RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Automatique

End-of-studies project dissertation

for obtaining the State Engineer's degree in Automatic Control

---

Fractional Order Control Strategies for Delta Robot

HATEM Yacine & IHADADENE Sidali

Under the supervision of **Pr. LADACI Samir**

Publicly presented and defended on the 31/05/2025 in front of the jury composed of :

President:	Prof. Fares BOUDJMA	ENP
Supervisor:	Prof. Samir LADACI	ENP
Examiner:	Dr. Hakim ACHOUR	ENP

ENP 2025

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Automatique

End-of-studies project dissertation

for obtaining the State Engineer's degree in Automatic Control

---

Fractional Order Control Strategies for Delta Robot

HATEM Yacine & IHADADENE Sidali

Under the supervision of **Pr. LADACI Samir**

Publicly presented and defended on the 31/05/2025 in front of the jury composed of :

President:	Prof. Fares BOUDJMA	ENP
Supervisor:	Prof. Samir LADACI	ENP
Examiner:	Dr. Hakim ACHOUR	ENP

ENP 2025

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Automatique

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en automatique

---

Stratégies de commande d'ordre fractionnaire pour robot Delta

---

HATEM Yacine & IHADADENE Sidali

Sous la direction de **Pr. LADACI Samir**

Présenté et soutenu publiquement le 31/05/2025 devant le jury composé de :

Composition du jury:

Président:	Pr. Fares BOUDJMA	ENP
Promoteur :	Pr. Samir LADACI	ENP
Examinateur :	Dr.Hakim ACHOUR	ENP

ENP 2025

الملخص

تُستخدم الروبوتات المتوازية، مثل الروبوت دلتا، بشكل متزايد في التطبيقات الصناعية المتطلبة نظرًا لأدائها الاستثنائي من حيث السرعة والدقة وقدرة التحميل. ومن أبرز التحديات في استخدامها هو الإدارة المثلى للتحكم فيها لضمان استقرار ودقة معززين في مواجهة الاضطرابات وعدم اليقين في النظام. تقترح هذه الأطروحة نهجًا مبتكرًا باستخدام التحكم من الرتبة الكسرية لتحسين التحكم التكيفي لهذا النوع من الروبوتات. تسمح تقنيات التحكم غير الخطية من الرتبة الكسرية بتنظيم أدق واستجابة أفضل للنظام، مع ضمان استقرار معزز. تثبت النتائج التجريبية صحة هذا النهج، حيث تظهر أنها تقدم أداءً متفوقًا مقارنة بالطرق التقليدية. تسلط هذه الدراسة الضوء على أهمية تطوير استراتيجيات التحكم لمواجهة التحديات الحالية للأنظمة الروبوتية في بيئات الصناعة الحديثة

الكلمات المفتاحية: روبوت دلتا، تتبع المسار، الحساب الكسرى، التحكم التكيفي المباشر وغير المباشر، طريقة باكستيبينغ، التحكم التكيفي المرجعي للنموذج، التحكم في الوضع الانزلاقي

Résumé

Les robots parallèles, tels que le robot Delta, sont de plus en plus utilisés dans des applications industrielles exigeantes en raison de leurs performances exceptionnelles en termes de rapidité, précision et charge utile. L'une des principales problématiques dans leur utilisation réside dans la gestion optimale de leur commande, afin d'assurer une stabilité et une précision accrues face aux perturbations et aux incertitudes du système. Ce mémoire propose une approche innovante en utilisant la commande d'ordre fractionnaire pour améliorer le contrôle adaptatif de ce type de robot. Les techniques de commande non linéaire fractionnaire permettent une régulation plus fine et une meilleure réactivité du système, tout en garantissant une stabilité renforcée. Les résultats expérimentaux valident cette approche, montrant qu'elle offre des performances supérieures aux méthodes traditionnelles. Ce travail met en évidence l'importance de l'évolution des stratégies de commande pour répondre aux défis actuels des systèmes robotiques dans les environnements industriels modernes.

Mots clés : Robot DELTA , Suivi de trajectoire , Calcul Fractionnaire , Commande adaptative directe et indirecte , Méthode de Backstepping , Commande adaptative par model de reference , Commande par mode glissant.

Abstract

Parallel robots, such as the Delta robot, are increasingly used in demanding industrial applications due to their exceptional performance in terms of speed, precision, and payload capacity. One of the main challenges in their use lies in the optimal management of their control to ensure enhanced stability and accuracy in the face of disturbances and system uncertainties. This thesis proposes an innovative approach using fractional-order control to improve the adaptive control of this type of robot. Nonlinear fractional-order control techniques allow for finer regulation and better system responsiveness, while ensuring reinforced stability. Experimental results validate this approach, showing that it offers superior performance compared to traditional methods. This work highlights the importance of evolving control strategies to meet the current challenges of robotic systems in modern industrial environments.

Keywords : Delta Robot, Trajectory Tracking, Fractional Calculus, Direct and Indirect Adaptive Control, Backstepping Method, Model Reference Adaptive Control, Sliding Mode Control.

Acknowledgments

First and foremost, we would like to express our deepest gratitude to Almighty Allah for granting us the strength, patience, and guidance to successfully complete this end-of-study project.

We would also like to extend our heartfelt thanks to our families for their continuous support, encouragement, and understanding throughout our academic journey. Their sacrifices and love have been an invaluable source of motivation.

Our sincere appreciation goes to our supervisor, **Pr Samir LADACI**, for their guidance, availability, and constructive feedback throughout the development of this project. Their expertise and commitment have greatly enriched our work.

We also thank the members of the jury for taking the time to evaluate our work and for their valuable insights and suggestions, which helped us to improve both technically and academically.

We are equally grateful to our professors and the entire teaching staff of **Ecole Nationale Polytechnique**, who have provided us with the knowledge and skills necessary to undertake this project.

A special thanks goes to our colleagues and fellow students for their collaboration, advice, and the friendly atmosphere we shared throughout our studies.

Finally, we express our gratitude to everyone who, directly or indirectly, contributed to the realization of this project.

Yacine and Sidali

Contents

List of Figures

List of Tables

List of Acronyms

- **FO** : Fractional-Order
- **PID** : Proportional–Integral–Derivative
- **MRAC** : Model Reference Adaptive Control
- **SMC** : Sliding Mode Control
- **DoF** : Degrees of Freedom
- **EE** : End Effector
- **IK** : Inverse Kinematics
- **FK** : Forward Kinematics
- **DIAC** : Direct and Indirect Adaptive Control
- **FOPID** : Fractional-Order PID (Proportional–Integral–Derivative)
- **SFC** : State Feedback Control
- **RMSE** : Root Mean Square Error
- **LTI** : Linear Time-Invariant
- **BIBO** : Bounded-Input, Bounded-Output (stability)
- **MIMO** : Multi-Input Multi-Output

General Introduction

Parallel robots, particularly Delta robots, have gained significant importance in recent years due to their high precision, speed, and ability to handle substantial payloads. These robots have found applications in various fields, such as industrial automation, biomedical procedures, and 3D printing, where rapid and accurate motion control is crucial. The Delta robot, with its closed-loop kinematics and efficient design, excels in high-performance tasks that require precise trajectory tracking and fast response times.

With the increasing complexity and demands placed on robotic systems, there has been a growing interest in improving their control mechanisms. Traditional integer-order controllers, although effective, often face limitations in terms of robustness, accuracy, and adaptability when dealing with uncertainties and non-linearities in dynamic environments. To overcome these limitations, fractional-order control techniques have emerged as a promising alternative. Fractional calculus, which extends traditional calculus to incorporate non-integer derivatives and integrals, offers enhanced flexibility in modeling and controlling complex systems. It allows for the adjustment of system dynamics over a broader range, providing better performance, especially in systems with inherent uncertainties or highly dynamic behavior.

This work aims to explore and compare various fractional-order control strategies applied to Delta robots. The goal is to develop robust and efficient control algorithms that can improve trajectory tracking and overall system performance. Specifically, we focus on three advanced fractional-order control methods: fractional-order adaptive backstepping, fractional-order sliding mode control (SMC), and **fractional-order model reference adaptive control (MRAC)**. Each of these methods offers unique benefits, such as robustness to disturbances, adaptability to changing environments, and the ability to provide precise control under varying operating conditions. By leveraging the advantages of fractional-order control, we seek to enhance the performance of Delta robots in real-world applications.

The document is structured as follows:

- **Chapter 1: Parallel Robots**
  This chapter provides a comprehensive overview of parallel robots, focusing on their design principles, advantages, and various types of parallel manipulators. We particularly highlight the Delta robot, describing its kinematic structure, operating principles, and typical applications. The chapter sets the foundation for understanding the unique challenges faced when controlling these systems and introduces the need for advanced control techniques to improve their performance.

- **Chapter 2: Fractional Calculus**
  In this chapter, we introduce the concept of fractional calculus, a key component of this research. We explain the mathematical foundation of fractional derivatives and integrals and their relevance to control theory. The chapter also covers the history and development of fractional calculus, highlighting its increasing importance in engineering fields, particularly in system modeling and control design. We emphasize how fractional-order

controllers offer superior flexibility and performance over traditional controllers.

- **Chapter 3: Modeling of the Delta Robot**
  This chapter presents a detailed mathematical model of the Delta robot, including its kinematics and dynamics. We describe the robot's structure, including the configuration of its arms, joints, and end-effector. The chapter outlines the equations governing the robot's motion, which serve as the basis for control design in subsequent chapters. A strong focus is placed on deriving accurate models to ensure that the control algorithms can be tested and evaluated effectively.

- **Chapter 4: Adaptive Fractional-Order Backstepping Control**
  Chapter 4 introduces the fractional-order adaptive backstepping control approach, which is designed to handle complex robotic systems with nonlinearities and uncertainties. The adaptive backstepping method is enhanced by incorporating fractional-order derivatives, offering improved robustness and performance in tracking trajectories. This chapter presents the theoretical foundations, the design of the controller, and its application to the Delta robot model.

- **Chapter 5:Adaptive Fractional-Order Sliding Mode Control (SMC)**
  In Chapter 5, we explore the implementation of fractional-order sliding mode control (SMC) with an adaptive framework for the Delta robot. The chapter discusses how SMC can enhance system robustness against disturbances and uncertainties. By incorporating fractional-order elements into the sliding mode controller, the resulting system offers better tracking accuracy and reduced chattering, making it ideal for high-performance applications. The chapter also presents simulation results demonstrating the effectiveness of the proposed approach.

- **Chapter 6: Model Reference Adaptive Control Based on Fractional-Order (MRAC)**
  This chapter focuses on the design of a fractional-order model reference adaptive controller (MRAC) for the Delta robot. MRAC is a powerful adaptive control strategy that ensures the system's output follows a reference model, even in the presence of parameter uncertainties and external disturbances. By using fractional-order elements, the MRAC is further refined to provide enhanced performance, especially in dynamic environments. We present the mathematical formulation of the controller and analyze its performance through simulations.

- **Chapter 7: Comparative Analysis and Conclusion**
  In the final chapter, we conduct a comparative analysis of the three control strategies—fractional-order adaptive backstepping, fractional-order SMC, and fractional-order MRAC. The performance of each method is evaluated based on several criteria, including tracking accuracy, robustness, and computational efficiency. The chapter concludes with a summary of the key findings and a discussion on the potential applications of fractional-order controllers in Delta robots and other robotic systems.

Through this work, we aim to demonstrate the practical advantages of fractional-order control in improving the trajectory tracking and robustness of Delta robots. We also explore how these advanced control strategies can contribute to the broader field of robotics, opening the door for more efficient and adaptive control techniques in industrial and scientific applications. The findings of this research are expected to advance the state of the art in both fractional-order control theory and robotic control systems.

Chapter 1

Parallel Robot Manipulators

1.1 Introduction

Parallel robots are manipulators composed of multiple kinematic chains (legs) arranged in parallel between a fixed base and a moving end-effector, forming closed-loop structures [1]. Each leg constrains the end-effector, and the combined constraints determine its pose. This architecture contrasts with serial robots, which have a single open chain of joints and links. Parallel designs concentrate actuators at the base, reducing moving mass and enabling exceptional dynamic performance and stiffness.[2]

Delta robots, pioneered by Clavel in the 1980s at EPFL, exemplify high-speed parallel mechanisms. A typical Delta robot uses three carbon-fiber arms connected via parallelogram linkages, guaranteeing pure translational motion of a triangular end-effector platform. The parallelogram ensures the platform remains parallel to the base throughout its workspace, simplifying kinematics and reducing actuation complexity [3]. Over the decades, Delta robots have evolved to achieve sub-0.3 s cycle times for pick-and-place tasks and payload capacities up to 8 kg, making them prevalent in food and electronics packaging lines [2]. Figure 1.1 shows a modern Delta robot deployed above a conveyor system, illustrating its typical overhead mounting and workspace coverage.

Figure 1.1: Modern Delta robot in a high-speed pick-and-place application.

1.2 Characteristics of Parallel Robots

Parallel manipulators possess distinct features arising from their closed-loop kinematic structure. Below, we analyze key characteristics with specific examples.

1.2.1 Structure and Design

In the Delta robot configuration, three identical kinematic chains connect the fixed base to the moving platform. Each chain consists of a base-mounted rotary actuator driving an upper link, which in turn drives two parallel lower links. These lower parallelogram links constrain the platform to pure translation, as they maintain constant orientation between the base and platform joints [4]. The resulting symmetric layout distributes loads uniformly and minimizes deflection under payload.

The Stewart platform uses six actuators, typically linear or rotary, each attached via universal joints to both base and platform. By coordinating leg extensions, the platform can achieve motion in all six degrees of freedom (three translations and three rotations) [5]. The hexapod's symmetric arrangement enhances stiffness and load capacity.

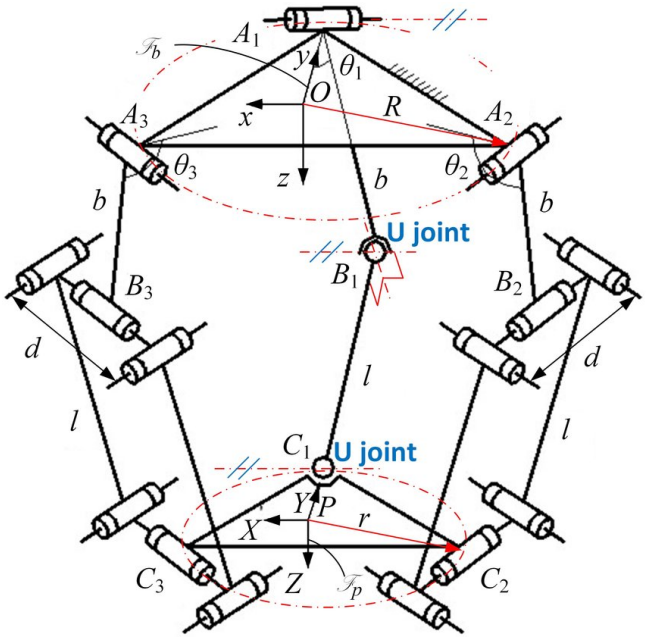


Figure 1.2: Schematic of Delta robot parallelogram linkages and Stewart platform leg arrangement.

1.2.2 Speed and Acceleration

Parallel robots achieve high dynamic performance due to low moving inertia: all heavy actuators are mounted on the base, leaving only lightweight links and end-effectors in motion. The Delta robot exemplifies this: modern FlexPicker models attain accelerations above 100 m/s^2 and linear speeds exceeding 10 m/s, enabling cycle rates upward of 300 picks per minute [6]. Distributed actuation allows each motor to contribute partially to the net end-effector movement, reducing torque demands and enabling rapid reversals with minimal overshoot.

Large-scale hexapods leverage similar principles: flight simulator platforms, weighing multiple

tons, produce realistic motion cues with accelerations up to 3 g, thanks to the coordinated action of base-fixed hydraulic actuators and low-inertia connecting links.

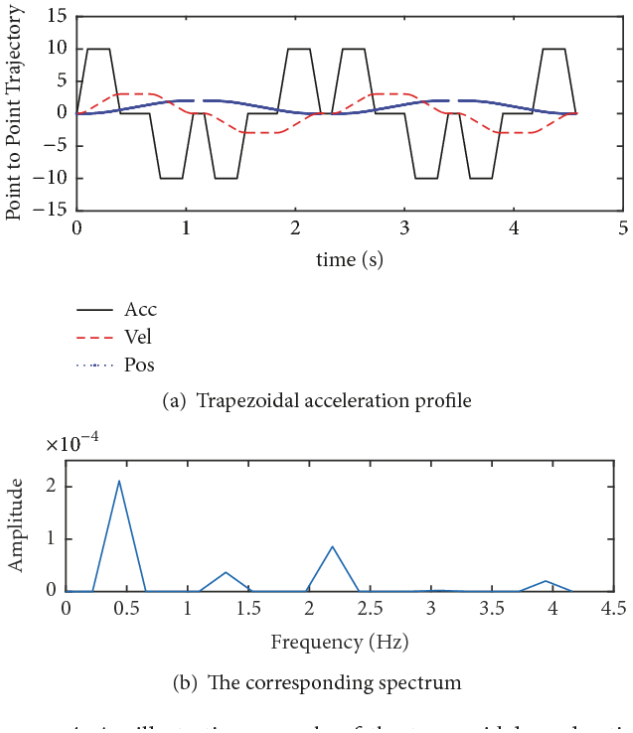


Figure 1.3: Representative speed and acceleration profile of a Delta robot performing pick-and-place cycles.

1.2.3 Precision and Rigidity

The closed-loop structure of parallel robots ensures that end-effector position errors are constrained by multiple linkages, averaging out individual joint errors and reducing compliance. Delta robots routinely achieve sub-millimeter repeatability at fast cycle rates, critical for tasks like microelectronics assembly and pharmaceutical dispensing [7]. Stewart platforms, used in telescope alignment and machine tool applications, maintain micrometer-level positioning under heavy loads, as all six legs oppose external forces simultaneously [2].

Advanced implementations use stiff materials (e.g., carbon fiber, hardened steel) and low-backlash joints, further improving accuracy. However, precision relies on meticulous calibration of leg lengths and joint offsets to align the physical robot with its kinematic model.

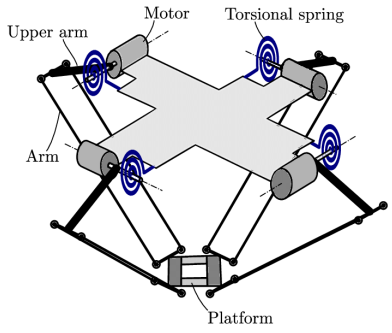


Figure 1.4: Comparison of positional deflection under load: parallel versus serial robot.

1.2.4 Degrees of Freedom

The number of independent actuators in a parallel robot defines its degrees of freedom (DOF):

- **3-DOF:** Classic Delta for pure translational motion along X, Y, and Z axes.
- **4-DOF:** Delta with an added rotary axis on the end-effector for yaw orientation.
- **5-DOF:** Delta plus two rotational axes (pitch and yaw) via a serial wrist.
- **6-DOF:** Stewart platform providing full translational and rotational control.

Workspace boundaries are determined by link geometry and joint limits, creating dome-shaped or cylindrical workspaces for translation-only Deltas and complex 6D volumes for Stewart platforms.

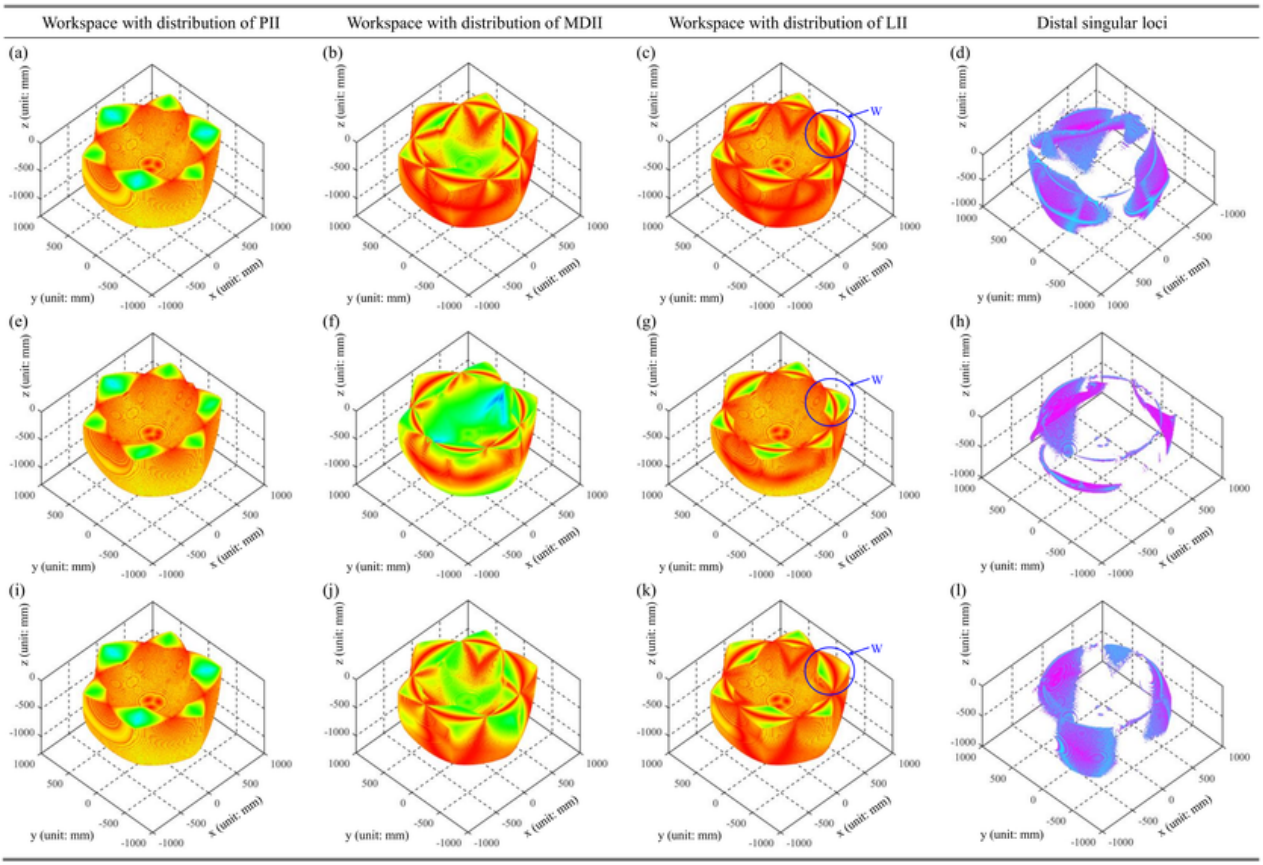


Figure 1.5: Workspace simulation examples for 3-DOF and 6-DOF parallel robots.

1.2.5 Constraints and Limitations

Parallel robots exhibit the following challenges:

- **Limited Workspace:** Defined by leg geometry; unable to reach beyond specific envelopes without encountering kinematic singularities.
- **Complex Control:** Forward kinematics require solving coupled non-linear equations; singularity avoidance and trajectory planning add software complexity.

- **Mechanical Complexity:** Multiple high-precision joints and symmetric assembly increase manufacturing and maintenance efforts.
- **Cost:** Custom designs, specialized components, and calibration processes result in higher upfront investment compared to off-the-shelf serial robots.

1.3 Applications of Parallel Robots

Parallel robots are deployed in domains that benefit from their unique capabilities. Key areas include:

1.3.1 Manufacturing and Logistics Industry

Delta robots revolutionized high-speed pick-and-place in packaging lines, handling up to 300 parts per minute with sub-millimeter accuracy [2]. Vision-guided Deltas sort food, pharmaceuticals, and electronic components on moving conveyors. Cable-driven parallel robots are emerging in warehouse automation, moving pallets and totes across large spans with speeds exceeding 5 m/s and payloads over 500 kg [8].

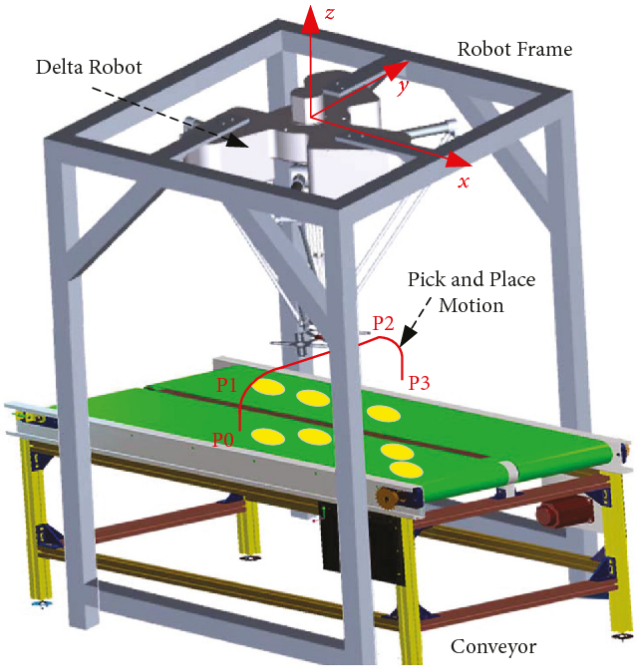


Figure 1.6: Delta robots in a high-speed packaging cell.

