})\dot{q} - (J^T G_p + \Gamma_{G_a}) + \Gamma_d = \Gamma \tag{2.19}$$

where Γ_d is an input disturbance term that accounts for uncertainties caused by non-modeled dynamics in addition to external disturbances. Equation (2.19) can be written in a standard form that is suitable for joint-space control as it is formulated in terms of the actuated joints' coordinates:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \Gamma_d = \Gamma \tag{2.20}$$

with:
$M(q) = I_a + J^T M_p J$ being the total mass matrix.
$C(q,\dot{q}) = J^T M_p \dot{J}$ being the Coriolis and centrifugal forces matrix.
$G(q) = -\left(J^T G_p + \Gamma_{G_a}\right)$ represents the gravitational force vector.

7 Parameters of the VELOCE robot model

The numerical values of the parameters for the VELOCE model used throughout this work were taken from [53]. These values were obtained through an identification process of the VELOCE prototype at the LIRMM laboratory, described in Section 2.2. The parameters include physical dimensions, inertia, and weights of the robot's parts, as presented in Table 2.1. A schematic drawing of this VELOCE prototype in the Cartesian configuration $X_0 = \begin{bmatrix} 0 & 0 & -0.6 & 0.010 \end{bmatrix}^T$ is shown in Fig 2.7.

Table 2.1: The main dynamic parameters of the VELOCE parallel robot

Parameter	Description	Value
l	Fore-arm length	530 mm
L	Rear-arm length	200 mm
m_{ar}	Rear-arm mass	0.541 Kg
m_{fa}	Fore-arm mass	0.080 Kg
I_a	Rear-arm inertia	$5.27 \times 10^{-3}\ Kg \cdot m^2$
m_{tp}	Traveling plate mass	0.280 Kg
m_s	Main screw mass	0.142 Kg
I_s	Main screw inertia	$7.6 \times 10^{-5}\ Kg \cdot m^2$
R	Frame radius	135 mm
r	Traveling plate radius	48 mm
h	Lower traveling plate length	78 mm
p	Helical joint pitch	$60\ mm \cdot rad^{-1}$

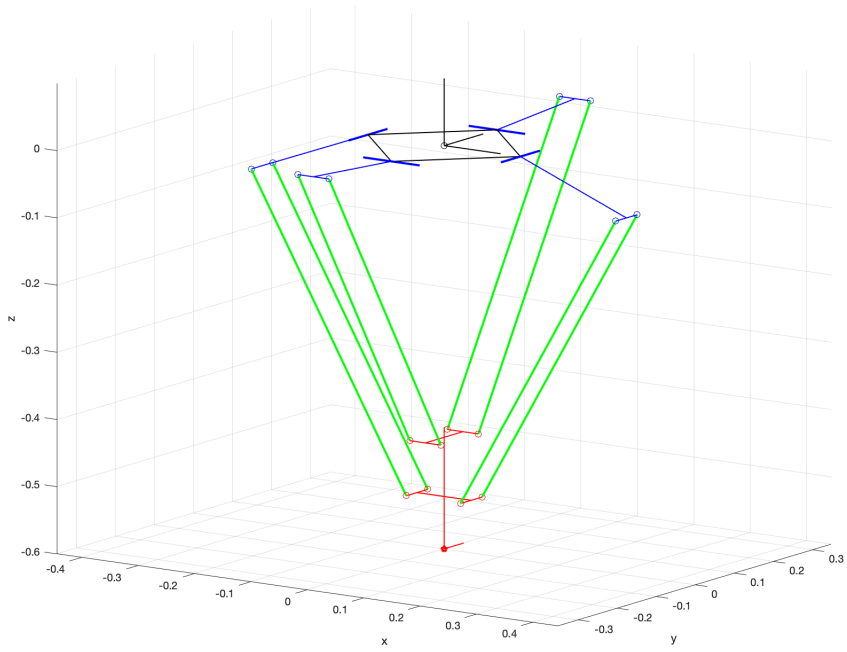


Figure 2.7: Schematic drawing of the VELOCE Robot

The total mass matrix M_p is calculated using the following expression:

$$M_p = diag\begin{bmatrix} M_{tot} & M_{tot} & M_{tot} & \frac{I_{tot}}{p} \end{bmatrix} \tag{2.21}$$

with $M_{tot} = m_{tp} + m_s + 4\frac{m_{fa}}{2}$ and $I_{tot} = I_s$.

8 Workspace and singularities

8.1 Singularities

A singular configuration in PKMs is a point in the workspace at which the end-effector may lose one or more degrees of freedom, these singularities occurs when the Jacobian matrix becomes poorly conditioned and may not be invertible, leading to infinite joint rates while the end-effector remains stationary [59].

According to [9], for parallel manipulators one can distinguish among three different types of singularities: *serial singularities*, *Parallel singularity* and *Combined singularities* . To determine the first two types of singularities it is sufficient to check the determinants of the Jacobian matrices as defined in Equation (2.22).

$$\boldsymbol{J}_x \dot{\boldsymbol{x}} = \boldsymbol{J}_q \dot{\boldsymbol{q}} \tag{2.22}$$

Where J_q and J_x are the joint and Cartesian Jacobian matrices respectively.

For the last type of singularities, according to [33], by validating that the relationship expressed in Equation (2.23) holds across the entire operational space, we can ensure that no "internal singularity" will occur.

$$(((d_1 \times l_1) \times (d_2 \times l_2)) \times ((d_3 \times l_3) \times (d_4 \times l_4)))^{\mathrm{T}} e_z \neq 0 \tag{2.23}$$

where:
d_i are vectors that links the two opposite parts of the travelling platform.
l_i are the vectors joining both extremities of the forearm.

However, for the scope of this thesis, which focuses more on control rather than path planning, the reference trajectories will be chosen to lie within the workspace of the VELOCE manipulator and avoid singularities without the need to use Equation (2.23). A detailed explanation of the trajectory generation process will be provided in Section 9.

8.2 Workspace

The workspace of the VELOCE robot is a crucial aspect that directly impacts its performance and capabilities. The VELOCE robot is designed to operate within a defined workspace that is determined by the physical constraints of its mechanical structure. This workspace typically consists of a three-dimensional volume where the robot can maneuver its end-effector. The dimensions and shape of this workspace are influenced by factors such as the length of the robot's arms, the configuration of its joints, and any physical obstacles in its environment.

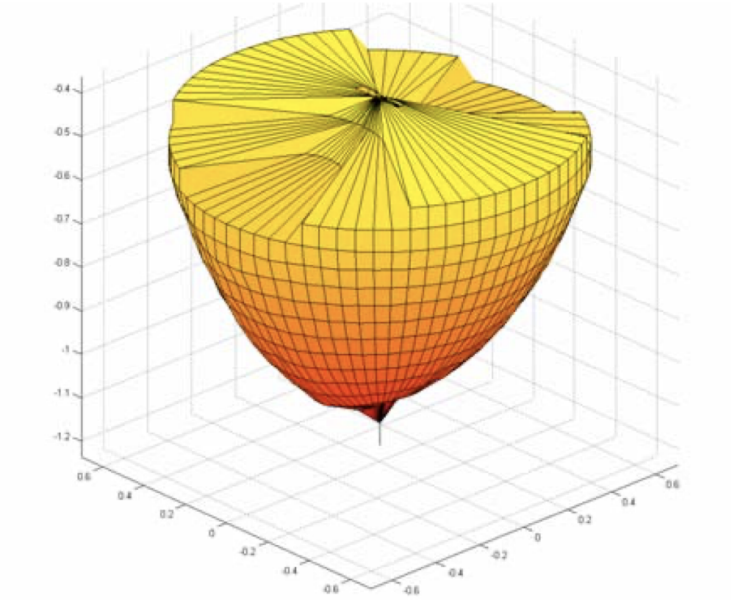


Figure 2.8: Workspace of VELOCE Robot, from [23]

9 Trajectory generation

The problem of path planning in robotics involves defining a sequence of points or a geometric curve. Finding a path that describes the displacement of the end-effector from an initial configuration M_i to a final configuration M_f can be approached in different ways. For some classes, the trajectory is better generated in the joint space, while for others, it is more suitable in the operational space (end-effector configuration). Each approach has its advantages and disadvantages, which are detailed in [52].

According to [10], using the point-to-point technique, the motion profiles of a parallel manipulator can be described by the following expressions:

$$\begin{aligned} x_d(t) &= x_d(t_i) + (x_d(t_f) - x_d(t_i))r(t) \\ \dot{x}_d(t) &= \dot{x}_d(t_i) + (\dot{x}_d(t_f) - \dot{x}_d(t_i))\dot{r}(t) \\ \ddot{x}_d(t) &= \ddot{x}_d(t_i) + (\ddot{x}_d(t_f) - \ddot{x}_d(t_i))\ddot{r}(t) \end{aligned} \tag{2.24}$$

By choosing 5^{th} order polynomials to calculate the path between each consecutive points, the expressions to calculate $r(t)$ is given as follow:

$$r(t) = 10\left(\frac{t-t_i}{T}\right)^3 - 6\left(\frac{t-t_i}{T}\right)^4 + 15\left(\frac{t-t_i}{T}\right)^5 \tag{2.25}$$

where t_i is the starting time and T is the period between each point-to-point movement.

The trajectory needs to be designed away from singularities and always remain within the reachable workspace of the VELOCE robot. An example of a desired trajectory for the end-effector is illustrated in Fig 2.9. The points of the sequence of movements are given in Appendix 6.

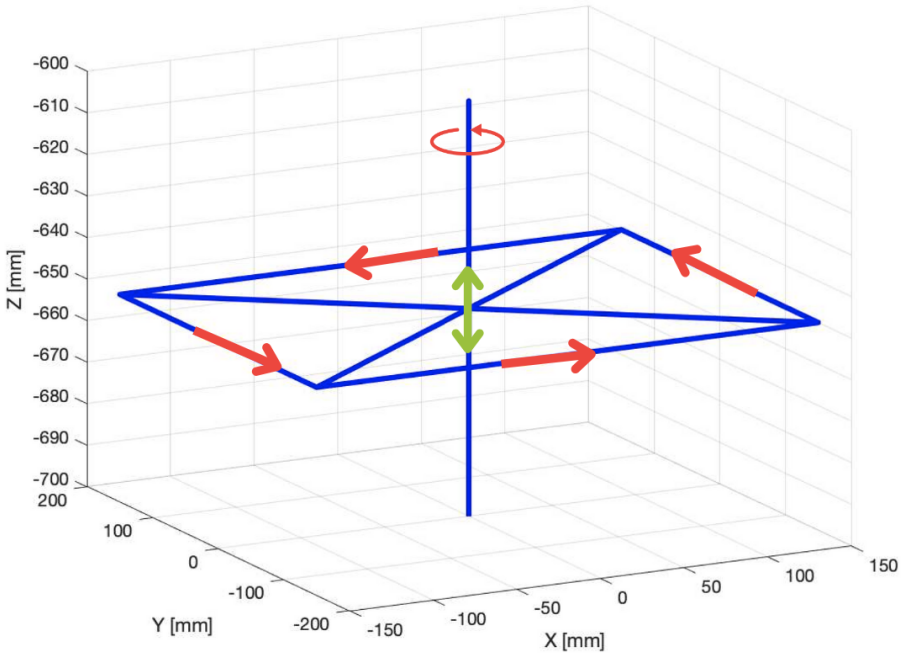


Figure 2.9: 3D view of the desired trajectory for the VELOCE robot

As shown, the trajectory comprises three parts: translational movements in the XY-plane, translation along the Z-plane, and rotations around the vertical axis. Subsequently, the IK model of the VELOCE manipulator will be used to compute the desired trajectory in the joint space.

10 PD-FF controller for VELOCE robot

Before presenting the design of the proposed control schemes for this work, we implement a simple Proportional Derivative (PD) controller with an additional feed-forward term based on the robot's model. The objective is to understand the nonlinear behavior of the VELOCE robot and the nature of its dynamics.

Furthermore, this implementation aims to showcase the limitations of the PD controller, even with a feed-forward term that compensates for most od the uncertainties and non-linearities, in comparison to robust or adaptive control methods. As stated in [38], achieving high performance with simple control methods requires precise and complex models of the manipulator's dynamics, which include modeling additional phenomena such as joint friction and actuator dynamics.

For a more in-depth comparison between various PD controllers applied to the VELOCE robot, [1] presents the advantages and disadvantages of each control scheme.

10.1 Control law

The control law is given by:

$$\Gamma(t) = \Gamma_{PD}(t) + \Gamma_{FF}(t) \tag{2.26}$$

with Γ_{PD} the PD control term and Γ_{FF} the feed-forward control term which are defined in Equations (2.27) and (2.28) respectively:

$$\Gamma_{PD} = K_p\, e + K_d\, \dot{e} \tag{2.27}$$

$$\Gamma_{FF} = M(q_d)\, \ddot{q}_d + C(\dot{q}_d, q_d)\, \dot{q}_d + G(q_d) \tag{2.28}$$

with $\begin{bmatrix} q_d & \dot{q}_d & \ddot{q}_d \end{bmatrix}$ is the desired actuator joints vector obtained using IK model. The tracking error is defined as $e = q_d - q$ and $\mathrm{K_p}, \mathrm{K_d} \in \mathbb{R}^{4\times4}$ are positive definite matrices representing the proportional and derivative gains respectively.

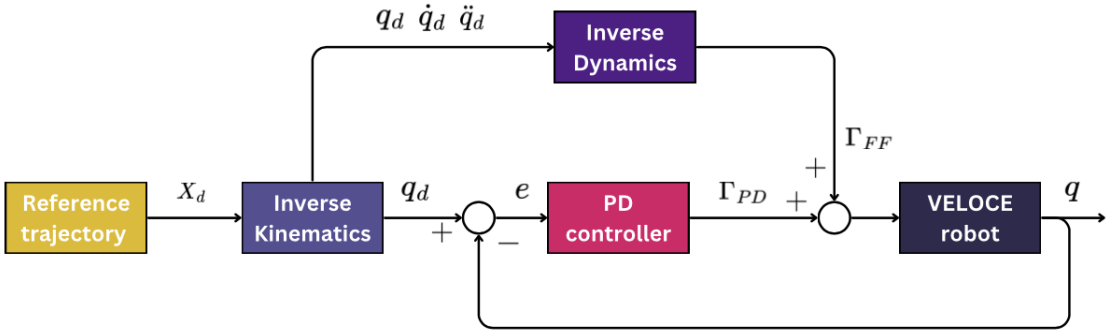


Figure 2.10: Diagram of PD-FF controller scheme

10.2 Simulation results

The proposed PD-FF control scheme will be validated through simulation of a trajectory tracking problem for the VELOCE robot. The reference trajectory presented in Section 9, which consists of a sequence of 22 points connected by 5^{th} order polynomial with a period $T = 0.5\,s$, will be used for these simulations. The values of the controller's parameters were inspired from [38] and are summarised in Table 2.2.

Table 2.2: Parameters of PD-FF controller

Parameters	Value
K_p	diag(600, 600, 600, 600)
K_d	diag(4, 4, 4, 4)

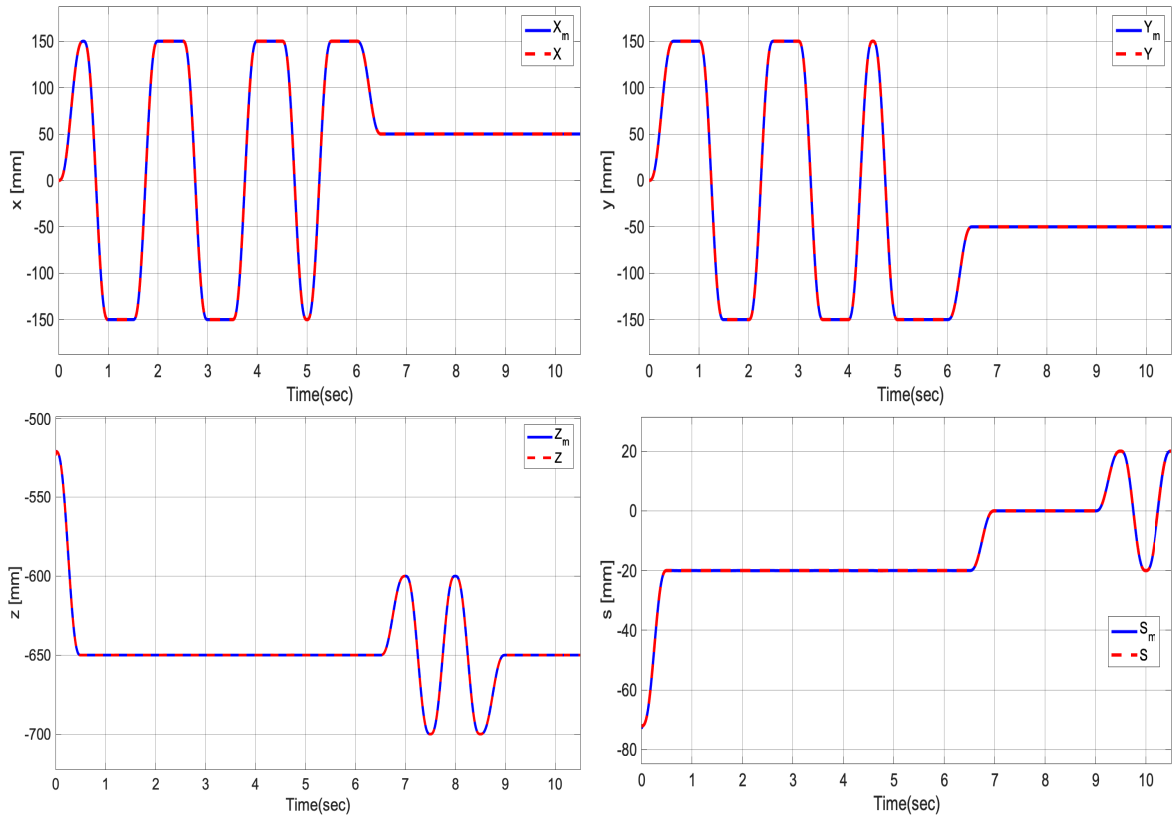


Figure 2.11: Evolution of Cartesian coordinates of the end-effector

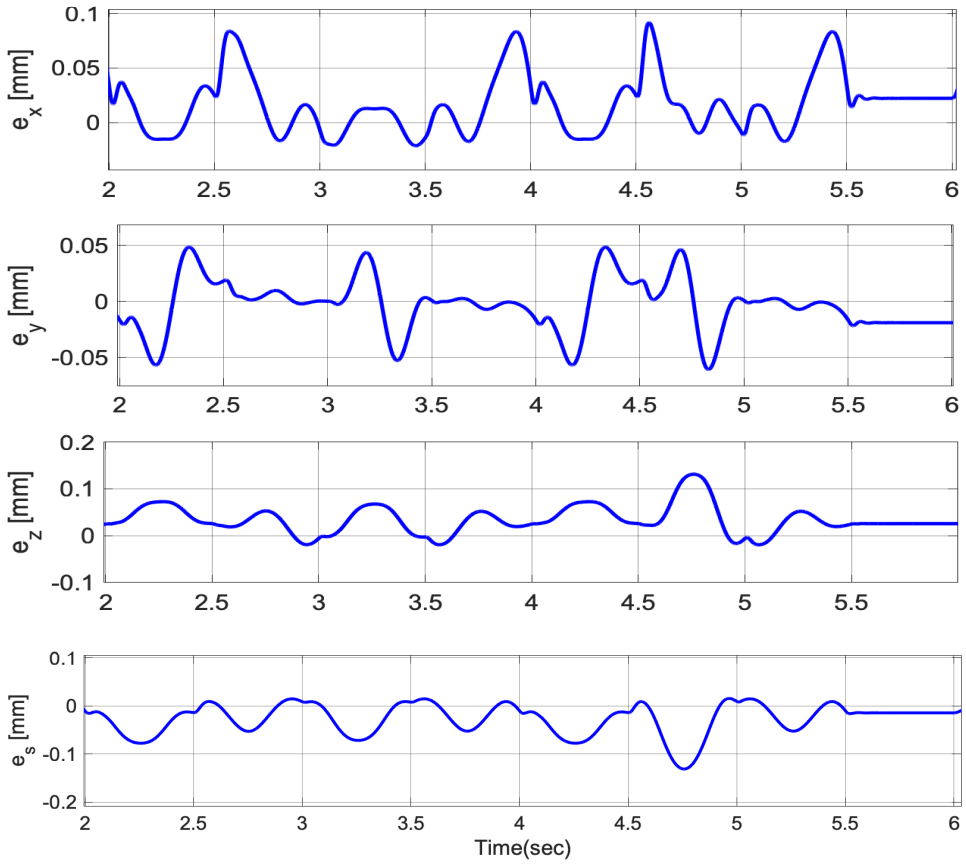


Figure 2.12: End-effector's Tracking Error

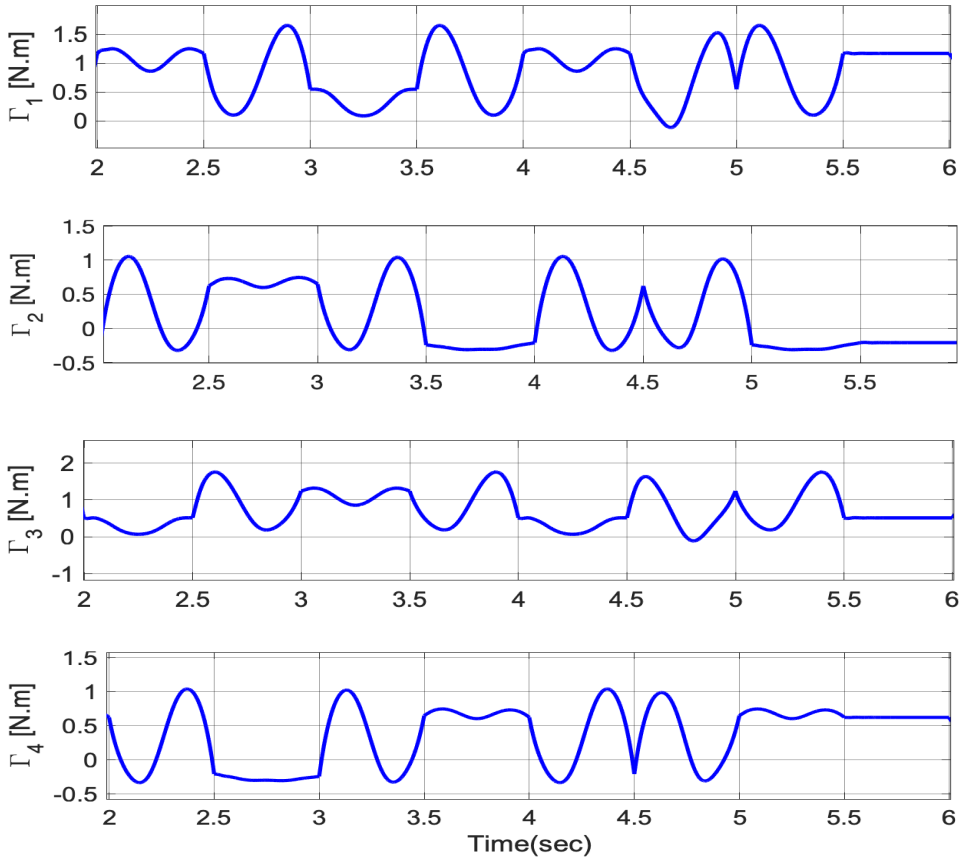


Figure 2.13: Evolution of the Control Inputs

10.3 Discussions and observations

The plots are focused on the interval $\begin{bmatrix}2 & 6\end{bmatrix}$ s to clearly illustrate the system's signal variations. We observe that the Cartesian tracking performance of the PD-FF controller is good, as we can see from Fig 2.11, the desired trajectory is followed by the manipulator with no apparent deviations. This is reflected by the acceptable ranges of the Cartesian tracking errors depicted in Fig 2.12. The control input torques for the actuators are illustrated in Fig 2.13, we can see that they are in the allowable ranges of the manipulator's actuators (maximum torque of 127 N.m) with the maximum value reached during the transition phase being 11 N.m.

The model-based term improves the tracking performance compared to using a PD controller alone and effectively reduces the control effort required from the PD controller, leading to smoother and more stable control actions. However, this approach has limitations when there are parametric variations in the system's model, as the feed-forward term is based on nominal dynamics and cannot compensate for parametric uncertainties.

Another issue is the unmodeled dynamics, which become more pronounced in critical operating conditions, such in the case of high speed, accelerations, and heavy payloads. This control scheme may not fully eliminate their effects. Additionally, one of the main drawbacks of classical controllers like PD controller is the requirement to redesign parameters when modifying the initial reference trajectory. This redesign process can be time-consuming and may not be practical for applications requiring frequent trajectory changes.

11 Conclusion

Through this chapter, we have presented both kinematics and dynamics models of the VELOCE robot. The inverse and direct kinematic models, which were designed based on the geometry of the manipulator, will be used to derive the desired trajectory from Cartesian to joint space and to determine the end-effector's coordinates based on the measurement of the actuator joints. After introducing some simplifying assumptions, the dynamic model of the VELOCE robot was presented in a form suitable for joint space control.

Regarding trajectory generation, 5th order polynomials were proposed for generating a sequence of displacements of the end-effector, designed to be singularity-free. Finally, the simulations of the VELOCE model with a PD-FF controller provided insight into the dynamics of the system, highlighting the coupling between the actuator joints and the nonlinearities in the robots dynamics. Because of the limitations of this approach exploring better strategies to handle uncertainties and disturbances, such as adaptive and robust control became a necessity.

In the next chapter, the theoretical framework for two adaptive control schemes proposed for the VELOCE robot will be presented, namely: Model Reference Control with a PD feedback (MRAC-PD) and $\mathcal{L}_1$ adaptive controller with a feed-forward term ($\mathcal{L}_1$-FF).

Chapter 3

Adaptive Control for PKMs

1 Introduction

According to Astrom and Wittenmark [60], "*an adaptive controller is a controller with adjustable parameters and a mechanism for adjusting the parameters*". Adaptive control is a well-researched topic in control theory that spans several decades and which has proven efficient in dealing with systems with uncertainty. Robot manipulators, particularly PKMs, are nonlinear systems with significant uncertainties, in addition, they are often used in environments with constantly changing characteristics. Consequently, numerous adaptive control strategies have been proposed and tested for these systems. A comprehensive overview of adaptive control schemes for robot manipulators can be found in [61].

In this chapter, the theoretical framework and necessary background on two adaptive control methods proposed for controlling the VELOCE robot are provided. After a general introduction to MRAC methods, including a brief history and basic concepts, the development of an MRAC control law with MIT-based adaptation rule for second-order systems is presented in detail, to be used later in Chapter 4 for VELOCE robot control. Following this, $\mathcal{L}_1$ adaptive control method is introduced, with a detailed layout of its architecture and an explanation of the design process of the control law which will be implemented in Chapter 6 for the VELOCE robot.

2 Model Reference Adaptive Control (MRAC)

In this section, we will present the fundamental concepts related to MRAC. Additionally, we will detail the design steps of an MRAC controller using the MIT-based adaptation rule for a second-order system.

2.1 Background on MRAC

2.1.1 Brief history of MRAC

The first emergence of adaptive control resulted from efforts to enhance the performance of autopilots in the aerospace industry during the 1950s [60]. With advancements in aircraft technology, the existing classic, constant-gain, linear feedback controllers proved inadequate for achieving desired performance across large flight envelopes. This inadequacy led to the development of the first Model Reference Adaptive Control (MRAC) designs by Whitaker [62]. However, a significant drawback for implementing these types of controllers was the absence of a theoretical framework to prove their stability. Consequently, the research community shifted its focus from adaptation problems to stability issues.

In the 1960s, Parks introduced the first MRAC adaptation law based on Lyapunov stability theory [63], and at the same time, the MIT research team developed and tested the MIT rule for parameters fitting. These advancements were made possible due to significant progress in stability theories such as Lyapunov theory, hyper-stability, and passivity theory, as well as advances in the field of system identification. Between the 1970s and 1990s, intensive research work to improve and understand the MRAC scheme

produced a relatively complete theory supported by a large body of literature.([64], [65], [60], [66]).

From the 1990s to the present, more advanced controllers have been developed that combine adaptive controllers with nonlinear control methods such as fuzzy control, neural networks, and sliding mode control. These developments have significantly improved the robustness of adaptive control systems in a variety of complex and changing environments.