1.3.2 Medical and Surgical Robotics

Parallel mechanisms provide the stiffness and precision required for neurosurgery and orthopedic alignment. A 5-DOF parallel device can hold surgical tools with sub-millimeter accuracy during brain biopsies [9]. Stewart platforms serve as patient-positioning tables in radiotherapy, ensuring exact tumor alignment under imaging guidance [10]. RCM-capable designs enable minimally invasive instruments to pivot around fixed incision points, improving safety and dexterity.

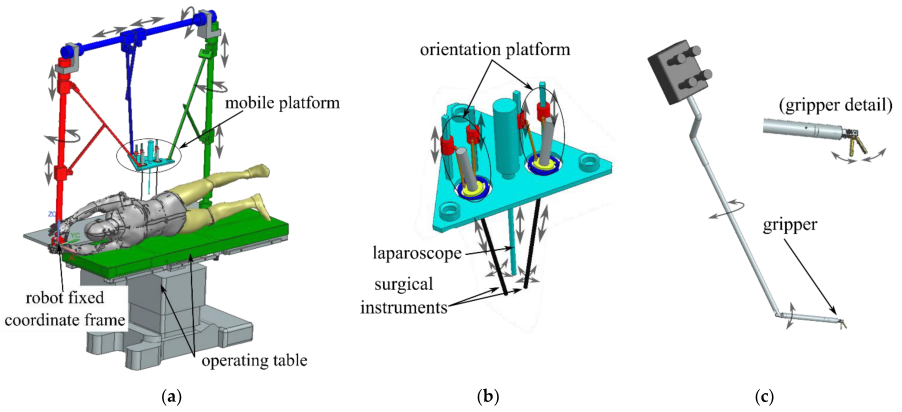


Figure 1.7: RCM-capable parallel manipulator used in laparoscopic surgery.

1.3.3 Motion Simulators

Stewart platforms underpin flight and driving simulators, replicating six-axis motions for pilot and driver training with accelerations up to 3 g. Civil engineering shake tables use similar hexapod rigs to subject full-scale structures to multi-directional seismic loads, vital for earthquake resilience research .

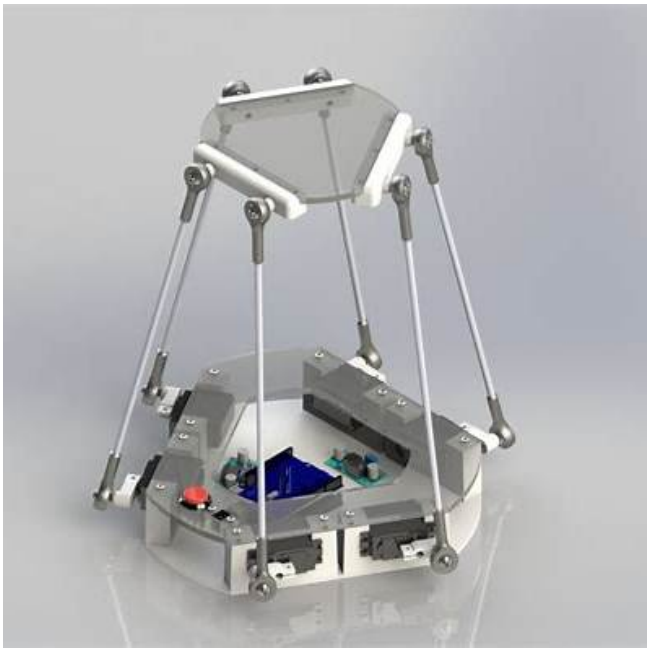


Figure 1.8: Flight simulator cockpit mounted on a Stewart platform.

1.3.4 Aerospace and Space Applications

Parallel kinematic machines assist in precision drilling and assembly of aircraft fuselages, maintaining hole placement tolerances of ±0.1 mm [11]. Hexapod devices calibrate satellite star trackers by executing precise orientation sequences. Cable-driven robots in microgravity, like ISS cargo handlers, use multiple winches to maneuver payloads without gravity-induced sag [12].

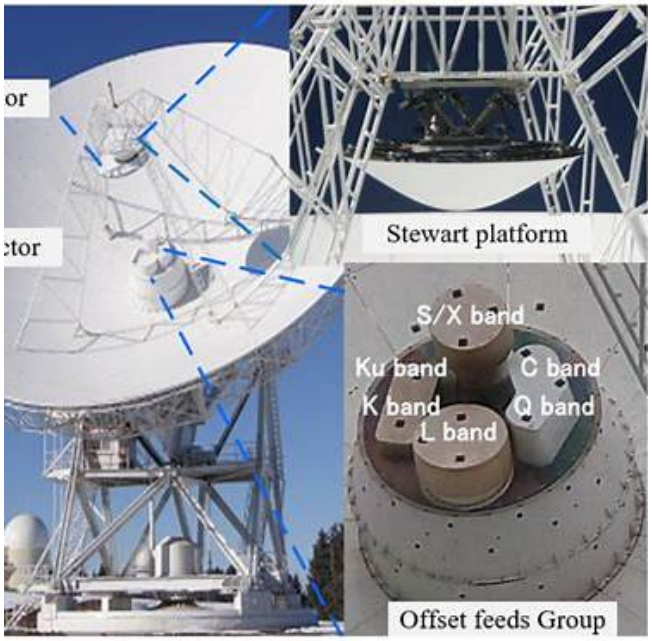


Figure 1.9: Hexapod platform used for satellite sensor calibration.

1.3.5 Construction and Civil Engineering

The HEPHAESTUS cable-driven robot installs glass façade panels on skyscrapers with 1 mm accuracy at heights exceeding 100 m, significantly reducing human exposure to hazardous conditions Prototype parallel masonry robots and large-scale 3D-printing cable systems explore autonomous bricklaying and concrete deposition for rapid construction

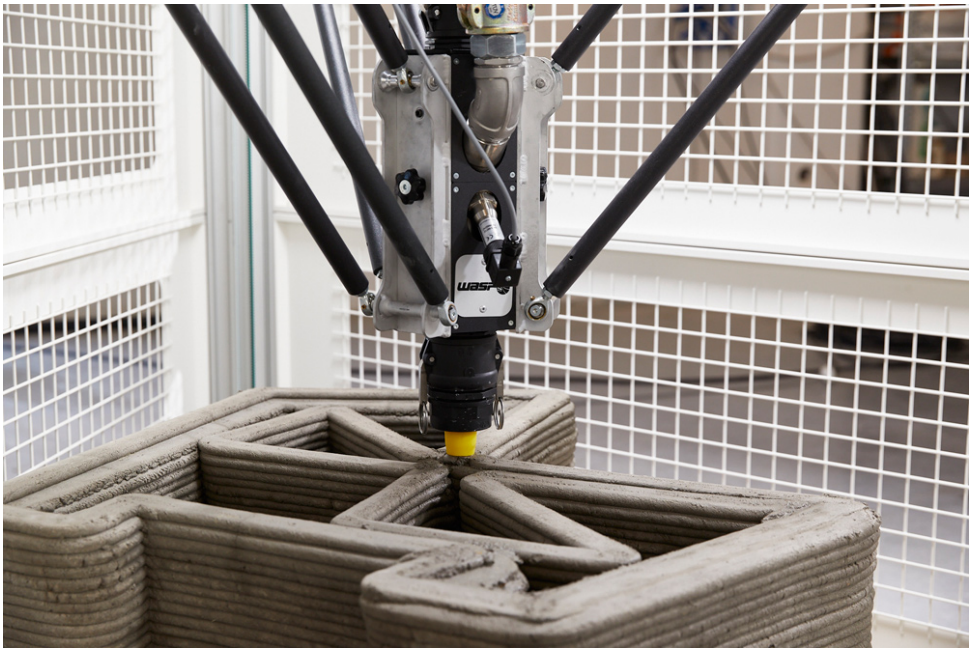


Figure 1.10: Delta WASP 3MT CONCRETE: Industrial-Scale Delta 3D Printer for Construction.

1.4 Comparison with Serial Robots

Consider the following table [13] :

Characteristic	**Parallel Robots**	**Serial Robots**
Workspace	Limited, bounded volume	Extensive, multi-directional reach
Speed/Accel.	Very high (Delta >100 m/s^2)	Moderate to high (inertia-limited)
Precision	High stiffness, micrometer repeatability	Lower inherent stiffness, requires compensation
Payload-to-Weight	High load sharing by legs	Lower ratio, heavy arm structures
Control	Complex inverse/forward kinematics, singularities	Simpler kinematics, well-established toolchains
Cost	Specialized, higher initial	Economies of scale, modular options

Table 1.1: Key differences between parallel and serial robot architectures.

1.5 Conclusion

Parallel manipulators particularly Delta and Stewart platforms offer unmatched dynamic performance, stiffness, and precision by virtue of their closed-loop kinematics and base-fixed actuation. Their specialized architectures serve high-speed pick-and-place, surgical precision, motion simulation, aerospace assembly, and large-scale construction tasks. Challenges remain in workspace limitations and control complexity, but advancements in reconfigurable designs, cable-driven systems, and hybrid parallel-serial solutions are expanding their applicability. As automation demands grow in speed, accuracy, and safety, parallel robots will continue to play essential roles alongside serial manipulators in the robotics ecosystem.

Chapter 2

Fractional Calculus

2.1 Introduction

Fractional calculus, dating back over 300 years, extends classical calculus to non-integer orders. Its origins trace to Leibniz and L'Hôpital (1695) and evolved through contributions from Euler, Lagrange, Laplace, Fourier, and Abel. Liouville (1832) and Riemann (1892) formalized key definitions, while 20th-century advancements enabled practical applications. Today, fractional calculus models diverse phenomena in physics, engineering, control systems, and bioengineering. It provides accurate descriptions of real-world systems, such as viscoelastic materials and anomalous diffusion. Modern numerical methods have facilitated its implementation, overcoming prior computational limitations. This chapter explores foundational concepts, mathematical formulations, and applications of fractional-order for control systems.

2.2 Applications for Control Systems and Robotics

Fractional calculus plays a crucial role in enhancing control strategies in robotics and automation. Fractional-order controllers (FOC), such as the fractional PID (FOPID), offer improved robustness and flexibility compared to classical PID controllers [14]. These controllers are particularly useful in robotic manipulators, where they enhance precision and adaptability to uncertainties [15].

Fractional calculus has gained significant attention in robotic manipulators due to its ability to model complex dynamics with memory and hereditary properties. Unlike traditional integer-order controllers, fractional-order controllers (FOCs) provide enhanced robustness, stability, and performance in nonlinear systems [15] :

- **Trajectory Control and Path Optimization:** Robotic manipulators require smooth and precise trajectory tracking to achieve accurate positioning. Fractional-order control improves trajectory tracking by incorporating memory effects, resulting in smoother transitions and better disturbance rejection [16]. This is particularly beneficial in industrial robots performing assembly or material handling tasks.

- **Impedance and Force Control:** In human-robot collaboration, impedance control adjusts the robot's stiffness and damping in response to external forces. Fractional

impedance controllers provide greater adaptability, making them effective in applications such as surgical robotics and rehabilitation systems [17].

- **Vibration and Noise Reduction:** Robotic arms often experience unwanted vibrations, especially in high-speed tasks such as pick-and-place operations. Fractional calculus-based controllers help reduce vibrations, leading to improved stability and lower energy consumption [18].

- **Future Directions:** The integration of fractional calculus with artificial intelligence and machine learning could enable adaptive and intelligent control strategies for robotic manipulators, further improving their autonomy and efficiency.

2.3 Fractional Order Operators

The integro-differential operator aDt^{α} where a and t are the limits of the operation is defined as:

$$aDt^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dx^{\alpha}}, & \Re(\alpha) > 0, \\ 1, & \Re(\alpha) = 0, \\ \int_a^t (d\tau)^{-\alpha}, & \Re(\alpha) < 0. \end{cases} \tag{2.1}$$

where α is the order of the operation, generally $\alpha \in \mathbb{R}$.

In the following, definitions of fractional-order operators are provided along with numerical approximation tools necessary for algorithm implementation.

2.4 Definitions

Several mathematical definitions exist for fractional-order integration and differentiation. These definitions do not always lead to identical results but are equivalent for a wide range of functions, particularly those considered in [19].

2.4.1 Riemann-Liouville Definition

The Riemann-Liouville integral is defined as:

Let C and $\mathbb{R}$ represent the complex and real number sets, respectively. The integral of order λ of a function f with a lower limit t_0 is defined as:

$$I_{t_0}^{\lambda} f(t) = \frac{1}{\Gamma(\lambda)} \int_{t_0}^{t} (t-\tau)^{\lambda-1} f(\tau) d\tau, \tag{2.2}$$

where $t \geq t_0$ and Γ is the Euler gamma function.

2.4.2 Caputo Definition

Caputo introduced another formulation of fractional-order differentiation:

$$ {}^{C}D_t^{\mu} f(t) = I_{t_0}^{n-\mu} \frac{d^n}{dt^n} f(t) = \frac{1}{\Gamma(n-\mu)} \int_{t_0}^{t} \frac{f^{(n)}(\tau)}{(t-\tau)^{\mu-n+1}} d\tau, \tag{2.3} $$

where n is the smallest integer such that $n-1 < \mu < n$.

This definition can also be expressed in terms of the Riemann-Liouville formulation as:

$$ {}^{C}D_t^{\mu} f(t) = {}^{RL}D_t^{\mu} f(t) - \sum_{k=0}^{n-1} \frac{t^k}{k!} f^{(k)}(t_0). \tag{2.4} $$

2.4.3 Grünwald-Letnikov Definition

The Grünwald-Letnikov derivative of order $\mu > 0$ is given by:

$$ {}^{GL}D_t^{\mu} f(t) = \lim_{h \to 0} h^{-\mu} \sum_{j=0}^{\infty} (-1)^j \binom{\mu}{j} f(t - jh), \tag{2.5} $$

where h is the sampling period, and the coefficients are:

$$ \omega_j^{(\mu)} = \frac{\Gamma(\mu+1)}{\Gamma(j+1)\Gamma(\mu-j+1)}. \tag{2.6} $$

2.4.4 Numerical Application

2.4.4.1 Fractional Derivatives Using Different Definitions

Table 2.1 presents the fractional derivatives of $f(x) = \sin(x) + x$ computed using the Caputo, Riemann-Liouville, and Letnikov definitions. The step size for the Letnikov approximation is set to $h = 0.005$.

Fractional Derivatives of $f(x)$				
α	x	Caputo	Riemann-Liouville	Letnikov ($h = 0.005$)
0.2	0.5	1.2031	1.2031	1.2023
0.5	10	2.5816	2.5816	5.8369
0.8	100	3.3566	3.3566	26.2013
1	1000	0	0	1.5644

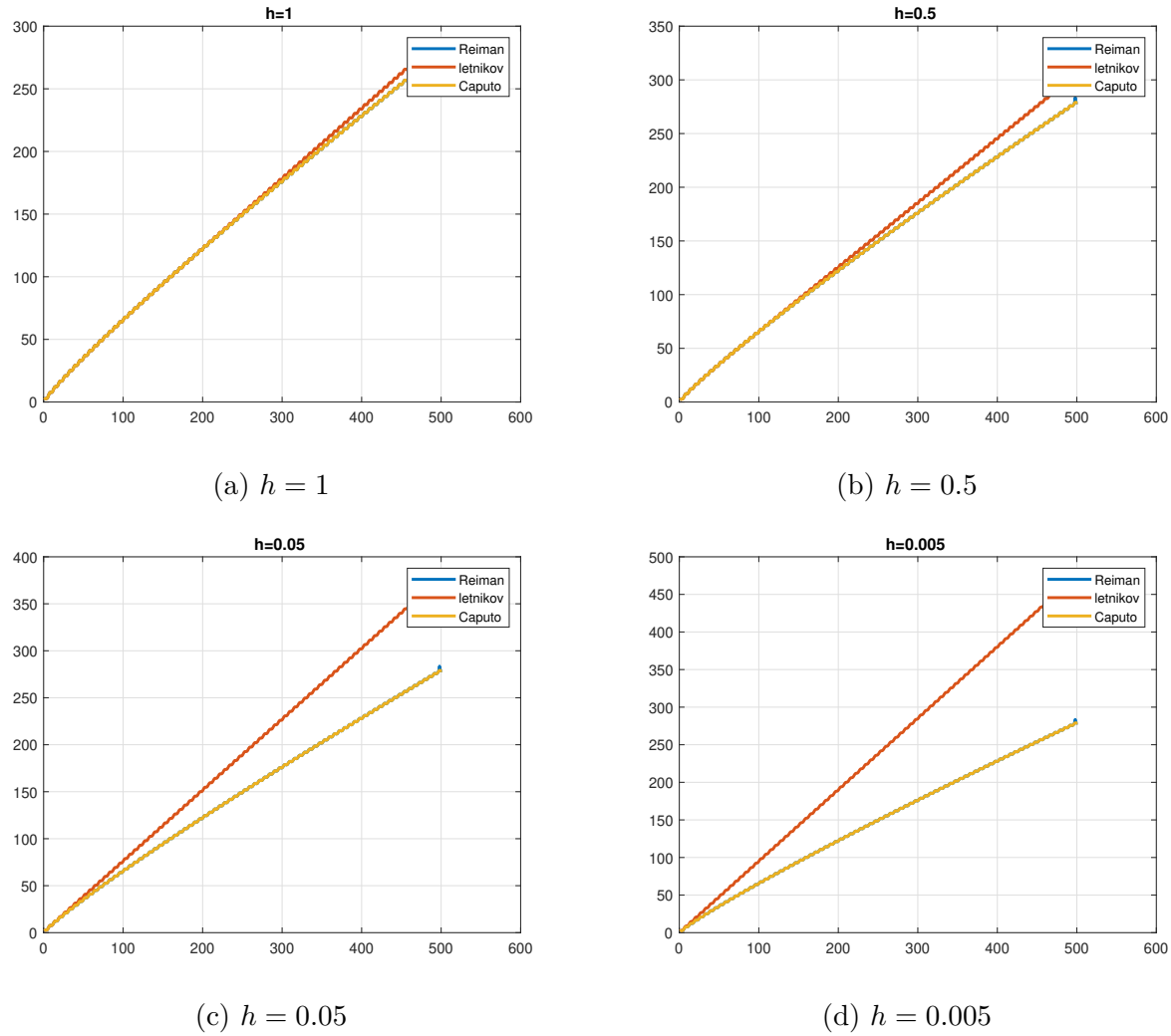
Table 2.1: Fractional derivatives of $f(x)$ using different definitions.

2.4.4.2 Letnikov Approximation Analysis

From Table 2.1, we observe a significant discrepancy between the Letnikov discrete approximation and the Riemann-Liouville-based formula, while the Caputo derivative remains consistent with the Riemann-Liouville result. To further analyze this difference, we investigate the influence of the step size h on the Letnikov approximation, as shown in Table 2.2.

Letnikov Approximation with Different Step Sizes				
α	x	$h = 0.05$	$h = 0.5$	$h = 1$
0.2	0.5	1.1942	1.1251	0.9794
0.5	10	2.6029	2.5980	2.6384
0.8	100	5.9174	3.2381	3.0429
1	1000	1.5828	1.7417	1.8533

Table 2.2: Effect of step size h on the Letnikov approximation.



(a) $h = 1$ (b) $h = 0.5$

(c) $h = 0.05$ (d) $h = 0.005$

Figure 2.1: Fractional derivatives of $f(x)$ using the Letnikov method for different step sizes h.

From Figure 2.1, we observe the following key phenomena:

- **For large h (e.g., $h = 1$):** The Letnikov approximation tends to stabilize but may introduce larger discretization errors due to the lower number of terms in the summation.

- **For moderate h (e.g., $h = 0.5$):** The approximation improves, providing more accurate values closer to the theoretical results.
- **For small h (e.g., $h = 0.05, h = 0.005$):** The method captures finer variations in the function but suffers from numerical instability, leading to significant variations in the computed values.
- **General conclusion:** Unlike Caputo and Riemann-Liouville definitions, the Letnikov approximation is highly sensitive to the choice of h, requiring careful tuning to balance accuracy and stability.

2.4.5 Properties of Fractional-Order Differentiation

Fractional-order differentiation has the following properties [8]:

1. If $f(t)$ is an analytic function, then the fractional-order differentiation ${}_0\mathcal{D}_t^\alpha f(t)$ is also analytic with respect to t.
2. If $\alpha = n$ and $n \in \mathbb{Z}_+$, then the operator ${}_0\mathcal{D}_t^\alpha$ can be understood as the usual operator d^n/dt^n.
3. Operator of order $\alpha = 0$ is the identity operator: ${}_0\mathcal{D}_t^0 f(t) = f(t)$.
4. Fractional-order differentiation is linear; if a, b are constants, then:
$$ {}_0\mathcal{D}_t^\alpha [af(t) + bg(t)] = a\,{}_0\mathcal{D}_t^\alpha f(t) + b\,{}_0\mathcal{D}_t^\alpha g(t). \tag{2.7}$$
5. For the fractional-order operators with $\Re(\alpha) > 0$, $\Re(\beta) > 0$, and under reasonable constraints on the function $f(t)$, it holds the additive law of exponents:
$$ {}_0\mathcal{D}_t^\alpha \left[{}_0\mathcal{D}_t^\beta f(t)\right] = {}_0\mathcal{D}_t^\beta \left[{}_0\mathcal{D}_t^\alpha f(t)\right] = {}_0\mathcal{D}_t^{\alpha+\beta} f(t). \tag{2.8}$$
6. The fractional-order derivative commutes with the integer-order derivative:
$$ \frac{d^n}{dt^n}\left(a\,{}_0\mathcal{D}_t^\alpha f(t)\right) = a\,{}_0\mathcal{D}_t^\alpha \left(\frac{d^n f(t)}{dt^n}\right) = a\,{}_0\mathcal{D}_t^{\alpha+n} f(t), \tag{2.9}$$
under the condition $t = a$ and $f^{(k)}(a) = 0, (k = 0, 1, 2, \ldots, n-1)$.

2.4.6 Laplace Transform for Fractional Derivatives

The Laplace transform of the fractional derivative of order α is given by:

$$ \mathcal{L}\left[{}_0\mathcal{D}_t^\alpha f(t)\right] = s^\alpha F(s) - \sum_{k=0}^{\lceil\alpha\rceil-1} s^{\alpha-k-1} f^{(k)}(0), \tag{2.10}$$

where s is the Laplace variable, and $f^{(k)}(0)$ are the initial conditions of the function $f(t)$.

The inverse Laplace transform allows recovering the function from the Laplace domain:

$$ f(t) = \mathcal{L}^{-1}\left[F(s)\right] = \frac{1}{j2\pi}\int_{c-j\infty}^{c+j\infty} e^{st} F(s) ds, \tag{2.11}$$

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

2.5 Differential Equations and Transfer Functions for Fractional Systems

2.5.1 Fractional Differential Equations

A fractional differential equation (FDE) is a generalization of classical differential equations where the order of differentiation can be a fractional number. The general form of a linear time-invariant fractional differential equation is:

$$\sum_{i=0}^{n} a_i {}_0\mathcal{D}_t^{\alpha_i} x(t) = \sum_{j=0}^{m} b_j {}_0\mathcal{D}_t^{\beta_j} u(t), \tag{2.12}$$

where ${}_0\mathcal{D}_t^{\alpha}$ represents the fractional derivative of order α, $x(t)$ is the system output, and $u(t)$ is the input. The coefficients a_i and b_j are real numbers.

2.5.2 Transfer Function of Fractional Systems

Taking the Laplace transform of a fractional differential equation and assuming zero initial conditions, we obtain the transfer function:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{j=0}^{m} b_j s^{\beta_j}}{\sum_{i=0}^{n} a_i s^{\alpha_i}}. \tag{2.13}$$

Unlike integer-order systems, the poles and zeros of fractional transfer functions do not necessarily lie on the real axis or the unit circle, leading to a richer and more flexible system behavior.

2.6 Frequency domain Approximation for Fractional systems

2.6.1 Oustaloup's Recursive Approximation

Oustaloup's method approximates fractional integrators and differentiators using a rational transfer function:

$$s^{\alpha} \approx K \prod_{k=1}^{N} \frac{s + \omega'_k}{s + \omega_k}, \tag{2.14}$$

where the parameters are defined as:

$$\omega'_k = \omega_b \cdot \omega_u^{(2k-1-\alpha)/N}, \tag{2.15}$$

$$\omega_k = \omega_b \cdot \omega_u^{(2k-1+\alpha)/N}, \tag{2.16}$$

$$K = \omega_h^{\alpha}, \quad \omega_u = \sqrt{\omega_h/\omega_b}. \tag{2.17}$$

Here: - α is the fractional order of differentiation or integration. - N is the number of poles and zeros used in the approximation. - ω_b and ω_h define the frequency range where the approximation is valid. - ω_k and ω_k' are the locations of the poles and zeros in the approximation.

2.6.2 Charef Approximation

The Charef approximation is a method used for approximating fractional-order differentiation and integration using integer-order transfer functions. It provides an efficient way to represent fractional operators in control systems and signal processing.[20]

The fractional derivative of order α is given by:

$$D^{\alpha} f(t) = \frac{d^{\alpha} f(t)}{dt^{\alpha}}, \quad 0 < \alpha < 1. \tag{2.18}$$

Charef's approximation represents this operator as a rational function of the Laplace variable s:

$$G(s) = \frac{1}{(1+\frac{s}{p})^{\alpha}} \approx \frac{\prod_{i=0}^{N-1}\left(1+\frac{s}{z_i}\right)}{\prod_{i=0}^{N}\left(1+\frac{s}{p_i}\right)} \tag{2.19}$$

The poles and zeros of the singularity function can be determined as follows:

$$p_i = (ab)^i p_0, \quad i = 1, 2, 3, ..., N \tag{2.20}$$

$$z_i = (ab)^i a p_0, \quad i = 2, 3, ..., N-1 \tag{2.21}$$

where:

$$p_0 = p_T 10^{\frac{c_p}{2.0\alpha}} \tag{2.22}$$

$$a = 10^{\frac{c_p}{10(1-\alpha)}} \tag{2.23}$$

$$b = 10^{\frac{c_p}{10\alpha}} \tag{2.24}$$

The fractional integrator is defined as:

$$H_I(s) = \frac{K_I}{\left(\frac{s}{\omega_c}\right)^{\alpha}} = \frac{K_I \omega_c^{\alpha}}{s^{\alpha}} = \frac{1}{s^{\alpha}} \tag{2.25}$$

The fractional differentiator is given by:

$$F_D(s) = K_D \left(\frac{s}{\omega_c}\right)^{\alpha} = \frac{K_D}{\omega_c^{\alpha}} s^{\alpha} = s^{\alpha} \tag{2.26}$$

The number of terms N is determined using:

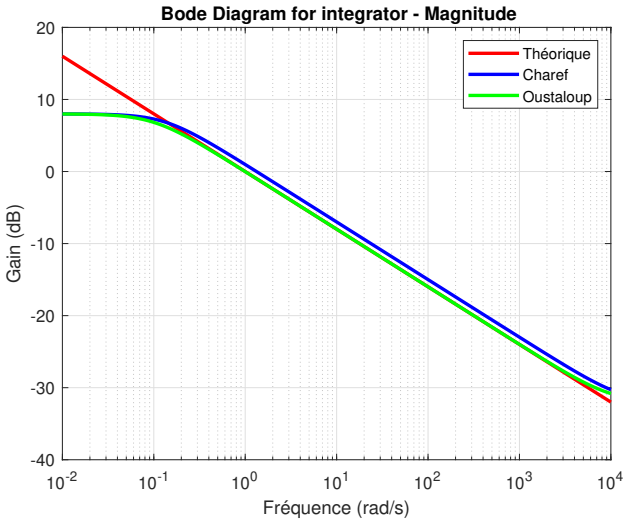
$$N = \lceil \frac{\log_{10}(\omega_m/p)}{\log_{10}(a \cdot b)} \rceil + 1 \tag{2.27}$$

2.6.3 Applications

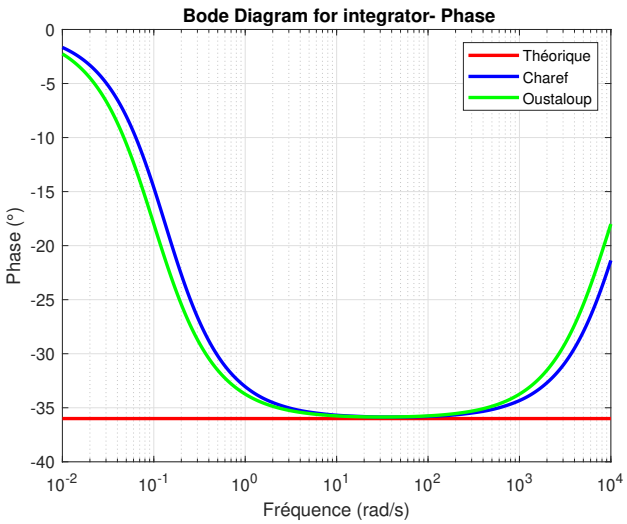
The fractional integrator (order 0.4) is defined as:

$$H_I(s) = \frac{1}{s^{0.4}} \tag{2.28}$$

We will then use the Oustaloup and Charef approximations to compare the efficiency of these approximations with the real fractional integrator. For this purpose, the Bode diagrams of the transfer functions for Oustaloup, Charef, and the real fractional integrator are plotted below.



(a) Maginute plot



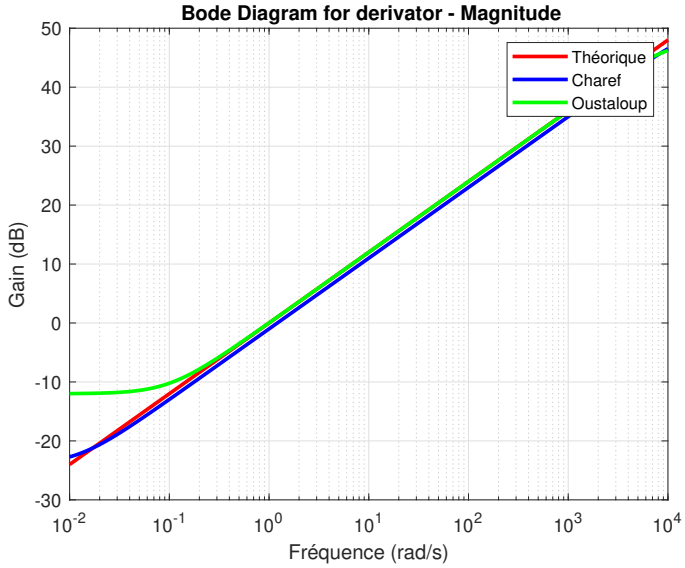
(b) Phase plot

Figure 2.2: Comparison of Oustaloup and Charef approximations for fractional integration ($\frac{1}{s^{0.4}}$).

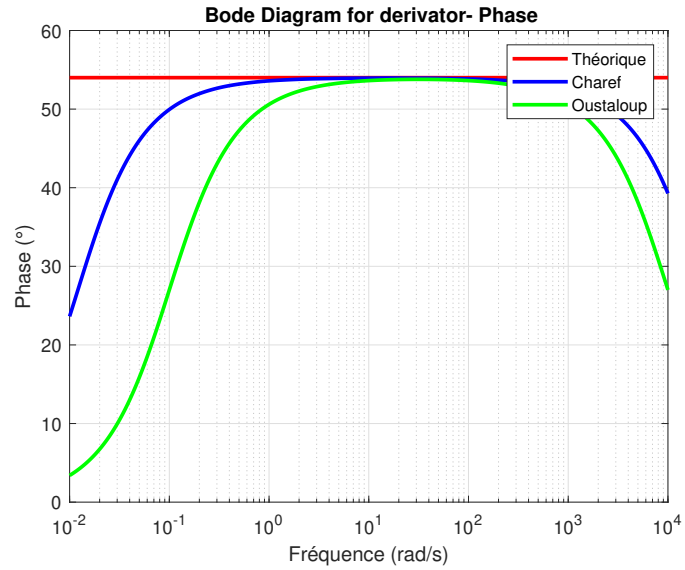
The fractional differentiator (order 0.6) is given by:

$$F_D(s) = s^{0.6} \tag{2.29}$$

We will use the Oustaloup and Charef approximations to evaluate their efficiency in approximating the real fractional differentiator. To achieve this, the Bode diagrams of the transfer functions for Oustaloup, Charef, and the real fractional differentiator are presented below.



(a) Maginute plot



(b) Phase plot

Figure 2.3: Comparison of Oustaloup and Charef approximations for fractional derivation ($s^{0.0.6}$).

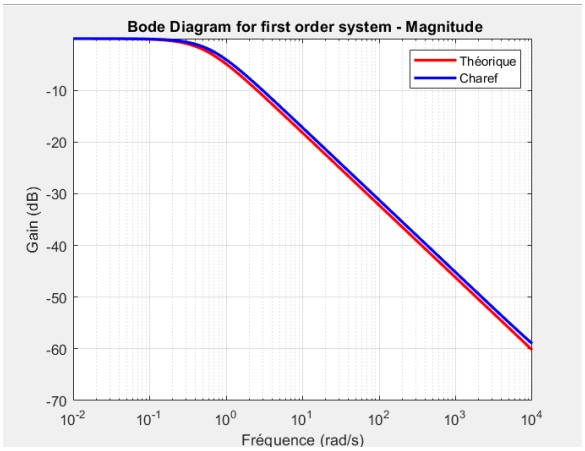
Both approximations provide a reasonable representation of fractional differentiation.The choice between these methods depends on the application requirements and computational constraints.

2.6.4 First-Order Transfer Function Approximation

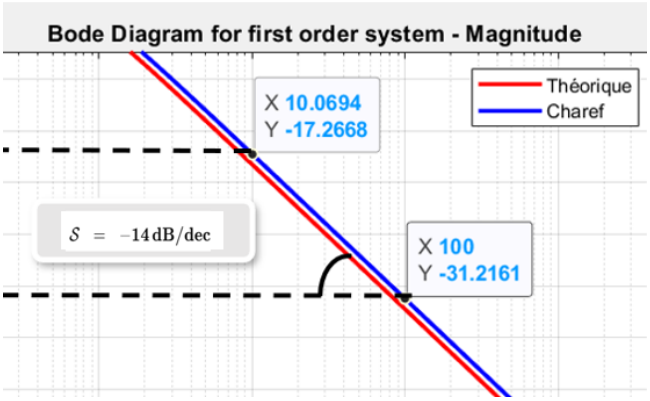
In this section, we study the approximation of the fractional-order transfer function

$$\frac{1}{(s+2)^{0.7}}$$

The aim is to find rational approximations that accurately represent the behavior of this function over a given frequency range. Using Charef method the results will be analyzed by plotting the Bode diagram for evaluating the accuracy of the approximations.



(a) Magnitude plot



(b) Magnitude plot

Figure 2.4: Bode diagram for first order system .

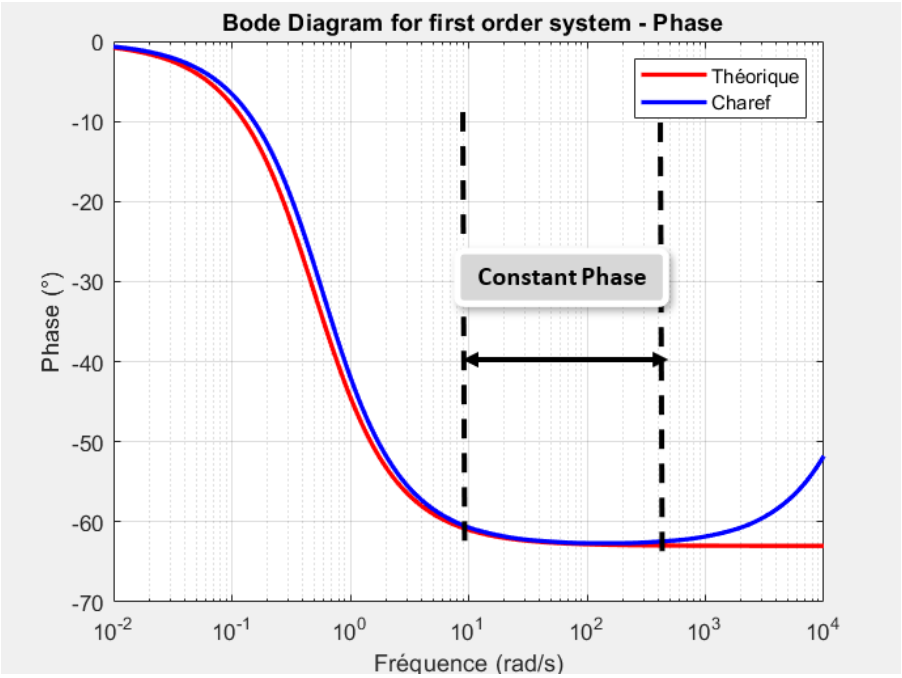


Figure 2.5: Phase plot

Comments and Conclusions for First-Order Fractional Systems

- Figures 3.4 and 3.5 illustrate the Bode diagrams of the first-order fractional system.
- The magnitude plot (Figure 3.4) shows that the slope of the gain is less than the classical -20 dB/dec, specifically $-\alpha \times 20$ dB/dec, due to the fractional nature of the system.
- The phase plot (Figure 3.5) highlights that the phase remains constant at $-\alpha \times \frac{\pi}{2}$, which is characteristic of fractional systems.
- The Charef approximation method was employed to obtain a rational approximation of the fractional-order system for implementation and analysis.
- The results demonstrate the effectiveness of the Charef method in approximating the frequency response of the real fractional system.

The characteristics of fractional-order systems, particularly the modified slope in the magnitude response and the constant phase shift, provide significant advantages in control applications:

- **Enhanced Robustness:** The gradual slope ($-\alpha \times 20$ dB/dec) allows for better robustness against parametric uncertainties compared to integer-order systems, which exhibit abrupt changes in gain.
- **Infinite Gain Margin:** Unlike classical integer-order systems, where gain margin is finite and can be a limiting factor in stability, the fractional system benefits from an infinite gain margin. This results in a more stable and resilient behavior in varying conditions.
- **Improved Phase Characteristics:** The constant phase shift of $-\alpha \times \frac{\pi}{2}$ ensures smoother phase transitions, reducing undesired resonances and oscillations in control loops.
- **Better Adaptability:** The flexibility of fractional-order models allows for more precise tuning of controllers to meet specific performance criteria, enhancing both stability and response time.

2.6.5 Second-Order Transfer Function Approximation

In this section, we study the approximation of the fractional-order transfer function :

$$\frac{1}{(s^2+2s+4)^{0.7}}$$

Using Charef approximation, the fractional transfer function is approximated and the the results will be analyzed accoring to Bode plots.

Let the transfer function $H_i(s)$ be defined as:

$$H_i(s) = \frac{1}{(1+\left(2\xi\frac{s}{\omega_n}\right)+\left(\frac{s}{\omega_n}\right)^2)^{\alpha}}$$

For the case where $\alpha < 0.5$, the approximation process is as follows:

1. Calculate the parameters η, a, and b:

$$\xi = B^{\alpha}, \quad \eta = 1-2\alpha, \quad a = 10^{\frac{\delta}{10(1-\eta)}}, \quad b = 10^{\frac{\delta}{10\eta}}$$

2. Compute the poles p_1 and z_1:

$$z_1 = \omega_n\sqrt{b}, \quad p_1 = az_1$$

3. Determine the number of factors N:

$$N = \lceil \log_{10}\left(\frac{\omega_m p_1}{ab}\right)+1\rceil + 1$$

4. For each factor k, the transfer function is updated as:

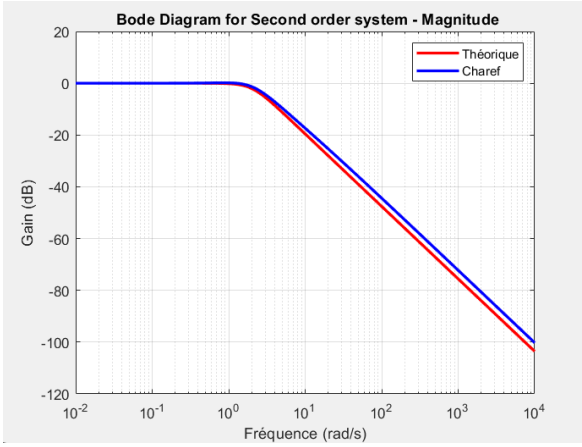
$$H_i(s) = \left(\frac{1 + \frac{s}{w_n}}{1 + \left(2B\frac{s}{\omega_n}\right) + \left(\frac{s}{\omega_n}\right)^2} \right) \cdot \frac{\prod_{i=0}^{N-1} \left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^{N} \left(1 + \frac{s}{p_i}\right)}$$

where:

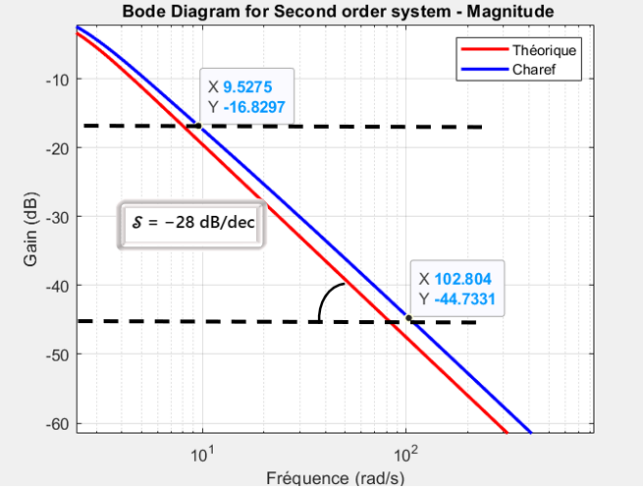
$$p_i = p_1 \left((ab)^{k-1}\right), \quad z_i = z_1 \left((ab)^{k-1}\right)$$

For the case where $\alpha \geq 0.5$, the approximation is similarly calculated, but with $\eta = -(1 - 2\alpha)$.

The overall transfer function is constructed by iterating over $k = 1, 2, \ldots, N - 1$.



(a) Magnitude plot



(b) Magnitude plot

Figure 2.6: Bode diagram for first order system .

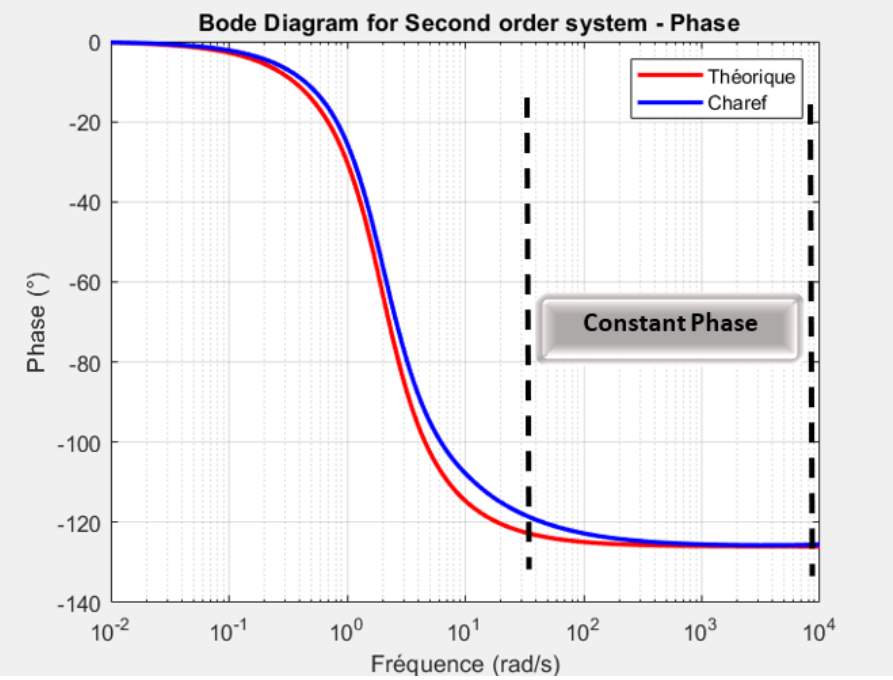


Figure 2.7: Phase plot

Comments and Conclusions for Second-Order Fractional Systems

- Figures 3.6 and 3.7 illustrate the Bode diagrams of the second-order fractional system.
- The magnitude plot (Figure 3.6) shows that the slope of the gain is less than the classical -40 dB/dec, specifically $-\alpha \times 40$ dB/dec, due to the fractional nature of the system.
- The phase plot (Figure 3.7) highlights that the phase remains constant at $-\alpha \times \pi$, which is characteristic of fractional systems.
- The Charef approximation method was employed to obtain a rational approximation of the fractional-order system for implementation and analysis.
- The results demonstrate the effectiveness of the Charef method in approximating the frequency response of the real fractional system.

The characteristics of second-order fractional-order systems, particularly the modified slope in the magnitude response and the constant phase shift, provide significant advantages in control applications:

- **Enhanced Robustness:** The gradual slope ($-\alpha \times 40$ dB/dec) allows for better robustness against parametric uncertainties compared to integer-order systems, which exhibit abrupt changes in gain.
- **Infinite Gain Margin:** Unlike classical integer-order systems, where gain margin is finite and can be a limiting factor in stability, the fractional system benefits from an infinite gain margin. This results in a more stable and resilient behavior in varying conditions.
- **Improved Phase Characteristics:** The constant phase shift of $-\alpha \times \pi$ ensures smoother phase transitions, reducing undesired resonances and oscillations in control loops.
- **Better Adaptability:** The flexibility of fractional-order models allows for more precise tuning of controllers to meet specific performance criteria, enhancing both stability and response time.

2.7 Performance Analysis of FO Systems

In this section, we analyze the performance of fractional-order systems by plotting the step response of both first-order and second-order fractional systems for different values of . The objective is to observe the impact of fractional differentiation on system dynamics, including response speed, overshoot, and stability.

2.7.1 Step Response Analysis

Figures 2.8 and 2.9 illustrate the step response of first-order and second-order fractional systems for different values of . The variations in influence the system's damping, transient response, and steady-state characteristics.

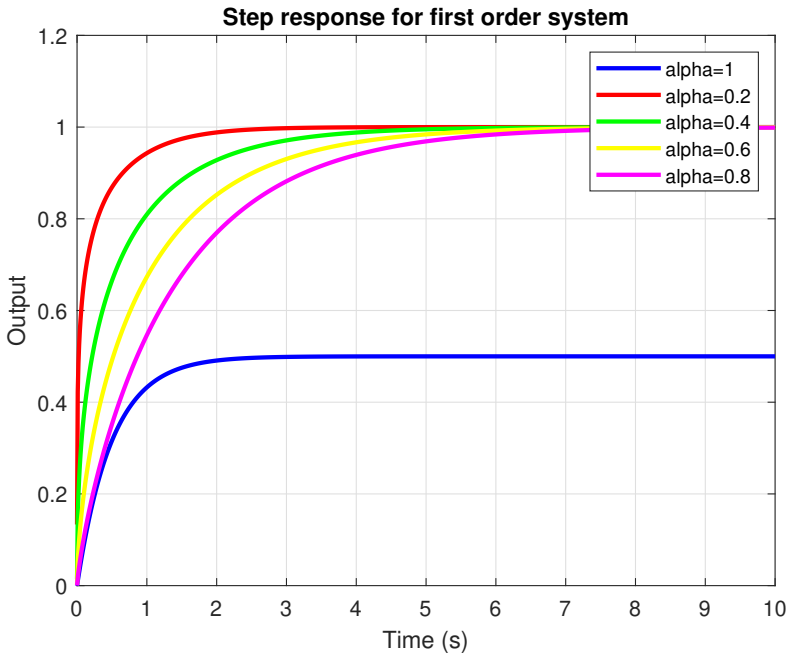


Figure 2.8: Step response of a first-order fractional system for different values.

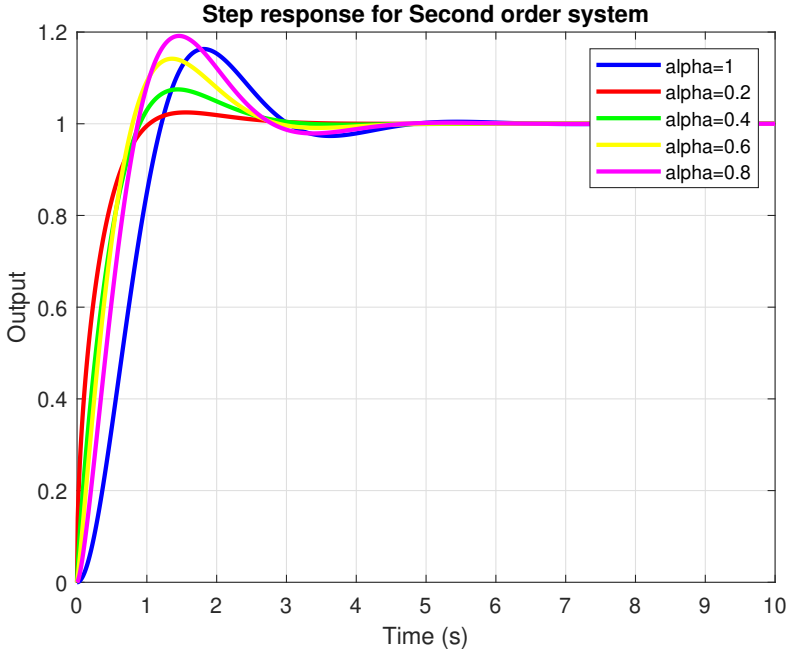


Figure 2.9: Step response of a second-order fractional system for different values.

2.7.2 Observations and Advantages

- As the fractional order α decreases, both the rise time and the overall response time become shorter, resulting in a faster system response.
- For the second-order system, the overshoot is reduced for lower values of α, which improves system stability and reduces oscillations.
- Fractional-order systems provide more flexibility in tuning system dynamics, allowing for improved trade-offs between response speed and stability.