2.1.2 Model Reference Adaptive Systems (MRAS)

Originally, the model-reference adaptive system (MRAS) was introduced to address performance requirements defined by a reference model, which generates an expected response based on a given command signal.

These systems comprises two primary loops: an internal loop and an external loop. The internal loop consists of the plant and controller arranged in a closed-loop feedback structure, while the external loop adjusts the controller's parameters to find their optimal values thus minimizing the error between the desired model and the actual plant system.

In implementing MRAS, it is assumed that the plants have known structures but unknown parameters. This means that for linear systems, the number of zeros and poles of the plant must be known, the equivalent for nonlinear systems is the structure of the dynamic equations

The main components of an MRAS are explained below based on [67], [68]:

- **Reference model**
  A reference model is utilized to specify the ideal response of the plant when receiving external command signals in terms of performance metrics, such as rise time and overshoot, in addition to robustness specifications such as phase and gain stability margins.

  Additionally, the selection of this reference model must take into account the assumptions about the plant's structure and its dynamics, including the plant's order and the presence of unstable zeros for non-minimum phase systems.

  Typically, a reference model are chosen as LTI systems aiming to bring the dynamic behaviour of a complex system to that of a simple and stable one.

- **Controller**
  The controller is dependent on the system's signals — such as tracking error, output, and reference signals— and some adjustable parameters.

  In most cases, the control law is linear in terms of the adaptation parameters to ensure stability and tracking convergence. Typically, the design of these mechanisms involves a trade-off between performance and robustness.

  The designed controller is supposed to accomplish ideal performance in the absence of uncertainty, which is referred to as nominal controller. However, when parameter variations or external disturbances occur, the adaptation mechanism adjusts the controller parameters to maintain the desired performance.

- **Adaptation mechanism**
  Adaptation mechanisms are designed to adjust control parameters in real-time, en-

suring the system output follows the desired reference model despite external disturbances and parameter variations.

These mechanisms aim to maintain system stability and convergence of the estimated parameters to their ideal values through various mathematical approaches (shown in Fig 3.1) , such as Lyapunov theory, the MIT rule, and passivity theory as detailed in [69].

While different approaches may offer unique advantages, their outcomes are often equivalent in terms of performance and stability. In this work, we will use the MIT rule-based method to develop and implement the adjustment mechanism.

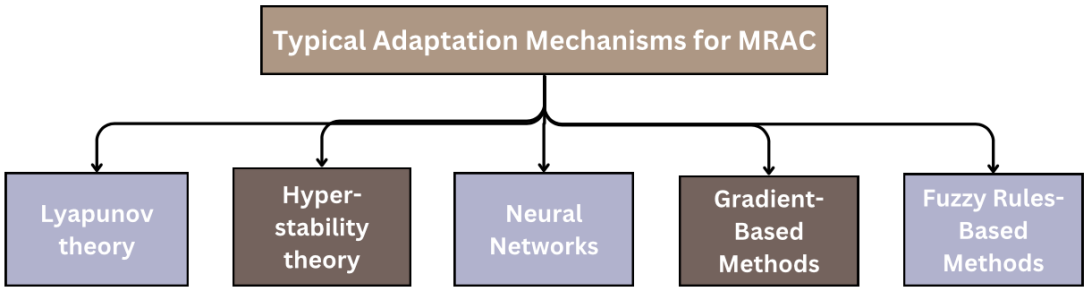


Figure 3.1: Typical adaptation mechanisms used for MRAC schemes

The MRAC schemes can be implemented using either direct or indirect approaches:

2.1.3 Direct MRAC

In the direct MRAC approach, the controller parameters are adjusted based on an adjustment laws that directly update the control gains to minimize the tracking error, ensuring that the system output converges to the desired trajectory.

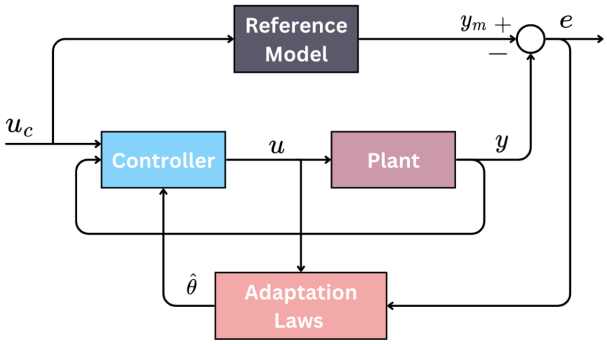


Figure 3.2: Direct MRAC structure

2.1.4 Indirect MRAC

This approach involves estimation of the system's parameters using an adaptive estimator based on identification techniques such as Recursive Least Squares and requires a linear parametrization of the dynamic model of the plant. The estimated parameters are then used to recompute the control gains, assuming the parameter estimate is the true value.

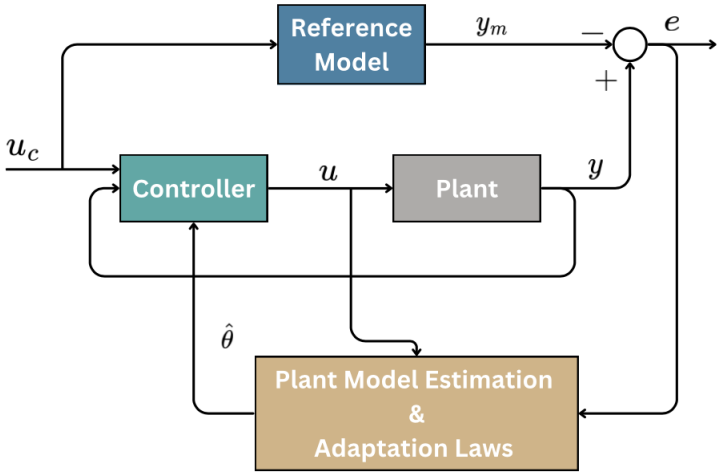


Figure 3.3: Indirect MRAC structure

2.2 MRAC with MIT-rule based adaptation

As previously mentioned, the VELOCE dynamic model are described using time varying, nonlinear differential equations as in Equation (2.20). However, for the purpose of designing the MRAC controller, we assume the manipulator's model to be a second-order system with unknown parameters. It can be expressed in the following form:

$$\ddot{y} = -a_1\dot{y} - a_2 y + u \tag{3.1}$$

with y being the system's output signal, u is the command signal and a_1 and a_2 are the unknown plant parameters.

We choose the reference model to be matched by the closed loop plant as:

$$\ddot{y}_m = -a_1^m\dot{y}_m - a_2^m y + u_c \tag{3.2}$$

where y_m represents the reference model output and u_c is the reference signal.

2.2.1 Adaptive control law

A general linear adaptive controller can have the following expression according to [60]:

$$Ru(t) = Tu_c(t) - Sy(t) \tag{3.3}$$

T, R and S are polynomials which degrees are determined such that the controller is both causal and has the minimum possible degree. According to [70], a valid control law for a second order plant to follow a second order reference model can be expressed as follow:

$$u(t) = u_c(t) - \theta_1\dot{y}(t) - \theta_2 y(t) \tag{3.4}$$

with θ_1 and θ_2 are the adjustable parameters of the adaptive controller.

2.2.2 Adaptation mechanism

The MIT rule, developed by researchers at the Massachusetts Institute of Technology (MIT) in the 1960s, was initially used to design the autopilot system for aircraft. It

ensures that the estimated parameters of the controller are adjusted to minimize a specific loss function, often a quadratic function of the tracking error $e = y - y_m$ as expressed below:

$$J(\theta) = \frac{1}{2}e^2 \tag{3.5}$$

The gradient descent method is used to derive the update rules for the estimated parameters, guaranteeing the minimization of the criteria in Equation (3.5). We end up with an adaptive law that is dependent on the partial derivative of the objective function with respect to the estimated parameters:

$$\dot{\theta} = -\gamma \nabla J(\theta) = -\gamma e \nabla e(\theta) \tag{3.6}$$

By replacing Equation (3.4) into (3.1), the tracking error becomes:

$$e = -a_1\dot{y} - a_2 y - \theta_1\dot{y} - \theta_2 y + a_1^m\dot{y}_m + a_2^m y \tag{3.7}$$

By using Equation (3.7) and (3.6), we derive the following adaptive laws:

$$\frac{d\theta_1}{dt} = -\gamma_1 e\frac{\partial y}{\partial \theta_1} \tag{3.8}$$

$$\frac{d\theta_2}{dt} = -\gamma_2 e\frac{\partial y}{\partial \theta_2} \tag{3.9}$$

Here, $\frac{\partial y}{\partial \theta_1}$ and $\frac{\partial y}{\partial \theta_2}$ are called the *sensitivity functions*, $\gamma_1 \in \mathbb{R}^+$ and $\gamma_2 \in \mathbb{R}^+$ are the adaptation gains.

Using Equations (3.4) and (3.1), we obtain:

$$\frac{\partial \ddot{y}}{\partial \theta 1} = -a_1\frac{\partial \dot{y}}{\partial \theta_1} - a_2\frac{\partial y}{\partial \theta_1} - \dot{y} - \theta_1\frac{\partial \dot{y}}{\partial \theta_1} - \theta_2\frac{\partial y}{\partial \theta_1} \tag{3.10}$$

$$\frac{\partial \ddot{y}}{\partial \theta 2} = -a_1\frac{\partial \dot{y}}{\partial \theta_2} - a_2\frac{\partial y}{\partial \theta_2} - y - \theta_1\frac{\partial \dot{y}}{\partial \theta_2} - \theta_2\frac{\partial y}{\partial \theta_2} \tag{3.11}$$

By assuming that the rate of change for the adaptive parameters is slow; i.e the changes of $\ddot{y}$ and $\dot{y}$ with respect to θ_1 and θ_2 are small, we can interchange the order of differentiation. Then applying the Laplace transform to Equations (3.10) and (3.11) result in the following expressions:

$$\frac{\partial y}{\partial \theta_1} = \frac{1}{s^2 + (\theta_1 - a_1)s + (\theta_2 - a2)}\dot{y} \tag{3.12}$$

$$\frac{\partial y}{\partial \theta_2} = \frac{1}{s^2 + (\theta_1 - a_1)s + (\theta_2 - a2)}y \tag{3.13}$$

The MIT rule approximates these sensitivity derivatives with their online estimates since a_1 and a_2 are unknown. However, global closed-loop stability and convergence of the tracking error to zero are not always provable when using approximations of the sensitivity functions. Nonetheless, simulations have shown that the MIT rule and other approximation techniques perform well with small adaptive gain and low-amplitude reference input signals [70].

The estimates of a_1 and a_2 are derived based on the following matching conditions:

$$\hat{a}_1 = a_1^m + \theta_1 \tag{3.14}$$

$$\hat{a}_2 = a_2^m + \theta_2 \tag{3.15}$$

Substituting Equations (3.14) and (3.15) in (3.12) and (3.13), the approximated sensitivity functions are obtained:

$$\frac{\partial y}{\partial \theta_1} = -\frac{1}{s^2 + a_1^m s + a_2^m}\dot{y} \tag{3.16}$$

$$\frac{\partial y}{\partial \theta_2} = -\frac{1}{s^2 + a_1^m s + a_2^m}y \tag{3.17}$$

Finally, the update rules of the adaptive parameters of the controller are given as follow:

$$\frac{d\theta_1}{dt} = \gamma_1 \frac{1}{s^2 + a_1^m s + a_2^m} e\dot{y} \tag{3.18}$$

$$\frac{d\theta_2}{dt} = \gamma_2 \frac{1}{s^2 + a_1^m s + a_2^m} ey \tag{3.19}$$

Since the response of the plant and stability of the MRAC system with the MIT-rule based adaptation depends upon the adaptation gain γ, the choice of this parameter becomes extremely important. In addition, according to [60] the value of γ depends on the amplitudes of the system's signals (such as the reference command u_c and the plant's output signal y). In order to mitigate this issue, it is preferable to use the normalized version of the MIT-rule.

The adjustment rules given in Equation (3.18) and (3.19) are modified to become:

$$\frac{d\theta_1}{dt} = \gamma_1 \frac{\phi_1}{\epsilon + \phi_1^2}\, e \tag{3.20}$$

$$\frac{d\theta_2}{dt} = \gamma_2 \frac{\phi_2}{\epsilon + \phi_2^2}\, e \tag{3.21}$$

with:

$$\phi_1 = \frac{1}{s^2 + a_1^m s + a_2^m}\, \dot{y} \tag{3.22}$$

$$\phi_2 = \frac{1}{s^2 + a_1^m s + a_2^m}\, y \tag{3.23}$$

ϵ is a positive number, which is introduced to remove the issue of possible division by zero if ϕ_1 or ϕ_2 have small values.

3 $\mathcal{L}_1$ adaptive control

3.1 Background on $\mathcal{L}_1$ adaptive control

In the late 1980s, several improvements to the stability and robustness of MRAC systems were introduced in various papers. Notable examples include the σ-modification [71] and the ϵ -modification [72] which emerged after thorough analysis of instability causes

in adaptive methods available at that time. These methods aim to address transient response issues and ensure boundedness of adaptive parameters by proposing limitations on the adaptation loop gain i.e limiting the bandwidth of the controller.

However, high adaptation rates are necessary in adaptive controllers to respond to changes in initial conditions, reference inputs, and uncertainties for satisfactory performance. Nevertheless, these methods left some issues unintended such as undesirable frequency characteristics and necessity for persistent parameter excitation for convergence leading to poor transient behavior [73].

As a result, the $\mathcal{L}_1$ adaptive control method was developed by Hovakimyan and Cao [74], it ensures decoupling between the adaptation and control loop hence allowing the selection of high adaptation gains, limited only by hardware constraints, without loss of robustness . It was first proposed for flight control as an alternative to gain scheduling guaranteeing uniform performance both in transient and steady-state operation thus eliminating the need for gain scheduling of the adaptation rates [75].

An other important feature is that this approach allows prediction of closed-loop system response a priori; i.e., ensuring that signals bounds are satisfied through controller design feature which makes $\mathcal{L}_1$ adaptive control suitable for numerous safety-critical applications. Theoretical proofs of guaranteed robustness bounds, stability margins, and tracking performance in transient and steady-state responses for the L1 adaptive control are well-established.

3.2 $\mathcal{L}_1$ adaptive control architecture

Like most adaptive controller the design process consists of determining an adaptive control law with adjustable parameters through an adaptation mechanism, along to these elements, this control scheme contains a state predictor for estimating the parameters of the control law and a low-pass filter that limits the control bandwidth for decoupling the control and adaptation loops allowing thereby for fast adaptation with guaranteed robustness (as shown in Fig 3.4). For the development of the $\mathcal{L}_1$ adaptive controller, we consider single-input single-output linear time-invariant systems described by the following state-space equations:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + bu(t), \quad x(0) = x_0 \\ y(t) &= c^\top x(t) \end{aligned} \tag{3.24}$$

where $x(t) \in \mathbb{R}^n$ is the states vector of the system, $u(t) \in \mathbb{R}$ is the control input, $b, c \in \mathbb{R}^n$ are the input and output vectors respectively (which are known), $y(t) \in \mathbb{R}$ is the output of the system.

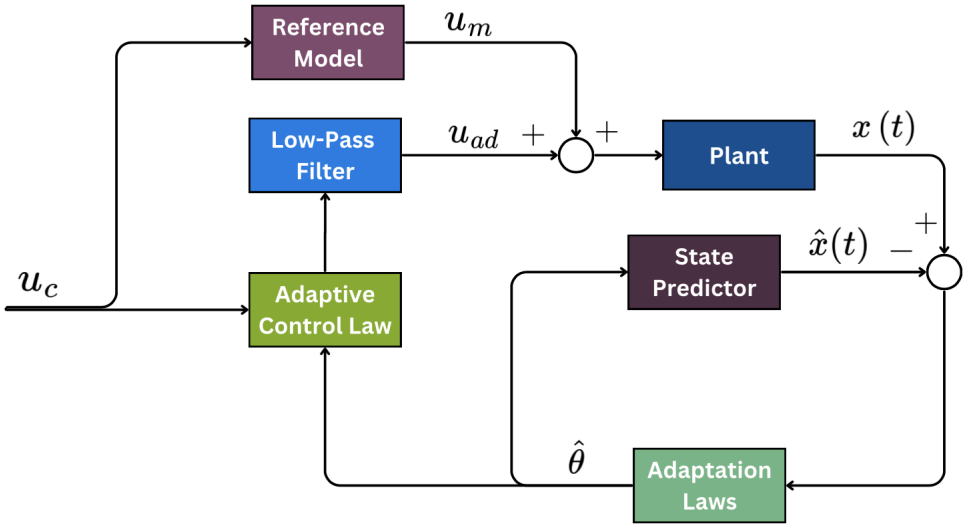


Figure 3.4: $\mathcal{L}_1$ adaptive control architecture

Since the $\mathcal{L}_1$ adaptive controller was inspired from the MRAC approach, they have similar control objectives. In the case of $\mathcal{L}_1$ control, he primary goal is to determine an appropriate control law $u(t)$ that ensures the system's output $y(t)$ tracks a specified reference signal $u_c(t)$ with desired performance specifications and guaranteed boundedness of all system signals. The design of the control $u(t)$ is based on the following assumption:

Assumption 3.2 With the system state matrix $A \in \mathbb{R}^{n\times n}$ considered unknown, we assume the existence of a Hurwitz matrix $A_m \in \mathbb{R}^{n\times n}$ and a vector $\theta \in \mathbb{R}^n$ of perfect parameters, with the pair (A_m, b) controllable and the difference between A_m and A can be expressed as $A_m - A = b\theta^T$. In addition, it is assumed that the parameter vector θ belongs to a given compact set Θ.

This assumption allows us to rewrite the system in 3.24 as follows:

$$\begin{aligned} \dot{x}(t) &= A_m x(t) + b\left(u(t) - \theta^\top x(t)\right), \quad x(0) = x_0 \\ y(t) &= c^\top x(t) \end{aligned} \tag{3.25}$$

3.2.1 Adaptive control law

In the case where the nominal dynamics of the system are known, the ideal controller would be:

$$u_{\text{ideal}}(t) = \eta(t) + k_g u_c(t) \tag{3.26}$$

where $\eta(t) = \theta^T x(t)$ and $k_g = 1/\left(c^T A_m^{-1} b\right)$ is the inverse of the dc gain of the system (3.25), which gives unit dc gain of the closed-loop system ensuring zero steady-state error.

Substituting Equation (3.26) in the system (3.25) gives the following desired reference system:

$$\begin{aligned} \dot{x}_m(t) &= A_m x_m(t) + b\, k_g\, u_c(t), \quad x(0) = x_0 \\ y_m(t) &= c^T x_m(t) \end{aligned} \tag{3.27}$$

However, this control law is problematic because the uncertainties in the system state matrix A are unknown and need to be estimated online. An adaptation mechanism is

required to ensure the convergence of the estimation errors to zero. Thus, the control law becomes:

$$u(t) = \hat{\eta}(t) + k_g u_c(t) \tag{3.28}$$

with $\hat{\eta}(t) = \hat{\theta}^T x(t)$.

The $\mathcal{L}_1$ adaptive scheme introduces an additional element compared to conventional adaptive controllers: a low-pass filter that separates the estimation and control bandwidths. The final adaptive control law, in its Laplace form, is given by:

$$u(s) = C(s)\left(\hat{\eta}(s) + k_g u_c(s)\right) \tag{3.29}$$

where $C(s)$ is a bounded-input bounded-output strictly proper transfer function with $C(0) = 1$.

To ensure the stability of the control system, the following $\mathcal{L}_1$-norm condition, as demonstrated in [76] must be satisfied:

$$\|G(s)\|_{\mathcal{L}_1} L < 1 \tag{3.30}$$

where:

$$G(s) = H(s)(1 - C(s)), \quad H(s) = (s\mathbb{I} - A_m)^{-1} b, \quad L = \max_{\theta \in \Theta} \|\theta\|_1$$

and $\|G(s)\|_{\mathcal{L}_1}$ the $\mathcal{L}_1$ norm of the system $G(s)$ (a definition of $\mathcal{L}_1$ norm is given in Appendix 2). This condition needs to be considered when selecting the reference model specifications and during the design of the low-pass filter.

3.2.2 State predictor

The disturbance estimation method for $\mathcal{L}_1$ adaptive control architecture uses a different methodology, instead of using the parametric estimation error in the adaptation laws, a state predictor is included in its structure and the prediction error is used for updating the adaptive control law. The following state predictor of the system in (3.25) is considered:

$$\begin{array}{ll} \dot{\hat{x}}(t) & = A_m \hat{x}(t) + b(u(t) - \hat{\theta} x(t)), \quad \hat{x}(0) = x_0 \\ \hat{y}(t) & = c^T \hat{x}(t) \end{array} \tag{3.31}$$

The dynamics of the prediction error, defined as $\tilde{x}(t) = \hat{x}(t) - x(t)$, can be obtained by subtracting the system dynamics from those of the predictor as follows:

$$\dot{\tilde{x}}(t) = A_m \tilde{x}(t) - b\tilde{\theta}(t) x(t) \tag{3.32}$$

From [76], the upper bound for the prediction error at any time is given by (the proof is provided in Appendix 3):

$$\|\tilde{x}\|_{L_\infty} \leq \sqrt{\frac{\theta_{\max}}{\lambda_{\min}(P)\gamma}}, \quad \theta_{\max} = 4 \max_{\theta \in \Theta} \|\theta\|^2 \tag{3.33}$$

with $\lambda_{min}(P)$ the minimum eigenvalue of P and γ the adaptation gain. It follows that by increasing the adaptation gain γ the tracking error can be made smaller.

3.2.3 Adaptation laws

A projection-based adaptation law is used to compute the uncertainties estimate $\hat{\theta}(t)$ as expressed:

$$\dot{\hat{\theta}}(t) = \gamma \,\mathrm{Proj}(\hat{\theta}(t), x(t)\tilde{x}(t)Pb), \quad \hat{\theta}(0) = \theta_0 \tag{3.34}$$

where $\gamma \in \mathbb{R}^+$is the adaptation gain and $P = P^T > 0$ the solution for the algebraic Lyapunov equation $A_m^T P + P A_m = -Q$, for an arbitrary choice of $Q = Q^T > 0$.

The projection operator (as defined in Appendix 4) is used to avoid the problem of parameters drift by making sure that the values of estimates remain within the compact set Θ.

4 Conclusion

This chapter established the theoretical foundation for implementing the proposed adaptive control methods on the VELOCE manipulator since adaptive control has proven essential for addressing the nonlinearities and uncertainties inherent in robotic systems. After providing historical context on MRAC approach, the key MRAC components were discussed, including the reference model, controller, and adaptation mechanism. The chapter then presented the adaptive control law and the associated adaptation mechanism based on a normalized version of the MIT rule, which helps avoid dependence between the adaptation gain value and the system's signal levels.

Additionally, the $\mathcal{L}_1$ adaptive control method was introduced as a technique enabling the application of adaptive control in safety-critical environments. This method's architecture, featuring a state predictor and low-pass filter, facilitates fast adaptation while maintaining guaranteed performance and robustness. The chapter offered a comprehensive overview of both adaptive control strategies, laying the groundwork for their implementation in subsequent chapters.

In the next chapter, we will present the fundamentals of Fractional Order Calculus and discuss its application to control theory. Additionally, a numerical approximation technique in the time domain for fractional order systems will be introduced. This will be accompanied by simulations that demonstrate the method's implementation and provide insights into the characteristics of fractional systems.

Chapter 4

Fractional Order Calculus

1 Introduction

In recent years, fractional calculus has been used in control theory and industrial automation and proved to bring significant improvements to control quality through increased precision in performance and energy consumption efficiency [77].

This chapter aims to present the fundamentals of fractional order (FO) calculus theory. Initially, it provides background on the origins and typical applications of fractional calculus. Approximation methods for FO systems are discussed then the Singularity method, also known as Charef's approximation [78], is presented. The chapter concludes with numerical applications using this method, highlighting the features of FO systems and illustrating the working principles of the Singularity technique.

2 Background of fractional order calculus

Fractional order calculus, a branch of mathematics that studies derivatives and integrals of arbitrary orders (whether real or complex), extends the concept of integer-order derivatives and integrals to non-integer orders [79]. The field's development dates back 300 years, with its theoretical foundation largely established in the nineteenth century. Initially, fractional order calculus was a niche subject primarily discussed among mathematicians. Early theoretical contributions were made by Euler and Lagrange in the eighteenth century, with systematic studies emerging in the early to mid-nineteenth century by scholars such as Liouville, Riemann, and Holmgren [80].

This lack of early attention can be attributed to several factors [81], including the absence of a standardized mathematical framework, the difficulty in providing simple geometric interpretations, and the complexity of solving fractional calculus problems. Additionally, integer-order calculus was often sufficient for simpler problems. However, fractional order calculus has proven essential for providing more accurate descriptions of complex dynamical systems. Consequently, it has found successful applications across various fields such as anomalous diffusion, control theory, signal and image processing, mechanics, dynamic systems, biology, and environmental science [82].

It has been particularly valuable in control engineering applications since fractional dynamics are best compensated with fractional controllers, [83]discusses applications related to fluid level control in multi-tank systems, magnetic levitation system control, and the control of ion-polymer metal composite actuators.

Interest in applying fractional calculus in system theory and feedback control has led to the development of several Fractional Order Controllers (FOC). The French research group CRONE, led by Oustaloup, created tools for identifying and controlling fractional dynamic systems [84] including the Oustaloup approximation method. Podlubny introduced the fractional-order PID (FOPID) controlle [85] which, with its additional "tuning knobs," allows precise adjustments that conventional integer-order systems cannot achieve. Other notable controllers include the Tilted Proportional and Integral (TID) Controller and

Fractional Lead-Lag Controller [86].

Several personal computer software packages have been developed to facilitate fractional system identification and controller design tasks. Notable MATLAB toolboxes include the CRONE toolbox, Ninteger toolbox, FOTF toolbox, as well as the Fractional Order Modelling and Control toolbox (FOMCON).

3 Fractional order operator

Fractional calculus extends the concepts of integration and differentiation to non-integer order using the fundamental diff-integral operator ${}_aD_t^\alpha$ given by:

$$ {}_aD_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & : R(\alpha) > 0 \\ 1 & : R(\alpha) = 0 \\ \int_a^t (d\tau)^{-\alpha} & : R(\alpha) < 0 \end{cases} \tag{4.1} $$

where a and t indicate the limits of the operation and α represents the fractional order.

Numerous definitions of the fractional operator have been proposed in the literature [82] , each developed based on distinct principles and within varying mathematical frameworks. However, their common objective remains consistent: to facilitate the incorporation of the differintegral operator in modeling or resolving real-world problems. The most frequently used definitions are Riemann-Louiville, Grünwald-Letnikov and Caputo. Since the Caputo definition is particularly popular in engineering applications, its definition is presented below.

Definition 3 (*Caputo Definition*)
As stated in [85], this definition allows initial conditions such as $y(0), y'(0)$, not like fractional condition such as $y^{0.5}(0)$. It is defined as follow:

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

where, n is an integer which satisfies the condition $n-1 < \alpha < n, \alpha$ is a real number and a and t are the limit of integration.

3.1 Properties of fractional order operators

The main properties of fractional derivatives and integrals are as follows:

1. If $f(t)$ is an analytic function of t, then its fractional-order derivative $D^\alpha f(t)$ is an analytic function of both t and α.
2. For $\alpha = n$, where n is an integer, the operation $D^\alpha f(t)$ yields the same result as classical integer-order differentiation n.
3. For $\alpha = 0$, the operation $D^\alpha f(t)$ is the identity operator: $D^0 f(t) = f(t)$.
4. Fractional-order differentiation and integration are linear operations:

$$ D^\alpha a f(t) + D^\alpha b g(t) = a D^\alpha f(t) + b D^\alpha g(t) $$

5. The additive law (property of the semi-group)

$$D^{\alpha}D^{\beta}f(t) = D^{\beta}D^{\alpha}f(t) = D^{\alpha+\beta}f(t)$$

is valid under certain constraints on the function $f(t)$.

6. The Laplace integral transform in an essential tool in dynamic system and control engineering. A function $F(s)$ of the complex variable s is called the Laplace transform of the original function $f(t)$ and defined as

$$F(s) = \mathscr{L}[f(t)] = \int_0^{\infty} \mathrm{e}^{-st} f(t)\mathrm{d}t$$

The original function $f(t)$ can be recovered from the Laplace transform $F(s)$ by applying the reverse Laplace transform defined as

$$f(t) = \mathscr{L}^{-1}[F(s)] = \frac{1}{j2\pi}\int_{c-j\infty}^{c+j\infty} \mathrm{e}^{st} F(s)\mathrm{d}s,$$

where c is greater than the real part of all the poles of function $F(s)$.