- The smoother transition in fractional systems helps mitigate abrupt changes in control applications, leading to better robustness against disturbances.

These characteristics make fractional-order controllers a powerful tool for optimizing dynamic system performance.

2.8 Stability Analysis of Fractional-Order Systems

In order to determine the stability of a fractional system given by (3.30), we consider the following theorem [8].

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{j=0}^{m} b_j s^{\beta_j}}{\sum_{i=0}^{n} a_i s^{\alpha_i}}. \tag{2.30}$$

2.8.1 Matignon's Stability Theorem

Theorem 1 (Matignon's stability theorem): The fractional transfer function

$$G(s) = \frac{Z(s)}{P(s)} \tag{2.31}$$

is stable if and only if the following condition is satisfied in the σ-plane:

$$|\arg(\sigma)| > \frac{q\pi}{2}, \quad \forall \sigma \in \mathbb{C}, \quad P(\sigma) = 0, \tag{2.32}$$

where $0 < q \leq 1$ and $\sigma := s^q$. When $\sigma = 0$ is a single root of $P(s)$, the system cannot be stable.

2.8.2 Stability Verification Algorithm

The algorithm for checking the stability of the system can be summarized as follows:

1. Find the commensurate order q of $P(s)$ and determine the coefficients $a_1, a_2, \ldots, a_n$ in :

$$H(\lambda) = \sum_{k=0}^{m} b_k \sigma_k \Big/ \sum_{k=0}^{n} a_k \sigma_k \tag{2.33}$$

2. Solve for σ the equation:

$$\sum_{k=0}^{n} a_k \sigma^k = 0. \tag{2.34}$$

3. If all obtained roots satisfy the condition in Theorem 1, the system is stable.

2.8.3 Stability Domain for Fractional LTI systems

The stability regions of a fractional-order system are illustrated in Figure 2.11. It is important to note that currently, there are no polynomial techniques, either Routh or Jury type, to analyze the stability of fractional-order systems [8].

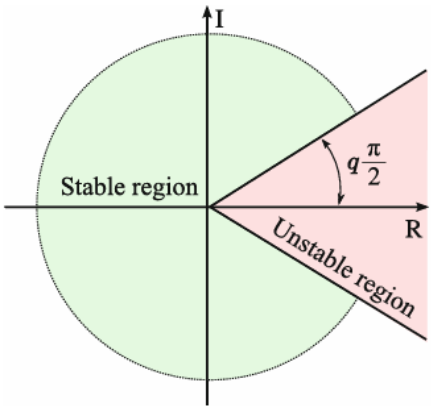


Figure 2.10: LTI fractional-order system stability region for $0 < q \leq 1$.

2.8.4 The Extended Lyapunov Second Theorem for Stability

Theorem 1 (Fractional-order extension of Lyapunov direct method).

Let $x = 0$ be an equilibrium point for the non-autonomous fractional-order system . We assume that there exists a Lyapunov function $V(x,t)$ and class-K functions c_i for $i = 1, 2, 3$ satisfying: [21]

$$c_1\|x\| \leq V(x,t) \leq c_2\|x\|, \quad t \geq t_0 \tag{2.35}$$

and

$$\frac{D^\alpha}{dt^\alpha}V(x,t) \leq -c_3\|x\|, \tag{2.36}$$

where $\alpha \in (0,1)$. Then, the origin of the system (3) is asymptotically stable.

Lemma 1
According to [[22],[23]] If $e(t)$ is a smooth function, then

$$\frac{1}{2}\,{}_0^C D_t^\alpha \left(e^T(t)e(t)\right) \leq e^T(t)\,{}_0D_t^\alpha e(t), \quad \forall t \in \mathcal{D}. \tag{2.37}$$

.

2.9 Fractional-Order PID Controller

The concept of a fractional-order PID (FOPID) controller was introduced by Podlubny in [24]. This generalized controller, also known as the $\mathbf{PI^\lambda D^\mu}$ controller, extends the traditional PID

controller by incorporating an integrator of fractional order λ and a differentiator of fractional order μ.

Podlubny demonstrated that the fractional-order controller offers superior performance compared to an integer-order PID controller when used in a control loop with a fractional-order plant. More recent research [25, 26] has confirmed that the fractional-order controller provides better adaptability and robustness in various applications.

In the Laplace domain, the parallel form of the **Fractional-Order PID (FOPID) controller** is given by:

$$C_{\text{FOPID}}(s) = K_p + K_i s^{-\lambda} + K_d s^{\mu}, \tag{2.38}$$

where: - K_p is the proportional gain, - K_i is the integral gain, - K_d is the derivative gain, - λ is the fractional order of the integrator ($0 < \lambda \leq 1$), - μ is the fractional order of the differentiator ($0 < \mu \leq 1$).

Clearly, when taking $\lambda = \mu = 1$, the controller reduces to the classical integer-order PID controller:

$$C_{\text{PID}}(s) = K_p + \frac{K_i}{s} + K_d s. \tag{2.39}$$

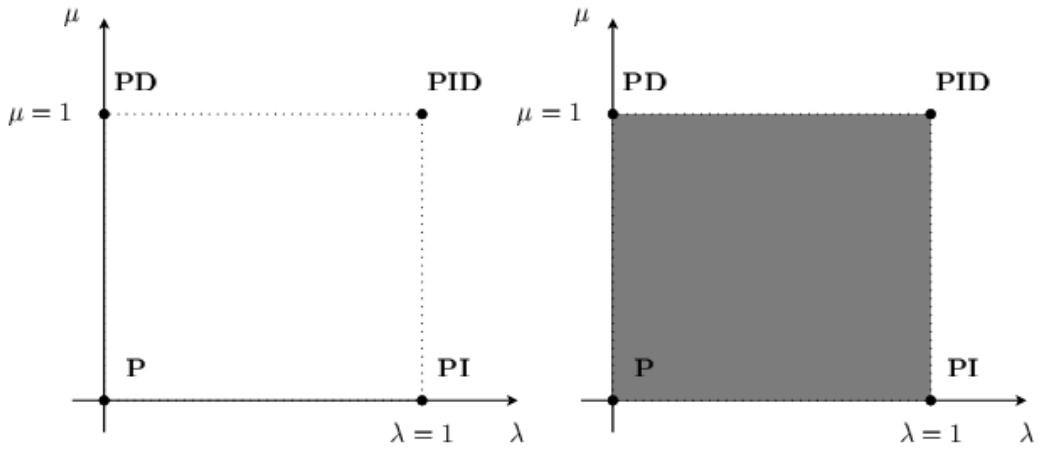


Figure 2.11: fractional-order PID Controller

2.10 State-Space Representation for Fractional Systems

The state-space representation of a fractional-order system is given by the following equations:

$$D^{\alpha} x(t) = Ax(t) + Bu(t)$$

where D^{α} represents the fractional derivative operator of order α, with $0 < \alpha < 1$.

$$y(t) = Cx(t) + Du(t)$$

It is important to highlight that the transfer function can only be derived from the state-space representation if the initial conditions allow it. If the Riemann-Liouville definition of the fractional derivative is used, the initial conditions may include fractional-order derivatives of functions. For this reason, the definitions of Caputo and Grünwald-Letnikov are often preferred for these calculations.

2.11 Controllable Canonical Form

Theorem 2. Transfer Function: The fractional transfer function is given by:

$$F(s) = \frac{\sum_{i=0}^{N} b_i s^i}{s^q + \sum_{i=0}^{N-1} a_i s^i}, \quad N, Q \in \mathbb{N}, \quad a_i, b_i \in \mathbb{R}. \tag{2.40}$$

This representation is equivalent to the following state-space form, called the controllable canonical form, assuming that the application conditions of Theorem 1 are met:

$$D^{\alpha} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} -a_{N-1} & -a_{N-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} b_{N-1} - a_{N-1} b_V \\ b_{N-2} - a_{N-2} b_V \\ \vdots \\ b_0 - a_0 b_V \end{bmatrix} u. \tag{2.41}$$

$$y = \begin{bmatrix} b_V - a_{N-1} b_V & b_{N-2} - a_{N-2} b_V & \cdots & b_0 - a_0 b_V \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + b_V u. \tag{2.42}$$

2.12 Observable Canonical Form

Theorem 3. The transfer function is equivalent to the following state-space representation, known as the observable canonical form, under the assumption that the conditions of Theorem 1 hold:[20]

$$D^{\alpha} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{N-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \begin{bmatrix} b_N - a_0 b_V \\ b_{N-1} - a_1 b_V \\ \vdots \\ b_1 - a_{N-1} b_V \end{bmatrix} u. \tag{2.43}$$

$$y = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + b_V u. \tag{2.44}$$

Different forms of the state representation are available in the literature.

2.13 Observability, Controllability, and Stability of Fractional Systems

The following results can be demonstrated similarly to the integer-order case.[20]

Theorem 4. A fractional-order system is observable if and only if the observability matrix:

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} \tag{2.45}$$

is of full rank.

Theorem 5. A fractional-order system is controllable if and only if the controllability matrix:

$$O = \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix} \tag{2.46}$$

is of full rank.

2.14 Fractional-State-Space Nonlinear Models

In the context of state-space representation, a fractional-state-space model can be used to describe the dynamics of a nonlinear system with fractional-order derivatives. The general form of a fractional-state-space nonlinear system is given by:

$$D^{\alpha}x(t) = f(x(t), u(t))$$

where: - D^{α} denotes the fractional derivative of order α (with $0 < \alpha < 1$), - $x(t) \in \mathbb{R}^n$ is the state vector, - $u(t) \in \mathbb{R}^m$ is the input vector, - $f(x(t), u(t))$ is a nonlinear function describing the system's dynamics,

For the output equation, we can express it as:

$$y(t) = h(x(t), u(t)$$

where: - $y(t)$ is the output vector,

The nonlinear function $f(x(t), u(t))$ could include various nonlinearities, such as saturation, dead zones, or hysteresis, depending on the system's characteristics.

2.15 Time domain fractional approximation

We use the Adams-Bashforth-Moulton (ABM) method, which is a predictor-corrector scheme for solving fractional differential equations.

$$y_h^P(t_{n+1}) = \sum_{k=0}^{\lceil\alpha\rceil-1} \frac{t_{n+1}^k}{k!} y^{(k)}(0) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y_h(t_j)) \tag{2.47}$$

$$y_h(t_{n+1}) = y_h^P(t_{n+1}) + \frac{h^\alpha}{\Gamma(\alpha+2)} f(t_{n+1}, y_h^P(t_{n+1})) \tag{2.48}$$

2.15.1 Simple Fractional Differential Equation

We consider the following fractional differential equation:

$$D^\alpha y + \lambda y + \cos(t) = 0, \quad y(0) = 1 \tag{2.49}$$

where α is the fractional order and λ is a positive parameter.

We choose $\lambda = 0.5$ and compare solutions for various values of α.

2.15.1.1 Results and Analysis

The results show that as α decreases, the exponential decay becomes slower, indicating a stronger memory effect.

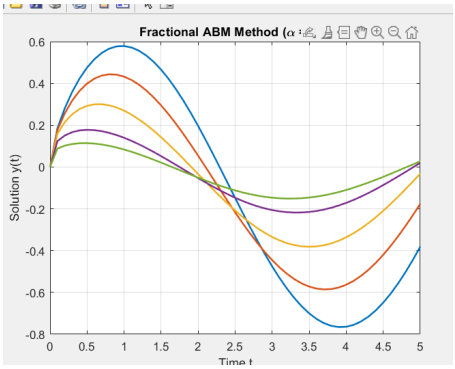


Figure 2.12: Dynamics of the fractional equation for different values of α.

2.15.2 Two-State Dynamic Model (η-model)

We consider here a two-state fractional dynamic system, commonly used to model viscoelastic, biological, or electrochemical phenomena with long-term memory [14, 27]. The system is defined by:

$$D^{\alpha_1} x(t) = ax(t) - by(t) \tag{2.50}$$

$$D^{\alpha_2} y(t) = cx(t) - dy(t) \tag{2.51}$$

where D^{α_i} denotes the fractional derivative of order $\alpha_i \in (0, 1]$, defined in the sense of Caputo or Grünwald–Letnikov. The parameters a, b, c, d are positive real constants representing the cross-interactions between the states.

This system belongs to the class of **multi-order linear time-invariant fractional systems**, extensively studied in the context of robust control and the stability of memory-affected systems [15, 28].

Two configuration cases are analyzed:

- **Standard case**:
  Parameters: $a = 1.2$, $b = 0.8$, $c = 0.5$, $d = 1.0$, with $\alpha_1 = \alpha_2 = 1$.
  This corresponds to a classical linear (non-fractional) system, which is stable and asymptotically convergent.

- **Fractional oscillatory case**:
  Parameters: $a = 1.5$, $b = 0.7$, $c = 0.6$, $d = 1.2$, with $\alpha_1 = 0.85$, $\alpha_2 = 0.88$.
  This configuration induces a slower dynamic, typical of memory-driven systems, and may exhibit prolonged damped oscillations due to the effect of fractional orders.

This type of model is relevant for analyzing memory effects in adaptive control systems and nonlocal dynamic behaviors.

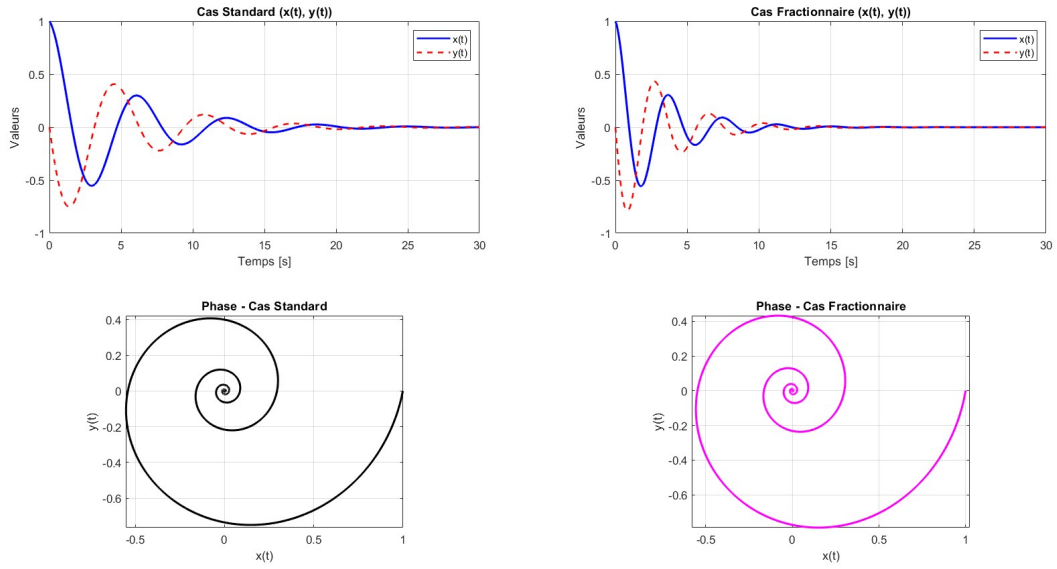


Figure 2.13: Comparison between standard and fractional dynamics:
(top) Time evolution of $x(t)$ and $y(t)$; (bottom) Phase portraits $x(t)$ vs $y(t)$.

Figure 2.13 illustrates the temporal dynamics and phase behavior of two linear two-state dynamic models: a standard integer-order system ($\alpha = 1$) and a non-integer fractional-order system ($\alpha_1 = 0.9$, $\alpha_2 = 0.88$). The parameters are selected to ensure both stability and a damped oscillatory behavior.

In the standard case (top left), the trajectories $x(t)$ and $y(t)$ display rapidly decreasing oscillations that converge to zero. The corresponding phase portrait (bottom left) forms a tight spiral, indicating fast amplitude decay—characteristic of a stable, weakly coupled system.

In contrast, in the fractional case (right), although the dynamic parameters remain the same, the oscillations persist longer and the damping is significantly reduced. The phase portrait

(bottom right) reveals a wider spiral, reflecting prolonged dynamic memory. This highlights the distinctive impact of fractional derivatives: their ability to capture long-memory dynamics, offering more realistic modeling of certain physical processes.

2.16 Conclusion

In conclusion, fractional-order systems offer a powerful and flexible framework for modeling and controlling complex dynamic systems. Their ability to capture memory and hereditary properties, which are not present in integer-order systems, makes them especially useful in fields such as engineering, physics, and biology. The extension of Lyapunov's direct method to fractional systems provides a valuable tool for analyzing the stability of such systems. The Charef approximation method further enhances the ability to approximate fractional-order transfer functions, aiding in practical design applications. As research in this area continues to grow, fractional-order models are expected to play an increasingly prominent role in the development of advanced control strategies and the design of resilient systems. Their integration into real-world applications could revolutionize industries by providing more accurate and efficient solutions for system behavior prediction and control. The complexity of fractional-order models, however, requires careful consideration of their computational and analytical challenges. Nonetheless, their potential for more accurate system representation and control remains a significant area for future exploration.

Chapter 3

Modeling of Delta robot

3.1 Introduction

The design of trajectory control for a delta robot requires comprehensive modeling that covers geometric, kinematic, and dynamic aspects. Each modeling approach serves to address different facets of the robot's operation, ensuring that all parameters are considered for optimal control. Geometric modeling focuses on the physical structure and configuration of the robot, helping to define its workspace and constraints. Kinematic modeling deals with the motion of the robot without considering forces, providing equations that describe the robot's movements in space. Dynamic modeling incorporates forces and torques, allowing for accurate control in real-world applications. This chapter will delve into these three types of modeling specifically applied to the delta robot, presenting the necessary equations, methodologies, and processes for each. The ultimate goal is to ensure that the trajectory control system can manage the robot's movement throughout its range of motion, achieving high precision and stability in various tasks.

3.2 Description of the Delta Robot

The Delta robot, as shown in Figure 2.1, is a parallel robot with three degrees of freedom (DOF), and it is specifically designed to perform high-speed, high-precision tasks such as picking and placing in industrial automation. The Delta robot consists of three identical kinematic chains, each comprising an arm and a forearm. The forearm is constructed from two parallel rods that are connected at one end to the movable platform, or "nacelle", and at the other end to the motors fixed to the robot's base.

These three kinematic chains are arranged symmetrically at 120° intervals, with each chain working in parallel to provide the desired motion of the movable platform. This design allows the Delta robot to perform three independent translational movements in the Cartesian space, providing precise control over the position of the platform. The combination of these movements results in the ability to manipulate the platform with a high degree of accuracy, making the Delta robot ideal for tasks requiring fine motion control.

One of the most crucial aspects of the Delta robot is the fact that the three arms work together to produce these translational movements, which are controlled by the motors fixed to the base. The inherent parallelism of the kinematic structure ensures that the Delta robot has a high payload-to-weight ratio, enabling it to carry out delicate operations at high speeds. In addition, the design of the robot enables the workspace to be relatively large, making it suitable for a

wide range of applications.

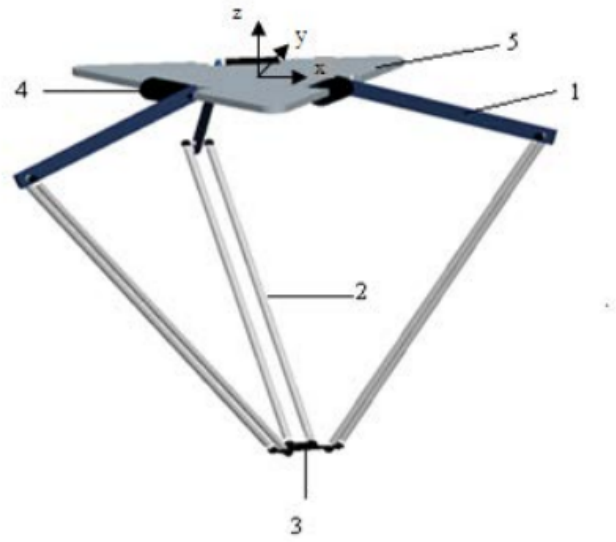


Figure 3.1: Description of Delta Robot

The workspace of the Delta robot is defined by the reach and flexibility of the three arms, as well as the geometry of the parallel linkage. Understanding the robot's workspace is essential for trajectory planning, as it dictates the points where the robot can position its end-effector within the operating environment. Additionally, the robot's ability to move with high accuracy and speed in this workspace is a result of its parallel structure, which minimizes the influence of gravity on the moving platform, making it ideal for precision tasks in assembly lines, 3D printing, and other fields requiring fine manipulations.

By fully understanding the geometrical and kinematic principles that govern the movement of the Delta robot, we can design precise trajectory control algorithms that ensure smooth and accurate operation within the robot's workspace. This chapter will outline the methodologies used to model the Delta robot and will detail the mathematical foundations behind its operation.

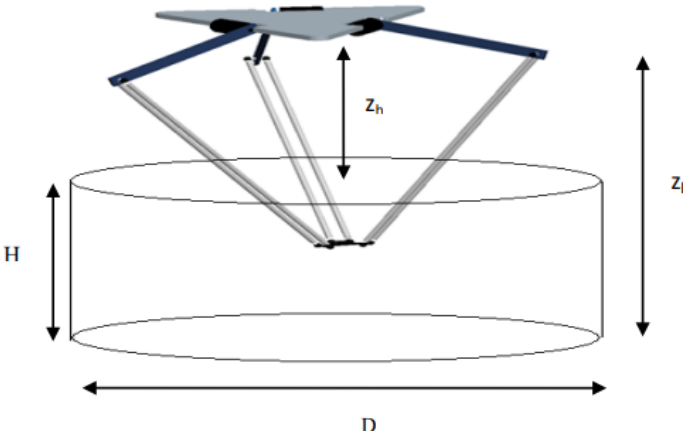


Figure 3.2: Description of Delta Robot

According to [29] The Table bellow summarizes the parameters of the robot with respect to the figure 2.1:

N°	Parameter	Notation	Value	Unit
1	The Arm	L_A	240	mm
2	The forearm	L_b	480	mm
3	The Diameter of the nacelle	D_n	30	mm
4	Motor's Inertia	I_m	$1.8 \cdot 10^{-4}$	Kg.m^2
5	The Diameter of the base	D_b	180	mm
6	The mass of the Nacelle	m_n	0.3815	Kg
7	The mass of the Forearm	m_b	0.2209	Kg

Table 3.1: Summary of Delta Robot Parameters

3.3 Mathematical modeling

3.3.1 Forward kinematics equations

Forward kinematics defines the relationship between the joint space and the end-effector space of a robot. Specifically, it involves determining the position and orientation of the end-effector (x, y, z) based on the given joint angles (θ). Therefore, defining a forward kinematics equation is essentially about finding the mathematical relationships that link the joint angles to the end-effector's position in Cartesian coordinates. This process allows us to predict where the end-effector will be located in space for any given set of joint parameters.

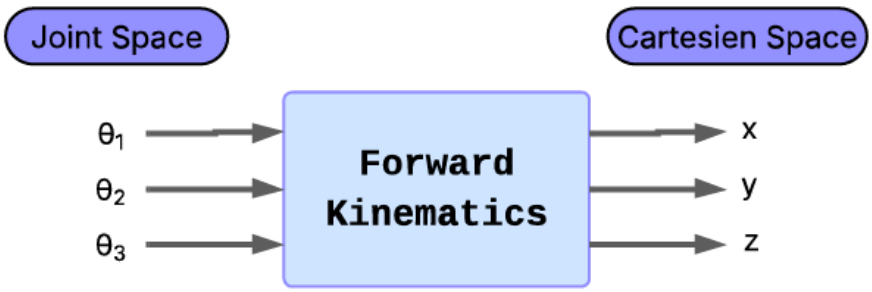


Figure 3.3: Direct geometric model

According to [2] for the top view, we fix a reference frame R_0 at the center of the base, and we also fix a reference frame R_i associated with each of the three motors. Thus, and due to symmetry, the rotation angles ϕ_1 , ϕ_2 and ϕ_3 are respectively equal to 0, $\frac{2\pi}{3}$, and $\frac{4\pi}{3}$.

The movement of the forearms describes a sphere with center P, which is the center of the nacelle, with a radius of L_B and passing through each of the points C_i connected to both of arms and forearms.

The points C_i will serve as intermediate points in the development of direct and inverse geometric models, with coordinates given by the following quantities: (θ_i are joint angles)

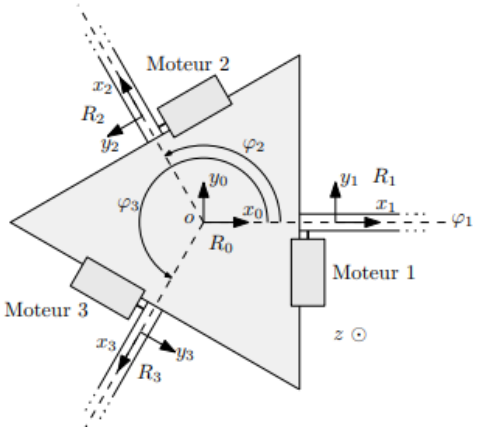


Figure 3.4: Top view of DELTA Robot

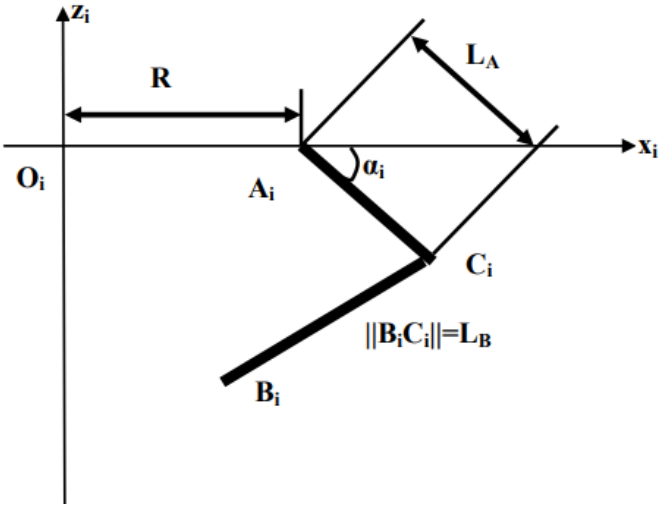


Figure 3.5: Front view of DELTA Robot

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_i} = \begin{pmatrix} R+\mathrm{L}_A \cdot \cos\theta_i \\ 0 \\ -\mathrm{L}_A \cdot \sin\theta_i \end{pmatrix} \tag{3.1}$$

Here, $i = 1, 2$, or 3. The points C_i belong to circles of radius L_B centered on the motor axis. Applying the change of coordinate system to equations (3.1):

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_0} = \begin{pmatrix} \cos\phi_i & -\sin\phi_i & 0 \\ \sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R+\mathrm{L}_A \cdot \cos\theta_i \\ 0 \\ -\mathrm{L}_A \cdot \sin\theta_i \end{pmatrix} = \begin{pmatrix} (R+\mathrm{L}_A \cdot \cos\theta_i)\cos\phi_i \\ (R+\mathrm{L}_A \cdot \cos\theta_i)\sin\phi_i \\ -\mathrm{L}_A \cdot \sin\theta_i \end{pmatrix}$$

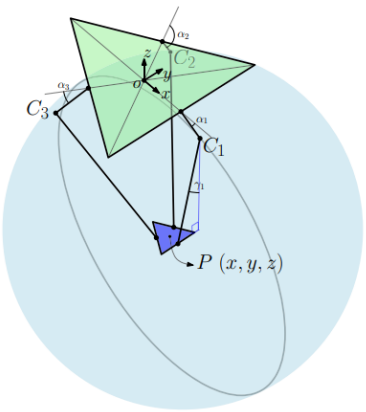


Figure 3.6: The sphere

Considering the equation representing the sphere ; while x,y and z are respectively the coordinate of the point P. (the center of the end effector) [7]

$$(X_{C_i} - x)^2 + (Y_{C_i} - y)^2 + (Z_{C_i} - z)^2 = L_B^2 \tag{3.2}$$

$$((R + L_A \cdot \cos\theta_i)\cos\phi_i - x)^2 + ((R + L_A \cdot \cos\theta_i)\sin\phi_i - y)^2 + (-L_A \cdot \sin\theta_i - z)^2 = L_B^2 \tag{3.3}$$

While,

$$R = R_b - R_n \tag{3.4}$$

When developing the formulas, we obtain the following system of equations :

$$A_i = -L_B^2 + L_A^2 + R^2 + 2RL_A\cos\theta_i \tag{3.5}$$

$$B_i = 2(R + L_A\cos\theta_i)\cos\phi_i \tag{3.6}$$

$$M_i = B_i\tan\phi_i \tag{3.7}$$

$$N_i = -2L_A\sin\theta_i \tag{3.8}$$

$$K_1 = B_1N_2 - B_1N_3 - B_2N_1 + B_2N_3 + B_3N_1 - B_3N_2 \tag{3.9}$$

$$K_2 = -B_1M_2 + B_1M_3 + B_2M_1 - B_2M_3 - B_3M_1 + B_3M_2 \tag{3.10}$$

$$K_3 = -B_1A_2 + B_1A_3 + B_2A_1 - B_2A_3 - B_3A_1 + B_3A_2 \tag{3.11}$$

$$K_4 = M_1A_2 - M_1A_3 - M_2A_1 + M_2A_3 + M_3A_1 - M_3A_2 \tag{3.12}$$

$$K_5 = -M_1N_2 + M_1N_3 + M_2N_1 - M_2N_3 - M_3N_1 + M_3N_2 \tag{3.13}$$

$$U = \frac{K_5^2 + K_1^2}{K_2^2} + 1 \tag{3.14}$$

$$V = \frac{2K_5K_4 + K_1K_3}{K_2^2} - \frac{K_5B_1 + K_1M_1}{K_2} - N_1 \tag{3.15}$$

$$W = \frac{K_4^2 + K_3^2}{K_2^2} - \frac{K_4B_1 + K_3M_1}{K_2} + A_1 \tag{3.16}$$

At least, we find:

$$z = \frac{-V \pm \sqrt{V^2 - 4UW}}{2U} \tag{3.17}$$

$$x = \frac{zK_5}{K_2} + \frac{K_4}{K_2} \tag{3.18}$$

$$y = \frac{zK_1}{K_2} + \frac{K_3}{K_2} \tag{3.19}$$

Because of the characteristic geometry of the robot, the positive solution of z is rejected, since the workspace is defined in the negative part of the z axis. Therefore, the equation for z becomes:

$$z = \frac{-V - \sqrt{V^2 - 4UW}}{2U} \tag{3.20}$$

3.3.2 Inverse kinematics equations

The inverse kinematic defines the relationship between the end-effector space and joint space of a robot. Specifically, it involves determining the necessery joint angles (θ) to achieve a desired position and orientation of the end-effector (x, y, z).

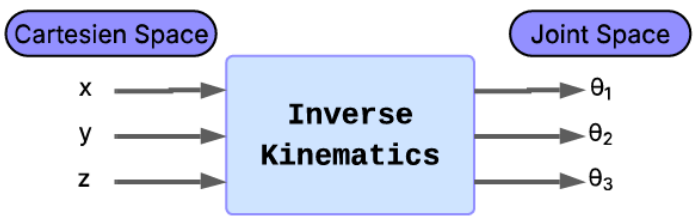


Figure 3.7: Inverse geometric model

According to Codourey's approach [30] : The forearms describe from the desired position $P = (x_0, y_0, z_0)$ of the nacelle in the reference frame R_i a sphere of radius L_B, which gives us equation .

$$\begin{pmatrix} x_{0i} \\ y_{0i} \\ z_{0i} \end{pmatrix}_{R_i} = \begin{pmatrix} \cos\phi_i & \sin\phi_i & 0 \\ -\sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}_{R_0}$$

The arm, centered at $(R, 0, 0)$, describes a circle of radius L_A (equation (3.1)). The point of intersection between the sphere and the circle lies in the plane $y = 0$ (equation (3.2)).

To solve the system let's define :

$$\begin{aligned} A &= L_A^2 - L_B^2 - R^2 + x_{0i}^2 + y_{0i}^2 + z_{0i}^2 \\ B &= 2x_{0i} - 2R \end{aligned}$$

$$\begin{aligned} a &= 4z_{0i}^2 + B^2 \\ b &= 4Rz_{0i}^2 + AB \\ c &= A^2 + 4R^2 z_{0i}^2 - 4z_{0i}^2 L_A^2 \end{aligned}$$

The negative solution of x is rejected, since it gives results out of the workspace of the robot, thus :

$$\begin{aligned} x &= \frac{b + \sqrt{b^2 - ac}}{a} \\ z &= \frac{A - Bx}{2z_{0i}} \end{aligned}$$

At least we have,

$$\begin{aligned} z &= -L_A \cdot \sin\theta_i \\ \theta_i &= -\arcsin(\frac{z}{L_A}) \end{aligned}$$

3.3.3 Dynamic model of delta robot

The dynamic model represents the system of equations that calculates the joint values based on a vector representing the injected torques.[29]

The following development is based on a set of simplifying assumptions:

- The mass of the forearms is distributed on the ends.

- The neglect of forearm's moment of inertia.
- All the frictions are neglect.

Taking into account the simplifying assumptions mentioned above, the dynamic model of the Delta robot is established in a straightforward manner. Each parallelogram can be replaced by a mass-less rod, as depicted in Figure (2.8).

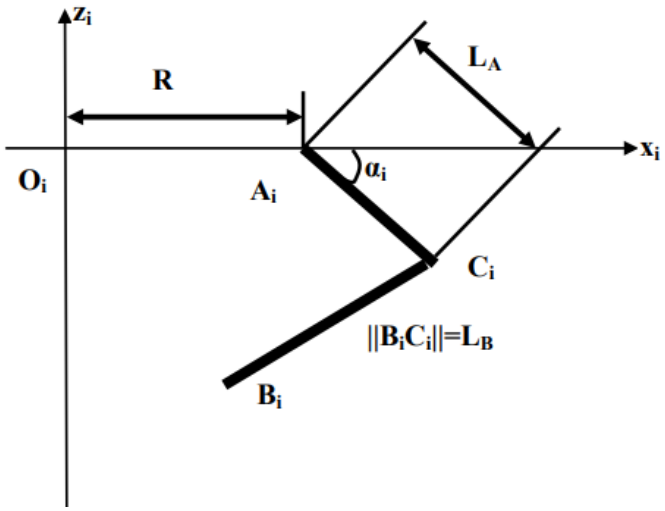


Figure 3.8: Front view of DELTA Robot

The development of the dynamic model is based on the Newton-Euler equations, which are respectively the sum of forces and moments, Therefore, the total body of the robot will be decomposed into several sub-bodies.

$$\sum_i \vec{F}_i = m_b \vec{a}_b \tag{3.21}$$

$$\sum_i \vec{\Gamma}_i = I_b \vec{\alpha}_b + \vec{\omega}_b \times (I_b \vec{\omega}_b). \tag{3.22}$$

In order to calculate the force vectors, we will first calculate the position vectors $\|AC\|_i$ and $\|BC\|_i$ respectively.

Director vector of the arm:

$$\overrightarrow{AC_i} = \begin{pmatrix} L_A \cos\theta_i \\ 0 \\ -L_A \sin\theta_i \end{pmatrix}$$

Director vector of the forearm:

$$\overrightarrow{B_iC_i} = \overrightarrow{B_iO_i} + \overrightarrow{O_iA_i} + \overrightarrow{A_iC_i}$$

$$\overrightarrow{B_iC_i} = -\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} R \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} L_A \cos\theta_i \\ 0 \\ -L_A \sin\theta_i \end{pmatrix}$$

Where $(x_i, y_i, z_i)^T$ is the vector expressing the coordinates of the end-effector in the frame R_i.

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}_{R_i} = \begin{pmatrix} \cos\phi_i & \sin\phi_i & 0 \\ -\sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{R_0}$$

By expanding the formulas, we find,

$$\overrightarrow{BC_i} = \begin{pmatrix} -x\cos\phi_i - y\sin\phi_i + R + L_A\cos\theta_i \\ x\sin\phi_i - y\cos\phi_i \\ -z - L_A\sin\theta_i \end{pmatrix}$$

In frame R, the director vector of each of the forearms is given by:

$$\overrightarrow{BC_i} = \begin{pmatrix} (L_A\cos\theta_i + R)\cos\phi_i - x \\ (L_A\cos\theta_i + R)\sin\phi_i - y \\ -z - L_A\sin\theta_i \end{pmatrix}$$

Application of Newton-Euler law on the forearms:

Isolating the forearm, considering its negligible mass and inertia, we have;

$$\vec{F_i} + \vec{F}_{fa_i} = \vec{0}$$

Thus, the two forces $\vec{F_i}$ and $\vec{F}_{fa_i}$ are opposite in direction and along the direction $\vec{BC}_i$ of the forearm.

- $\vec{F_i}$: The contact force between the nacelle and the forearm.
- $\vec{F}_{fa_i}$: The contact force between the arm and the forearm.

Expanding upon the formulas we have, correct

$$\vec{F_i} = F_i \cdot \frac{\vec{BC}_i}{|\vec{BC}_i|} = F_i \cdot \frac{\vec{BC}_i}{|L_B|} = F_i \cdot \vec{P_i} \tag{3.23}$$

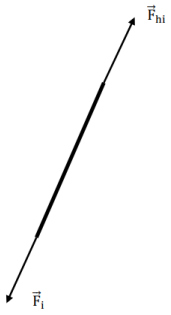


Figure 3.9: The isolation of the Forearms

While,

$$\vec{P}_i = \begin{pmatrix} P_{ix} \\ P_{iy} \\ P_{iz} \end{pmatrix} = \begin{pmatrix} \frac{(L_A \cos\theta_i + R)\cos\phi_i - x}{LB} \\ \frac{(L_A \cos\theta_i + R)\sin\phi_i - y}{LB} \\ \frac{-z - L_A \sin\theta_i}{LB} \end{pmatrix}$$

Application of Newton-Euler law on the nacelle:

We isolate the nacelle and add the constraint forces $\vec{F_i}$ (Figure 3.15). Due to the design of the Delta robot, the platform can only undergo translational movements in the operational space. Therefore, the Newton-Euler equations reduce to the equations of translational dynamics (Newton's equations). In the reference frame R, we obtain:

$$\vec{g}_n - \sum_{i=1}^{3} \vec{F_i} = m_n \ddot{\vec{X}}_n$$

Where:

- $\vec{F_i}$: Constraint forces between the nacelle and the forearms, expressed in reference frame R.

- $\ddot{\vec{X}}_n = \begin{pmatrix} \ddot{x} & \ddot{y} & \ddot{z} \end{pmatrix}^T$: Acceleration of the nacelle expressed in reference frame R.

- $\vec{g}_n = \begin{pmatrix} 0 & 0 & -m_n g \end{pmatrix}^T$: Gravitational force acting on the nacelle.

- m_n: Total mass of the nacelle, including the transported mass and the mass of the forearms attached to the platform.

- g: Acceleration due to gravity.

While injecting the equation (3.3.8) in (3.3.9) we have;

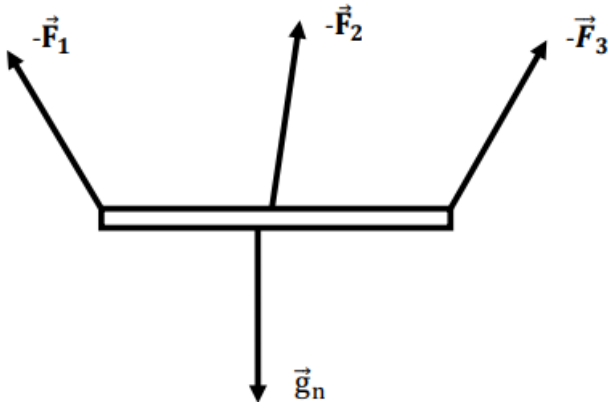


Figure 3.10: The isolation of the nacelle

$$\mathbf{F}_1 \begin{pmatrix} P_{x1} \\ P_{y1} \\ P_{z1} \end{pmatrix} + \mathbf{F}_2 \begin{pmatrix} P_{x2} \\ P_{y2} \\ P_{z2} \end{pmatrix} + \mathbf{F}_3 \begin{pmatrix} P_{x3} \\ P_{y3} \\ P_{z3} \end{pmatrix} = \begin{pmatrix} -m_n\ddot{x} \\ -m_n\ddot{y} \\ m_n[-g-\ddot{z}] \end{pmatrix} \tag{3.24}$$

Equation (3.24) can be expressed as:

$$\mathbf{P} \cdot \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \end{pmatrix} = \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} \tag{3.25}$$

where:

$$\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} -m_n\ddot{x} \\ -m_n\ddot{y} \\ m_n[-g-\ddot{z}] \end{pmatrix} = -m_n \cdot \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} - m_n \cdot \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \tag{3.26}$$

And

$$\mathbf{P} = \begin{pmatrix} P_{x1} & P_{x2} & P_{x3} \\ P_{y1} & P_{y2} & P_{y3} \\ P_{z1} & P_{z2} & P_{z3} \end{pmatrix} \tag{3.27}$$

Then,

$$\mathbf{F} = \mathbf{P^{-1}} \cdot \mathbf{M} \tag{3.28}$$

At least we got,

$$\mathbf{F} = \mathbf{K_n} \cdot \ddot{\mathbf{X}} - \mathbf{G_n} \tag{3.29}$$

While,

$$\mathbf{K_n} = -m_n \cdot \mathbf{P^{-1}} \tag{3.30}$$

$$\mathbf{G_n} = m_n \cdot \mathbf{P^{-1}} \cdot \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \tag{3.31}$$

Application of Newton-Euler law on the Arms:

The following figure shows the isolation of the arms. .

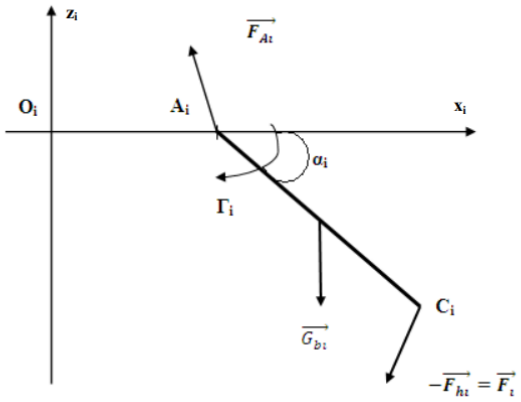


Figure 3.11: The isolation of the Arm

Due to the rotation around the y-axis Euler's equation can be written as:

$$\vec{AC}_i \times \vec{F}_i + \vec{r}_a \times \vec{G}_{\mathrm{a}_i} + \vec{\Gamma}_i = \mathbf{I}_{\mathrm{a}_i} \cdot \ddot{\vec{\theta}}_i \tag{3.32}$$

where:

- $\mathbf{r}_a$: Vector expressing the position of the center of gravity of arm i.
- $\mathbf{G}_a = \begin{pmatrix} 0 & 0 & -m_a g \end{pmatrix}^T$: Gravity forces of arm i expressed in reference frame R_i.
- m_b: Total mass of the arm, including the mass of the attached forearms.
- $\mathbf{\Gamma}_i$: Torque around point A_i.
- $\mathbf{I}_a$: Moment of inertia of the arm around point A_i.
- $\ddot{\boldsymbol{\theta}}_{\boldsymbol{i}}$: Angular acceleration of arm i.

The vectors are given by :

$$\vec{AC}_i = \begin{pmatrix} L_A \cos\theta_i \\ 0 \\ -L_A \sin\theta_i \end{pmatrix} \quad \text{and} \quad \vec{r}_a = \begin{pmatrix} r_a \cos\theta_i \\ 0 \\ -r_a \sin\theta_i \end{pmatrix} \tag{3.33}$$

Therefore, we obtain the following expressions for the vector products $\vec{AC}_i \times \vec{F}_i$ and $\vec{r}_a \times \vec{G}_a$:

$$\vec{AC}_i \times \vec{F}_i = \begin{pmatrix} -\frac{L_A}{L_B} \sin\theta_i \cdot F_i \cdot y_i \\ \frac{F_i}{L_B} \left[z_i \cos\theta_i - (R - x_i) \sin\theta_i \right] \\ -\frac{L_A}{L_B} \cos\theta_i \cdot F_i \cdot y_i \end{pmatrix} \tag{3.34}$$

$$\vec{r}_a \times \vec{G}_a = \begin{pmatrix} 0 \\ r_a \cdot m_a \cdot g \cdot \cos\theta_i \\ 0 \end{pmatrix} \tag{3.35}$$

Since the rotation of the arms occurs around the y-axis, only the following equation along the y-axis is useful to us. Developing it, we find:

$$\Gamma_i = I_a \ddot{\theta}_i - F_i \frac{L_A}{L_B} \left[z_i \cos\theta_i - (R - x_i) \sin\theta_i \right] - r_a m_a g \cos\theta_i \tag{3.36}$$

Also,

$$\Gamma_i = I_a \ddot{\theta} - K_a F - G_{aai} \tag{3.37}$$

While,

$$\mathbf{\Gamma} = \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{pmatrix}, \quad \ddot{\theta} = \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} \tag{3.38}$$

$$\mathbf{I}_a = \begin{pmatrix} I_{a1} & 0 & 0 \\ 0 & I_{a2} & 0 \\ 0 & 0 & I_{a3} \end{pmatrix}, \quad \mathbf{K_a} = \frac{L_A}{L_B} \cdot \begin{pmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{pmatrix} \tag{3.39}$$

With

$$K_i = z_i \cdot \cos\theta_i - (R - x_i) \cdot \sin\theta_i, \quad i = 1, 2, 3 \tag{3.40}$$

$$\mathbf{G}_{aai} = \begin{pmatrix} r_a \cdot m_a \cdot g \cdot \cos\theta_1 \\ r_a \cdot m_a \cdot g \cdot \cos\theta_2 \\ r_a \cdot m_a \cdot g \cdot \cos\theta_3 \end{pmatrix} \tag{3.41}$$

At least, we replace the expression of the force according the equation (3.37);

$$\mathbf{\Gamma} = \mathbf{I}_a\ddot{\theta} - \mathbf{K}_a\mathbf{K}_n\ddot{X} + \mathbf{K}_a\mathbf{G}_n - \mathbf{G}_{aai} \tag{3.42}$$

Given the Jacobian matrix J:

$$J = \begin{bmatrix} \frac{\partial X}{\partial \theta_1} & \frac{\partial X}{\partial \theta_2} & \frac{\partial X}{\partial \theta_3} \\ \frac{\partial Y}{\partial \theta_1} & \frac{\partial Y}{\partial \theta_2} & \frac{\partial Y}{\partial \theta_3} \\ \frac{\partial Z}{\partial \theta_1} & \frac{\partial Z}{\partial \theta_2} & \frac{\partial Z}{\partial \theta_3} \end{bmatrix} \tag{3.43}$$

We substitute this into the equation (3.42) by the relation :

$$\ddot{X} = J'\dot{\theta} + J\ddot{\theta} \tag{3.44}$$

$$\mathbf{\Gamma} = (\mathbf{I}_a - \mathbf{K}_a\mathbf{K}_n J)\ddot{\theta} - (\mathbf{K}_a\mathbf{K}_n\dot{J})\dot{\theta} + \mathbf{K}_a\mathbf{G}_n - \mathbf{G}_{aai} \tag{3.45}$$

$$\mathbf{\Gamma} = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta)\dot{\theta} + \mathbf{G}(\theta) \tag{3.46}$$

3.4 State Space Model of Delta Robot

The state-space representation is a mathematical framework used to describe the dynamics of a dynamic system over time. It is particularly useful for modeling and analyzing systems, defined from the differential equation governing them.

$$\dot{\mathbf{x}}(t) = f(x, u)$$
$$\mathbf{y}(t) = h(x)$$

- **State vector ($\mathbf{x}(t)$)**: Represents the internal state of the system at time t.
- **Input vector ($\mathbf{u}(t)$)**: Represents inputs applied to the system at time t.
- **Output vector ($\mathbf{y}(t)$)**: Represents measurable outputs of the system at time t.

Based on the equations governing the dynamic model of the robot, which consist of three second-order differential equations, we define the state vector x(t) of order 6 as follows:

$$\mathbf{x} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 \end{bmatrix}^T$$

Also, the input vector is;

$$u = \begin{pmatrix} \Gamma_1 & \Gamma_2 & \Gamma_3 \end{pmatrix}^T$$

Hence, the state space representation of the delta robot is given by ;

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \mathbf{M}(\theta)^{-1}\left(\mathbf{\Gamma} - \mathbf{C}(\theta)\dot{\theta} - \mathbf{G}(\theta)\right) \end{bmatrix} \tag{3.47}$$

3.5 Fractional Order state space model for delta robot

Fractional order modeling is essential to accurately capture the dynamic behavior of the Delta robot, as it accounts for the effects of memory and nonlocal properties inherent in robotic systems. Unlike integer-order models, fractional derivatives provide enhanced flexibility in representing system dynamics [31, 32]. This section proposes a fractional-order state-space representation of the Delta robot, offering a more precise framework for control design.

The state vector for the fractional-order model is defined as:

$$\mathbf{x} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & D^\alpha\theta_1 & D^\alpha\theta_2 & D^\alpha\theta_3 \end{bmatrix}^T$$

Hence, the fractional-order state-space representation of the Delta robot is given by:

$$D^\alpha\mathbf{x}(t) = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \mathbf{M}(\theta)^{-1}\left(\mathbf{\Gamma} - \mathbf{C}(\theta)D^\alpha\theta - \mathbf{G}(\theta)\right) \end{bmatrix}$$

Where;

- The non-integer order α is set to 0.8 to closely approximate the integer-order model while capturing the intrinsic fractional dynamics of robots.

3.6 Open loop simulation

In order to analyze the dynamic behavior of the delta robot and assess the impact of model order on system response, an open-loop simulation is conducted for both the integer-order and fractional-order models. Open-loop simulation, by definition, refers to system analysis without the use of feedback control. This method is particularly useful in identifying inherent dynamic characteristics such as rise time, settling time, oscillatory behavior, and system stability.