4 Approximation of fractional order systems

A crucial step in implementing a fractional order transfer function (as defined in 4.1) digitally is the numerical evaluation or discretization of the fractional-order differentiator in the Laplace domain s^r. There are two general discretization methods: direct and indirect.

Indirect methods, as described in [87], involve two steps. Firstly, frequency domain fitting is performed in the continuous-time domain, followed by discretizing the fitted s-transfer function. On the other hand, direct methods rely on approximating fractional transfer functions using either integer or digital transfer functions - an approach more practical and cost-effective due to ease of implementation with integer transfer functions. A comprehensive review of various approximation methods and techniques for continuous and discrete fractional-order models was conducted in [88]. In this work, we will adopt the singularity function approximation technique developed by Charef in [78].

4.1 Approximation using Singularity function (Charef's Method)

Charef's approach aims to approximate the fractional order Laplace operator (defined in 4.1) using rational functions of first order. These functions can be visualized as cascaded branches (Fig), with multiple pole-zero pairs, allowing for an approximation of the fractional-order function across a specified frequency range [78].

Definition 4.1 Assuming zero initial conditions, the Laplace transform is defined as follows:

$$\mathscr{L}\left[D^{\alpha}f(t)\right] = s^{\alpha}F(s). \tag{4.2}$$

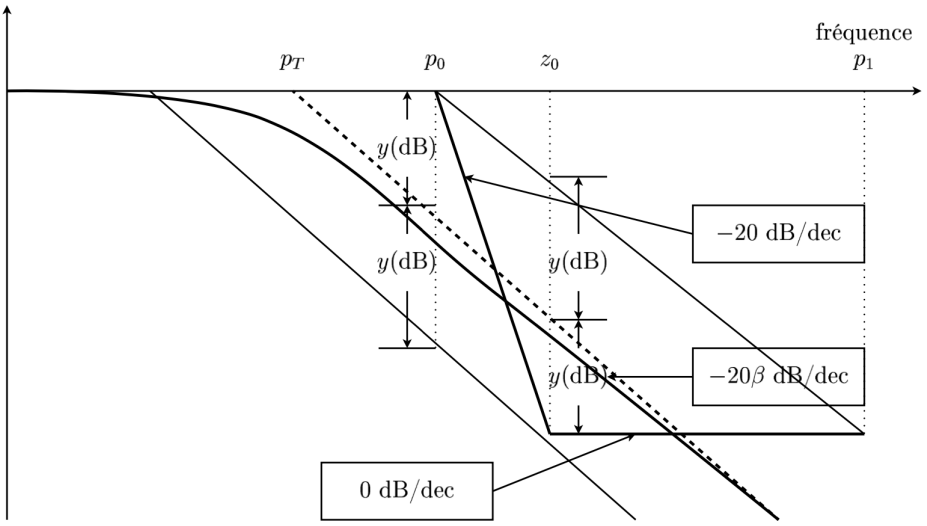


Figure 4.1: Bode diagram of fractional order system and its approximation using Charef's method

4.2 Fractional order integrator

The transfer function of the fractional integration operator is presented below:

$$H_I(s) = \frac{1}{s^\beta} \tag{4.3}$$

where β is a positive number ranging from 0 to 1. For a given frequency interval $[\omega_l, \omega_h]$ this operator can be approximated by a first order fractional system as follow:

$$H_I(s) = \frac{K_I}{\left(1 + \frac{s}{\omega_c}\right)^\beta} \tag{4.4}$$

We need to choose the frequency interval $[\omega_l, \omega_h]$ such that $\omega > \omega_c$, $H_I(s)$ can be written as:

$$H_I(s) = \frac{K_I}{\left(\frac{s}{\omega_c}\right)^\beta} = \frac{K_I \omega_c^\beta}{s^\beta} = \frac{1}{s^\beta}$$

The expression of K_I is given by $\frac{1}{\omega_c^\beta}$ where ω_c is the cut-off frequency and can be calculated as $\omega_c = \sqrt{10^{\frac{y}{10}} - 1}$ with y representing the maximal error between the fractional order slope and the integer order one.

$$H_I(s) = \frac{K_I}{\left(1 + \frac{s}{\omega_c}\right)^\beta} = K_I \frac{\prod_{i=0}^{N-1}\left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^{N}\left(1 + \frac{s}{p_i}\right)} \tag{4.5}$$

The pairs of poles and zeros (p_i, z_i) of the rational transfer functions presented in Equation (4.5) are then calculated after selecting the approximation error y (in dB) and the approximation bandwidth ω_{max} - it should be 100 times greater than ω_h .

$$\begin{aligned} p_i &= (ab)^i p_0, \quad \text{pour } i = 0, 1, \ldots, N \\ z_i &= (ab)^i a p_0, \quad \text{pour } i = 0, 1, \ldots, N-1 \end{aligned} \tag{4.6}$$

where a and b are the position ratios between p_i and the next z_i , their expressions are given below:

$$a = 10^{\frac{y}{10(1-\beta)}}, \quad b = 10^{\frac{y}{10\beta}} \tag{4.7}$$

with

$$p_0 = \omega_c\sqrt{b}, \quad z_0 = ap_0 \tag{4.8}$$

The total number of singularities is determined based on the limit frequency of the approximation bandwidth is equal to $N + 1$:

$$N = ceil\left(1 + \frac{\ln\left(\frac{\omega_{\max}}{p_0}\right)}{\ln(ab)}\right) \tag{4.9}$$

4.3 Fractional order systems

A fractional-order continuous-time dynamic system can be expressed by a fractional differential equation of the following form [82]:

$$\begin{gathered} a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t) = \\ b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t), \end{gathered}$$

where $(a_i, b_j) \in \mathbb{R}^2$ and $(\alpha_i, \beta_j) \in \mathbb{R}_+^2$.

Applying the Laplace transform with zero initial conditions, the transfer function obtained is as follows:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}}.$$

4.3.1 First order systems

For a fractional first order system with the following transfer function:

$$G(s) = \frac{1}{\left(1 + \frac{s}{p_t}\right)^{\beta}} \tag{4.10}$$

with $1/p_t$ being the time constant of the system and β the fractional order coefficient (between 0 and 1).

The approximation using the Singularity method of $G(s)$ is presented below:

$$G(s) = \frac{1}{\left(1 + \frac{s}{p_t}\right)^{\beta}} = \frac{\prod_{i=0}^{N-1}\left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^{N}\left(1 + \frac{s}{p_i}\right)} \tag{4.11}$$

The poles and zeros are calculated using Equation (4.6) , (4.8) and (4.7) by replacing ω_c with p_t. The number of singularities needed is obtained using Equation (4.9).

4.3.2 Second order systems

For second order systems having transfer functions in this form:

$$G(s) = \frac{1}{\left(\frac{s^2}{\omega_n^2} + 2\xi\frac{s}{\omega_n} + 1\right)^{\beta}} \tag{4.12}$$

where ω_n is the resonance frequency, ξ the damping coefficient and β the fractional order coefficient (between 0 and 1). We distinguish two cases:

- **Case 1:** $0 < \beta < 0.5$
  We can rewrite Equation (4.12) as follow:

$$G(s) = \frac{\left(\frac{s}{\omega_n} + 1\right)\left(\frac{s}{\omega_n+1}\right)^{\eta}}{\left(\frac{s^2}{\omega_n^2} + 2\alpha\frac{s}{\omega_n} + 1\right)} \tag{4.13}$$

  with $\alpha = \xi^{\beta}$ and $\eta = 1 - 2\beta$. We can approximate Equation (4.13) using the following expression:

$$G(s) \approx \frac{\left(\frac{s}{\omega_n} + 1\right)}{\left(\frac{s^2}{\omega_n^2} + 2\alpha\frac{s}{\omega_n} + 1\right)} \frac{\prod_{i=1}^{N-1}\left(1 + \frac{s}{z_i}\right)}{\prod_{i=1}^{N}\left(1 + \frac{s}{p_i}\right)} \tag{4.14}$$

  The poles and zeros are calculated from the following expressions:

$$\begin{aligned} p_i &= (ab)^{i-1} a z_1 \quad i = 1, 2, 3, \ldots, N \\ z_i &= (ab)^{i-1} z_1 \quad i = 2, 3, \ldots, N-1 \end{aligned} \tag{4.15}$$

  with:

$$\begin{aligned} b = 10^{\frac{y}{10\eta}}, \quad a = 10^{\frac{y}{10(1-\eta)}} \\ z_1 = \omega_n\sqrt{b}, \quad \eta = \frac{\log(a)}{\log(ab)} \end{aligned} \tag{4.16}$$

  The number of poles and zeros can be calculated as in the previous case using Equation (4.9).

- **Case 2:** $0.5 < \beta < 1$
  The approximation of the transfer function in Equation (4.12) is given as follow:

$$G_e(s) = \frac{\left(\frac{s}{\omega_n} + 1\right)}{\left(\frac{s^2}{\omega_n^2} + 2\alpha\frac{s}{\omega_n} + 1\right)\left(\frac{s}{\omega_n+1}\right)^{\eta}} \tag{4.17}$$

  with $\alpha = \xi^{\beta}$ and $\eta = 2\beta - 1$.
  Here the poles and zeros are obtained from the previous Equation (4.15) and (4.16) by replacing z_1 by p_1.

4.4 Numerical applications

To illustrate the working principle of the method described above, we will present numerical examples through simulation code in MATLAB for different values of β, the fractional order coefficient and y the approximation error in dB. The parameters' values used are shown in Table 4.1.

System	Transfer function	y (dB)	ω_{max} (rad)
First order system	$G(s) = \frac{1}{(1+\frac{s}{100})^{\beta}}$	1	10^5
Second order system	$G(s) = \frac{1}{(\frac{s^2}{10^4}+2\frac{0.9}{100}s+1)^{\beta}}$	1	10^6
Integrator	$G(s) = \frac{1}{s^{\beta}}$	1	10^6

Table 4.1: Simulation parameters for fractional first, second order systems and fractional integrator

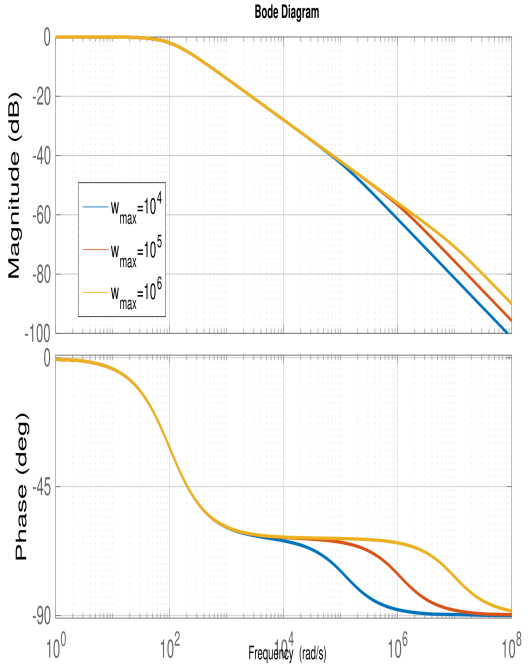


Figure 4.2: Bode diagrams for different values of maximum approximation frequency ω_{max}

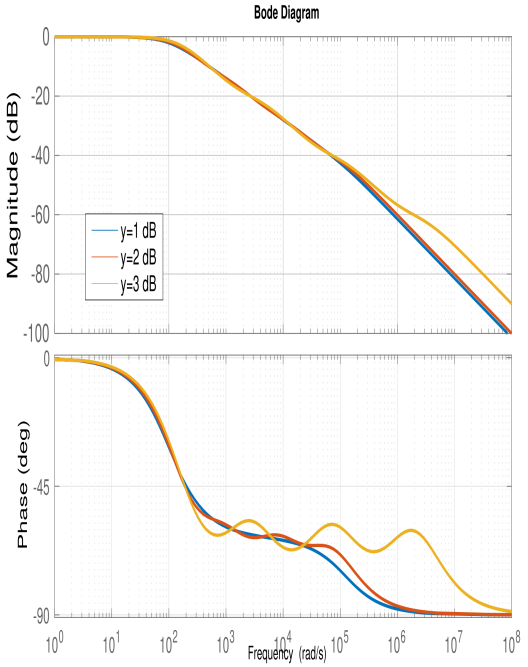
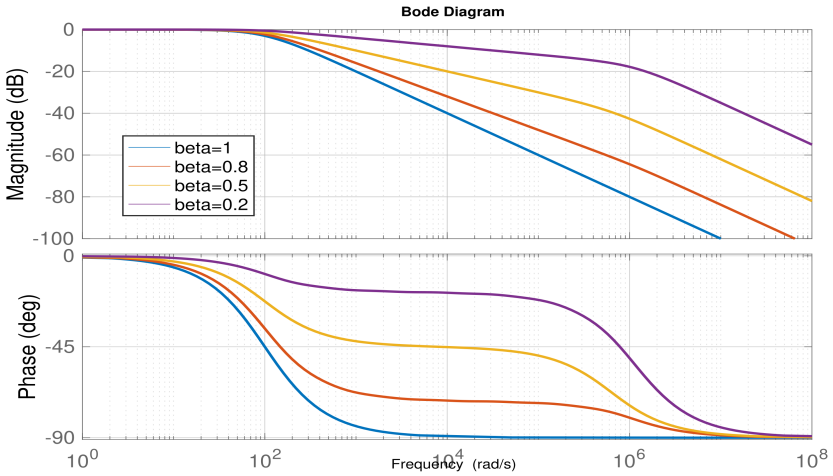


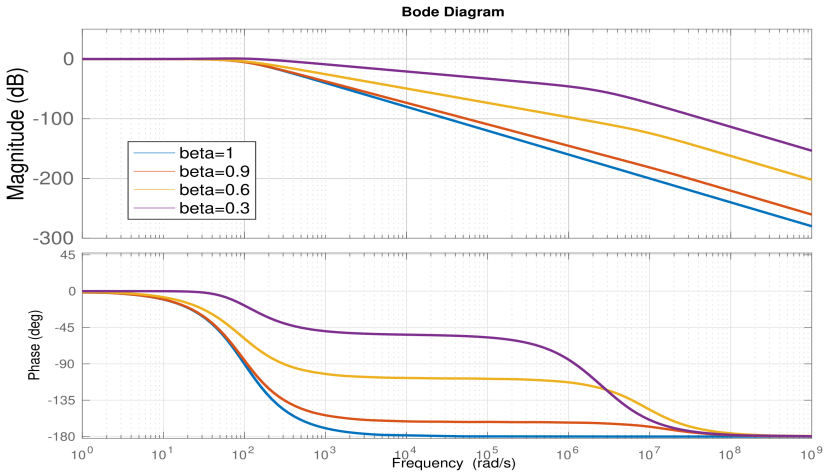
Figure 4.3: Bode diagrams for different values of maximum approximation error y

The results shown in Fig 4.4 indicate that the fractional-order characteristics, including a fractional slope of $-20\beta, dB/dec$ and a constant phase, are only approximated within the specified frequency range (limited by ω_{max}).

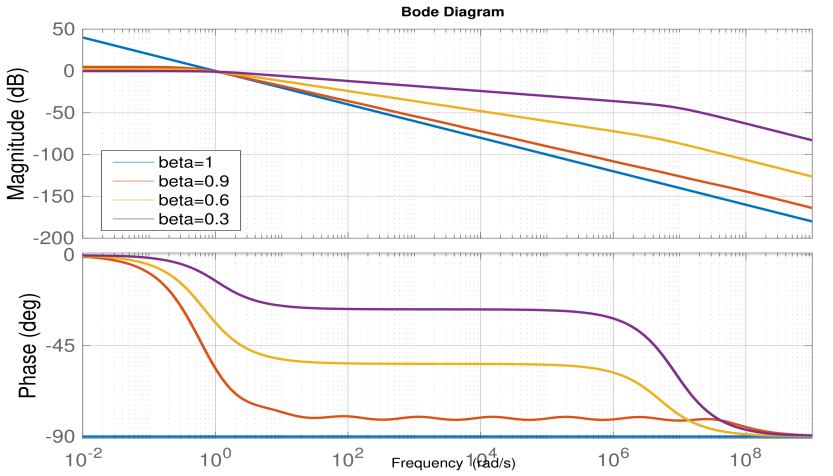
Additionally, it is noteworthy that increasing the value of ω_{max} leads to a higher-order approximating transfer function, which in turn increases the computational complexity. Furthermore, from 4.3 we remark that the approximation of fractional-order systems is more accurate when the approximation error y is smaller.



(a): 1^{st} order system



(b): 2^{nd} order system



(c): Fractional integrator

Figure 4.4: Bode diagrams for different values of fractional order coefficient β

5 Conclusion

In this chapter, we have covered the fundamental aspects of fractional order (FO) calculus and highlighted its abundant applications in control theory thanks to an important body of research that provided a comprehensive framework for FO control applications.

A critical aspect in applying FO calculus in control and automation is the development of accurate numerical approximation methods. We presented the Singularity method, an approximation technique in the frequency domain, and detailing the steps for approximating fractional systems of first and second order as well as fractional integrators.

Numerical simulations were provided to demonstrate the effects of various parameters on the accuracy of the approximation, enhancing our understanding of the method's practical implications. The insights and techniques discussed in this chapter will be utilized in subsequent chapters to develop fractional adaptive controllers for the VELOCE robot.

Chapter 5

MRAC-PD controller for VELOCE manipulator

1 Introduction

In this chapter, we design a Fractional Order Model Reference Adaptive Controller (MRAC) with a Proportional-Derivative (PD) component for the VELOCE robot. We begin by outlining the motivation for selecting this specific control scheme, and a clear statement of the control objectives. Next, we present the theoretical foundation of the control law and explain the integration of Fractional Order (FO) adjustment rules into this control scheme. Once the control design is established, we describe the simulation setup and present the results under both nominal conditions and in the presence of payload variations.

2 Motivation and objectives

The dynamics of robotic manipulators are often modeled by second-order differential equations. As a result, numerous studies have focused on developing and implementing classical linear controllers, such as PD controllers, due to their simplicity and easy implementation especially in industrial settings [40], [36]. However, especially in the case of PKMs, the nonlinearities and couplings between joint motions become more pronounced in applications requiring high speeds and accelerations, causing the PD controller to be insufficient in achieving the required performance and accuracy, as discussed in Section 2.10. Therefore, other alternatives have been considered, such as adaptive control, more precisely the Model Reference Adaptive Control (MRAC) scheme.

The capability of MRAC approach to improve the performance of robot manipulators was demonstrated in multiple works, for instance, in [44] an an adaptive control scheme based on MRAC consisting of a joint space PD controller with adjustable gains was proposed to control a 6-DOF parallel manipulator based on the Stewart platform prototype used to emulate space operations. Moreover, [89] provided an analysis of MRAC application in robotic manipulators, particularly focusing on adaptation based on Lyapunov stability implemented in operational. In another study by [90] MRAC was employed to control a 4-DOF robot manipulator, outperforming traditional controllers such as PID and model-independent controllers based on time-delay estimation techniques. Furthermore, [91] explored the integration of MRAC with Active Inference Control (AIC) to achieve high performance even in the presence of large unmodeled dynamics for a 7-DOF robot arm.

Instead of using classical and adaptive controllers separately, a more appealing approach is to combine the simplicity and ease of implementation of the PD controller with an adaptive element based on MRAC theory to compensate for the uncertainties and nonlinearities in the manipulator's dynamics. The objective in this section is to develop an MRAC control scheme for the VELOCE robot comprising two main parts: a PD feedback component with constant gains alongside an adaptive control law with adjustable parameters updated through MIT-based adaptation mechanism. This approach was tested for marginally stable second order processes with dead time [92], [93] and has proven to be efficient in reducing overshooting, enhancing tracking performance, and improving robustness to payload variations. The parameters of the PD controller, in addition to the adaptation gains, will be calculated using the PSO algorithm aiming to attain optimal

performance.

3 Design of MRAC-PD controller for VELOCE robot

The controller is designed in the joint space, with the actuators' torques representing the command signal of the system. The joint coupling is not explicitly considered because it is treated as part of the nonlinearities that the adaptive component of the controller is designed to address. As a result, the robot system is modeled as four independent linear time-varying systems. The control law, expressed in terms of the actuators' torques, consists of two main components (Fig 5.1):

$$\Gamma = \Gamma_{PD} + \Gamma_{AD} \tag{5.1}$$

with Γ_{PD} representing the PD feedback control signal and Γ_{AD} representing the adaptive controller output.
The reference model is a second-order system as expressed in Equation(5.2):

$$\ddot{q}_m = -a_1^m \dot{q}_m - a_2^m q_m + b_m q_c \tag{5.2}$$

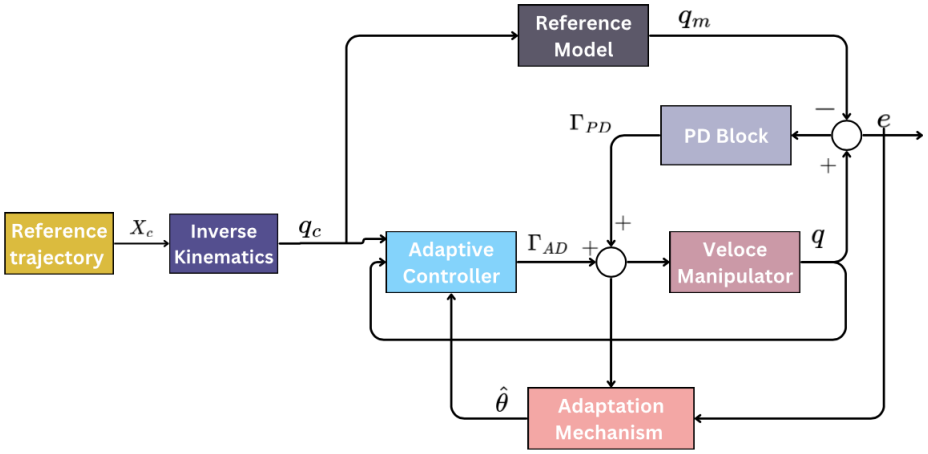


Figure 5.1: Structure of the (MRAC-PD) control system

Based on Equation (3.3), we derive the following adaptive control law:

$$\Gamma_{AD} = q_c - \theta_1 \dot{q} - \theta_2 q \tag{5.3}$$

The PD controller is based on the error between the system and the reference model, as shown below:

$$\Gamma_{PD} = K_p(q_m - q) + K_d(\dot{q}_m - \dot{q}) \tag{5.4}$$

with K_p and K_d are the proportional and derivative gains, respectively. Given that the error is defined as $e = q - q_m$, the final control law becomes as follows:

$$\Gamma = q_c - \theta_1 \dot{q} - \theta_2 q - K_p e - K_d \dot{e} \tag{5.5}$$

The adjustment rules of the adaptive controller are given in Equation (3.18) and (3.19).

4 Fractional order MRAC with PD controller

There have been significant contributions regarding fractional order adaptive controllers in recent years. Many approaches have been proposed for the integration of fractional calculus into MRAC schemes. Some methods involve using fractional order reference models to improve the set of candidates to be used as a model and allow for more flexibility when it comes to setting control specifications, as implemented in [94], [95]. This control scheme was referred to as FOMRAC.

An alternative approach to addressing the fractional adaptation problem has been modifying the adaptation mechanism to include fractional order adjustment rules. For example, using fractional integration for MIT-rule-based adaptation [96] showcased better speed of convergence and stability compared with integer adjustment, along with high noise rejection capabilities, as reported in [97], [98].

The adaptation laws derived previously in Equation (3.18) and (3.19) can be modified to include fractional order calculus as follow:

$$\frac{d^{\alpha}\theta_1}{d^{\alpha}t} = \gamma_1 \frac{\phi_1}{\epsilon + \phi_1^2} e \Longrightarrow \theta_1 = D^{-\alpha}\left(\gamma_1 \frac{\phi_1}{\epsilon + \phi_1^2} e\right) \tag{5.6}$$

$$\frac{d^{\alpha}\theta_2}{d^{\alpha}t} = \gamma_2 \frac{\phi_2}{\epsilon + \phi_2^2} e = D^{-\alpha}\left(\gamma_2 \frac{\phi_2}{\epsilon + \phi_2^2} e\right) \tag{5.7}$$

with $D^{-\alpha}$ the fractional diff-integral operator introduced in Equation (4.1). The temporal evaluation of θ_1 and θ_2 will involve fractional integration which will be implemented using the Singularity method presented previously.

5 Simulation setup

The controller's parameters for this scheme were determined using the PSO algorithm (complete description found in Appendix 5). The optimisation algorithm was given as variables the parameters of the PD controller K_p and K_d , in addition to the adaptation gains of the MRAC controller γ_1 and γ_2. Distinct adaptation gains were utilized for θ_1 and θ_2 in the control law, to account for different dynamics of position and velocity. The parameters used for the PSO algorithm are presented in Table 5.1.

Table 5.1: Parameters of PSO algorithm

Parameters	Description	Value
C_1	Cognitive learning factor	1.6
C_2	Social learning factor	1.4
ω	Inertia coefficient	0.8
N	Swarm size	12
n	Number of iterations	80

The cost function used for PSO has a term to account for the tracking error as well

as the energy of the control signals as expressed below:

$$\mathrm{J} = \int_0^T \alpha_1 |e(t)| + \alpha_2 |\Gamma(t)| \, dt \tag{5.8}$$

with $\alpha_1 = 1$ and $\alpha_2 = 0.02$.

Additionally, we selected the reference model to have a fast rising time (0.0076 s) with minimum overshoot (4.32 %). It has the following form:

$$G_m(s) = \frac{\omega_0^2}{s^2 + 2\delta\omega_0 s + \omega_0^2}$$

The controller's parameters obtained using the PSO algorithm, in addition to the parameters of the reference model are presented in Table 6.1

Table 5.2: MRAC with PD controller parameters

Parameters	**Value**
K_p	490
K_d	16
γ_1	12×10^3
γ_2	8×10^3
ω_0	282.84
δ	0.70

6 Simulation results

Using the parameters given above, we run simulations using Simulink to to assess the performance of the proposed controller under nominal conditions without disturbances, as well as considering the case of parametric variations. The simulation results are presented and discussed in the following sections.

6.1 Case 1: Nominal conditions

Under nominal conditions, the manipulator's model is assumed to be perfectly known, with no external disturbances or parametric variations occurring. However, in this scenario, the controller still need to compensate for internal disturbances related to the nonlinear dynamics and coupling between the joints are still present.
Both IO and FO adaptation laws are considered, their simulation results presented and at the end discussed and compared.

6.1.1 Case 1.1: Integer order adaptation law

To assess the controller's ability to track different references (like the ones in Fig 2.9), we first evaluated its performance for a single point-to-point displacement.

Trajectory 1: Single point-to-point motion

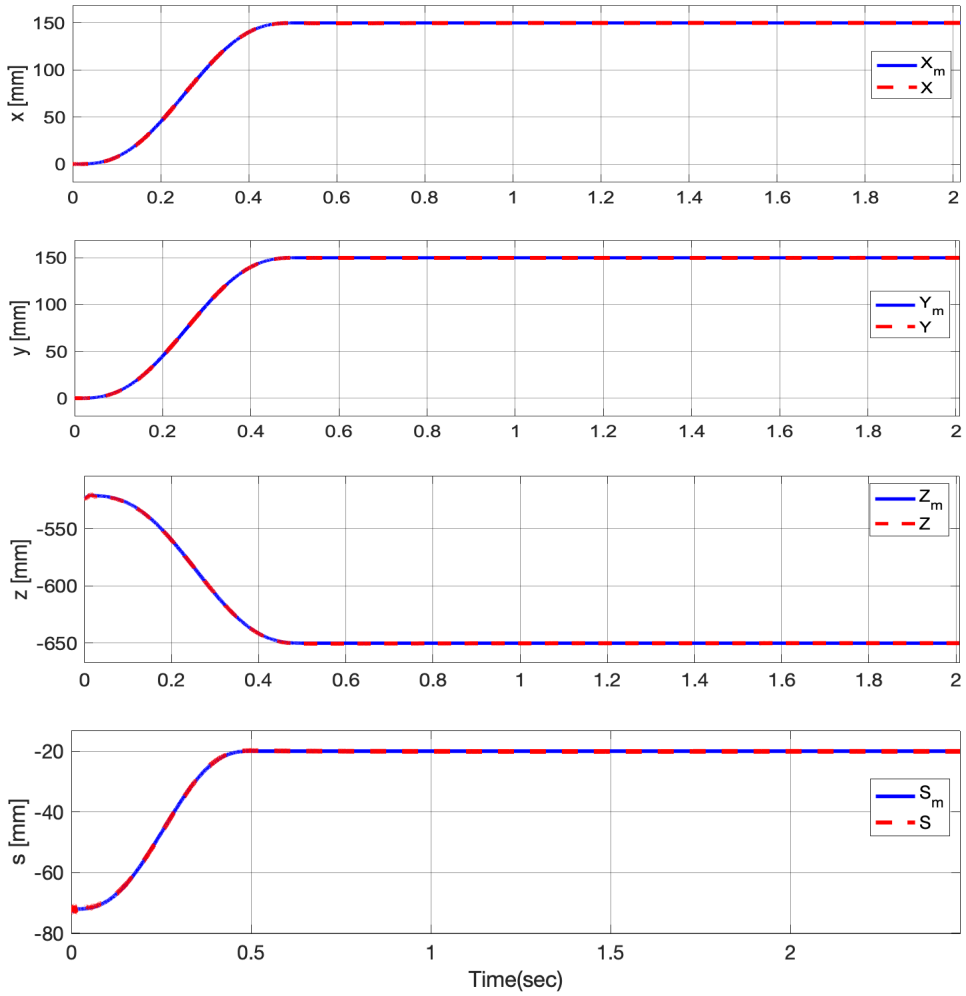


Figure 5.2: Case 1.1- Evolution of the end-effector's coordinates for trajectory 1

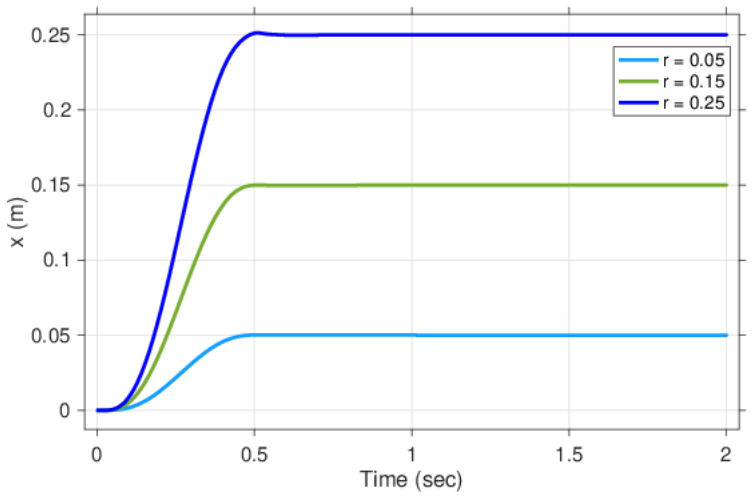


Figure 5.3: Case 1.1 - Evolution of x coordinate for different references

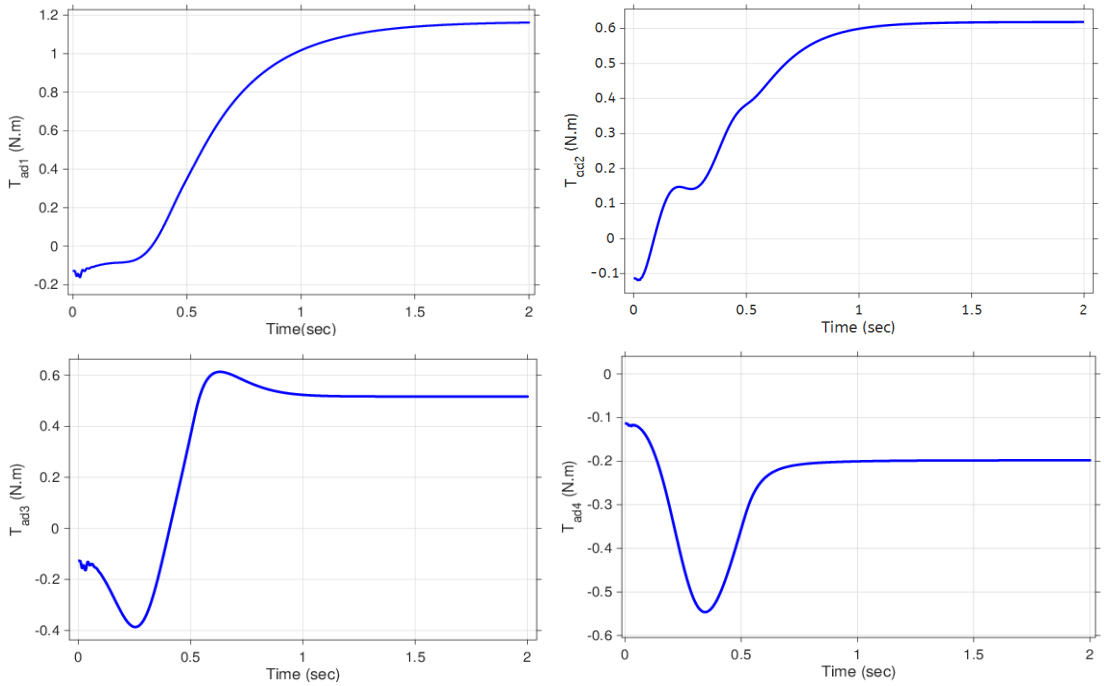


Figure 5.4: Case1.1 - Evolution of the adaptive control input torques $\Gamma_{AD}(t)$ for trajectory 1

Trajectory 2: Sequence of point-to-point motion

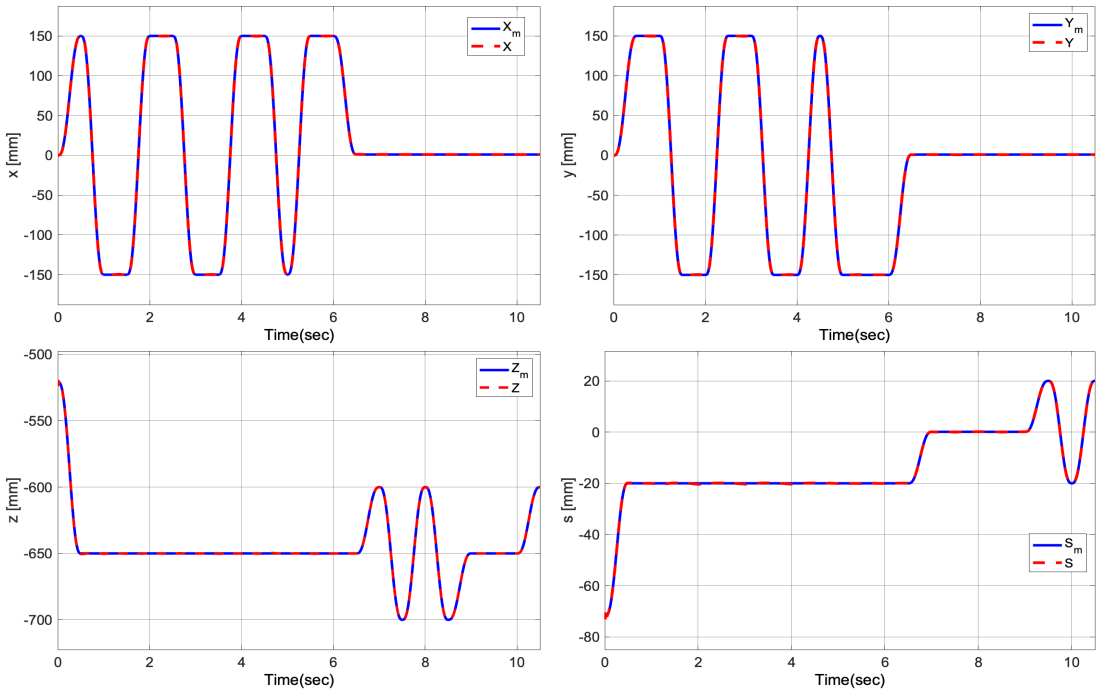


Figure 5.5: Case 1.1 - Evolution of Cartesian coordinates for trajectory 2

The trajectory proposed here is similar to the example represented in Fig 2.9,consisting of a sequence of points with a time interval of $T = 0.5\,s$ for each displacement. This trajectory more closely resembles a real-world scenario of pick and place task. The sequence of points used to generate it are given in Appendix 6.

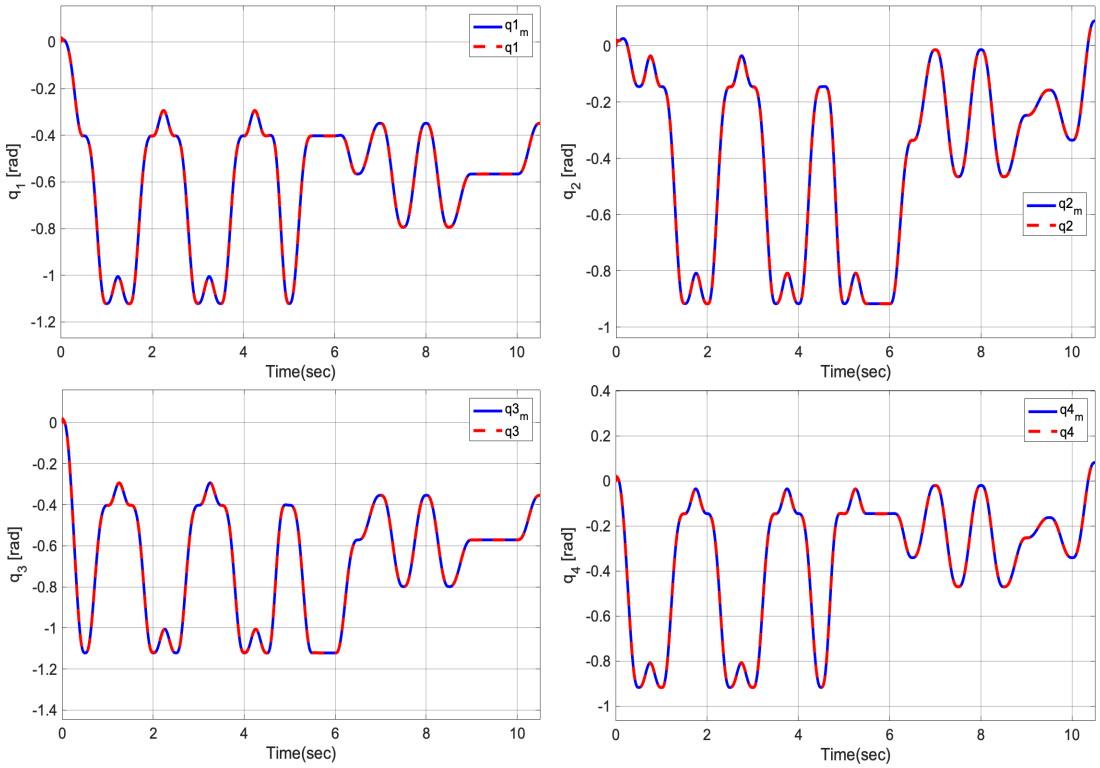


Figure 5.6: Case 1.1 - Evolution of the actuators' joints for trajectory 2

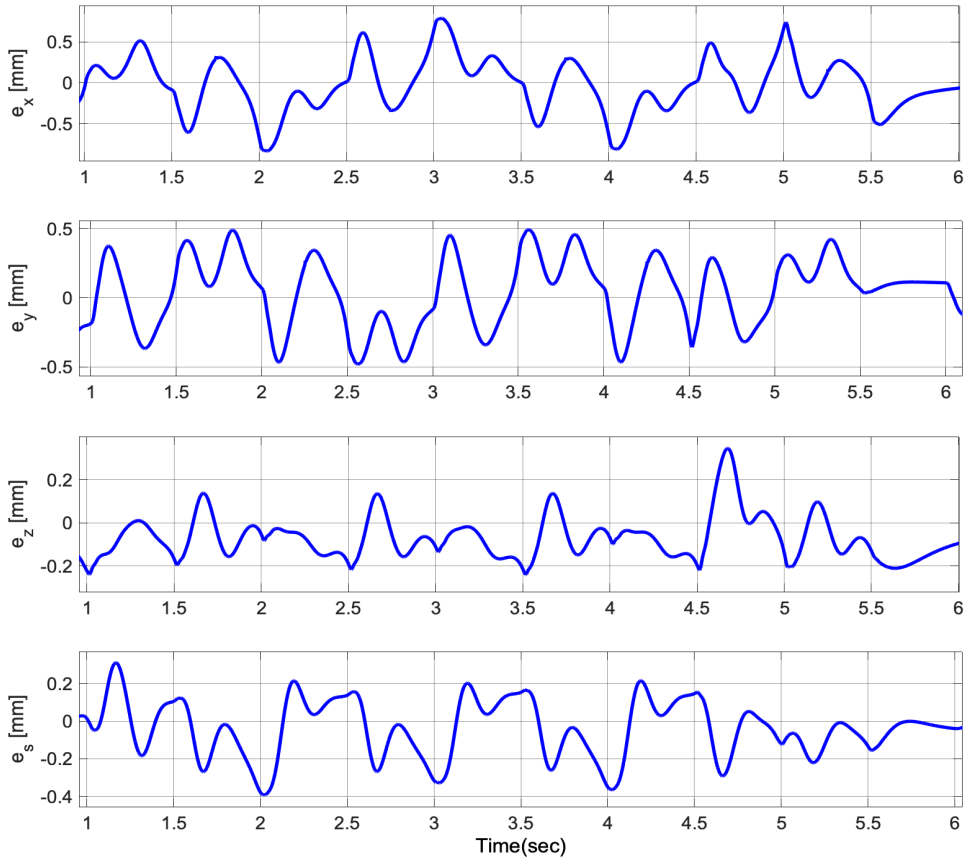


Figure 5.7: Case 1.1 - Evolution of Cartesian tracking errors for trajectory 2

6.1.2 Case 1.2: Fractional order adaptation law

The fractional coefficient is set for $\beta = 0.8$.

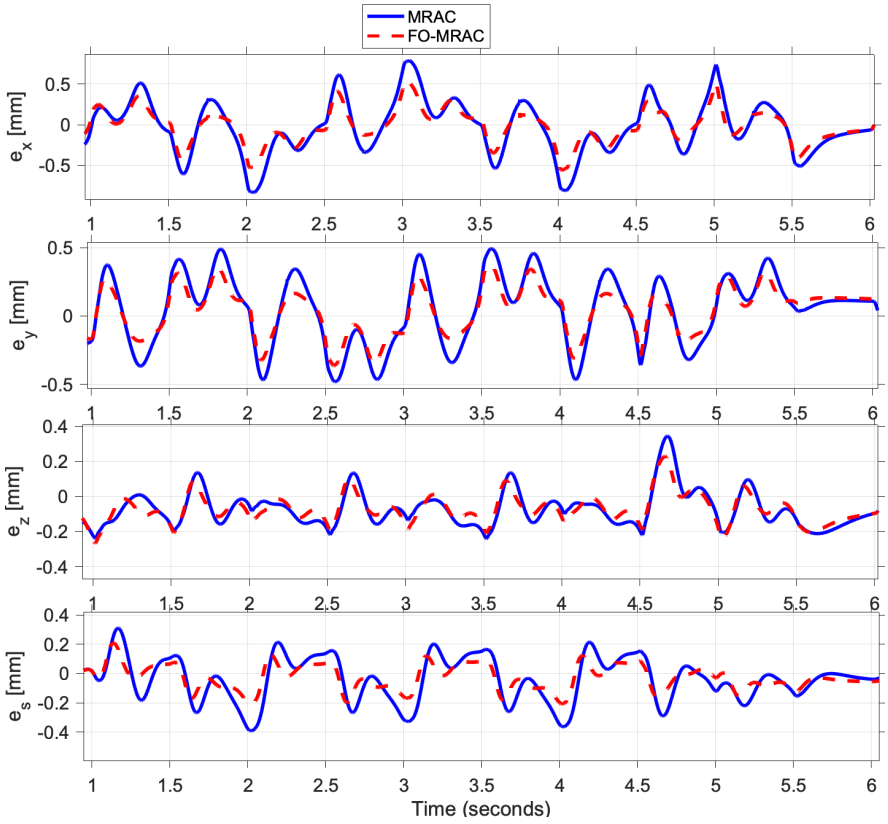


Figure 5.8: Case 1.2 - Evolution of end-effector's coordinate error

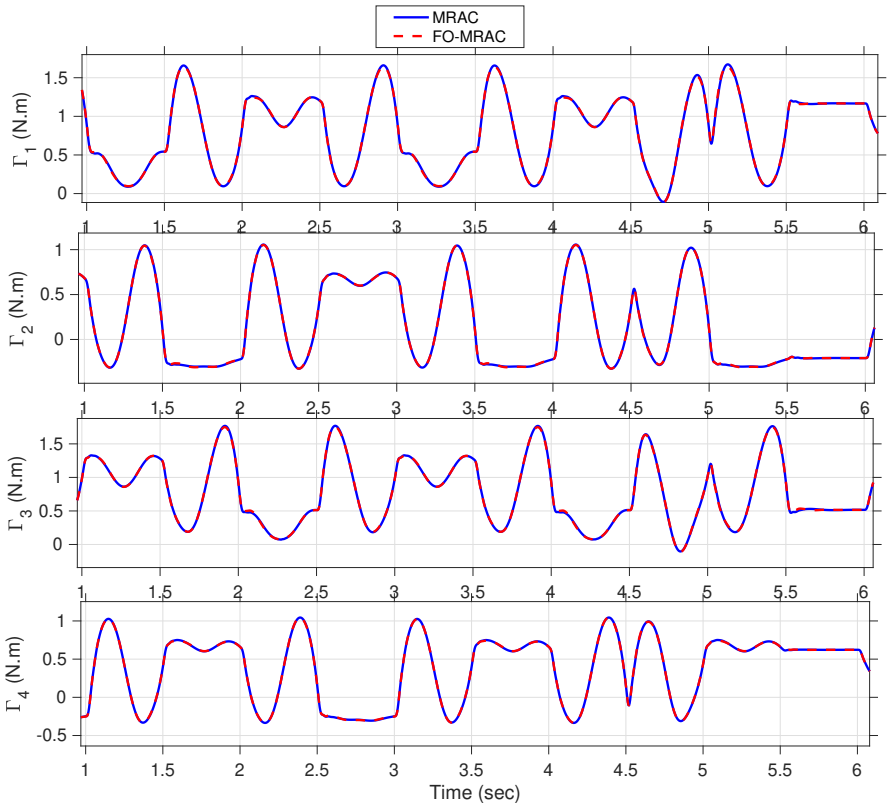


Figure 5.9: Case 1.2- Evolution of the control input torques $\Gamma(t)$

The performance in the previous IO and FO cases can be quantified using the performance metrics as defined in Appendix 7. The indices for the joints are evaluated in *deg* and for the Cartesian coordinates in *mm*.

Table 5.3: Performance indices for MRAC-PD controller for nominal conditions

Case	$\mathbf{IAE}_c$	$\mathbf{IAE}_J$	$\mathbf{ITAE}_c$	$\mathbf{ITAE}_J$	$\mathbf{RMSE}_c$	$\mathbf{RMSE}_J$	**IUE**
IO case	5.9	1.5126	22	6.0734	0.3984	0.0974	26.2650
FO case	4.7	1.2204	17.2	4.9217	0.3150	0.0802	26.2580

6.2 Results discussion

The results of applying the MRAC-PD controller to the VELOCE robot can be summarized and discussed through the following key observations points:

- The MRAC-PD controller demonstrated acceptable tracking performance across all tested cases. The simulation graphs reveal that there were no significant deviations from the reference trajectories, indicating precise tracking. This is further supported by the reduced range of tracking errors, measured in millimeters, compared to the manipulator's range of motion, which is in the order of 100 millimeters.

- The dynamics of the adaptive term of the controller is seen in Fig 5.4 showing the evolution of the adaptive input torques.The convergence rate of the adaptive input is approximately 1 second, which is slower than the convergence rate of the tracking error. This can be attributed to the PD controller's ability to initially reduce the error to smaller values, allowing the adaptive controller to compensate for the remaining nonlinearities and achieve greater positional accuracy.

- The controller successfully managed to maintain stability and tracking performance for different reference points as illustrated in Fig 5.3, which indicates the robustness of the MRAC-PD controller in handling different operational scenarios.

- The introduction of FO integrator in the adjustment rules has proven to bring improvements in performance by reducing the peaks of the position errors (as shown in Fig 5.8). This enhancement is reflected in the performance indices, with a 20% reduction in both $\mathbf{RMSE}_c$ and $\mathbf{IAE}_c$ indices.

- The FO adaptive controller achieved superior tracking results without increasing the energy of the control signals. Both IO and FO cases maintained control signal energies within acceptable ranges, not exceeding $127\,N.m$.

6.3 Case 2: Payload variations

This case is studied because PKMs are frequently used in applications where they transport payloads from one place to an other, leading to disturbances in the form of parametric variations (i.e., changes in the inertia and mass of the manipulator). Here a mass of 2 Kg is added to the moving platform of the robot with a period of 2s.

6.3.1 Case 2.1: Integer order adaptation law

Trajectory 1: Single point-to-point motion

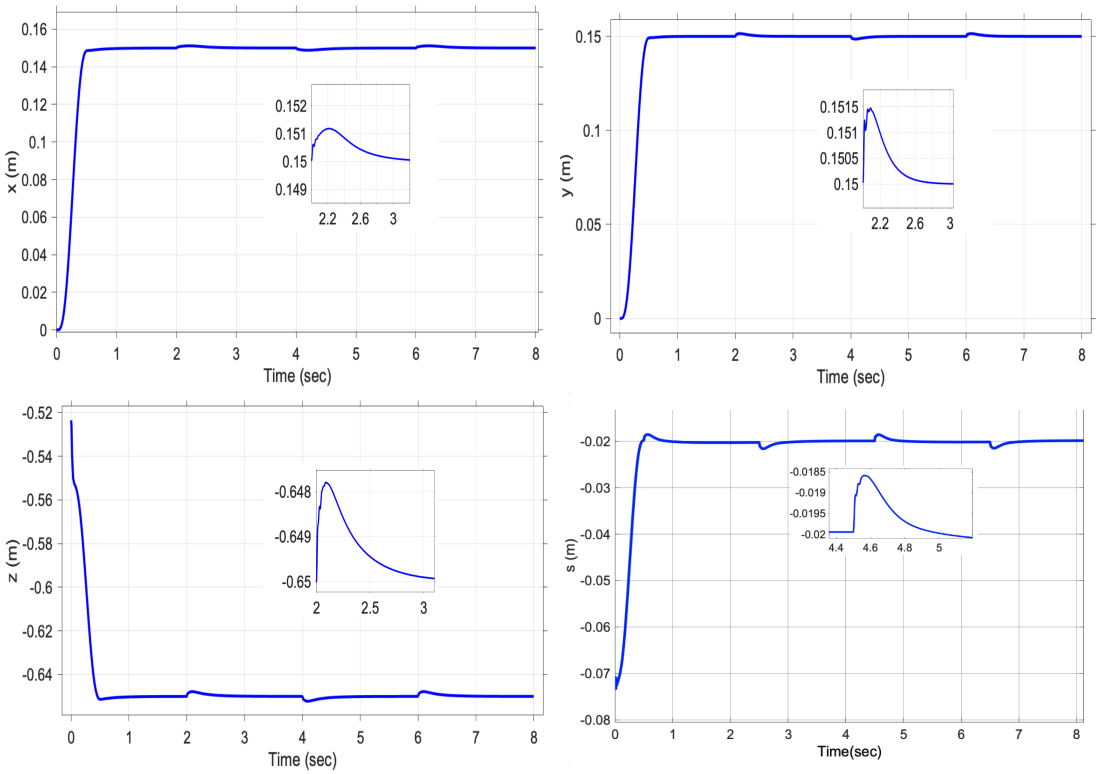


Figure 5.10: Case 2.1 - Evolution of the end-effector's coordinates

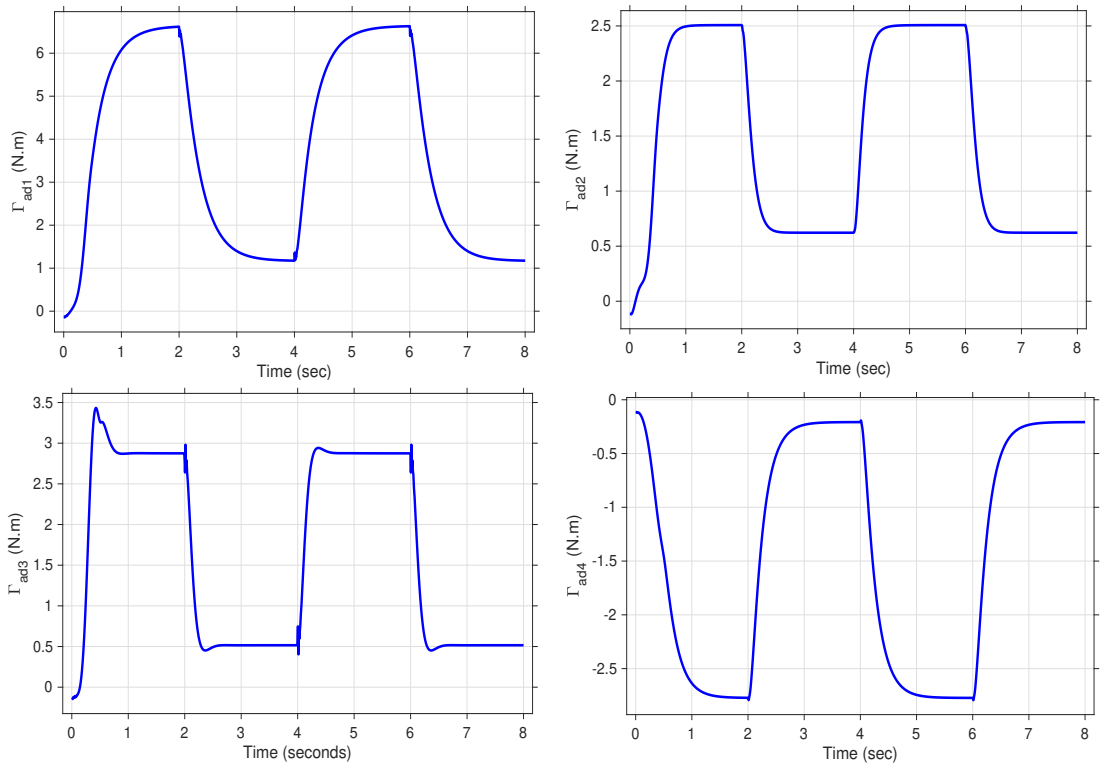


Figure 5.11: Case 2.1 - Evolution of the adaptive control input torques $\Gamma_{AD}(t)$

Trajectory 2: Sequence of point-to-point motion

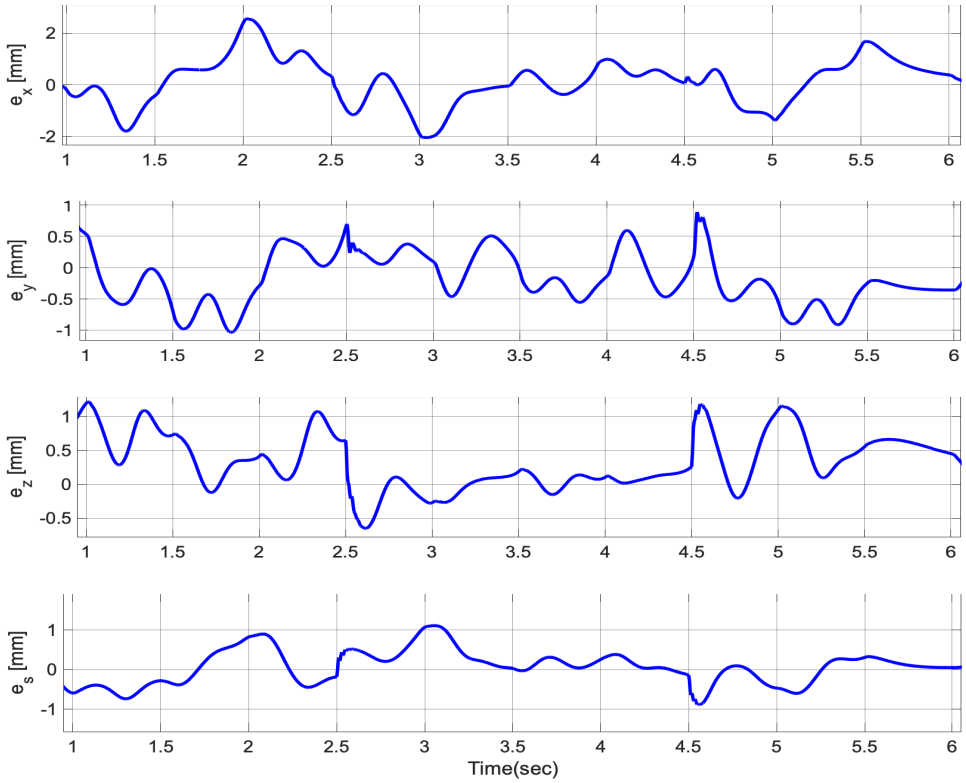


Figure 5.12: Case 2.1 - Evolution of Cartesian tracking errors for trajectory 2

6.3.2 Case 2.2: Fractional order adaptation law

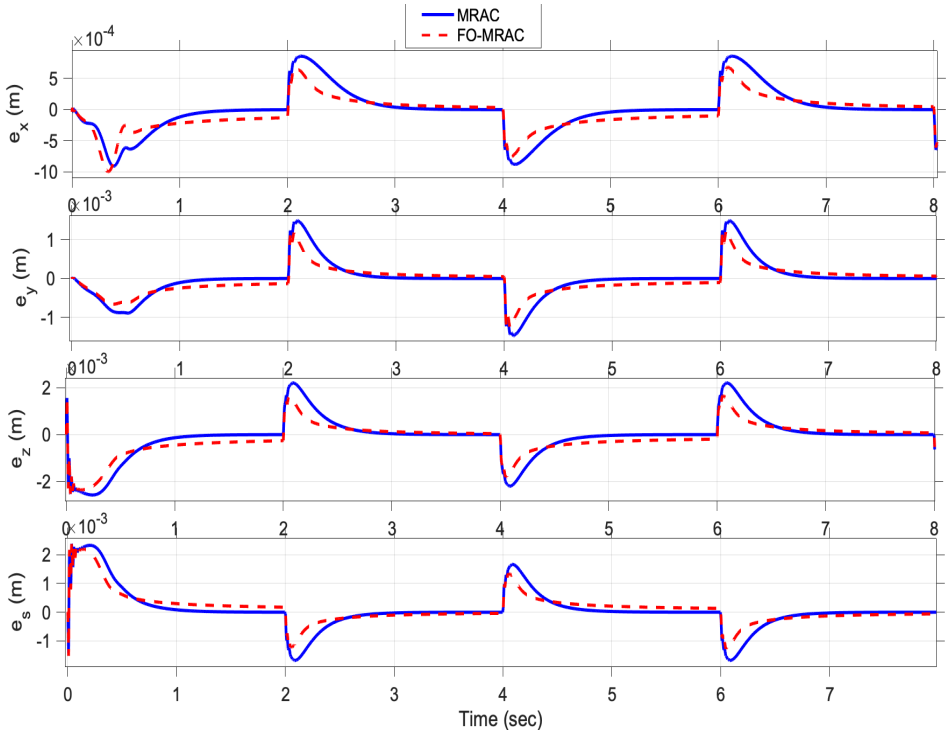


Figure 5.13: Case 2.2 - Evolution of Cartesian tracking error trajectory 1

In this case we retained the same fractional order coefficient as in Case 1.2.