From Figure 3.12, we observe that the robot model is highly unstable. When a torque of 5 Nm is applied at the input, the joint dynamics diverge and tend toward infinity within a few seconds.

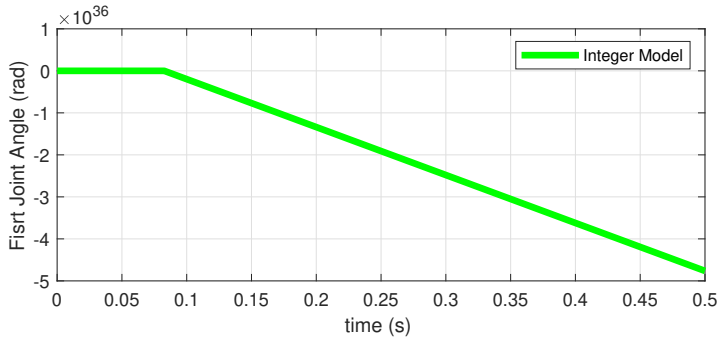


Figure 3.12: Open loop first joint response

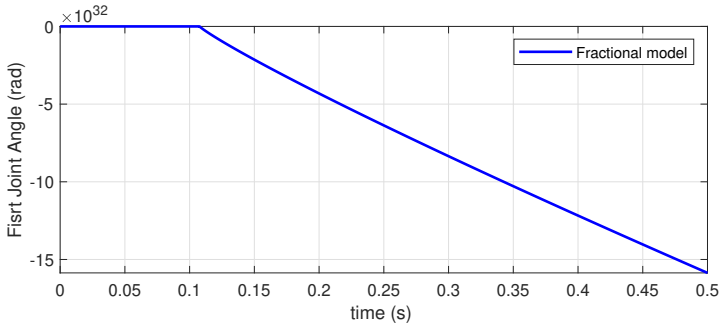


Figure 3.13: Open loop first joint response

Figure 3.13 shows the response of the proposed fractional model under the same conditions. Indeed, the behavior was practically similar to that of the integer model in terms of divergence and instability.

3.7 Advanced Fractional Control Strategies For Delta Robot

This section provides an overview of recent advanced Fractional control strategies applied to delta robots, including both model-based and data-driven approaches. By analyzing various techniques such as adaptive control, robust control, and intelligent controllers.

Fractional-Order classical Controller

- This study [33] proposes the implementation of a tracking control for a Delta-type parallel robot using fractional-order PID controllers in conjunction with the computed torque control strategy. The performance is contrasted with integer-order PID controllers, demonstrating improved robustness and disturbance rejection.

- This paper [34] deals with piezoelectric actuators control. A fractional order fuzzy PID controller is designed for this class of systems with the help of particle swarm optimization (PSO) algorithm.

- This thesis [3] presents a global modeling for the Delta robot ISIR88, considering several control structures including PD, fractional-order PD, and Time Delay Control (TDC). The study involves both simulation and experimentation, demonstrating the robustness of these control laws in managing modeling uncertainties.

Fractional-Order Adaptive Controllers

- The Adaptive Backstepping Fractional-Order Non-Singular Terminal Sliding Mode Control proposed by [35] combines adaptive backstepping with fractional-order non-singular terminal sliding mode control to achieve precise trajectory tracking of Delta parallel robots. The approach addresses model uncertainties and external disturbances by employing an adaptive law that estimates and compensates for these uncertainties in real-time.

- The Adaptive Super-Twisting Fractional-Order Sliding Mode Control proposed by [36] introduces an adaptive super-twisting fractional-order sliding mode control approach for redundantly actuated cable-driven parallel robots. The controller incorporates a fractional-order sliding surface and an adaptive law to estimate and compensate for uncertainties and disturbances.

- This paper introduces [37] a Fractional Order Model Reference Adaptive Control (FOMRAC) strategy for trajectory tracking of SCARA-type robotic manipulators. The approach integrates fractional-order systems into the classical MRAC framework to enhance the system's robustness and adaptability to model uncertainties and external disturbances. The proposed controller demonstrates improved tracking performance and stability compared to traditional integer-order MRAC methods..

Learning based Fractional-Order Intelligent Controllers

- This study presents [38] n inverse kinematic controller using neural networks for trajectory controlling of a delta robot in real-time. The developed control scheme is purely data-driven and does not require prior knowledge of the delta robot kinematics. Moreover, it can adapt to the changes in the kinematics of the robot. For developing the controller, the kinematic model of the delta robot is estimated by using neural networks. Then, the trained neural networks are configured as a controller in the system. The parameters of the neural networks are updated while the robot follows a path to adaptively compensate for modeling uncertainties and external disturbances of the control system.

3.8 Conclusion

In this chapter, we analyzed the modeling of the Delta robot using both integer and fractional-order representations. The open-loop simulations revealed high instability in both models, highlighting the need for advanced control strategies. A review of the literature showed increasing interest in fractional and intelligent adaptive control methods for managing complex robot dynamics and uncertainties

Chapter 4

FO Indirect Lyapunov Based Adaptive Backstepping Controller

4.1 Introduction

This chapter aims at designing a Fractional Order Lyapunov-based Indirect Adaptive Backstepping Control strategy for the precise trajectory tracking of a DELTA robot. The proposed approach integrates fractional calculus with Lyapunov stability theory to enhance the robustness and adaptability of the controller in the presence of system uncertainties and external disturbances. Unlike conventional backstepping controllers [39] , the indirect adaptive mechanism estimates unknown system parameters online, improving control performance without requiring precise dynamic modeling.

4.2 Problem Formulation

The objective of this strategy is to design a control law for a Delta robot capable of accurately performing trajectory tracking tasks in the presence of undesired dynamics (uncertainties and external disturbances, and the resistive torques applied by the load).

The control design will be based on the fractional-order state-space model [40] introduced in the previous chapter, defined as follows:

$$\mathbf{x_1} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \tag{4.1}$$

$$\mathbf{x_2} = \begin{bmatrix} D^\alpha\theta_1 & D^\alpha\theta_2 & D^\alpha\theta_3 \end{bmatrix}^T \tag{4.2}$$

$$D^\alpha \mathbf{x}(t) = \begin{bmatrix} x_2 \\ M(x_1)^{-1}(\mathbf{u} - C(x_1, x_2)x_2 - G(x_1) + \Delta(t)) \end{bmatrix} \tag{4.3}$$

Where;

- $\Delta(t)$ represents the unknown dynamics to be estimated and compensated by the controller.

4.3 Control law design

We first define the tracking error vector:

$$e_i(t) = x_i(t) - x_{ir}(t), i = 1, 2 \tag{4.4}$$

The estimation error is given by,

$$\tilde{\Delta}(t) = \Delta(t) - \hat{\Delta}(t) \tag{4.5}$$

Assuming the slow dynamics of uncertainties ,

$$D^{\alpha}\Delta(t) \approx 0 \tag{4.6}$$

Considering the plant described by Eq. (4.3), with order $0 < \alpha < 1$, *a positive gain* $\lambda > 0$, *and a symmetric positive definite matrix* $\Gamma > 0$. *we propose the respectively virtual, adaptation, and global following control laws:*

$$u_v = -\lambda e_1(t) + D^{\alpha}x_{1r} \tag{4.7}$$

$$D^{\alpha}\hat{\Delta}(t) = \Gamma^{-1}M^{-T}e_2 \tag{4.8}$$

$$\mathbf{u} = M\left(D^{\alpha}u_v - e_1 - \lambda e_2\right) + CD^{\alpha}x_1 + G - \hat{\Delta}(t) \tag{4.9}$$

4.4 Proof and Stability Analysis

Let consider the following lyapunov candidate function:

$$V_1(t, e_1(t)) = \frac{1}{2}e_1(t)^T e_1(t) \tag{4.10}$$

According to Lemma 1 , which provides an upper bound on the fractional derivative of a quadratic form [22] , we have:

$$D^{\alpha}V_1 \leq e_1(t)^T D^{\alpha}e_1(t) \tag{4.11}$$

To ensure that $D^{\alpha}V_1 < 0$, we design the virtual control such that:

$$e_1(t)^T D^{\alpha}e_1(t) < 0 \tag{4.12}$$

This leads to:

$$e_1(t)^T D^{\alpha}e_1(t) = e_1(t)^T \cdot (D^{\alpha}x_1 - D^{\alpha}x_{1r}) \tag{4.13}$$

Then we propose to choose the following virtual control signal to stabilize the position error ,

$$u_v = -\lambda e_1(t) + D^\alpha x_{1r} \tag{4.14}$$

where $\lambda > 0$ is a positive control gain.

To further analyze stability, we define the following composite Lyapunov function [41] :

$$V_2 = V_1 + \frac{1}{2} e_2(t)^T e_2(t) + \frac{1}{2} \tilde{\Delta}(t)^T \Gamma \tilde{\Delta}(t) \tag{4.15}$$

where Γ is a symmetric positive definite matrix.

According to Lemma 1, the fractional derivative of V_2 satisfies [23] :

$$D^\alpha V_2 \leq e_1^T D^\alpha e_1 + e_2^T D^\alpha e_2 + \tilde{\Delta}(t)^T \Gamma D^\alpha \tilde{\Delta}(t) \tag{4.16}$$

According to (4.6) :

$$D^\alpha \tilde{\Delta}(t) = -D^\alpha \hat{\Delta}(t) \tag{4.17}$$

We aim to design the control input $\mathbf{u}$ to make $D^\alpha V_2 < 0$. From the previous results:

$$e_1^T D^\alpha e_1 = e_1^T(-\lambda e_1 + e_2) \tag{4.18}$$

Substituting into the expression for $D^\alpha V_2$, we obtain:

$$D^\alpha V_2 \leq -\lambda \|e_1\|^2 + e_2^T(e_1 + D^\alpha e_2) - \tilde{\Delta}(t)^T D^\alpha \hat{\Delta}(t) \tag{4.19}$$

The derivative $D^\alpha e_2$ can be computed as:

$$D^\alpha e_2 = M(x_1)^{-1}\left(\mathbf{u} - Cx_2 - G + \Delta(t)\right) - D^\alpha u_v \tag{4.20}$$

The proposed control law is then:

$$\mathbf{u} = M\left(D^\alpha u_v - e_1 - \lambda e_2\right) + Cx_2 + G - \hat{\Delta}(t) \tag{4.21}$$

Substituting this into the expression for $D^\alpha V_2$, we get:

$$D^\alpha V_2 \leq -\lambda \|e_1\|^2 - \lambda \|e_2\|^2 + e_2^T M^{-1} \tilde{\Delta}(t) - \tilde{\Delta}(t)^T \Gamma D^\alpha \hat{\Delta}(t) \tag{4.22}$$

To eliminate the disturbance term, acording to [41] we define the adaptation law as:

$$D^\alpha \hat{\Delta}(t) = \Gamma^{-1} M^{-T} e_2 \tag{4.23}$$

Thus, the final expression becomes:

$$D^\alpha V_2 \leq -\lambda \|e_1\|^2 - \lambda \|e_2\|^2 < 0 \tag{4.24}$$

This ensures the exponential stability of the closed-loop system, in accordance with Lemma 1 using the extended Lyapunov stability theorem.

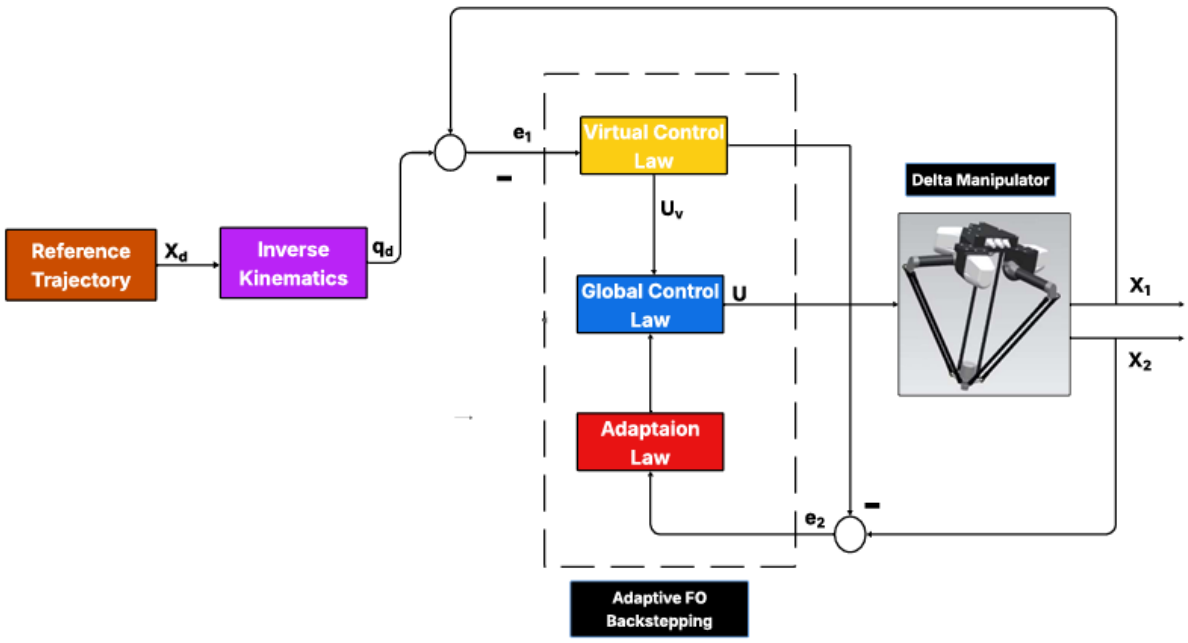


Figure 4.1: Structure of the adaptive FO Backstepping control system

4.5 Simulation results

To validate the effectiveness of the proposed fractional-order backstepping control strategy, simulations (For $\alpha = 0.8$) are performed in MATLAB/Simulink. The Delta robot's end-effector is tasked with a sinusoidal trajectories on it's joints angular position.This trajectory is carefully designed to ensure smooth motion, along with continuous velocity and acceleration profiles, which are crucial for precise tracking and stability analysis.

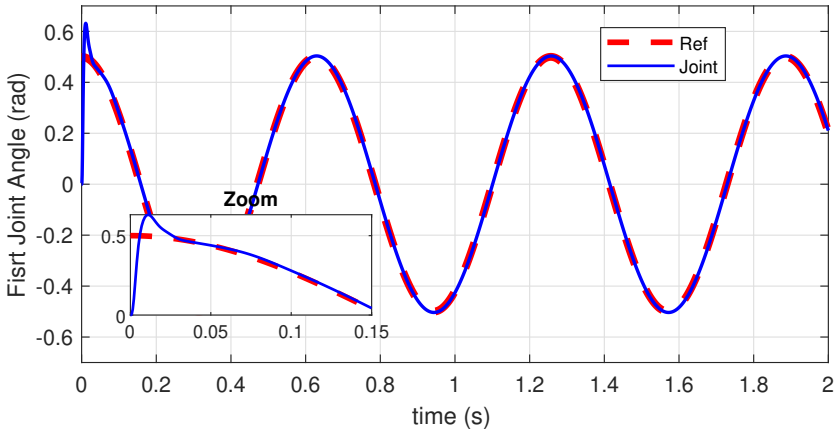


Figure 4.2: First Joint angle

By observing the joint position curve in Fig. 4.1, 4.2 and 4.3, we notice a good tracking of the reference. The FO Backstepping controller delivered excellent performance with a significantly low rise time (about 0.05 s), highlighting the stability of the closed-loop system.

The tracking performance of the system can also be evaluated by analyzing the curves along the 3d space trajectory, in Fig.4.4. As observed, the end-effector response exhibits excellent

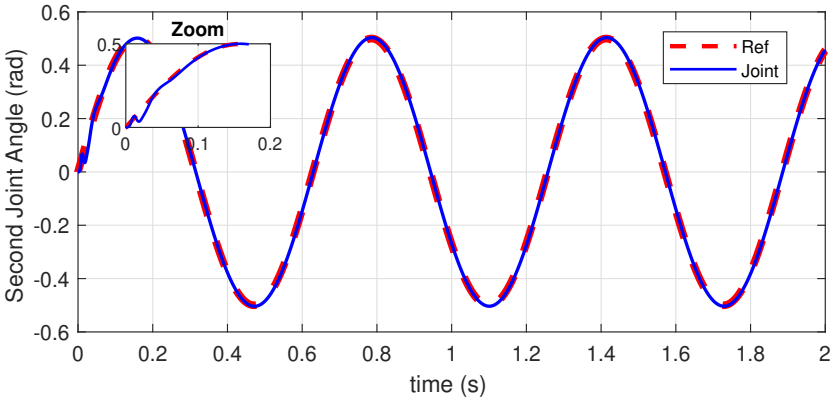


Figure 4.3: Second Joint angle

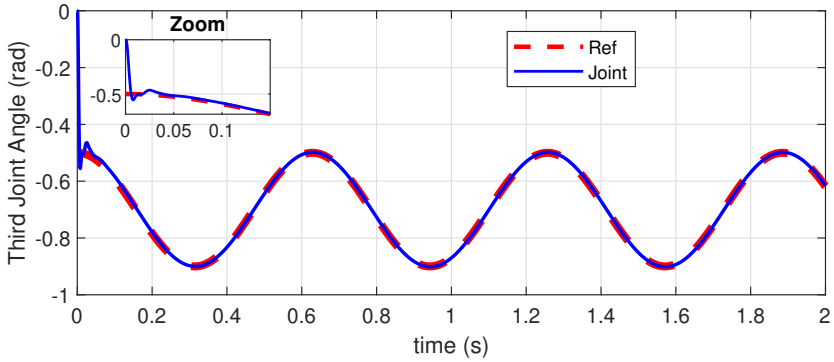


Figure 4.4: Third Joint angle

tracking of the reference trajectory, demonstrating the effectiveness of the control strategies implemented.

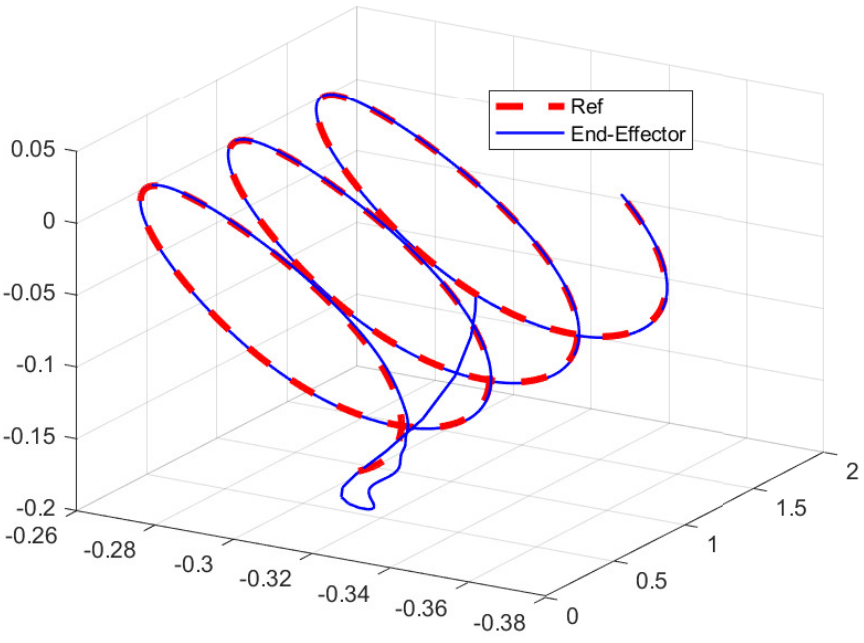


Figure 4.5: End effector trajectory

To analyze the performances of the control law, the root mean square error (RMSE) of the

trajectories for both controllers is summarized in Table 7.1.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ir} - x_i)^2} \tag{4.25}$$

Table 4.1: RMSE values

Controller	RMSE (x-axis)	RMSE (y-axis)	RMSE (z-axis)
FO Backstepping	1.8×10^{-4}	3.2×10^{-10}	6.4×10^{-5}

The RMSE values highlight the tracking accuracy of the control law along the x, y, and z axes, confirming that the control objective is achieved.

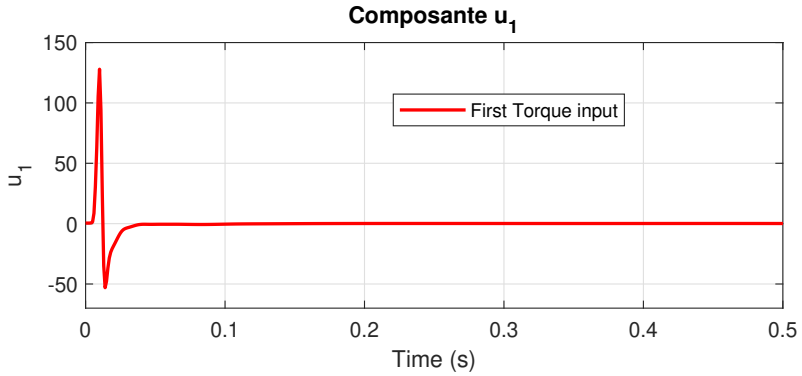


Figure 4.6: Induced Control Signal

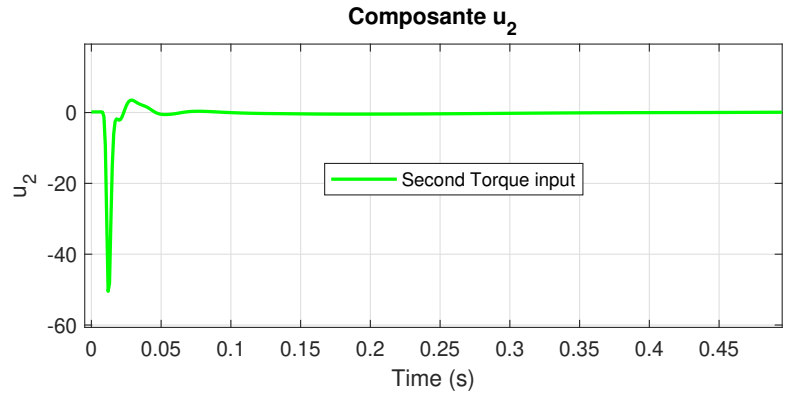


Figure 4.7: Induced Control Signal

The figures 4.5, 4.6 and 4.7 illustrate the torque evolution over time for the FO Backstepping controller. Initially, a significant torque overshoot is observed during the transient phase due to the influence of the fractional dynamics on rise time and performance. However, it is rapidly stabilized around 0.2 in the steady state.

4.6 Robustness Tests

This section aims to examine the robustness of the controller against uncertainties, undesired dynamics, and perturbations. This study is conducted in two steps: first, by injecting an external perturbation signal, and second, by introducing an uncertain parameter variation

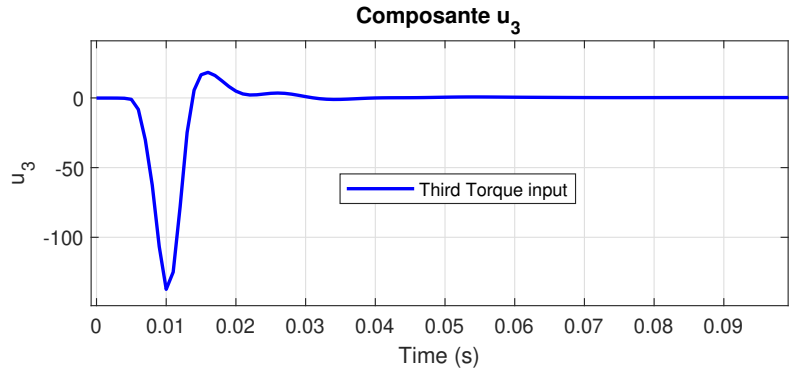


Figure 4.8: Induced Control Signal

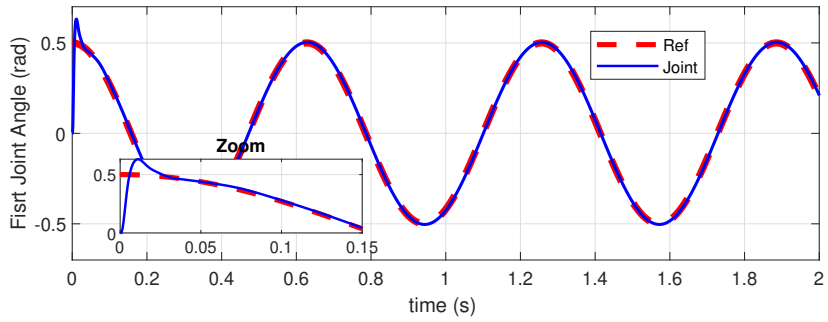


Figure 4.9: Joint angle In presence of uncertainty

Figure 4.8 shows the joint angle response in the presence of bounded uncertainty. Due to the adaptation term in the controller, the additive perturbation is estimated and compensated. This explains the good performance in terms of tracking and stability.

We now evaluate the controller's ability to reject external disturbances, such as those arising from sensor noise or environmental factors. As illustrated in Figures 4.9 and 4.10, the system response remains stable and converges effectively, confirming the robustness of the control law, which benefits from the inherent memory effect of fractional-order systems.