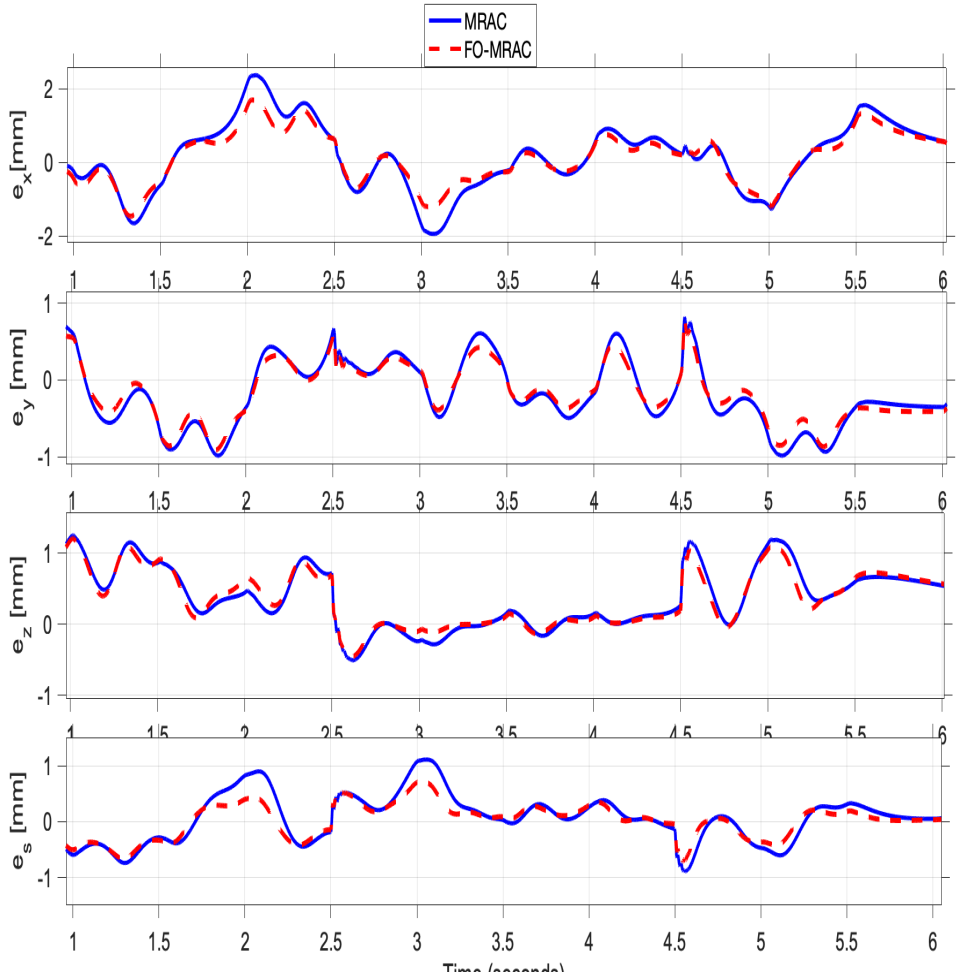


Figure 5.14: Case 2.2 - Evolution of Cartesian tracking error trajectory 2

In Table 5.4, the performance indices evaluated for both joint and Cartesian tracking errors are computed in *mm* and *deg* respectively.

Table 5.4: Performance indices for MRAC-PD controller in the presence of payload variations

Case	IAE_c	IAE_J	$ITAE_c$	$ITAE_J$	$RMSE_c$	$RMSE_J$	IUE
IO case	9.2	1.9882	42.3	8.5543	0.9854	0.2177	99.7900
FO case	8.8	1.8621	40.0	8.0959	0.7194	0.1604	99.7829

6.4 Results discussion

We have assessed the robustness capabilities of the MRAC-PD controller through this case study. The key observations are discussed below:

- The effect of a time-varying payload on the moving platform is evident in the end-effector's coordinates shown in Fig 5.10. Although there is a noticeable deviation from the reference point, the percentage of the overshoot is very small, which indicates the efficiency of the controller despite the added payload..
- The MRAC-PD controller can achieve faster disturbance attenuation and smaller overshoot by increasing the adaptation rates. However, this may introduce oscil-

lations and potentially lead to system instability. Therefore, a trade-off must be made between achieving high performance and maintaining system stability.

- We see from the results that An alternative method to reduce the impact of disturbances is by integrating FO adjustment rules. Performance indices indicate an improvement in tracking error with a drop of 2% in $\mathbf{IAE}_c$, which is less than in the nominal case. However, the $\mathbf{RMSE}_c$ indicates a reduction in the mean value of the Cartesian errors of 27%.
- The integration of FO adjustment rules affects the attenuation rate, as seen in Fig 5.13 as the controller takes more time to compensate for parametric variations. However, tuning the fractional coefficient provides greater flexibility in the control design, allowing for the specification of more tailored control requirements.

7 Conclusion

In this chapter, we designed and evaluated an MRAC-PD controller for the VELOCE robot which demonstrated robust tracking performance with minimal deviations from reference trajectories, even with time-varying payloads. Significant improvements were achieved by incorporating Fractional Order (FO) integration into the adaptation mechanisms. This integration resulted in enhanced tracking performance, increased robustness to disturbances, and greater flexibility in meeting control specifications.

If the tracking performance of this controller is compared the PD-FF controller presented in Chapter 2, we observed that the PD-FF controller achieved lower tracking errors. However, the MRAC-PD controller's strength lies in its minimal requirement for prior knowledge about the controlled system and its inherent adaptive structure. This makes it particularly well-suited for applications involving PKMs in dynamic and constantly changing operating environments.

The MRAC-PD controller is ideal for applications where extreme precision and accuracy are not critical, due to its simplicity in development and reasonable hardware computational requirements. However, for safety-critical applications where higher precision is necessary, more sophisticated control strategies may be required to ensure optimal performance and reliability.

In the next chapter, an $\mathcal{L}_1$ adaptive controller with a model based feed-forward term will be designed for the VELOCE robot, with simulation results presented and discussed.

Chapter 6

$\mathcal{L}_1$ adaptive controller with model-based feed-forward for VELOCE manipulator

1 Introduction

This chapter aims to design a Fractional Order $\mathcal{L}_1$ adaptive controller with a feedforward component based on the reference trajectory. The rationale for selecting this specific control approach and the control objectives are first outlined. Subsequently, the theoretical basis of the control law is presented, along with the integration of Fractional Order adaptation rules into the control scheme. Once the control design is finalized, the simulation setup and results are described, considering both nominal conditions and the presence of payload variations.

2 Motivation and objectives

Robot manipulators are commonly used in safety-critical applications such as medical surgery and flight control. These applications require control systems that can quantify transient and steady-state performance while ensuring accurate tracking of reference trajectories by canceling internal and external disturbances.

The $\mathcal{L}_1$ adaptive approach meets most of these requirements, especially the robustness guarantee under fast adaptation rates to meet performance specifications. It has been tested for various robot manipulators to design controllers with marginal stability in the presence of time delays [99]. Additionally, it has been used for safe motion planning for autonomous robots [100], as well as increasing the robustness of a pre-trained agent with reinforcement learning (RL) techniques to achieve active compensation of dynamic variations [101].

Moreover, adding a model-based term with the $\mathcal{L}_1$ adaptive controller can compensate for modeled non-linear dynamics, further improving tracking performance. As obtaining an accurate dynamic model has become relatively easier recently, taking advantage of this knowledge can lead to improved tracking performance and lower adaptation rates, allowing less overshoot in the control signal and reduced sensitivity to time delays [102], [45].

The objective of this section is to design a $\mathcal{L}_1$ adaptive controller for the VELOCE robot to satisfy control specifications related to transient and steady-state performance. The choice of the filter structure will determine the trade-off between performance and robustness. Additionally, integrating model-based control with the adaptive controller will enable compensation for modeled nonlinearities. This control scheme leverages the robustness of the $\mathcal{L}_1$ controller and improves accuracy, even in high-speed applications, thanks to the model-based feedforward term. Furthermore, a fractional-order filter will be employed in this control scheme to achieve better bandwidth selectivity, and a fractional adaptation law will be implemented to reduce improve performance and enhance stability margins.

3 Design of $\mathcal{L}_1$-FF adaptive controller for VELOCE robot

In the following section, we will build from the theory presented in Chapter 3 and present an extension of $\mathcal{L}_1$ adaptive control to PKMs that was developed in [53], in addition a model-based term will be added to the control law.

3.1 Control law

The dynamics of PKMs can be described by Equation (6.1) as explained in Section 2.6, where $q \in \mathbb{R}^4$ is the vector of joint angles:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \Gamma_D(t) = \Gamma_T(t) \tag{6.1}$$

The the tracking error used here is a combination of position and velocity errors in the joint space as expressed in Equation (6.2):

$$r(t) = (\dot{q} - \dot{q}_d) + \Lambda(q - q_d) \tag{6.2}$$

with $\Lambda \in \mathbb{R}^{4\times4}$ a symmetric positive-definite weighting matrix for the position tracking error.

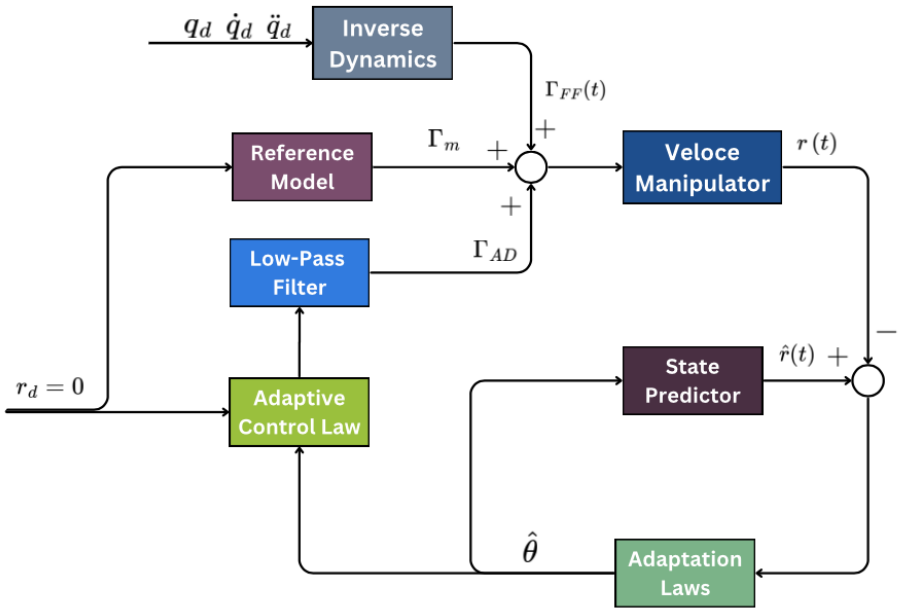


Figure 6.1: $\mathcal{L}_1$ adaptive control scheme with model-based feed-forward compensation

The control law is composed of three distinct terms as follow:

$$\Gamma(t) = \Gamma_m(t) + \Gamma_{ad}(t) + \Gamma_{FF} \tag{6.3}$$

with $\Gamma_m(t) = A_m r(t)$, a linear state-feedback control and A_m denotes a Hurwitz matrix which is introduced to shape the transient dynamics of the tracking error $r(t)$,
$\Gamma_{FF} = M(q_d)\ddot{q}_d + C(q_d,\dot{q}_d)\dot{q}_d + G(q_d)$ is the feed-forward term computed from the desired

trajectory and is Γ_{ad} the adaptive part of the controller.
Leveraging knowledge about the manipulator's dynamics will result in improved tracking performance. Consequently, the adaptive control component will have a lower magnitude since the estimate $\eta(t, \hat{\zeta}(t))$ will only need to account for unstructured uncertainties such as joint friction, actuator dynamics, and external disturbances. This approach helps avoid high-frequency and high-gain control efforts.

To design $\Gamma_{AD}(t)$, we first start by substituting the control law Equation (6.3) into the dynamics of the parallel manipulator Equation (6.1) and solving for $\ddot{q}$ to derive the derivative of the tracking error $r(t)$:

$$\dot{r}(t) = A_m r(t) + \Gamma_{AD}(t) - \eta(t, \zeta(t)), \quad r(0) = r_0 \tag{6.4}$$

where $\eta(t, \zeta(t))$ is a nonlinear function that gathers all the non-linearities of the system including uncertainties and external disturbances and is given by the following expression:

$$\begin{aligned} \eta(t, \zeta(t)) = M^{-1}(q) \left(\tilde{M}(q)\ddot{q}_d + \tilde{N}(q, \dot{q}) \right) \\ + \left(\mathbb{I} - M^{-1}(q) \right) \left(A_m r + \Gamma_{AD} \right) \\ - \Lambda r(t) + \Lambda^2 \left(q - q_d \right) \end{aligned} \tag{6.5}$$

where $\tilde{M}(q) \triangleq M(q) - \hat{M}\left(q_d\right), \tilde{N}(q, \dot{q}) \triangleq N(q, \dot{q}) - \hat{N}\left(q_d, \dot{q}_d\right)$ and $\mathbb{I} \in \mathbb{R}^{4\times4}$ denotes the identity matrix.

Some assumptions are imposed about the properties of $\eta(t, \zeta(t))$ to give it a more general structure than its expression given in Equation (6.5):

Assumption 3.1 (***Uniform boundedness of*** $\eta(t, 0)$)
There exist $B > 0$ such that $\forall t \geq 0, \|\eta(t, 0)\| \leq B$.

Assumption 3.1 ***(Semi-global uniform boundedness of partial derivatives of*** $\eta(t, \zeta(t))$ ***)***
The unknown nonlinear function $\eta(\mathrm{t}, \zeta(\mathrm{t}))$ is continuous with respect to its arguments, and for arbitrary $\delta > 0$, there exist $\mathrm{d}_{\eta t}(\delta)$ and $\mathrm{d}_{\eta x}(\delta)$ such that

$$\left\| \frac{\partial \eta(t, \zeta)}{\partial t} \right\|_\infty \leq d_{\eta_t}(\delta), \quad \left\| \frac{\partial \eta(t, \zeta)}{\partial \zeta} \right\|_\infty \leq d_{\eta_\zeta}(\delta)$$

Assumption 3.1
For $\mathrm{t} \geq 0, \|r_\tau\|_{\mathcal{L}_\infty} \leq \rho$ and $\|\dot{r}_\tau\|_{\mathcal{L}_\infty} \leq \mathrm{d_r}$, for some positive constants ρ and $\mathrm{d_r}$.

According to [76], it follows from the assumptions 3.1, 3.1 and 3.1 that the unknown nonlinear function $\eta(t, \zeta(t))$ can be expressed for $\forall t \in [0, \tau]$ as follows:

$$\eta(t, \zeta(t)) = \theta(t) \|r_\tau\|_{\mathcal{L}_\infty} + \sigma(t) \tag{6.6}$$

where $\theta(t), \sigma(t) \in \mathbb{R}^4$ are continuous unknown functions, with their first derivatives uniformly bounded on $[0, \tau]$, $\|\|_{\mathcal{L}_\infty}$ represent the $\mathcal{L}_\infty$-norm and $r_\tau(t)$ is the truncation of $r(t)$ defined as follow:

$$r_\tau(t) = \begin{cases} r(t), & 0 \leq t \leq \tau \\ 0, & t > \tau \end{cases} \tag{6.7}$$

The estimate of the non-linear function $\hat{\eta}(t, \zeta(t))$ is computed in real-time. The dynamics of the predicted tracking error $\hat{r}(t)$ in terms of the estimated parameters are described by the following expression:

$$\dot{\hat{r}}(t) = A_m \hat{r}(t) + \Gamma_{ad}(t) - \left(\hat{\theta}(t) \left\|r_t\right\|_{\mathscr{L}_\infty} + \hat{\sigma}(t)\right) - K\tilde{r}(t), \quad \hat{r}(0) = r_0 \tag{6.8}$$

where the prediction error is defined as $\tilde{r}(t)= \hat{r}(t) - r(t)$, $\hat{\theta}(t)$ and $\hat{\sigma}(t)$ are estimates of $\theta(t)$ and $\sigma(t)$, respectively. The matrix K contains loop-shaping parameters to characterise the dynamics of the estimation error and reject high frequency noise.

The dynamics of the prediction error are obtained by substituting Equation (6.4) from Equation (6.8) as follow:

$$\dot{\tilde{r}}(t) = A_o \tilde{r}(t) - \left(\tilde{\theta}(t) \left\|r_t\right\|_{\mathscr{L}_\infty} + \tilde{\sigma}(t)\right) \tag{6.9}$$

The matrix $A_o = (A_m - K)$ is chosen to guaranty convergence of the prediction error to zero, $\tilde{\theta}$ and $\tilde{\sigma}$ are the errors between the estimate and real values of θ and σ respectively.

Finally, the adaptive element is chosen to ensure the desired closed-loop dynamics of the prediction error $r(t)$ with an addition of low-pass filter:

$$\Gamma_{\text{ad}}(s) = C(s)\hat{\eta}(s) \tag{6.10}$$

The estimates of $\hat{\theta}$ and $\hat{\sigma}$ are calculated by the following projection-based adaptation laws:

$$\begin{aligned} \dot{\hat{\theta}}(t) &= \gamma \operatorname{Proj}\left(\hat{\theta}(t), P\tilde{r}(t) \left\|r_t\right\|_{\mathcal{L}_\infty}\right), \hat{\theta}(0) = \hat{\theta}_0 \\ \dot{\hat{\sigma}}(t) &= \gamma \operatorname{Proj}(\hat{\sigma}(t), P\tilde{r}(t)), \quad \hat{\sigma}(0) = \hat{\sigma}_0 \end{aligned} \tag{6.11}$$

where $\gamma \in \mathbb{R}^+$is the adaptive gain, $P = P^T > 0$ is the solution to the algebraic Lyapunov equation $A_m^T P + PA_m = -Q$ for some arbitrary matrix $Q = Q^T > 0$.

3.2 Filter design

The adaptive control term is passed through a low-pass filter, as expressed in Equation (6.10) that limits the bandwidth of the controller enabling the decoupling of the adaptation and control loops. This filter's structure is carefully designed to ensure optimal performance while accounting for hardware constraints and stability specifications in terms of the $\mathcal{L}_1$ gain requirement as defined in Equation (3.30).

The low-pass filter serves multiple roles. It helps prevent high frequencies in the control signals, which can cause wear and deterioration (mechanical vibrations and oscillations), while also shaping the nominal response of the closed-loop system. Increasing the bandwidth of the low-pass filter has been observed to render performance bounds (the norm $\|G(s)\|_{\mathcal{L}_1}$) arbitrarily small [76].

However, low-pass filters with high bandwidths require high adaptation rates, potentially leading to high-gain feedback and resulting in closed-loop systems with overly small robustness margins that are susceptible to measurement noise. Conversely, properly designed higher order filters have been proven to achieve similar performance bounds with smaller bandwidth compared to first-order filters [103].

4 Fractional order $\mathcal{L}_1$ adaptive controller

Papers and works available focusing on fractional order adaptive control have mostly centered around integrating fractional order calculus for MRAC schemes. Considering the benefits achieved by using fractional order with MRAC, it is intriguing to explore its potential application in the L1 adaptive control method to see if similar improvements can be obtained. There are two approaches to consider: either using a fractional order low-pass filter as demonstrated in [104], or applying fractional order adjustment rules within the adaptation mechanism, akin to what was done with MRAC.

4.1 Fractional order adaptation laws

Just like in Section 5.4, the adaptation rules will be modified to incorporate a fractional order integration. The new adjustment rules of the estimates of $\hat{\theta}$ and $\hat{\sigma}$ will become:

$$\hat{\theta}(t) = D^{-\alpha}\left(\gamma \,\mathrm{Proj}(\hat{\theta}(t), P\tilde{r}(t)\,\|r_t\|_{\mathcal{L}_\infty})\right) \tag{6.12}$$

$$\hat{\sigma}(t) = D^{-\alpha}\left(\gamma \,\mathrm{Proj}(\hat{\sigma}(t), P\tilde{r}(t))\right) \tag{6.13}$$

with α representing the fractional order coefficient. The Singularity method is used for the implementation of the fractional adaptation rules.

4.2 Fractional order filter

Fractional-order filters offer several advantages by allowing for better selectivity in tuning bandwidths. They are commonly used in applications with extremely complex systems, offering solutions to significant problems such as reducing high control values, filtering noises effectively, and minimizing time lag [104]. A first order FO filter can be defined as follow:

$$C(s) = \frac{1}{(1 + \frac{s}{\omega_c})^\alpha} \tag{6.14}$$

with ω_c being the cut-off frequency and α (between 0 and 1) the fractional orders.

5 Simulation setup

In this section, we detail the selected parameters for the controller, designed to achieve the control objectives of accurate trajectory tracking and robustness against disturbances.

The filter and model reference are selected according to the $\mathcal{L}_1$-norm condition stated in Equation (3.30), but before this we need to find a maximum bound for the control input gain b.

We can rearrange Equation(2.20) of the manipulator's dynamics in the following way:

$$\ddot{q}(t) = M(q)^{-1}\Gamma(t) - M(q)^{-1}C(q,\dot{q})\dot{q} - M(q)^{-1}G(q) \tag{6.15}$$

Thus determining the maximum value of b comes to finding the $||M(q)^{-1}||_\infty$ norm at the limit of the reachable workspace where it reaches its highest values. The $||M(q)^{-1}||_\infty$ is plotted in Fig 6.2 , we conclude that a possible bound is $b = 160$.

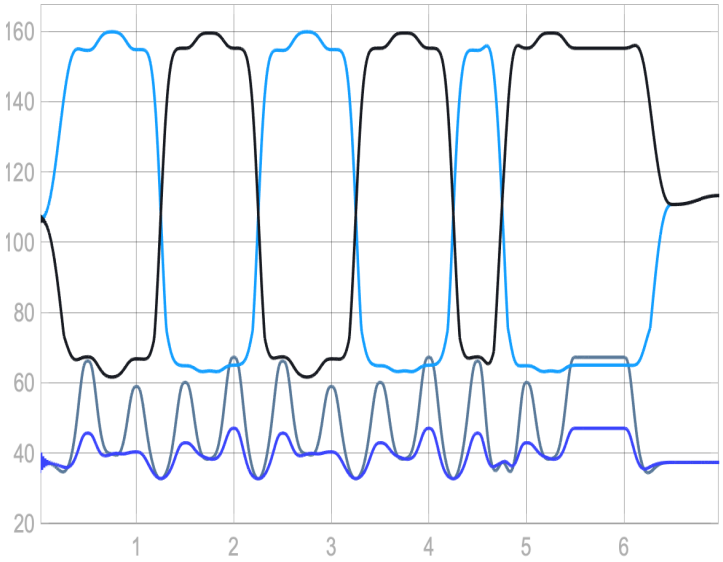


Figure 6.2: Maximum value for control input gain

If we check the $\mathcal{L}_1$-norm condition for the controller's parameters given in Table 6.1, we obtain:

$$G(s) = \frac{160}{s+320}\left(1 - \frac{1}{0.02s+1}\right) = \frac{3.2s}{0.02s^2 + 7.4s + 320} \tag{6.16}$$

As a result we obtain $\|G(s)\|_{\mathcal{L}_1} L = 0.70263 < 1$.

Table 6.1: $\mathcal{L}_1$-FF controller parameters

Parameters	**Value**
Λ	diag(40, 40, 40, 40)
A_m	diag(-8, -8, -8, -8)
γ	10^6
Q	diag(1 ,1, 1, 1)
K	$10^3 \times$diag(3, 3, 3, 3)
θ_{max}	30
σ_{max}	30
$C(s)$	$\frac{50}{s+50}$

6 Simulation results

To validate the $\mathcal{L}_1$-FF controller developed here we proceed into implementing the control law on Simulink and run a series of simulations. These simulations are conducted under both nominal conditions and in the presence of external disturbances, such as payload variations. The objective is to assess both the performance and robustness. Additionally, we analyze the implications of using FO adaptation rules and the effects of employing fractional-order filters.

6.1 Case 1: Nominal condition

The referenced trajectory used for the simulations in this case is given in Section 3.29. Since the $\mathcal{L}_1$ adaptive controller was designed to be used in fast applications, the time period between each displacement is set to $T = 0.2\,s$. The law-pass filter is taken as an IO first system ($\alpha = 1$).

6.1.1 Case 1.1: Integer order adaptation law

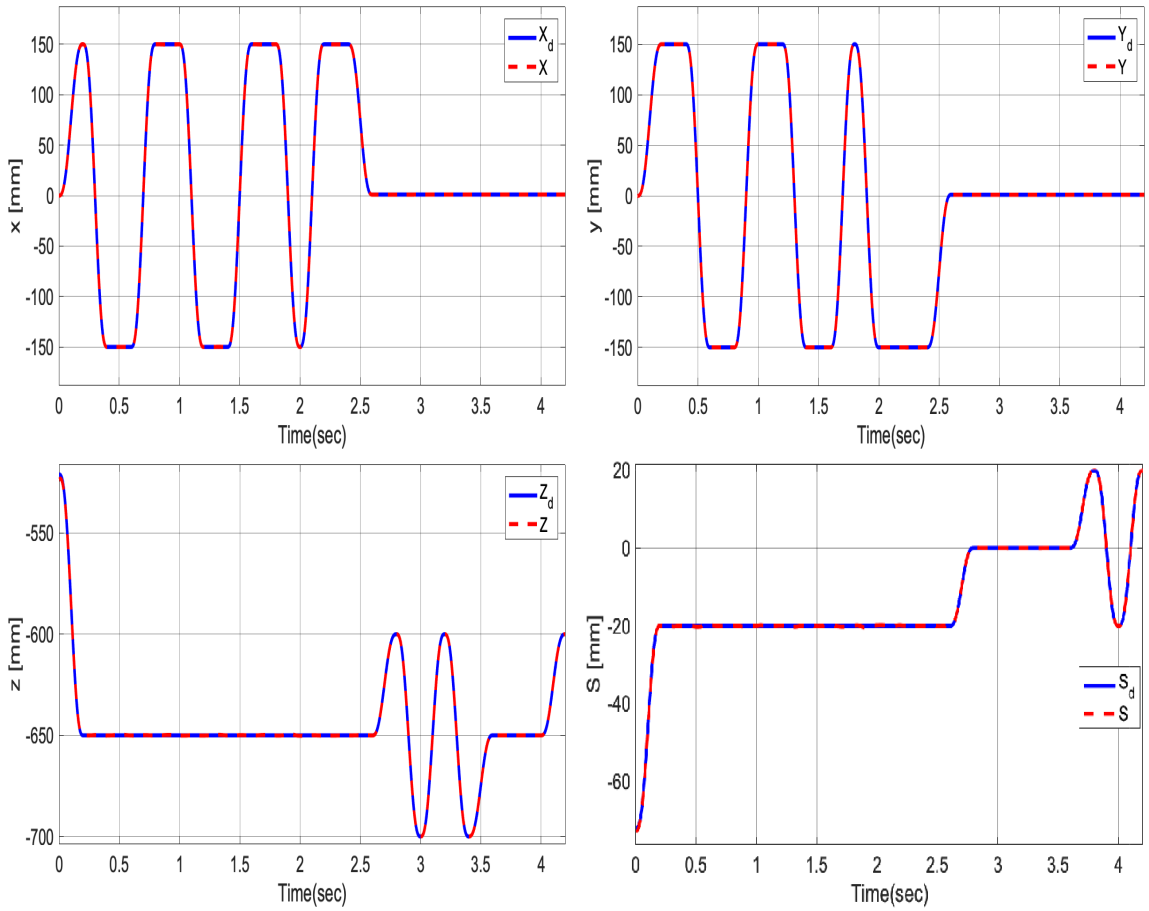


Figure 6.3: Case 1.1 - Evolution of end-effector coordinates

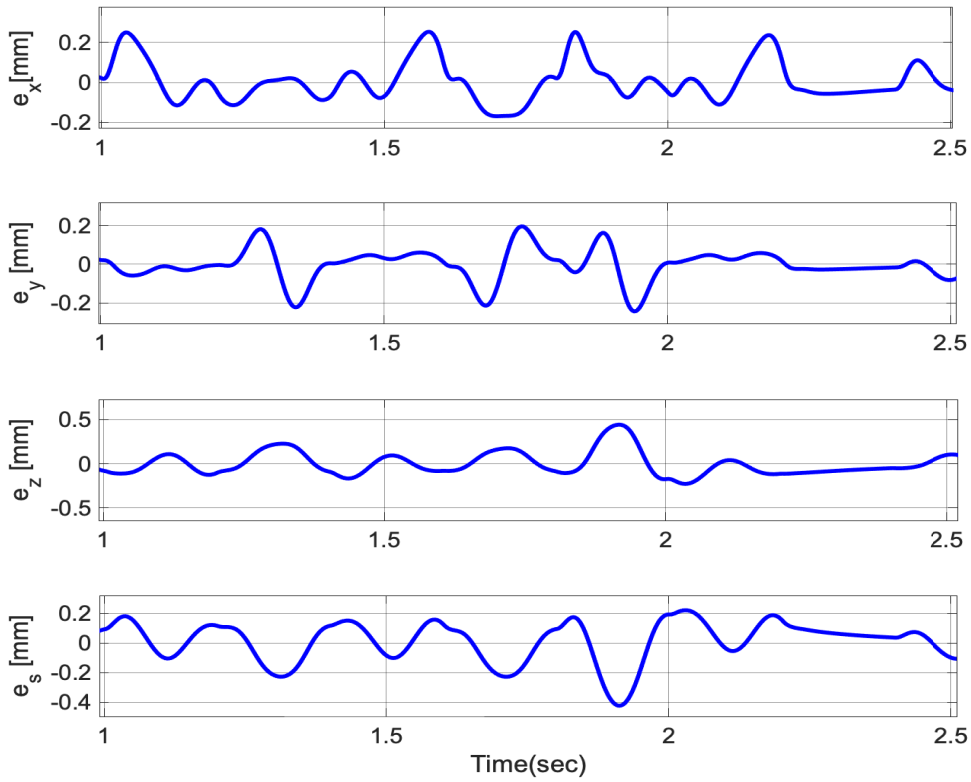


Figure 6.4: Case 1.1 - Evolution of the end-effector coordinates tracking errors

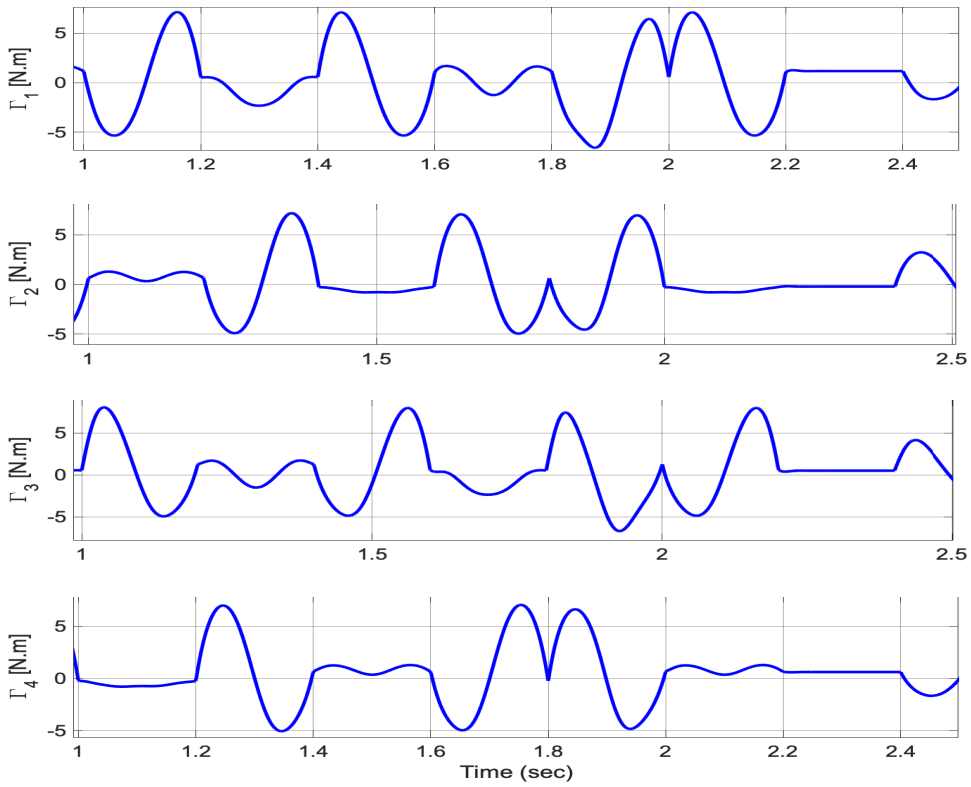


Figure 6.5: Case 1.1 - Evolution of the control inputs torques $\Gamma(t)$

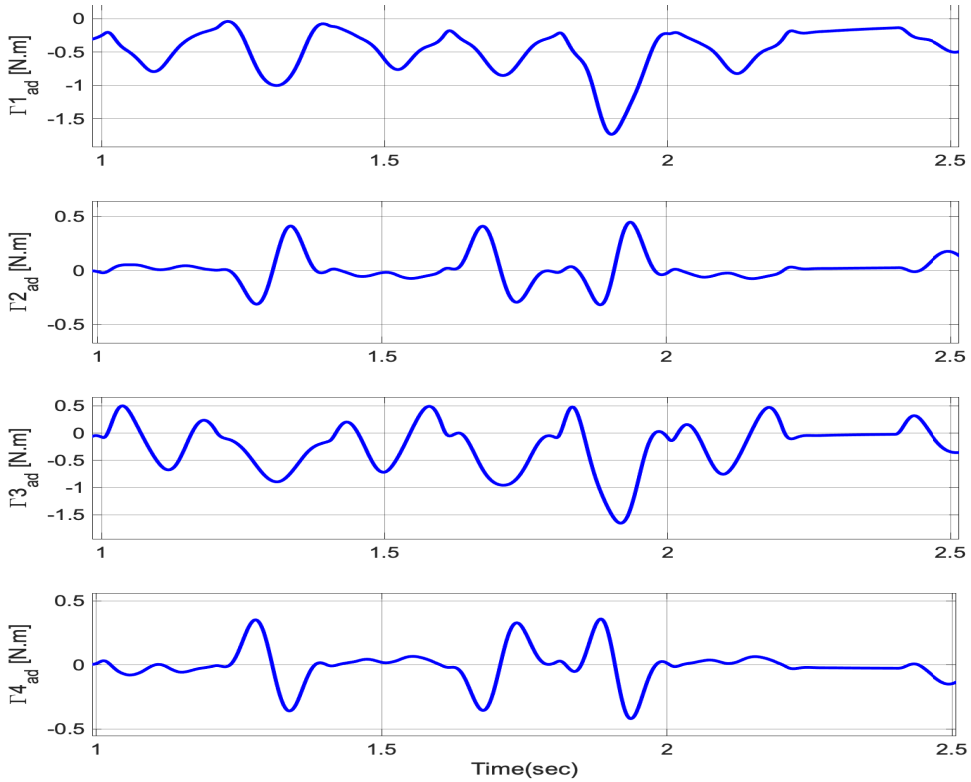


Figure 6.6: Case 1.1 - Evolution of the adaptive control input torques $\Gamma_{AD}(t)$

6.1.2 Case 1.2: Fractional order adaptation law

For the sake of this simulation, the fractional coefficient is chosen as $\beta = 0.75$.

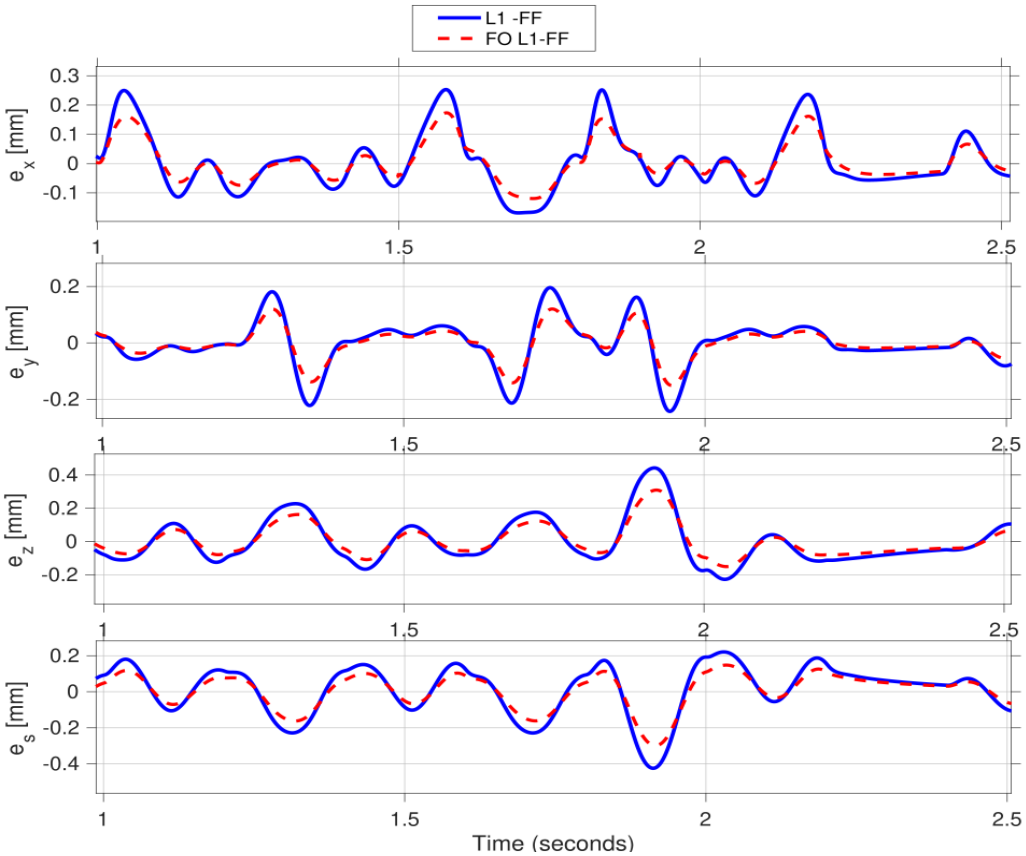


Figure 6.7: Case 1.2 - Evolution of Cartesian tracking errors

6.1.3 Case 1.3: Fractional order filter

The cut-off frequency of the FO filter is reduced to $\omega_c = 37\,rad.s^{-1}$, and the fractional order is set to $\beta = 0.55$.

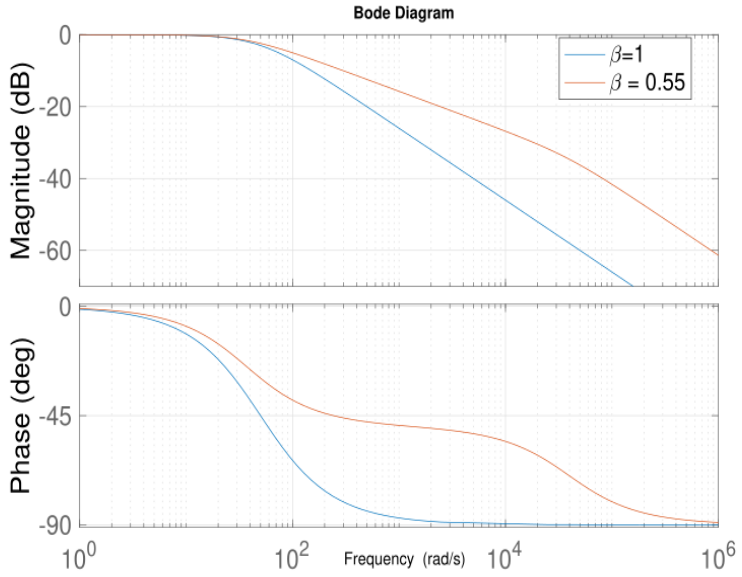


Figure 6.8: Case 1.3 - Comparison between FO ($\omega_c = 37\,rad$) and IO ($\omega_c = 50\,rad$) filters

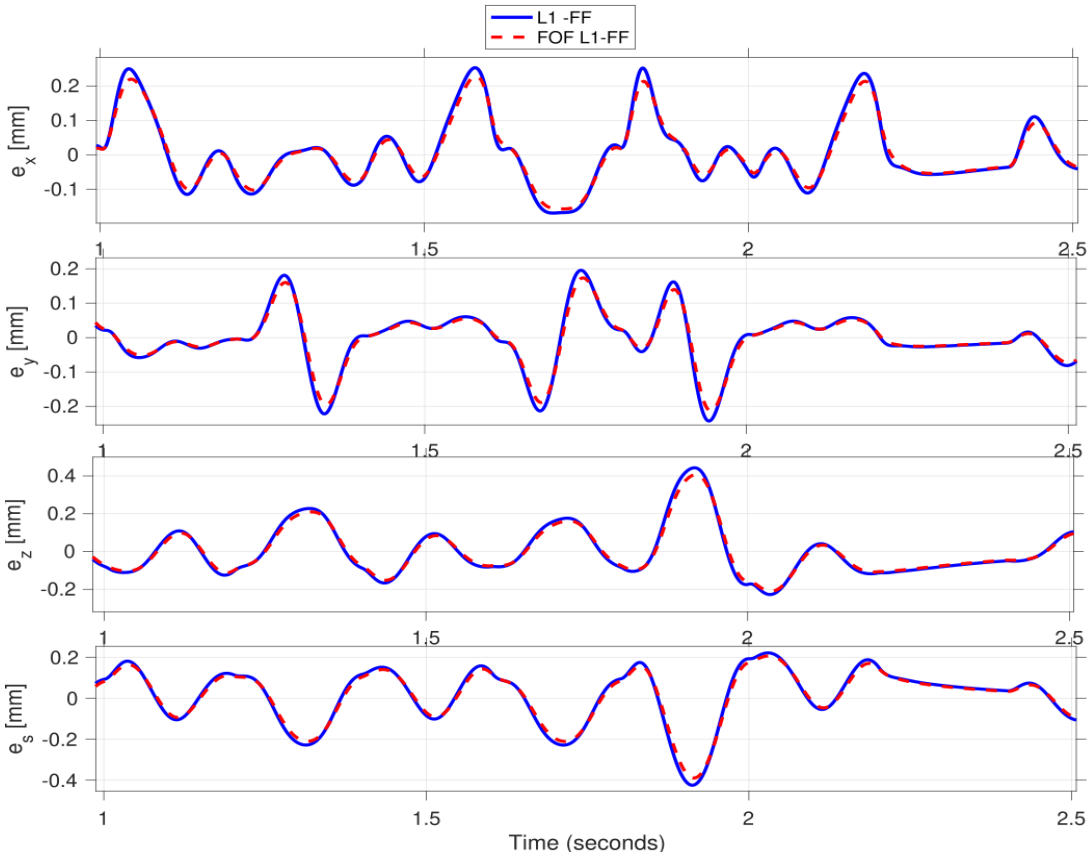


Figure 6.9: Case 1.3 - Evolution of Cartesian tracking errors

Table 6.2: Performance indices for $\mathcal{L}_1$-FF controller for nominal conditions

Case	$\mathbf{IAE}_c$	$\mathbf{IAE}_J$	$\mathbf{ITAE}_c$	$\mathbf{ITAE}_J$	$\mathbf{RMSE}_c$	$\mathbf{RMSE}_J$	**IUE**
IO adaptation	1.6	0.3023	1.0492	0.2693	0.2487	0.1203	28.3998
FO adaptation	0.7269	0.2370	0.8547	0.1822	0.2084	0.1159	28.3738
FO filter	0.9706	0.2868	1.016	0.2475	0.2380	0.1191	28.3918

6.2 Results discussion

From the simulation results for the three cases presented above, some relevant observations are made and discussed here:

- The tracking objective both in joint and Cartesian space (Fig 6.3) is achieved with great accuracy by the $\mathcal{L}_1$-FF controller, even with relatively fast displacement timest. This can be attributed to the decoupled architecture of the $\mathcal{L}_1$ control which allows for fast adaptation, in addition to the feedforward term that cancels most of the non-linearities, thereby reducing the effort required by the adaptive part of the controller

- The control input torques were always within the allowed range (maximum value 127 N.m), in addition the effect of the low-pass filter can be observed on the adaptive torques input presented in Fig 6.6, as there are no sharp or aggressive oscillations, even during fast adaptation.

- The fractional adaptation law dampens the overshoot of the system's tracking error and improve the accuracy and precision. This is reflected in an improvement of 50% in the accumulation of tracking error $\mathbf{IAE}_c$ as well as a decrease in the mean of the Cartesian error by 16% ($\mathbf{RMSE}_c$).

- When choosing the FO filter, we decreased its cutting frequency compared to the IO filter in order to preserve the system's ability to filter noise outside its bandwidth, since the slope of FO filters attenuating rate is less than their IO counterpart. From Fig 6.9, we observe that FO filter managed to achieve slightly better performance even with smaller bandwidth. In addition, FO filters improve the margin stability of the system because of their particular feature of having constant phase.

6.3 Case 2: Payload variation case

An important aspect in the control of uncertain systems that are subject to external changes leading to parametric variations is to have high rejection capabilities of disturbances. To test the robustness of $\mathcal{L}_1$-FF to payload variations, a time varying payload was introduced that has a mass of 2 Kg and changes every 2.8 seconds. A different reference trajectory is used here to highlight disturbance rejection during transient phase and steady state.

6.3.1 Case 2.1: Integer order adaptation law

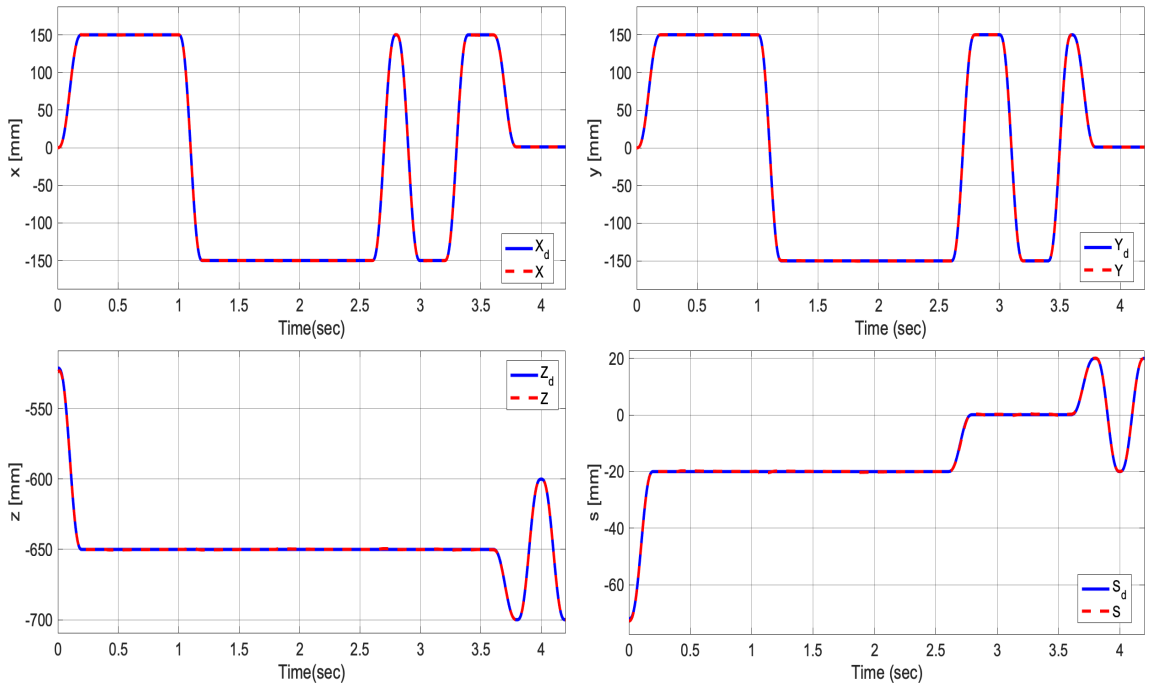


Figure 6.10: Case 2.1 - Evolution of end-effector coordinates

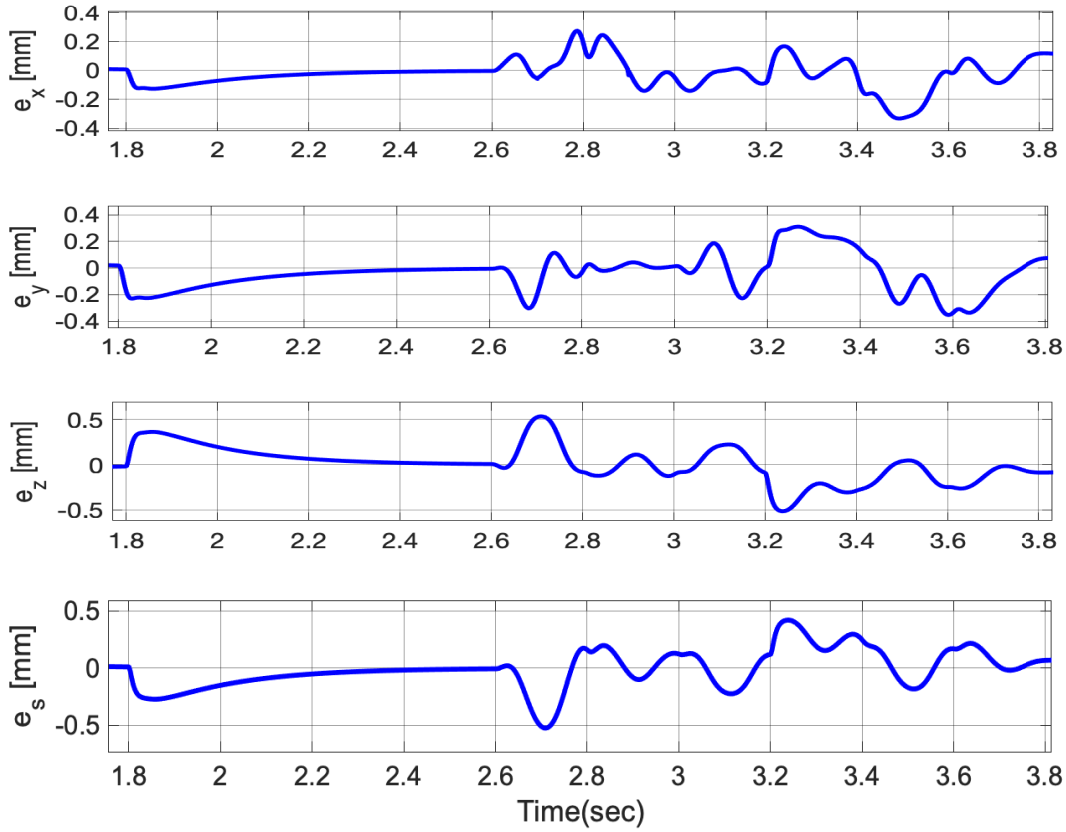


Figure 6.11: Case 2.1 - Evolution of the end-effector coordinates tracking errors

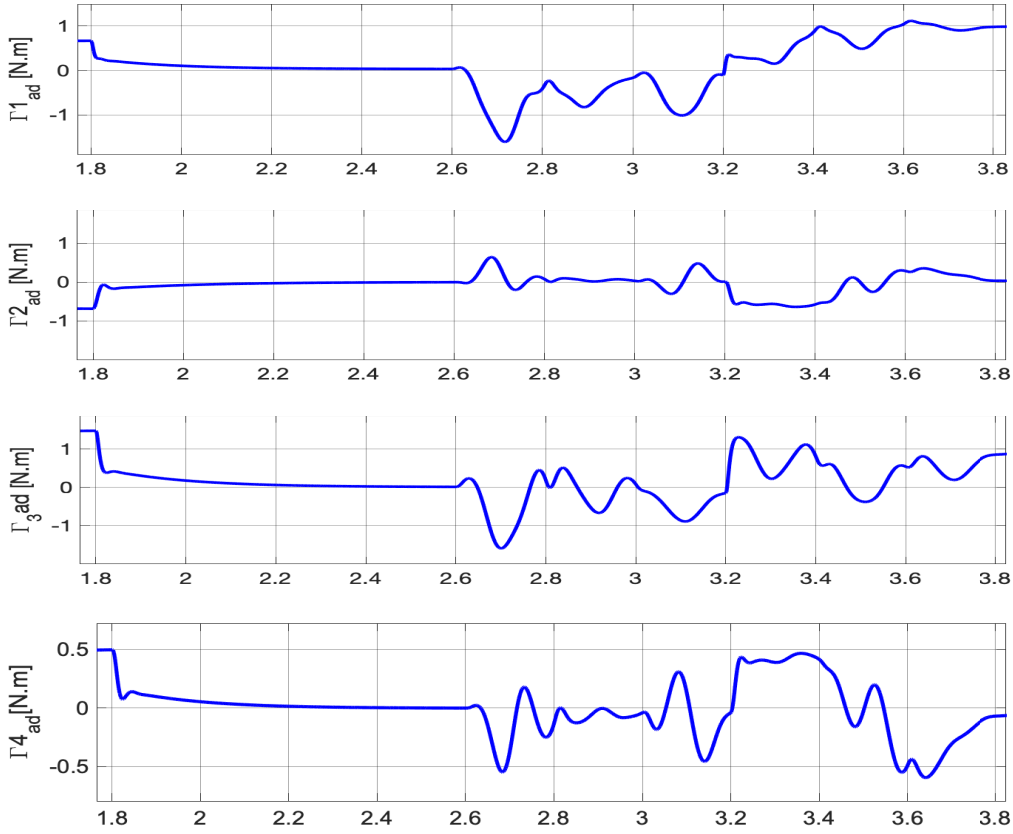


Figure 6.12: Case 2.1 - Evolution of the adaptive control input torques $\Gamma_{AD}(t)$

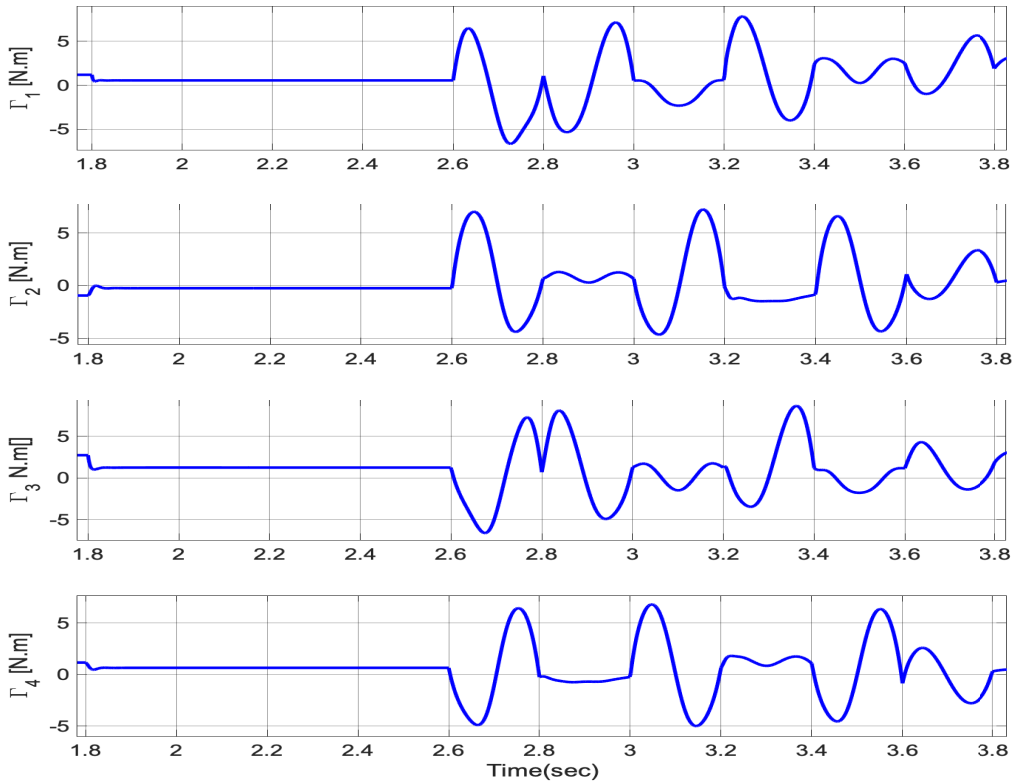


Figure 6.13: Case 2.1 - Evolution of the control inputs torques $\Gamma(t)$

6.3.2 Case 2.2: Fractional order adaptation law

The same fractional coefficient as used in Case 1.2 is kept here.