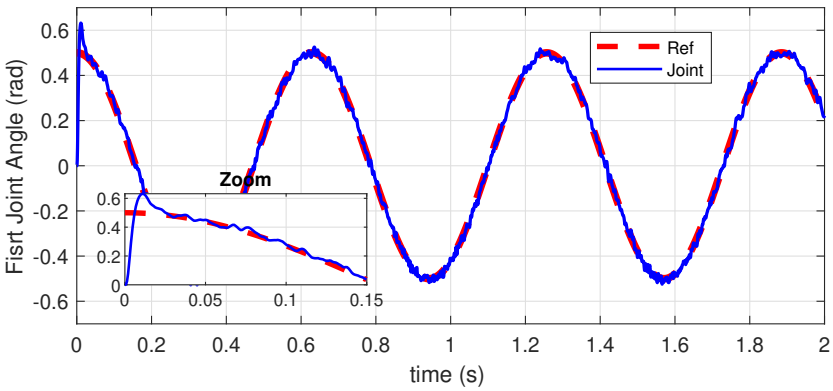


Figure 4.10: Joint angle In presence of uncertainty

At least, we will examine the controller's ability to stabilize the system in the presence of nonlinear uncertain parameters. The system is then considered as:

$$D^{\alpha}\mathbf{x}(t) = \begin{bmatrix} x_2 \\ (M(x_1) + \Delta M)^{-1}(\mathbf{u} - (C(x_1, x_2) + \Delta C)x_2 - G(x_1)) \end{bmatrix} \tag{4.26}$$

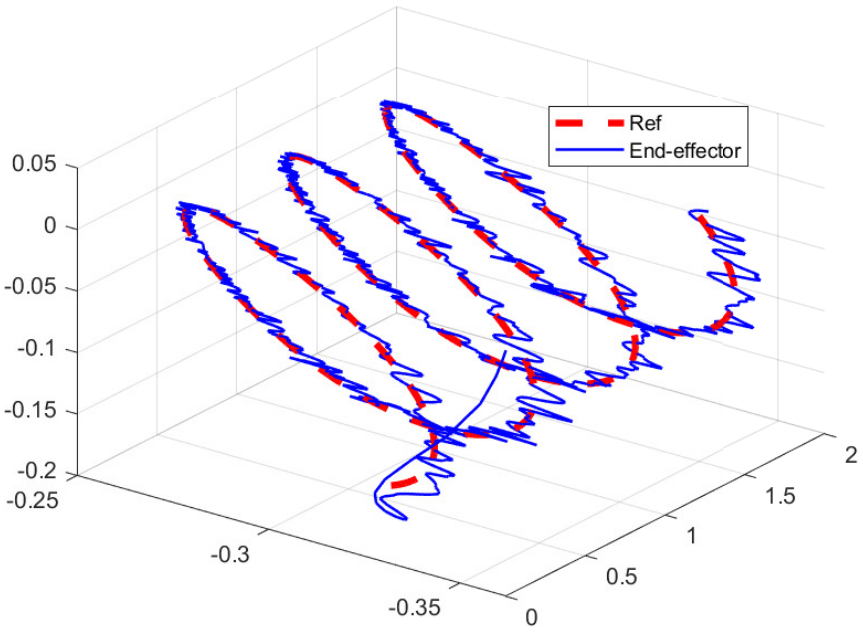


Figure 4.11: 3D End-Effector Trajectory Tracking under noise

Where , ΔM and ΔC are constant bounded uncertainties.

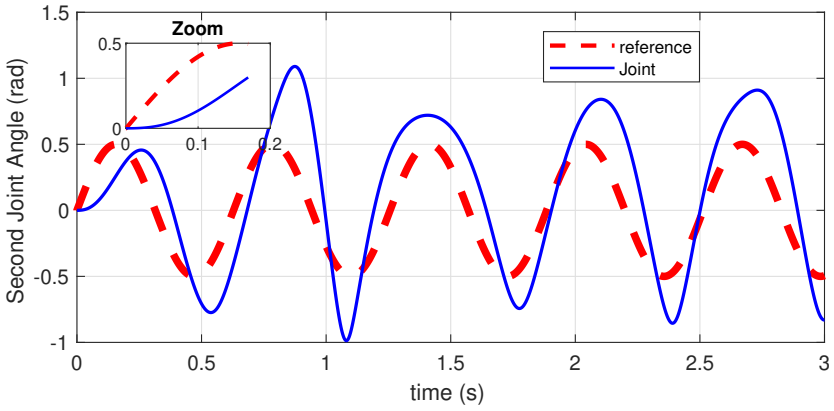


Figure 4.12: Joint angle In presence of uncertainty

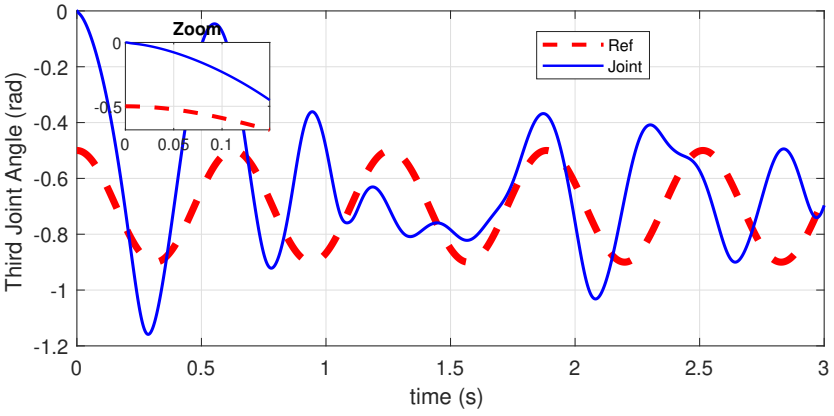


Figure 4.13: Joint angle In presence of uncertainty

The impact of nonlinear modeling errors is illustrated in Figures 4.11 and 4.12. Although the system maintains overall stability, a significant degradation in performance is observed, particularly in terms of accurate tracking of the reference trajectory. This highlights the sensitivity of the fractional-order backstepping controller to modeling inaccuracies, as the absence of adaptation mechanisms limits its ability to compensate for nonlinear uncertainties.

4.7 Conclusion

In this chapter, an adaptive fractional-order backstepping controller was proposed for the Delta robot. The simulation results demonstrated good overall performance, particularly in terms of robustness and adaptability to external perturbations. The fractional-order design introduced a relatively high rise time, which is a known characteristic of such systems, but it also contributed to smoother control actions. While the controller effectively compensates for bounded uncertainties and partially known dynamics, it shows limitations when dealing with systems with completely unknown dynamics. Therefore, its application is best suited for scenarios where partial system knowledge is available.

Chapter 5

FO Sliding Mode Control Based on Super-Twisting Algorithm

5.1 Introduction

This chapter aims at designing a robust Fractional Order Sliding Mode Control (FO-SMC) strategy using the Super-Twisting Algorithm for the precise trajectory tracking of a DELTA robot. The approach [42] integrates fractional calculus with Lyapunov stability theory to enhance robustness and convergence in the presence of modeling uncertainties and external disturbances. Unlike classical SMC, [28] the super-twisting mechanism reduces chattering and introduces more flexible tuning. This controller is particularly beneficial in dealing with strong nonlinearities and uncertainty that commonly affect parallel robot dynamics.[43, 27]

5.2 Problem Formulation

The objective of this strategy is to design a control law for a Delta robot capable of accurately performing trajectory tracking tasks in the presence of undesired dynamics (uncertainties, external disturbances, and the resistive torques applied by the load).[44]

The control design is based on the following fractional-order state-space model:

$$\mathbf{x_1} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \tag{5.1}$$

$$\mathbf{x_2} = \begin{bmatrix} D^\alpha\theta_1 & D^\alpha\theta_2 & D^\alpha\theta_3 \end{bmatrix}^T \tag{5.2}$$

$$D^\alpha \mathbf{x}(t) = \begin{bmatrix} M(\mathbf{x_1})^{-1}(\mathbf{u} - C(\mathbf{x_1}, \mathbf{x_2})\mathbf{x_2} - G(\mathbf{x_1}) + \Delta(t)) \end{bmatrix} \tag{5.3}$$

Where represents the unknown disturbance and model uncertainty to be estimated and compensated. [45][46]

5.3 Control Law Design

We define the tracking error vector:

$$\mathbf{e}(t) = \mathbf{x_1}(t) - \mathbf{x}_{1r}(t) \tag{5.4}$$

Then we define the sliding surface:

$$\mathbf{S}(t) = \lambda \mathbf{e}(t) + D^{\alpha}\mathbf{e}(t) \tag{5.5}$$

Let the estimation error be:

$$\tilde{\Delta}(t) = \Delta(t) - \hat{\Delta}(t) \tag{5.6}$$

Assuming slow variation of the uncertainties:

$$D^{\alpha}\Delta(t) \approx 0 \tag{5.7}$$

We consider the following Lyapunov candidate:

$$V(t) = \frac{1}{2}\mathbf{S}^T\mathbf{S} + \frac{1}{2}\tilde{\Delta}^T\Gamma\tilde{\Delta} \tag{5.8}$$

Then its derivative is:

$$D^{\alpha}V(t) \leq \mathbf{S}^T D^{\alpha}\mathbf{S} + \tilde{\Delta}^T\Gamma D^{\alpha}\tilde{\Delta} \tag{5.9}$$

We impose the following super-twisting law:

$$D^{\alpha}\mathbf{S} = -\lambda_1 \operatorname{sign}(\mathbf{S}) \cdot |\mathbf{S}|^{1/2} - \lambda_2\mathbf{S} + M^{-1}\tilde{\Delta}(t) \tag{5.10}$$

Then the control law is:

$$\mathbf{u} = M\left(D^{2\alpha}\mathbf{x}_{1r} - \lambda D^{\alpha}\mathbf{e} - \lambda_1 \operatorname{sign}(\mathbf{S}) \cdot |\mathbf{S}|^{1/2} - \lambda_2\mathbf{S}\right) + C\mathbf{x_2} + G - \hat{\Delta}(t) \tag{5.11}$$

The adaptation law is:

$$D^{\alpha}\hat{\Delta}(t) = \Gamma^{-1}M^T\mathbf{S} \tag{5.12}$$

Substituting and simplifying:

$$D^{\alpha}V(t) \leq -\lambda_1||\mathbf{S}||^{3/2} - \lambda_2||\mathbf{S}||^2 < 0 \tag{5.13}$$

Summary: *Considering the plant described by Eq. (5.3), with order* $0 < \alpha < 1$*, and positive gains* $\lambda > 0$*,* $\lambda_1 > 0$*, and* $\lambda_2 > 0$*, we propose the following:*

$$\mathbf{S}(t) = \lambda \mathbf{e}(t) + D^{\alpha}\mathbf{e}(t) \tag{5.14}$$

$$D^{\alpha}\hat{\Delta}(t) = \Gamma^{-1}M^T\mathbf{S} \tag{5.15}$$

$$\mathbf{u} = M\left(D^{2\alpha}\mathbf{x}_{1r} - \lambda D^{\alpha}\mathbf{e} - \lambda_1 \operatorname{sign}(\mathbf{S})|\mathbf{S}|^{1/2} - \lambda_2\mathbf{S}\right) + C\mathbf{x_2} + G - \hat{\Delta}(t) \tag{5.16}$$

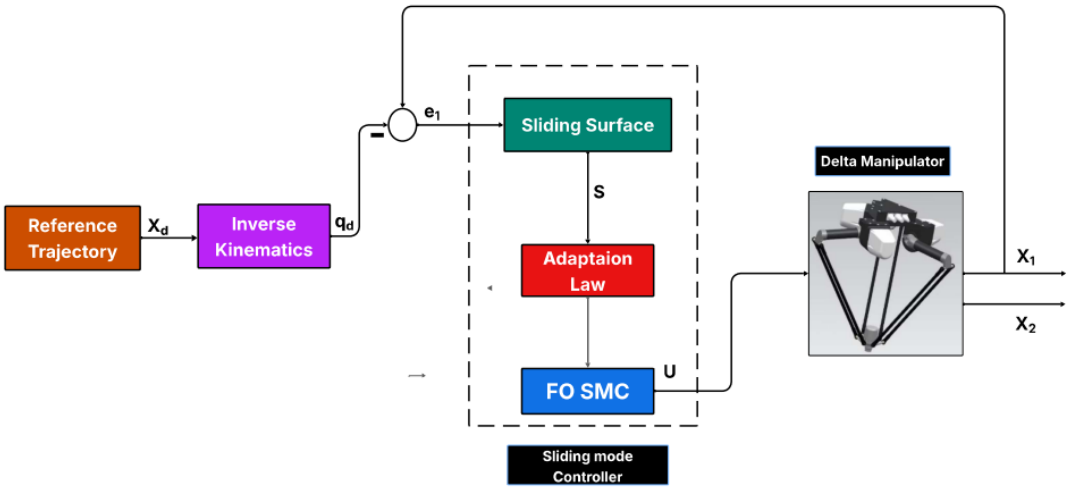


Figure 5.1: Structure of the adaptive Sliding Mode FO control system

5.4 Simulation and Discussions

To validate the effectiveness of the proposed fractional-order Sliding Mode control strategy, simulations (For $\alpha = 0.8$) are performed in MATLAB/Simulink. The Delta robot's end-effector is tasked with a cubic trajectories on it's joints angular position.This trajectory is carefully designed to ensure smooth motion, along with continuous velocity and acceleration profiles, which are crucial for precise tracking and stability analysis.[43][27]

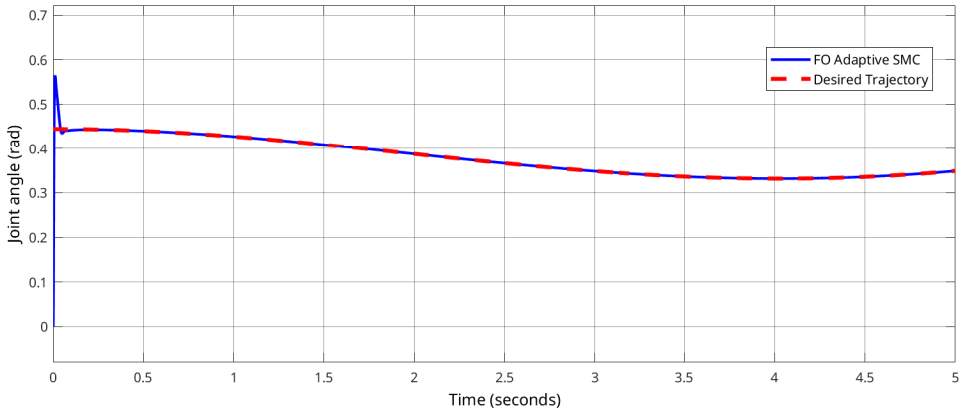


Figure 5.2: First Joint angle

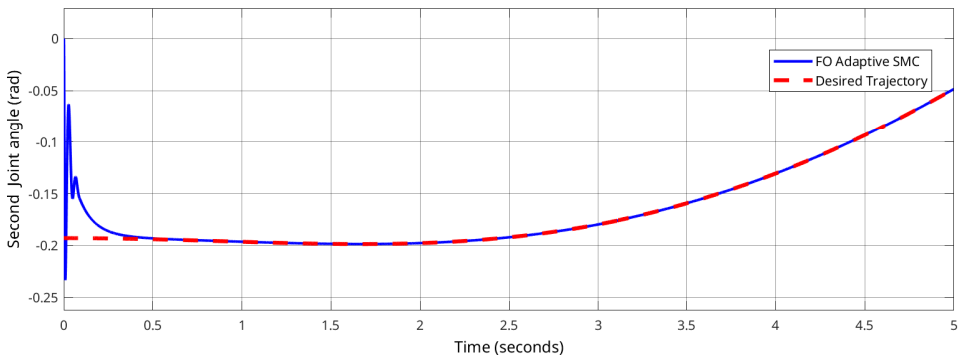


Figure 5.3: Second Joint angle

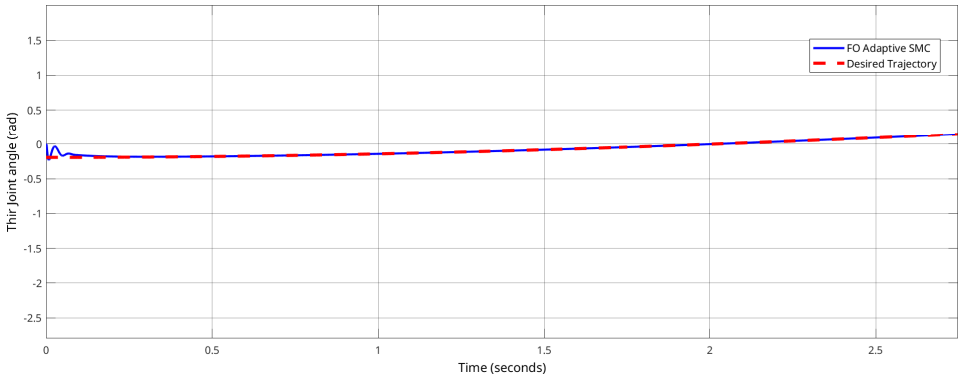


Figure 5.4: Third Joint angle

By observing the joint position curve in Fig. 5.2, 5.3 and 5.4, we notice a virtually perfect tracking of the reference trajectories (dashed) by the actual joint responses (solid). In each case the fractional-order sliding-mode controller delivers excellent performance, with a remarkably low rise time of about 0.25 s, negligible overshoot and virtually zero steady-state error. Such behavior highlights the robustness and stability of the closed-loop system, as well as the high level of precision typically required for advanced robotic applications.

The tracking performance can be further appreciated in the 3D end-effector plot of Fig. 5.5. Here, the actual path (solid line) coincides almost exactly with the desired spatial trajectory (dashed line), confirming the effectiveness of the fractional super-twisting sliding-mode strategy in coordinating all three joints to steer the end-effector along complex three-dimensional paths.

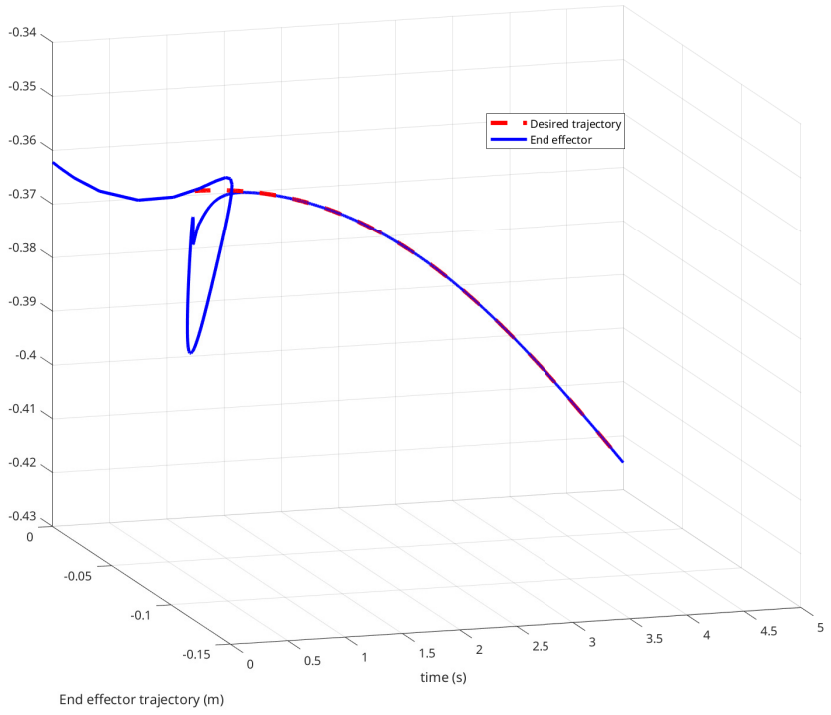


Figure 5.5: 3D End-Effector Trajectory Tracking

To analyze the performances of the control law, the root mean square error (RMSE) of the trajectories for both controllers is summarized in Table 7.1.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ir} - x_i)^2} \tag{5.17}$$

Table 5.1: RMSE values

Controller	RMSE (x-axis)	RMSE (y-axis)	RMSE (z-axis)
FO Sliding mode	9.96×10^{-4}	2.24×10^{-10}	1.02×10^{-5}

The RMSE values highlight the tracking accuracy of the control law along the x, y, and z axes, confirming that the control objective is achieved.

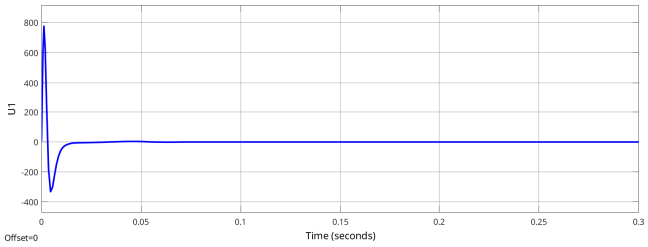


Figure 5.6: Induced Control Signal

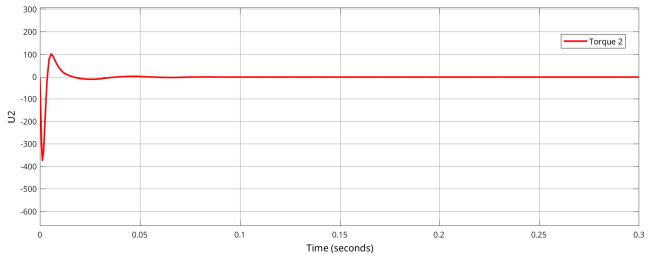


Figure 5.7: Induced Control Signal

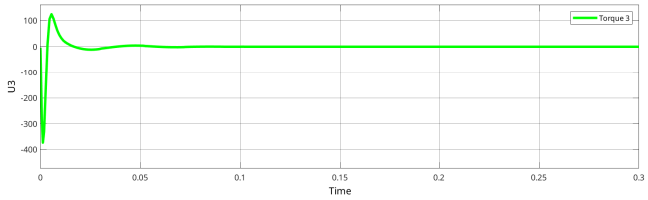


Figure 5.8: Induced Control Signal

- **High initial overshoot:**
  - Joint 1 (Fig. 5.6) peaks at approximately +800N.m and dips to −300N.m within the first 0.01 s.
  - Joint 2 (Fig. 5.7) reaches about +120N.m before settling.
  - Joint 3 (Fig. 5.8) attains roughly +100N.m at its maximum.

  This transient spike is due to the fractional dynamics acting on rise-time performance.

- **Rapid stabilization:** All three torques converge to a small steady-state value (0.2 s) in under 0.05s, demonstrating excellent damping and disturbance rejection.

- **Absence of chattering:** Thanks to the Super-Twisting algorithm, the control signals are completely smooth—no high-frequency oscillations (chattering) are observed in any torque trace.

These results confirm that the proposed controller delivers both *high-bandwidth* performance (fast rise, controlled overshoot) and *robust, chatter-free* operation, which are essential for precision robotic tasks.

5.5 Robustness Tests

This section aims to examine the robustness of the controller against uncertainties, undesired dynamics, and perturbations. This study is conducted in two steps: first, by injecting an external perturbation signal, and second, by introducing an uncertain parameter variation

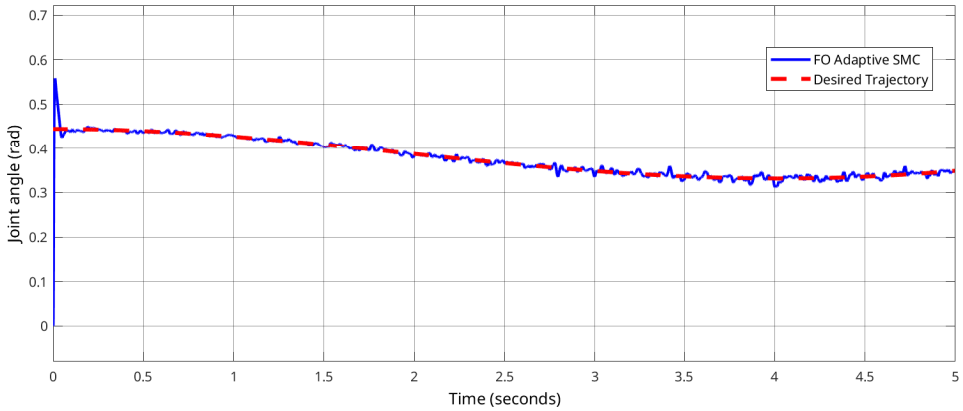


Figure 5.9: Joint angle In presence of noise

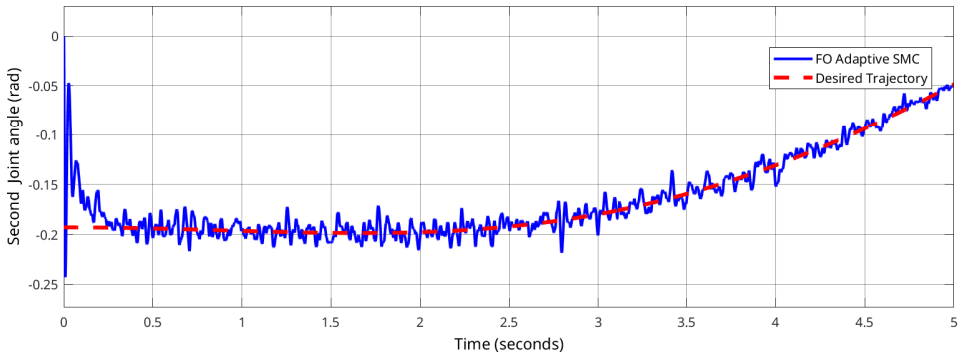


Figure 5.10: Joint angle In presence of noise

Joint 1 and 2 angle in the presence of uncertainty. Despite the injected perturbations, the FO–Adaptive SMC (blue) closely follows the desired trajectory (red), with only small oscillations (± 0.01 rad) around the set-point. This demonstrates the controller's ability to reject disturbances almost immediately.