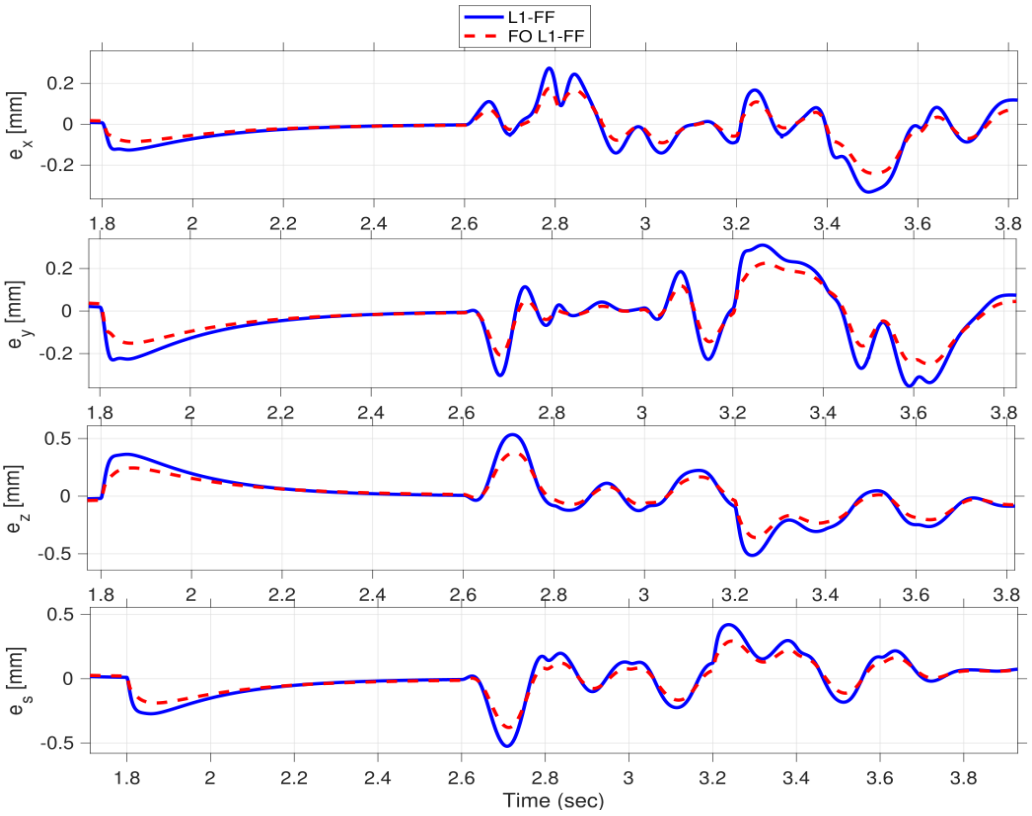


Figure 6.14: Case 2.3 - Evolution of Cartesian tracking errors

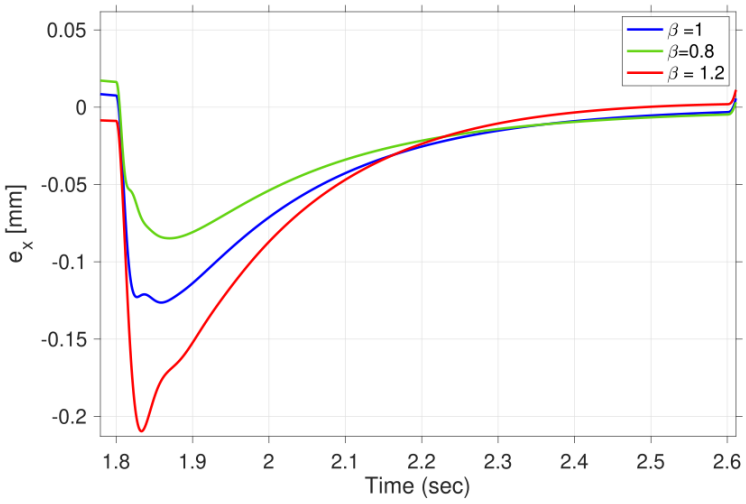


Figure 6.15: Case 2.2 - Disturbance effect on x coordinate for different values of β

6.3.3 Case 2.3: Fractional order filter

The same FO filter used in Case 1.3 was kept for these simulations.

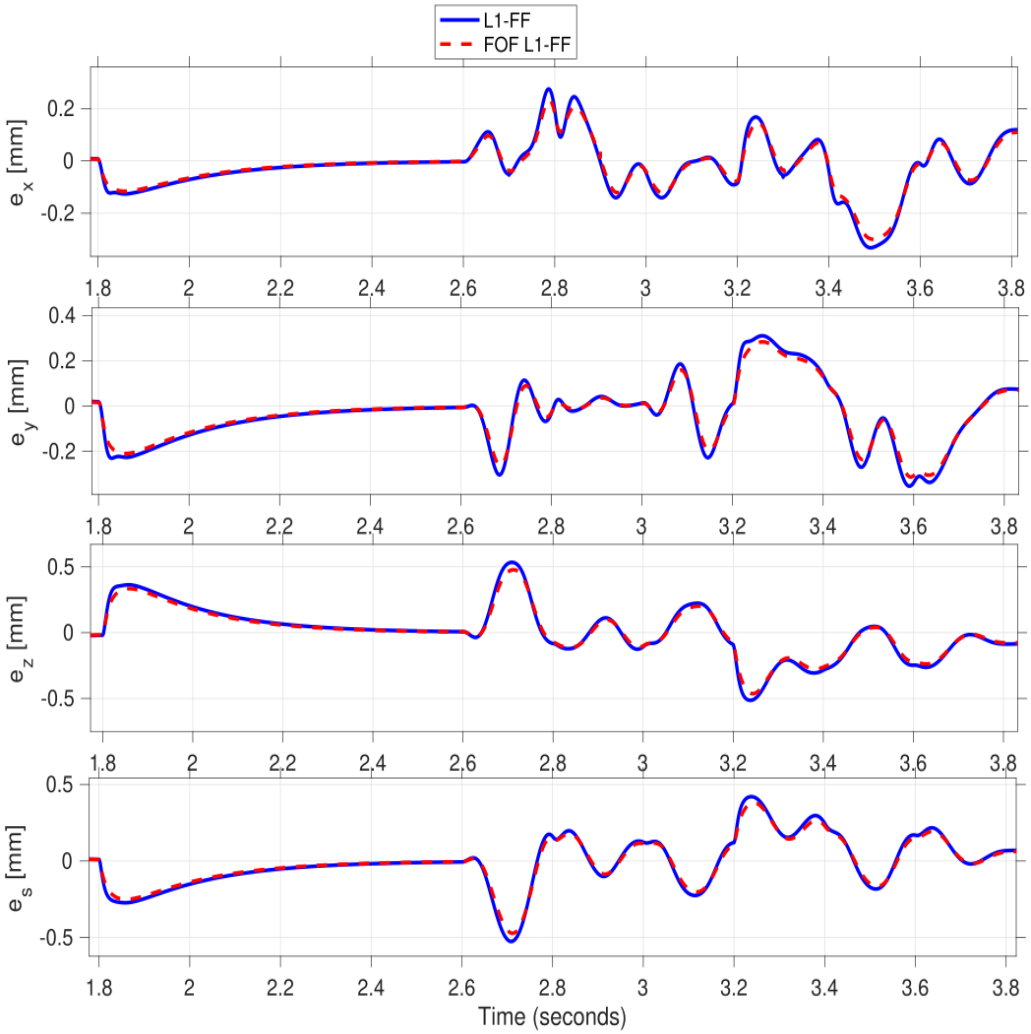


Figure 6.16: Case 2.2 - Evolution of Cartesian tracking errors

Table 6.3: Performance indices for $\mathcal{L}_1$-FF controller in the presence of payload variations

Case	$\mathbf{IAE}_c$	$\mathbf{IAE}_J$	$\mathbf{ITAE}_c$	$\mathbf{ITAE}_J$	$\mathbf{RMSE}_c$	$\mathbf{RMSE}_J$	IUE
IO adaptation	2.5293	0.6260	3.5930	0.8100	0.4938	0.1505	34.6105
FO adaptation	1.9247	0.4944	2.8212	0.6341	0.3759	0.1336	34.5996
FO filter	2.3454	0.5876	3.3170	0.7489	0.4610	0.1456	34.6049

6.4 Results discussion

From the simulation of the payload variation scenario for both IO and FO cases, we can discuss the following observations:

- The effect of the disturbance on the system's performance is very limited. As can be seen from Fig. 6.11, the reference tracking errors deviate from zero by less than 0.3 mm when the robot is in steady state with a 2 kg payload. For the parts where the manipulator is in motion, the maximum values of the errors are slightly higher compared to the nominal case.

- The system's signals remained bounded under parametric variation, as the controller provides bounds for the system signals determined by the $\mathcal{L}_1$ norm. These bounds can be made arbitrarily smaller by increasing the bandwidth of the filter; however, this may result in high-gain feedback, leading to closed-loop systems with overly small robustness margins and increased susceptibility to measurement noise.

- The FO adaptive mechanisms are more effective in attenuating the disturbance effect, achieving faster responses with smaller deviation amplitudes. Compared to the IO case, the Cartesian tracking errors are reduced by approximately 20% according to the indices in Table 6.3. Notably, higher values of the FO coefficient β result in greater deviation but faster convergence to the reference trajectory. This indicates that incorporating FO into the adaptive scheme allows for more flexible fine-tuning of the controller to meet diverse control requirements.

- The FO filter provides the same accuracy and precision, as shown in Fig 6.16, effectively rejecting disturbances even with a smaller bandwidth. A restricted bandwidth is advantageous because controllers with high bandwidths require more powerful hardware with high computational capability and are more sensitive to high-frequency noise and vibrations.

7 Conclusion

In this chapter, we designed and evaluated an FO-$\mathcal{L}_1$ adaptive controller with a model-based feedforward component for the VELOCE manipulator. Our simulations demonstrated that the $\mathcal{L}_1$-FF controller achieved precise tracking in both joint and Cartesian spaces, even with fast displacement times. The decoupled architecture of the $\mathcal{L}_1$ control and the feedforward term effectively canceled most of the nonlinearities, thereby reducing the adaptive controller's effort.

Under payload variations, the $\mathcal{L}_1$-FF controller maintained system performance, exhibiting minimal deviations from the reference tracking errors. The FO adaptive mechanisms proved more effective in disturbance attenuation, offering faster and more precise convergence to reference trajectories. Overall, the $\mathcal{L}_1$-FF controller demonstrated robustness and flexibility, making it suitable for dynamic environments and varied control requirements.

Conclusion and future works

8 General conclusion

In this study, our focus was on developing advanced control methodologies, specifically Fractional Order (FO) adaptive controllers to enhance the performance and robustness of non-linear, time-varying systems, specifically Parallel Kinematic Manipulators (PKMs). The VELOCE robot was selected for applying and testing these controllers. The primary goal of this research was to explore the potential benefits of FO adaptive control for PKMs, aiming to reduce tracking errors, rapidly adapt to external disturbances, and offer greater flexibility in meeting customized control requirements.

When comparing these controllers against traditional approaches like PD-FF control, it was obvious their superior ability to achieve precise tracking with reduced error under varying conditions, including time-varying payloads and dynamic operational environments.

The MRAC-PD controller, while simpler and requiring minimal prior system knowledge, exhibited robust performance and adaptability suitable for applications where extreme precision is not critical but adaptability to changing conditions is essential. It leveraged Fractional Order integration to enhance its adaptation capabilities, leading to improved disturbance rejection and faster convergence to desired trajectories. This adaptability makes it particularly well-suited for PKMs operating in dynamic and unpredictable environments.

In contrast, the FO $\mathcal{L}_1$ adaptive controller with model-based feedforward showed superior performance in terms of precise tracking in both joint and Cartesian spaces, emphasizing its effectiveness in mitigating nonlinearities and disturbances. By decoupling control and compensation for system uncertainties, the $\mathcal{L}_1$-FF controller demonstrated robustness and flexibility, maintaining high performance even under significant payload variations and rapid displacement scenarios. These attributes highlight its potential for applications demanding high precision and reliability, such as those found in aerospace and medical robotics.

Overall, the integration of Fractional Order calculus into adaptive control strategies for PKMs has not only addressed inherent challenges like nonlinearities and uncertainties but also significantly enhanced operational efficiency and reliability. The findings from this study underscore the significant impact of FO adaptive controllers in advancing the capabilities of PKMs across diverse industrial sectors.

9 Future works

As the field of adaptive control for Parallel Kinematic Manipulators (PKMs) continues to advance, there are several promising avenues for future research and development that can build upon the findings and contributions of this thesis. The following areas are w further exploration:

- While simulations provide valuable insights, validating the developed controllers on real PKM prototypes is essential. Real-world implementations can verify the effectiveness of controllers under actual operating conditions, accounting for dynamics and uncertainties not fully captured in simulations.
- Enhancing the performance of $\mathcal{L}_1$ adaptive controllers remains an open question. Exploring and evaluating various optimization techniques from the literature can optimize control parameters to achieve improved tracking accuracy and disturbance rejection capabilities.
- Developing accurate models that capture the complex dynamics of PKMs is critical for robust control design. Data-driven modeling techniques, leveraging artificial intelligence (AI), can improve model accuracy by learning from real-world data, enabling more precise analysis and evaluation of control schemes.
- The theoretical underpinnings of stability for fractional order adaptive controllers require further exploration. Research efforts should focus on deriving conditions that guarantee stability and bounded response, addressing the complexities introduced by fractional calculus in control theory.
- Addressing challenges related to feedforward terms based on nominal system dynamics is crucial. Implementing adaptive feedforward strategies can mitigate uncertainties in model identification, improving disturbance rejection capabilities. Integration of fractional order integration can further enhance adaptive feedforward approaches.
- Investigating the application of Fractional Order (FO) filters can optimize closed-loop responses of PKMs. By balancing robustness and performance, FO filters offer potential improvements in control design, facilitating better trade-offs in dynamic environments.

Bibliography

[1] H. Saied, A. Chemori, M. E. Rafei, C. Francis, and F. Pierrot, "From non-model-based to model-based control of pkms: A comparative study," in *Mechanism, Machine, Robotics and Mechatronics Sciences*, ser. Mechanisms and Machine Science, R. Rizk and M. Awad, Eds. Cham: Springer, 2019, vol. 58, ch. 12.

[2] A. J. Humaidi and H. A. Hussein, "Adaptive control of parallel manipulator in cartesian space," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, 2019, pp. 1–8.

[3] S. Ladaci, F. Azzaza, K. Arbatni, and A. Charef, "Commande auto-ajustable à modèle fractionnaire d'un bras artificiel," in *SNAS'02*, Département d'Electronique, Université Badji Mokhtar, Annaba, Algeria, October 2002.

[4] T. Seghiri, S. Ladaci, and S. Haddad, "Fractional order pi-iλdμ regulator design for high-accuracy position control of an industrial robot," in *The Electrical Engineering International Conference, EEIC'19*, Bejaia, Algeria, December 04-05 2019.

[5] E. A. Mohamed, E. M. Ahmed, A. Elmelegi, M. Aly, O. Elbaksawi, and A.-A. A. Mohamed, "An optimized hybrid fractional order controller for frequency regulation in multi-area power systems," *IEEE Access*, vol. 8, pp. 213 899–213 915, 2020.

[6] T. Seghiri, S. Ladaci, and S. Haddad, "Fractional order adaptive MRAC controller design for high-accuracy position control of an industrial robot arm," *Int. J. Advanced Mechatronic Systems*, vol. 10, no. 1, pp. 8–20, 2023.

[7] B. Khoumeri and S. Ladaci, "Robust fractional-order adaptive control design for a single-link flexible robot arm," in *The 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC'24)*. Setif, Algeria: IEEE, May 12-14 2024, pp. 1–6.

[8] T. Leinonen, "Terminology for the theory of machines and mechanisms," *Mechanism and Machine Theory*, vol. 26, no. 5, 1991.

[9] J. P. Merlet, *Parallel Robots*, ser. Solid Mechanics and Its Applications. Springer Science & Business Media, 2006, vol. 128.

[10] S. Briot and W. Khalil, *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements*, ser. Mechanisms and Machine Science. Springer, 2015.

[11] M. Weck and D. Staimer, "Parallel kinematic machine tools – current state and future potentials," *CIRP Annals*, vol. 51, no. 2, pp. 671–683, 2002.

[12] G. Pritschow, "Parallel kinematic machines (pkm) – limitations and new solutions," *CIRP Annals - Manufacturing Technology*, vol. 49, no. 1, 2000.

[13] J. Craig, *Introduction to Robotics: Mechanics and Control*, ser. Addison-Wesley series in electrical and computer engineering: control engineering. Pearson/Prentice Hall, 2005.

[14] J. E. Gwinnett, "Amusement device," US Patent 1 789 680A, 1931. [Online]. Available: https://patents.google.com/patent/US1789680A

[15] W. L. V. Pollard, "Position controlling apparatus," US Patent 2 286 571A, 1942. [Online]. Available: https://patents.google.com/patent/US2286571A

[16] V. E. Gough and S. G. Whitehall, "Universal tyre test machine," in *Proceedings of 9th International Congress FISITA*, May 1962, pp. 117–137.

[17] D. Stewart, "A platform with six degrees of freedom," *Aircraft Engineering and Aerospace Technology*, vol. 38, no. 4, pp. 30–35, 1966.

[18] R. Clavel, "Device for the movement and positioning of an element in space," US Patent 4,976,582, December 1985. [Online]. Available: https://patents.google.com/patent/US4976582A

[19] S. Reviews, "Parallel robots market research 2031," https://www.linkedin.com/pulse/parallel-robots-market-research-2031-shopping-reviews-fjhvf/, 2023.

[20] M. Russo, D. Zhang, X.-J. Liu, and Z. Xie, "A review of parallel kinematic machine tools: Design, modeling, and applications," *International Journal of Machine Tools and Manufacture*, vol. 196, p. 104118, 03 2024.

[21] R. Zavala-Yoe, R. Ramirez-Mendoza, and J. Ruiz-Garcia, "Mechanical and computational design for control of a 6-pus parallel robot-based laser cutting machine," *Advances in Military Technology*, 07 2015.

[22] A. P. Pashkevich, A. B. Dolgui, and O. A. Chumakov, "Multiobjective optimization of robot motion for laser cutting applications," *International Journal of Computer Integrated Manufacturing*, vol. 17, no. 2, pp. 171–183, 2004.

[23] V. Nabat, M. de la O Rodriguez, O. Company, S. Krut, and F. Pierrot, "Par4: Very high speed parallel robot for pick-and-place," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada, 2005, pp. 553–558.

[24] Y. Li and Q. Xu, "Design and development of a medical parallel robot for cardiopulmonary resuscitation," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 265–273, 2007.

[25] K. K, S. A. S. Srinath. A, M. Harish, P. Vijay, and K. Bhaskar, "Simulation of four arm parallel manipulator for medical applications," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 9, pp. 1802–1809, 2017.

[26] J.-A. Leal-Naranjo, M. Ceccarelli, C.-R. Torres-San-Miguel, L.-A. Aguilar-Perez, G. Urriolagoitia-Sosa, and G. Urriolagoitia-Calderón, "Multi-objective optimization of a parallel manipulator for the design of a prosthetic arm using genetic algorithms," *Latin American Journal of Solids and Structures*, vol. 15, no. 3, p. 26, 2018.

[27] S. He, Z. Xu, X. Wang, A. Li, and Q. Huo, "Design and testing of a parallel manipulator for space micro-vibration simulation," in *Towards Autonomous Robotic Systems. TAROS 2017*, ser. Lecture Notes in Computer Science, Y. Gao, S. Fallah, Y. Jin, and C. Lekakou, Eds. Springer, 2017, vol. 10454.

[28] L. A. M.-C. Héctor Fabio Quintero-Riaza and M. Díaz-Rodríguez, "Synthesis of planar parallel manipulators including dexterity, force transmission and stiffness index," *Mechanics Based Design of Structures and Machines*, vol. 47, no. 6, pp. 680–702, 2019.

[29] R. A. Srivatsan, S. Bandyopadhyay, and A. Ghosal, "Analysis of the degrees-of-freedom of spatial parallel manipulators in regular and singular configurations," *Mechanism and Machine Theory*, vol. 69, pp. 127–141, 2013.

[30] I. A. Bonev, D. Chablat, and P. Wenger, "Working and assembly modes of the agile eye," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2317–2322.

[31] F. Pierrot and O. Company, "H4: A new family of 4-dof parallel robots," in *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399)*, Atlanta, GA, USA, 1999, pp. 508–513.

[32] S. Krut, M. Benoit, H. Ota, and F. Pierrot, "I4: A new parallel mechanism for scara motions," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, Taipei, Taiwan, 2003, pp. 1875–1880.

[33] S. Krut, O. Company, V. Nabat, and F. Pierrot, "Heli4: A parallel robot for scara motions with a very compact traveling plate and a symmetrical design," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 1656–1661.

[34] G. Sartori Natal, A. Chemori, and F. Pierrot, "Dual-space control of extremely fast parallel manipulators: Payload changes and the 100g experiment," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1520–1535, July 2015.

[35] H. Saied, "On control of parallel robots for high dynamic performances: From design to experiments," Ph.D. dissertation, Université Montpellier; Université Libanaise, Nov 2019.

[36] M. I. Hosseini, S. A. Khalil, and H. D. Taghirad, "Practical robust nonlinear pd controller for cable-driven parallel manipulators," *Nonlinear Dynamics*, vol. 106, pp. 405–424, 2021.

[37] W. Shang, S. Cong, Z. Li, and S. Jiang, "Augmented nonlinear pd controller for a redundantly actuated parallel manipulator," *Advanced Robotics*, vol. 23, pp. 1725–1742, Sep 2009.

[38] H. Saied, A. Chemori, M. E. Rafei, C. Francis, and F. Pierret, "Actuator and friction dynamics formulation in control of pkms: From design to real-time experiments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5634–5639.

[39] A. Dumlu and K. Erenturk, "Trajectory tracking control for a 3-dof parallel manipulator using fractional-order (pid mu)-d-lambda control," *Industrial Electronics, IEEE Transactions on*, vol. 61, pp. 3417–3426, 07 2014.

[40] Z. Lv, D. Ning, P. Liu, and P. Liao, "Pd control with gravity compensation for 3-rps parallel manipulator," in *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, 05 2023, pp. 1744–1749.

[41] M. Jafarinasab, M. Keshmiri, H. Azizan, and M. Danesh, "Sliding mode control of a novel 6-dof parallel manipulator with rotary actuators," in *2011 16th International Conference on Methods & Models in Automation & Robotics*, 2011, pp. 218–223.

[42] Y. Ye, Z. Yue, and B. Gu, "Adrc control of a 6-dof parallel manipulator for telescope secondary mirror," *Journal of Instrumentation*, vol. 12, pp. T03 006 – T03 006, 2017.

[43] K. V. Sancak and Z. Y. Bayraktaroglu, "Nonlinear computed torque control of 6-dof parallel manipulators," *International Journal of Control, Automation and Systems*, vol. 20, pp. 2297–2311, 2022.

[44] C. C. Nguyen, S. S. Antrazi, Z.-L. Zhou, and C. E. Campbell, "Adaptive control of a stewart platform-based manipulator," *J. Field Robotics*, vol. 10, pp. 657–687, 1993.

[45] M. Bennehar, A. Chemori, F. Pierrot, and V. Creuze, "Extended model-based feedforward compensation in l1 adaptive control for mechanical manipulators: Design and experiments," *Frontiers in robotics and AI*, vol. 2, pp. 1–11, 12 2015.

[46] W.-W. Shang, S. Cong, and Y. Ge, "Adaptive computed torque control for a parallel manipulator with redundant actuation," *Robotica*, vol. 30, no. 3, pp. 457–466, 2012.

[47] M. Bennehar, A. Chemori, and F. Pierrot, "A new revised desired compensation adaptive control for enhanced tracking: application to ra-pkms," *Advanced Robotics*, pp. 1–16, 07 2016.

[48] M. Bennehar, G. El-Ghazaly, A. Chemori, and F. Pierrot, "A novel adaptive terminal sliding mode control for parallel manipulators: Design and real-time experiments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 6086–6092.

[49] J. Escorcia, A. Chemori, and H. Aguilar-Sierra, "Adaptive rise feedback control for robotized machining with pkms: Design and real-time experiments," *Control Systems Technology, IEEE Transactions on*, pp. 1–16, 05 2022.

[50] G. S. Natal, A. Chemori, and F. Pierrot, "Nonlinear control of parallel manipulators for very high accelerations without velocity measurement: stability analysis and experiments on Par2 parallel manipulator," *Robotica*, vol. 34, no. 01, pp. 43–70, Jan. 2016.

[51] F. Tajdari, "Adaptive time-delay estimation and control of optimized stewart robot," *Journal of Vibration and Control*, vol. 29, no. 23-24, pp. 5511–5531, 2023.

[52] E. Dombre and W. Khalil, *Robot Manipulators: Modeling, Performance Analysis and Control*, ser. ISTE. Wiley, 2013.

[53] M. Bennehar, A. Chemori, and F. Pierrot, "$\mathcal{L}_1$ adaptive control of parallel kinematic manipulators: Design and real-time experiments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1587–1592.

[54] Y. Fitas, A. Chemori, J. Lamaury, and F. Pierrot, "A new time-varying adaptive feedforward sliding mode control of pkms," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 621–626, 2022, 14th IFAC Workshop on Adaptive and Learning Control Systems ALCOS 2022.

[55] R. Kouki, A. Chemori, and F. Bouani, "A new fast nmpc scheme for parallel kinematic manipulators: Design and real-time experiments," in *2019 International Conference on Signal, Control and Communication (SCC)*, vol. 130, 2019, pp. 69–75.

[56] A. Codourey, "Dynamic modeling of parallel robots for computed-torque control implementation," *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1325–1336, 1998.

[57] D. Corbel, M. Gouttefarde, and O. Company, "Towards 100g with pkm. is actuation redundancy a good solution for pick and place?" in *Proc. IEEE International Conference on Robotics and Automation (ICRA'10)*, Anchorage, Alaska, May 2010, pp. 4675–4682.

[58] V. Nabat, "Robots parallèles à nacelle articulée, du concept à la solution industrielle pour le pick-andplace," Ph.D. dissertation, Université Montpellier II - Sciences et Techniques du Languedoc, Mar. 2007.

[59] L.-W. Tsai, *Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators*. New York; NY; USA: John Wiley& Sons, Inc., 1999.

[60] K. J. Astrom and B. Wittenmark, *Adaptive Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[61] D. Zhang and B. Wei, *Adaptive Control for Robotic Manipulators*. CRC Press, 2017.

[62] H. Whitaker, J. Yamron, A. Kezer, and M. I. of Technology. Instrumentation Laboratory, *Design of Model Reference Adaptive Control Systems for Aircraft*, ser. Report Massachusetts Institute of Technology Instrumentation Laboratory R. M.I.T. Instrumentation Laboratory, 1958.

[63] P. Parks, "Liapunov redesign of model reference adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 362–367, 1966.

[64] B. Egardt, Ed., *Stability of Adaptive Controllers*, 1st ed., ser. Lecture Notes in Control and Information Sciences. Springer Berlin, Heidelberg, 1979, vol. 20.

[65] G. Goodwin and K. Sin, *Adaptive Filtering Prediction and Control*, ser. Dover Books on Electrical Engineering. Dover Publications, 2009.

[66] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*, ser. Dover Books on Electrical Engineering Series. Dover Publications, 2011.

[67] N. T. Nguyen, *Model-Reference Adaptive Control*, 1st ed., ser. Advanced Textbooks in Control and Signal Processing. Springer Cham, 2018.

[68] J. Slotine and W. Li, *Applied Nonlinear Control*, ser. Prentice-Hall International Editions. Prentice-Hall, 1991.

[69] A. Shekhar and A. Sharma, "Review of model reference adaptive control," in *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)*, 2018, pp. 1–5.

[70] P. Ioannou and J. Sun, *Robust Adaptive Control*, ser. Dover Books on Electrical Engineering Series. Dover Publications, Incorporated, 2012.

[71] P. Ioannou and P. Kokotović, "Robust redesign of adaptive control," *Automatic Control, IEEE Transactions on*, vol. 29, pp. 202 – 211, 04 1984.

[72] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *1986 American Control Conference*, pp. 1067–1072, 1986.

[73] C. E. Rohrs, L. Valavani, M. Athans, and G. Stein, "Robustness of adaptive control algorithms in the presence of unmodeled dynamics," in *1982 21st IEEE Conference on Decision and Control*, 1982, pp. 3–11.

[74] C. Cao and N. Hovakimyan., "Design and analysis of a novel l1 adaptive controller, part i: Control signal and asymptotic stability," in *Proceedings of the American Control Conference*, 2006, pp. 3397–3402.

[75] C. Cao and N. Hovakimyan, "Stability margins of l1 adaptive control architecture," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 480–487, 2010.

[76] N. Hovakimyan and C. Cao, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, ser. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2010.

[77] A. Tepljakov, *Fractional-order Modeling and Control of Dynamic Systems*, 1st ed., ser. Springer Theses. Springer Cham, 2017.

[78] A. Charef, H. Sun, Y. Tsao, and B. Onaral, "Fractal system as represented by singularity function," *IEEE Transactions on Automatic Control*, vol. 37, no. 9, pp. 1465–1470, 1992.

[79] K. Oldham and J. Spanier, *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order*, ser. ISSN. Elsevier Science, 1974.

[80] I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, 1st ed. San Diego, CA: Academic Press, Oct 1998, vol. 198.

[81] J. T. M. Ricardo Enrique Gutiérrez, João Maurício Rosário, "Fractional order calculus: Basic concepts and engineering applications," *Mathematical Problems in Engineering*, 2010.

[82] C. Monje, Y. Chen, B. Vinagre, D. Xue, and V. Feliu, *Fractional Order Systems and Control - Fundamentals and Applications*, 01 2010.

[83] H. Sun, Y. Zhang, D. Baleanu, W. Chen, and Y. Q. Chen, "A new collection of real world applications of fractional calculus in science and engineering," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 64, pp. 213–231, 2018.

[84] A. Oustaloup, J. Sabatier, and P. Lanusse, "From fractal robustness to the crone control," *Fractional Calculus and Applied Analysis*, vol. 2, no. 1, pp. 1–30, 1999.

[85] I. Podlubny, "Fractional-order systems and fractional-order controllers," *Institute of Experimental Physics, Slovak Academy of Sciences, Kosice*, vol. 12, no. 3, pp. 1–18, 1994.

[86] H.-F. Raynaud and A. Mokraoui, "State-space representation for fractional order controllers," *Automatica*, vol. 36, 07 2000.

[87] A. Oustaloup, F. Levron, B. Mathieu, and F. Nanot, "Frequency-band complex noninteger differentiator: characterization and synthesis," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 1, pp. 25–39, 2000.

[88] S. Emad, A. M. Hassanein, A. M. AbdelAty, A. H. Madian, A. G. Radwan, and L. A. Said, "A study on fractional power-law applications and approximations," *Electronics*, vol. 13, no. 3, 2024.

[89] B. Wei, "Adaptive control design and stability analysis of robotic manipulators," *Actuators*, vol. 7, no. 4, p. 89, 2018.

[90] M. S. Tran and et al., "Model reference adaptive control strategy for application to robot manipulators," in *AETA 2018 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, ser. Lecture Notes in Electrical Engineering. Springer, Cham, 2020, vol. 554, pp. 559–566.

[91] C. Pezzato, R. Ferrari, and C. H. Corbato, "A novel adaptive controller for robot manipulators based on active inference," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2973–2980, 2020.

[92] R. Sengupta, U. M. Nath, and C. Dey, "Design and performance analysis of a modified mrac for second-order processes," in *2017 4th International Conference on Power, Control & Embedded Systems (ICPCES)*, 2017, pp. 1–5.

[93] S. Manna and A. K. Akella, "Designing of an updated mrac with pd feedback for marginally stable second-order process with dead time," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, 2020, pp. 63–66.

[94] S. Ladaci, "Adaptive fractional piλ controller," 11 2010.

[95] Y. fia, S. Ladaci, K. Khettab, and A. Chemori, "Fractional order model reference adaptive control for scara robot trajectory tracking," *International Journal of Industrial and Systems Engineering*, vol. 30, 01 2017.

[96] S. Ladaci and A. Charef, "Mit adaptive rule with fractional integration," 07 2003.

[97] S. Coman, C. Boldisor, and L. Floroian, "Fractional adaptive control for a fractional - order insuline - glucose dynamic model," in *2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017*, 2017, pp. 887–892.

[98] T. Seghiri, S. Ladaci, and H. Salim, "Fractional order adaptive mrac controller design for high-accuracy position control of an industrial robot arm," *International Journal of Advanced Mechatronic Systems*, vol. 10, pp. 8–20, 01 2023.

[99] K.-D. Nguyen, H. Dankowicz, and N. Hovakimyan, "Marginal stability in 1-adaptive control of manipulators," vol. 7, 08 2013.

[100] A. Lakshmanan, A. Gahlawat, and N. Hovakimyan, "Safe feedback motion planning: A contraction theory and $\mathcal{L}_1$-adaptive control based approach," 04 2020.

[101] Y. Cheng, P. Zhao, M. Gandhi, B. Li, E. Theodorou, and N. Hovakimyan, "Robustifying reinforcement learning policies with l1 adaptive control," 06 2021.

[102] R. Zhu, M. Xie, Z. Guo, W. Gai, and G. Tang, "Nonlinear l1 adaptive control with feedforward control action and its application in wind tunnel," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 386–391.

[103] K. K. K. Kim, E. Kharisov, and N. Hovakimyan, "Filter design for l1 adaptive output-feedback controller," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 5653–5658.

[104] B. Maalej, R. Jallouli, C. Mhiri, M. Elleuch, and N. Derbel, "L 1 adaptive fractional control optimized by genetic algorithms with application to polyarticulated robotic systems," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–14, 04 2021.

[105] E.-G. Talbi, *Metaheuristics: From Design to Implementation.* Hoboken: John Wiley & Sons, Ltd, 2009, vol. 74.

[106] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[107] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 12, pp. 2531–2561, 2022.

Appendices

1 Inverse Kinematic model of VELOCE robot

To develop the inverse kinematic model the following frames will be used, the fixed-base frame $O-xyz$ and the four frames attached to each actuator $A_i - u_i v_i z$, with the points A_i representing the actuated joints locations whose coordinates in the fixed-base frame are given by Equation (1):

$$A_i = R \begin{bmatrix} cos(\beta_i) & sin(\beta_i) & 0 \end{bmatrix}^T \tag{1}$$

with R representing the radius of the fixed platform of the robot.
The vectors u_i and v_i composing the $A_i - u_i v_i z$ frame are expressed as follow:

$$u_i = \begin{bmatrix} cos(\beta_i) & sin(\beta_i) & 0 \end{bmatrix}^T \tag{2}$$

$$v_i = \begin{bmatrix} -sin(\beta_i) & cos(\beta_i) & 0 \end{bmatrix}^T \tag{3}$$

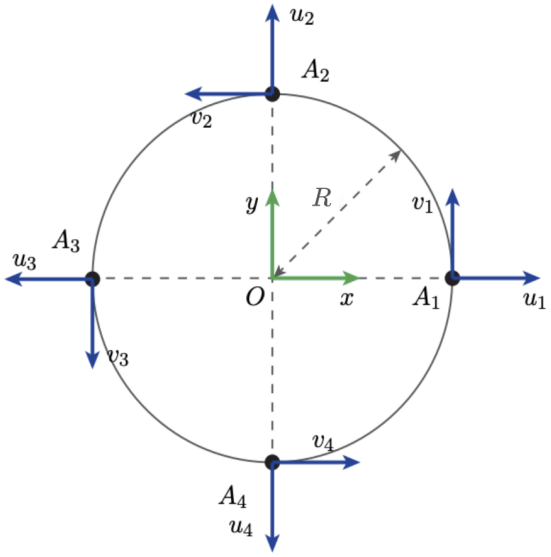


Figure 0.17: Top view of VELOCE robot with corresponding frames

The points B_i are attached to the passive joints linking the rear-arms and the forearms of the manipulator (Fig 0.18), to derive their expressions in the base frame, we define the rotation matrix between the base frame and the actuators frames:

$$R_O^{A_i} = \begin{bmatrix} cos(\beta_i) & -sin(\beta_i) & 0 \\ sin(\beta_i) & cos(\beta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

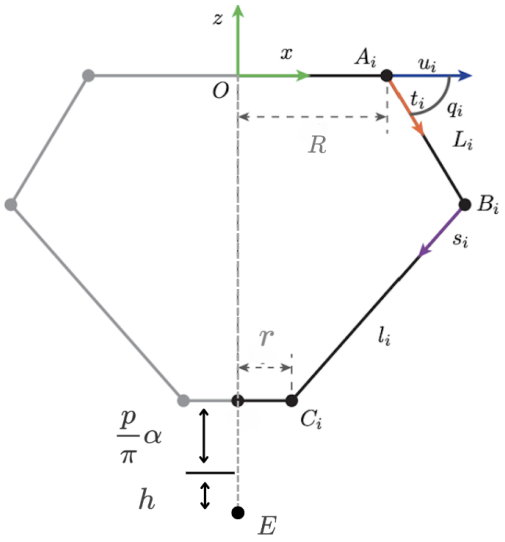


Figure 0.18: Side view of VELOCE robot

From Fig 0.18 we can derive the coordinates of the points B_i in the $A_i - u_i v_i z$ frame:

$$B_i = \begin{bmatrix} R + Lcos(q_i) & 0 & Lsin(q_i) \end{bmatrix}^T \tag{5}$$

where L is the length of the rear-arms.

Using the rotation matrix R_O^A obtained above, we get the coordinates of points B_i in the $O - xyz$ frame:

$$B_{i_{/O}} = R_O^{A_i} \, B_{i_{/A_i}} = \begin{bmatrix} cos(\beta_i) & -sin(\beta_i) & 0 \\ sin(\beta_i) & cos(\beta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R + L\,cos(q_i) \\ 0 \\ L\,sin(q_i) \end{bmatrix} \tag{6}$$

$$B_{i_{/O}} = A_i + L \begin{bmatrix} cos(\beta_i)cos(q_i) & cos(\beta_i)sin(q_i) & sin(q_i) \end{bmatrix}^T \tag{7}$$

Finally, we geometrically derive the coordinates of points C_i which locates the joints between a given forearm and either the lower or upper part of the traveling-plate using Fig 0.18:

$$C_i = E + r\,u_i + (h + \frac{p_i}{\pi}\alpha)\,e_z \tag{8}$$

with $E = \begin{bmatrix} x & y & z \end{bmatrix}^T$ the position of the end-effector in the base-frame and h is the distance between the lower part of the moving platform and the end-effector.
We end up with the final coordinates of the points C_i:

$$C_i = \begin{bmatrix} x + r\,cos(\beta_i) & y + r\,sin(\beta_i) & z + h + \frac{p_i}{\pi}\alpha \end{bmatrix}^T \tag{9}$$

with p_i defined as follow:

$$p_i = \begin{cases} 0 & i = 1, 3 \\ p & i = 2, 4 \end{cases}$$

According to [56], finding the inverse kinematics equations of the VELOCE robot involves solving for the intersection of a circle and a sphere, which represents the coordinates of the passive joints B_i in the actuators' frame. In other words, this requires considering the geometric constraints of the closed-loop system composed of the kinematic chains and the traveling plate. For the *i-th* kinematic chain, we can write the geometric constraint related to the constant length of the forearm as follows:

$$||B_iC_i||^2 - l^2 = 0 \quad \forall i = 1, \cdots, 4 \tag{10}$$

This constraint results in an equation of a sphere with its center at C_i and l, the length of the forearms, as its radius:

$$(x_{B_i} - x_{C_i})^2 + y_{C_i}^2 + (z_{B_i} - z_{C_i})^2 = l^2 \tag{11}$$

From the motion of one rear arm, we can obtain the equation of a circle with A_i as its center and L as its radius:

$$x_{B_i}^2 + z_{B_i}^2 = L^2 \tag{12}$$

To solve for the intersection of the sphere and the circle, we subtract Equation (12) from Equation (11) to obtain the following expression:

$$2x_{C_i}x_{B_i} + 2z_{C_i}z_{B_i} = l^2 - L^2 + x_{C_i}^2 + y_{C_i}^2 + z_{C_i}^2 \tag{13}$$

Solving Equation (13) for z_{B_i}, we find:

$$z_{B_i} = \frac{S_i - 2x_{C_i}x_{B_i}}{2z_{C_i}} \tag{14}$$

with $S_i \triangleq l^2 - L^2 + x_{C_i}^2 + y_{C_i}^2 + z_{C_i}^2$. Substituting Equation (14) into Equation (12) yields:

$$4\left(z_{C_i}^2 + x_{C_i}^2\right)x_{B_i}^2 - 4S_ix_{C_i}x_{B_i} + \left(S_i^2 - 4z_{C_i}^2L^2\right) = 0 \tag{15}$$

Solving for x_{B_i} gives:

$$x_{B_i} = \frac{S_ix_{C_i} \pm \sqrt{\left(S_ix_{C_i}\right)^2 - \left(z_{C_i}^2 + x_{C_i}^2\right)\left(S_i^2 - 4z_{C_i}^2L^2\right)}}{2\left(z_{C_i}^2 + x_{C_i}^2\right)} \tag{16}$$

From Fig 2.5, it is evident that the actuated joint angles can be derived geometrically from the coordinates of the points B_i. Once these coordinates are determined, we can calculate q_i as follows:

$$q_i = \text{atan2}\left(z_{B_i}, x_{B_i}\right) \tag{17}$$

The function atan2 is preferred in order to avoid divisions by zero and to obtain more precise values.

2 $\mathcal{L}_1$-norm of a system

he $\mathcal{L}_1$-norm of a system sets the relation between the peak values of the system's input and output. The $\mathcal{L}_1$-norm is also called the peak-to-peak gain of a system.

Let $G(s)$ be a proper and exponentially stable system. Assume zero initial conditions. Then, for the bounded input $u(t)$, its output $y(t)$ can be written as

$$y(t) = g(t) * u(t) = \int_0^t g(t-\tau)u(\tau)d\tau,$$

where " $*$ " denotes the convolution operator, and $g(t)$ is the impulse response of $G(s)$. Letting

$$\|y\|_{\mathcal{L}_\infty} \triangleq \sup_{t\geq 0} |y(t)|,$$

we obtain the bound:

$$\begin{aligned} |y(t)| &= \left| \int_0^t g(\tau)u(t-\tau)d\tau \right| \\ &\leq \int_0^t |g(\tau)\|u(t-\tau)|d\tau \\ &\leq \int_0^\infty |g(\tau)|d\tau \|u\|_{\mathcal{L}_\infty}. \end{aligned}$$

The $\mathcal{L}_1$-norm of $G(s)$ is defined as:

$$\|G(s)\|_{\mathcal{L}_1} \triangleq \int_0^\infty |g(\tau)|d\tau \tag{18}$$

3 $\mathcal{L}_1$ adaptive controller: stability proof of prediction error

To prove the boundedness of the prediction error, the following Lyapunov function is considered:

$$V(\tilde{x}(t), \tilde{\theta}(t)) = \tilde{x}^\top(t)P\tilde{x}(t) + \frac{1}{\gamma}\tilde{\theta}^\top(t)\tilde{\theta}(t)$$

Lemma 3 :
The prediction error in of the sate predictor is uniformly bounded:

$$\|\tilde{x}\|_{L_\infty} \leq \sqrt{\frac{\theta_{\max}}{\lambda_{\min}(P)\gamma}}, \quad \theta_{\max} \triangleq 4 \max_{\theta\in\Theta} \|\theta\|^2,$$

where γ is the adaptation rate and $\lambda_{\min}(P)$ is the minimum eigenvalue of P.
Proof:
Using properties of the projection operator, we can upper bound the derivative of the

Lyapunov function along the trajectories of the system as

$$\begin{aligned}\dot{V}(t) &= \dot{\tilde{x}}^\top(t)P\tilde{x}(t) + \tilde{x}^\top(t)P\dot{\tilde{x}}(t) + \frac{1}{\gamma}\left(\dot{\tilde{\theta}}^\top(t)\tilde{\theta}(t) + \tilde{\theta}^\top(t)\dot{\tilde{\theta}}(t)\right)\\ &= \tilde{x}^\top(t)\left(A_m^\top P + PA_m\right)\tilde{x}(t) + 2\tilde{x}^\top(t)Pb\tilde{\theta}^\top(t)x(t) + \frac{2}{\gamma}\tilde{\theta}^\top(t)\dot{\hat{\theta}}(t)\\ &= -\tilde{x}^\top(t)Q\tilde{x}(t) + 2\tilde{x}^\top(t)Pb\tilde{\theta}^\top(t)x(t) + 2\tilde{\theta}^\top(t)\,\mathrm{Proj}\left(\hat{\theta}(t), -x(t)\tilde{x}^\top(t)Pb\right)\\ &= -\tilde{x}^\top(t)Q\tilde{x}(t) + 2\tilde{\theta}^\top(t)\left(x(t)\tilde{x}^\top(t)Pb + \mathrm{Proj}\left(\hat{\theta}(t), -x(t)\tilde{x}^\top(t)Pb\right)\right)\\ &\leq -\tilde{x}^\top(t)Q\tilde{x}(t)\end{aligned}$$

which implies that $\tilde{x}(t)$ and $\tilde{\theta}(t)$ are uniformly bounded. Next, since $\tilde{x}(0) = 0$, it follows that

$$\lambda_{\min}(P)\|\tilde{x}(t)\|^2 \leq V(t) \leq V(0) = \frac{\tilde{\theta}^\top(0)\tilde{\theta}(0)}{\gamma}.$$

The projection operator ensures that $\hat{\theta}(t) \in \Theta$, and therefore

$$\frac{\tilde{\theta}^\top(0)\tilde{\theta}(0)}{\gamma} \leq \frac{4\max_{\theta\in\Theta}\|\theta\|^2}{\gamma},$$

which leads to the following upper bound:

$$\|\tilde{x}(t)\|^2 \leq \frac{\theta_{\max}}{\lambda_{\min}(P)\gamma}$$

Since $\|\cdot\|_\infty \leq \|\cdot\|$, and this bound is uniform, the bound above yields

$$\|\tilde{x}_\tau\|_{\mathscr{L}_\infty} \leq \sqrt{\frac{\theta_{\max}}{\lambda_{\min}(P)\gamma}}$$

4 Projection operator

A convex compact set with a smooth boundary given by

$$\Omega_c = \{\boldsymbol{\theta} \in R^n \mid f(\boldsymbol{\theta}) \leq c\}, \quad 0 \leq c \leq 1,$$

where $f : R^n \to R$ is the following smooth convex function:

$$f(\theta) = \frac{\theta^{\mathrm{T}}\theta - \theta_{\max}^2}{\varepsilon_\theta \theta_{\max}^2} \tag{19}$$

where $\boldsymbol{\theta}_{\max}$ is the norm bound imposed on the vector $\boldsymbol{\theta}$ and $0 < \varepsilon_{\boldsymbol{\theta}} < 1$ stands for the projection tolerance bound of our choice. For any given $\mathbf{y} \in R^n$, the projection operator is defined as

$$\mathrm{Proj}(\theta, \mathbf{y}) = \begin{cases} \mathbf{y}, & \text{if } f(\theta) < 0\\ \mathbf{y}, & \text{if } f(\theta) \geq 0, \nabla f^{\mathrm{T}}\mathbf{y} \leq 0\\ \mathbf{y} - \mathbf{g}(f, \mathbf{y}), & \text{if } f(\theta) \geq 0, \nabla f^{\mathrm{T}}\mathbf{y} > 0, \end{cases}$$

where $\nabla f(\boldsymbol{\theta})$ is the gradient vector of $f(\cdot)$ evaluated at $\boldsymbol{\theta}$ and

$$\mathbf{g}(f, \mathbf{y}) = \frac{\nabla f \nabla f^{\mathrm{T}}\mathbf{y}}{\|\nabla f\|^2} f(\theta)$$

This operator only modifies y when θ takes values outside the allowed range.

5 Background on Particle Swarm Optimisation (PSO) algorithm

Swarm Intelligence (SI) techniques are inspired by the simple behaviors and self-organizing interactions of agents in nature, where the components of an initially disordered system interact locally to produce coordination or global order. Examples include phenomena such as fish schooling, honey bee activity, bacterial growth, animal herding, bird flocking, and ant colony foraging. Common SI algorithms include Ant Colony Optimization and Particle Swarm Optimization (PSO) [105].
PSO, a swarm-based stochastic algorithm, was originally proposed by Kennedy and Eberhart [106]. In this algorithm, each potential solution is represented as a particle with a certain position and velocity, moving through the problem space similarly to a flock of birds. One of the main advantages of PSO is having few tuning parameters. However, even though PSO can find the best solution through the interaction of particles, it converges relatively slowly towards the global optimum in high-dimensional search spaces [107] and thus it sometimes fail to find the optimal solution.

5.1 Particles neighborhood

In a basic swarm, consisting of N particles in a D- dimensional search space, each particle i represents a candidate solution to the problem and is referred to by the vector x_i containing its own position and velocity which indicates its moving direction. Each particle successively adjusts its position x_i toward the global optimum based on two factors: the best position visited by itself (p_{best_i}) and the best position visited by the entire swarm (g_{best}).

A neighborhood must be defined, following different principles, for each particle that denotes and limits the social influence between the particles. The PSO algorithm presented in [105] and used in this work is based on the g_{best} method, which considers the neighborhood as encompassing the entire population of particles.

5.2 Fitness score

Every member of the population is allocated a fitness score, which is determined based on its performance and assessed against a defined objective function. This fitness score is utilized to rank the candidate solutions and to direct the particles' movement towards the optimal global position g_{best}. The fitness score can be calculated using various methods, such as the Euclidean norm, F-measure, or SVM fitness score.

5.3 Update laws

At each iteration the value associated to every particle, including the velocity and position are updated along the best position found so far. The update is performed in a parallel manner i.e; all particles are updated simultaneously. A pseudo code of the steps describe below is presented in Fig 0.19.

Velocity update: The velocity determines the rate and direction of change for every particle and in which direction it will be moving, its update rule is expressed in Equ 20.

$$v_i(t) = \omega v_i(t-1) + \rho_1 C_1(p_{best_i} - x_i) + \rho_2 C_2(g_{best} - x_i(t-1)) \tag{20}$$

where:
ω represents the inertia coefficient.
ρ_1 and ρ_2 are two random numbers ranging from 0 to 1.
C_1 is the cognitive learning factor that represents the attraction that a particle has toward its own success.
C_2 is the social learning factor that represents the attraction that a particle has toward the success of its neighbors.

Position update: The next step is to update the position of the particles based on the new calculated velocity.

$$x_i(t) = x_i(t-1) + v_i(t) \tag{21}$$

Best local solution update: The final step involves comparing the solutions found so far for each particle based on their fitness scores and then updating the best local solution. Additionally, the global position of the entire swarm is updated.

Algorithm Template of the particle swarm optimization algorithm.

Random initialization of the whole swarm ;
Repeat
  Evaluate $f(x_i)$;
  For all particles i
    Update velocities:
      $v_i(t) = v_i(t-1) + \rho_1 \times (p_i - x_i(t-1)) + \rho_2 \times (p_g - x_i(t-1))$;
    Move to the new position: $x_i(t) = x_i(t-1) + v_i(t)$;
    If $f(x_i) < f(pbest_i)$ **Then** $pbest_i = x_i$;
    If $f(x_i) < f(gbest)$ **Then** $gbest = x_i$;
    Update(x_i, v_i) ;
  EndFor
Until Stopping criteria

Figure 0.19: Pseud-code of the PSO algorithm

6 Trajectory generation

The reference trajectory has 22 points linked by 5^{th} order polynomials, and a given T time period between each displacement.

Point	x[mm]	y[mm]	z[mm]	s[mm]
X_0	0	0	0	0
X_1	150	150	-650	-20
X_2	-150	150	-650	-20
X_3	-150	-150	-650	-20
X_4	150	-150	-650	-20
X_5	150	150	-650	-20
X_6	-150	150	-650	-20
X_7	-150	-150	-650	-20
X_8	150	-150	-650	-20
X_9	150	150	-650	-20
X_{10}	-150	-150	-650	-20
X_{11}	-150	-150	-650	-20
X_{12}	150	-150	-650	-20
X_{13}	0	0	-650	0
X_{14}	0	0	-600	0
X_{15}	0	0	-700	0
X_{16}	0	0	-600	0
X_{17}	0	0	-700	0
X_{18}	0	0	-650	0
X_{19}	0	0	-650	20
X_{20}	0	0	-650	-20
X_{21}	0	0	-650	20

Table 0.4: Sequence of points for the reference trajectory

7 Performance evaluation metrics

To analyze and compare the control schemes proposed for the VELOCE robot in this work, we will use the following performance indices defined as follow:

- **Integral Absolute error**:

$$\text{IAE} = \int_{t_1}^{t_2} |e(t)| \mathrm{d}t \tag{22}$$

- **Integral of Time-weighted Absolute Error:**

$$\text{ITAE} = \int_{t_1}^{t_2} t|e(t)|\, dt \tag{23}$$

- **Root Mean Squared Error:**

$$\text{RMSE} = \left(\frac{1}{N} \sum_{i=1}^{4} e_i^2 \right)^{\frac{1}{2}} \tag{24}$$

- **Integral of Absolute Torques:**

$$\text{IAU} = \int_{t_1}^{t_2} |\Gamma(t)| \mathrm{d}t \tag{25}$$