At least, we will examine the controller's ability to stabilize the system in the presence of nonlinear uncertain parameters. The system is then considered as:

$$D^{\alpha}\mathbf{x}(t) = \begin{bmatrix} x_2 \\ (M(x_1) + \Delta M)^{-1}(\mathbf{u} - (C(x_1, x_2) + \Delta C)x_2 - G(x_1)) \end{bmatrix} \tag{5.18}$$

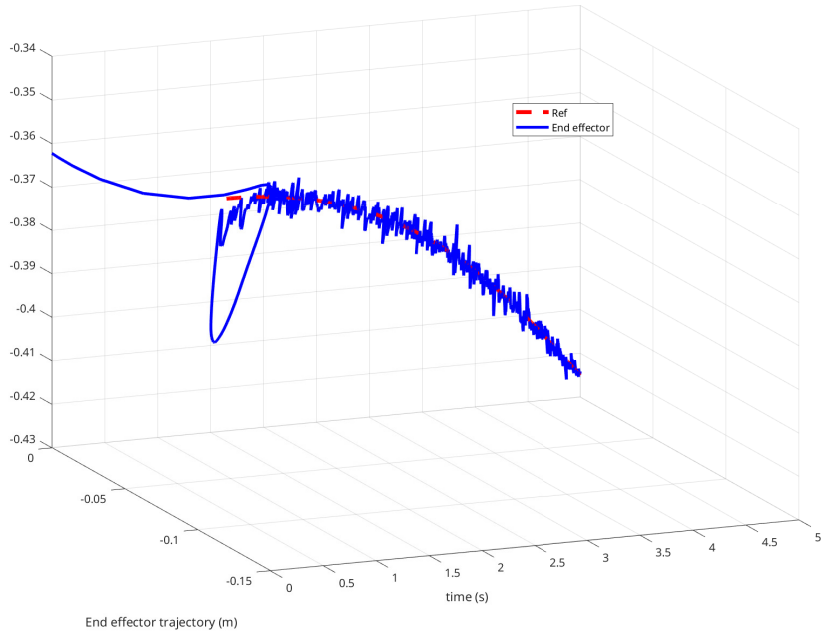


Figure 5.11: 3D End-Effector Trajectory Tracking under noise

Where , ΔM and ΔC are constant bounded uncertainties.

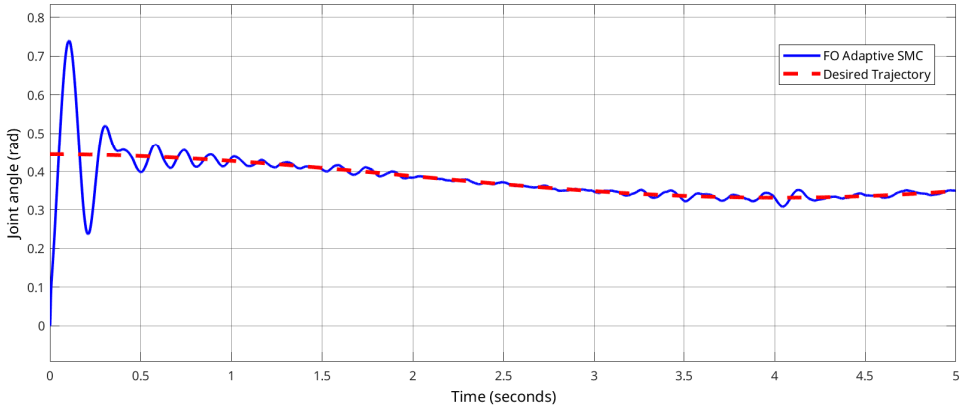


Figure 5.12: Joint angle In presence of uncertainty

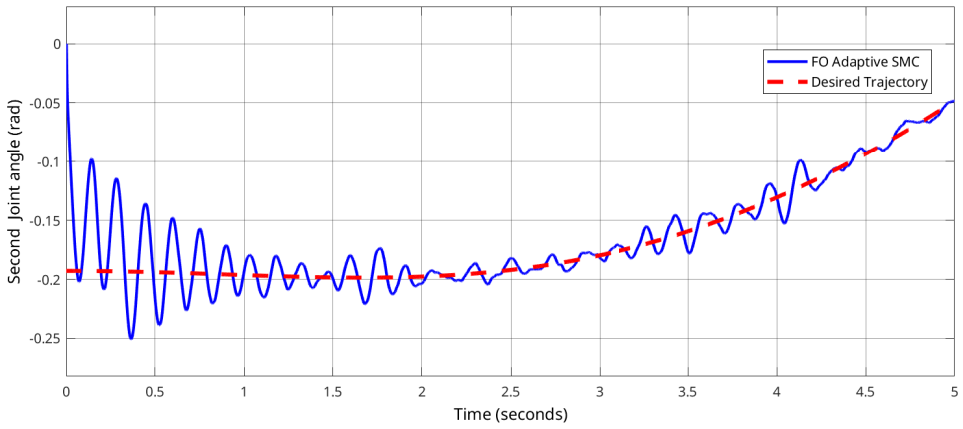


Figure 5.13: Joint angle In presence of uncertainty

The impact of nonlinear modeling errors is illustrated in Figures 5.11 and 5.12. Although the system maintains overall stability, a significant degradation in performance is observed,

particularly in terms of accurate tracking of the reference trajectory. This highlights the sensitivity of the fractional-order backstepping controller to modeling inaccuracies, as the absence of adaptation mechanisms limits its ability to compensate for nonlinear uncertainties.

5.6 Conclusion

In this chapter, an Fractional Order Sliding Mode controller was proposed for the Delta robot. The simulation results demonstrated good overall performance, particularly in terms of robustness and adaptability to external perturbations. The fractional-order design introduced a relatively high rise time, which is a known characteristic of such systems, but it also contributed to smoother control actions. While the controller effectively compensates for bounded uncertainties and partially known dynamics, it shows limitations when dealing with systems with completely unknown dynamics. Therefore, its application is best suited for scenarios where partial system knowledge is available.[27]

Chapter 6

Fractional Order Lyapunov based MRAC with State feedback Controller

6.1 Introduction

This chapter gives the design of a Fractional Order direct adaptive model reference [47] Control strategy for the precise tracking of reference model for the DELTA robot. The proposed approach integrates fractional calculus with Lyapunov stability theory to enhance the robustness and adaptability of the controller in the case of the fully unknown system's dynamics [13]. The direct adaptive mechanism estimates the controller parameters online, improving control performance without requiring precise dynamic modeling.

6.2 Problem Formulation

The objective of this strategy is to design a control law for a Delta robot capable of accurately tracking the dynamics and performances of a reference model [40] [39] , while estimating the controller parameters online.

The control design will be based on the fractional-order state-space model introduced in the previous chapter, defined as follows: :

$$\mathbf{x_1} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \tag{6.1}$$

$$\mathbf{x_2} = \begin{bmatrix} D^\alpha\theta_1 & D^\alpha\theta_2 & D^\alpha\theta_3 \end{bmatrix}^T \tag{6.2}$$

$$D^\alpha \mathbf{x}(t) = \begin{bmatrix} x_2 \\ M(x_1)^{-1}(\mathbf{u} - C(x_1, x_2)x_2 - G(x_1)) \end{bmatrix} \tag{6.3}$$

The reference model is obtained through feedback linearization of the nonlinear system, followed by stabilization using state feedback control with a precompensator. It is described by the following equation:

$$D^{\alpha}\mathbf{x_r}(t) = (A - BK) \cdot x_r + B \cdot P \cdot V \tag{6.4}$$

where:

- A and B are the state-space matrices in canonical form, ensuring full controllability and observability.
- K is the state feedback gain matrix used to assign the desired closed-loop dynamics.
- P is the reference precompensator matrix, designed for trajectory tracking.
- V is the reference input or desired trajectory.

6.3 Control Law Design

We begin by defining the feedback linearization-based control law for tracking the reference model through model identification:

$$u = -\alpha^{\star} - \beta^{\star}\left(-Kx + PV\right) \tag{6.5}$$

where:

$$\alpha^{\star} = -C(x_1, x_2)\, x_2 - G(x_1) \tag{6.6}$$

$$\beta^{\star} = -M(x_1) \tag{6.7}$$

Assuming the system dynamics [48] are completely unknown, we aim to estimate the controller parameters α and β, and apply their estimated values to the system in Equation (6.3):

$$u = u_{m1} - u_{m2} \cdot u_{SFC} \tag{6.8}$$

With u_{m1}, u_{m2} representing the adaptive term in the controller and u_{SFC} representing the state feedback control signal.

$$u = -\hat{\alpha} - \hat{\beta}\left(-Kx + PV\right) \tag{6.9}$$

Substituting the control input into the system, the dynamics become:

$$D^{\alpha}\mathbf{x}(t) = \begin{bmatrix} x_2 \\ M(x_1)^{-1}\left(-\hat{\alpha} - \hat{\beta}(-Kx + PV) - C(x_1, x_2)x_2 - G(x_1)\right) \end{bmatrix} \tag{6.10}$$

By adding and subtracting $\alpha^{\star}$ and $\beta^{\star}$, we rewrite:

$$D^{\alpha}\mathbf{x}(t) = \begin{bmatrix} x_2 \\ -Kx + PV + M^{-1}\tilde{\alpha} + M^{-1}\tilde{\beta}(-Kx + PV) \end{bmatrix} \tag{6.11}$$

Let the estimation errors of the controller parameters be defined as:

$$\tilde{\alpha} = \alpha^{\star} - \hat{\alpha} \tag{6.12}$$

$$\tilde{\beta} = \beta^{\star} - \hat{\beta} \tag{6.13}$$

Thus, Equation (6.10) can be expressed in matrix form as:

$$D^{\alpha}\mathbf{x}(t) = (A - BK)x + BPV + BM^{-1}\tilde{\alpha} + BM^{-1}\tilde{\beta}\Phi \tag{6.14}$$

where:

$$\Phi_{(3,1)} = -Kx + PV = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} \tag{6.15}$$

$$\tilde{\beta}_{(3,3)} = \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 & \tilde{\beta}_3 \end{bmatrix} \tag{6.16}$$

This leads to:

$$D^{\alpha}\mathbf{x}(t) = (A - BK)\,\mathbf{x}(t) + BPV + BM^{-1}\tilde{\alpha} + BM^{-1}\sum_{i=1}^{3}\tilde{\beta}_i\Phi_i \tag{6.17}$$

The tracking error is defined as:

$$e(t) = x(t) - x_r(t) \tag{6.18}$$

$$D^{\alpha}e(t) = D^{\alpha}x(t) - D^{\alpha}x_r(t) \tag{6.19}$$

Substituting Equations (6.4) and (6.16), we obtain:

$$D^{\alpha}e(t) = (A - BK)e + BM^{-1}\tilde{\alpha} + BM^{-1}\sum_{i=1}^{3}\tilde{\beta}_i\Phi_i \tag{6.20}$$

6.4 Stability Analysis

Let us consider the following Lyapunov candidate function:

$$V_1(t, e(t)) = e(t)^T e(t) + \frac{1}{2}\tilde{\alpha}^T M^{-T}\tilde{\alpha} + \sum_{i=1}^{3} \tilde{\beta}_i^T M^{-T}\tilde{\beta}_i \tag{6.21}$$

According to Lemma 1, which provides an upper bound on the fractional derivative of a quadratic form, we have:

$$D^\alpha V_1 \leq e(t)^T D^\alpha e(t) + \tilde{\alpha}^T M^{-T} D^\alpha \tilde{\alpha} + \sum_{i=1}^{3} \tilde{\beta}_i^T M^{-T} D^\alpha \tilde{\beta}_i \tag{6.22}$$

To ensure $D^\alpha V_1 < 0$, we design the adaptation laws such that:

$$D^\alpha \hat{\alpha}(t) = 2B^T e(t) \tag{6.23}$$

$$D^\alpha \hat{\beta}_i(t) = 2\Phi_i B^T e(t) \tag{6.24}$$

Thus, the final inequality becomes:

$$D^\alpha V_1 \leq -e(t)^T Q e(t) < 0 \tag{6.25}$$

Where :

Q is a positive defined matrix.

This guarantees the exponential stability of the closed-loop system according to Lemma 1 and the extended Lyapunov stability theorem.

Summary:
Considering the plant described by Eq. (6.3) with order $0 < \alpha < 1$*, and given a state feedback gain matrix* K *and a reference precompensator* P*, we propose the following adaptive control laws:*

$$u = -\hat{\alpha} - \hat{\beta}(-Kx + PV) \tag{6.26}$$

$$D^\alpha \hat{\alpha}(t) = 2B^T e(t) \tag{6.27}$$

$$D^\alpha \hat{\beta}_i(t) = 2\Phi_i B^T e(t), \quad i = 1, 2, 3 \tag{6.28}$$

6.5 Simulations and Discussions

To validate the effectiveness of the proposed fractional-order MRAC control strategy, simulations (For $\alpha = 0.8$) are performed in MATLAB/Simulink. The Delta robot's end-effector is tasked with a Cubic spline trajectories on it's joints angular position.This trajectory is carefully designed to ensure smooth motion, along with continuous velocity and acceleration profiles, which are crucial for precise tracking and stability analysis.

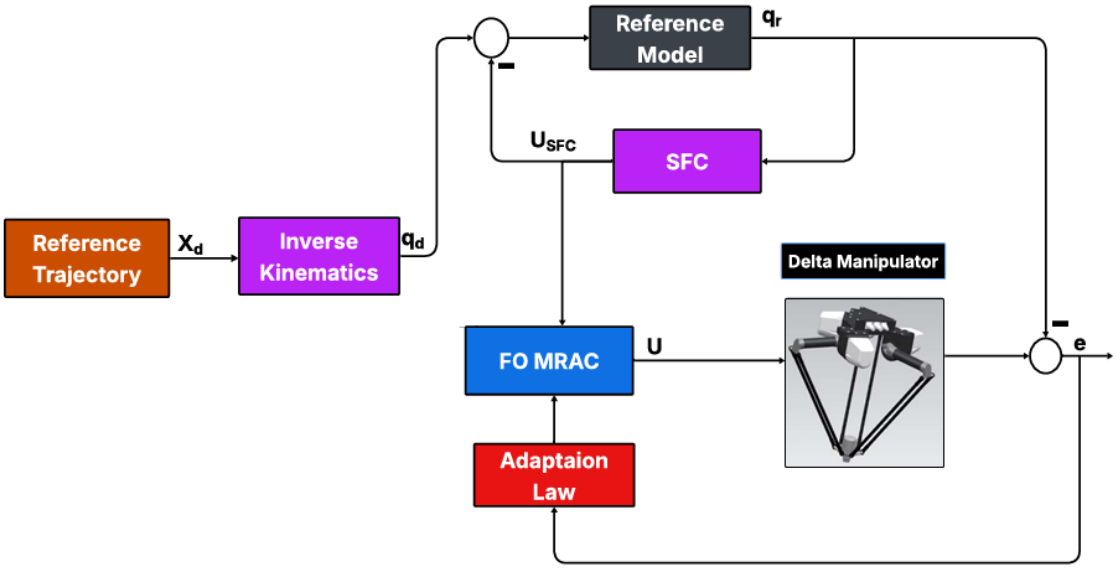


Figure 6.1: Structure of the adaptive FO MRAC control system

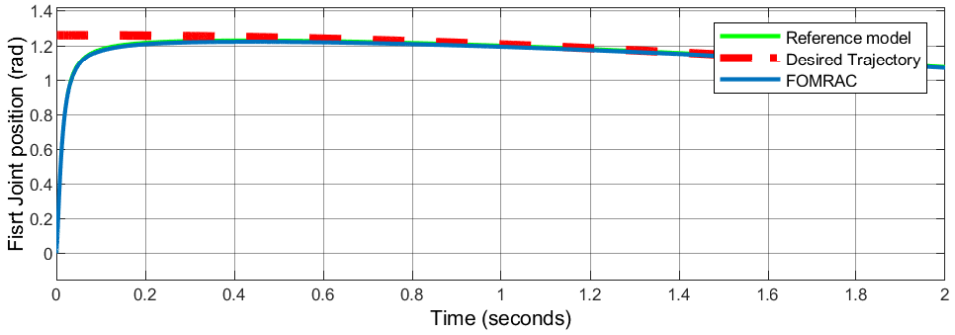


Figure 6.2: First Joint angle

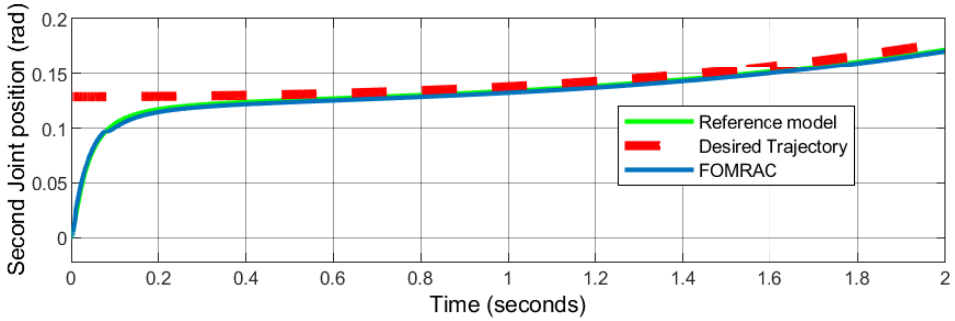


Figure 6.3: Second Joint angle

By observing the joint position curves in Figures 6.2, 6.3, and 6.4, we notice that the reference tracking is very good. The FO MRAC controller exhibits satisfactory performance, with a relatively low rise time (approximately 0.05 s), which reflects the overall stability of the closed-loop system. Moreover, the controller shows a level of performance that aligns well with typical requirements in robotic applications.

The tracking performance of the system is further evaluated by analyzing the end-effector trajectory in 3D space, as shown in Fig. 6.4. The plotted trajectories indicate that the system

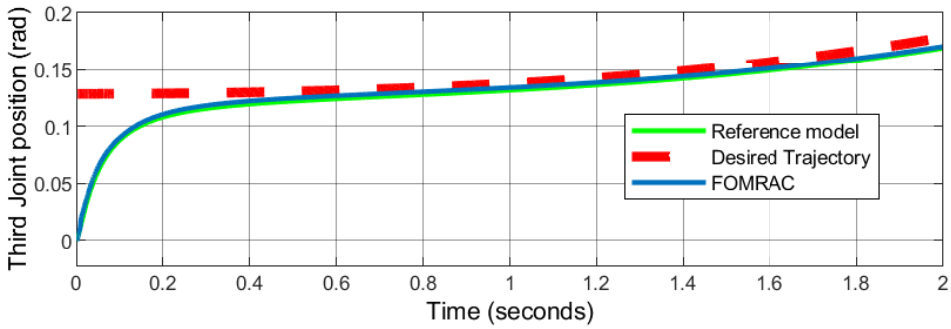


Figure 6.4: Third Joint angle

is capable of following the reference path with good accuracy. The end-effector's response closely follows the desired trajectory, suggesting that the implemented control strategies provide effective path tracking in a three-dimensional environment.

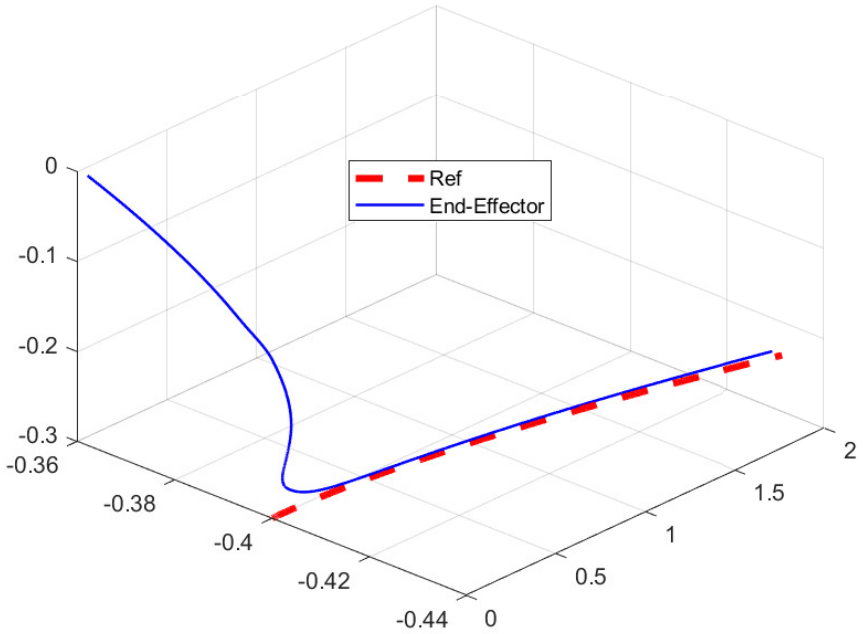


Figure 6.5: End-effector trajectory

To analyze the performances of the control law, the root mean square error (RMSE) of the trajectories for both controllers is summarized in Table 7.1.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ir} - x_i)^2} \tag{6.29}$$

Table 6.1: RMSE values

Controller	RMSE (x-axis)	RMSE (y-axis)	RMSE (z-axis)
FO MRAC	7.86×10^{-3}	7.75×10^{-3}	4.3×10^{-3}

The RMSE values highlight the tracking accuracy of the control law along the x, y, and z axes, confirming that the control objective is achieved.

Figure 6.6 shows the error dynamics between the reference model and the robot, demonstrating very good tracking performance with respect to the reference model.

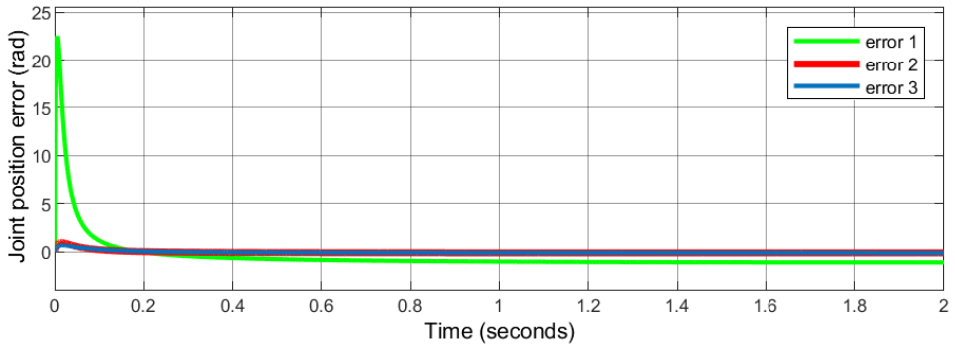


Figure 6.6: Errors Dynamic

Figure 6.7 presents the time evolution of the torque generated by the FO MRAC controller. At the beginning of the response, a noticeable overshoot occurs during the transient phase, which can be attributed to the effect of fractional dynamics on system behavior. Nonetheless, the torque stabilizes quickly, reaching a steady state around 0.05 seconds.

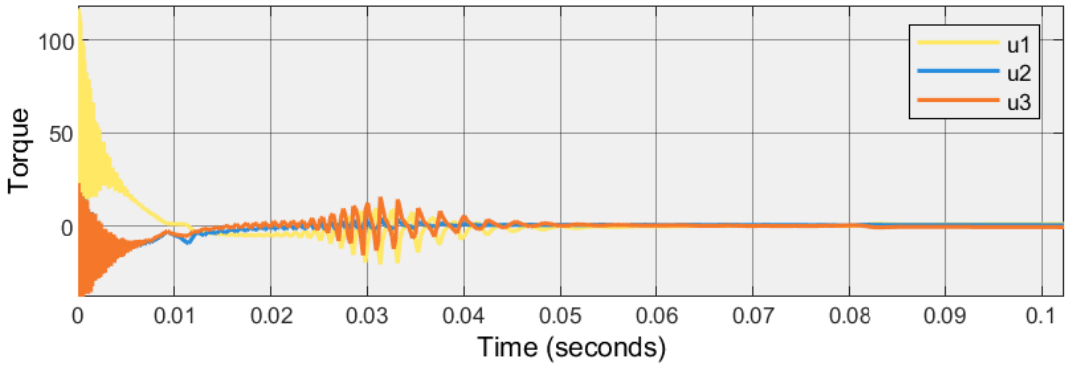


Figure 6.7: Induced Control Signal

6.6 Robustness Tests

This section aims to examine the robustness of the controller against uncertainties, undesired dynamics, and perturbations. This study is conducted in two steps: first, by injecting an external perturbation signal, and second, by introducing an uncertain parameter variation

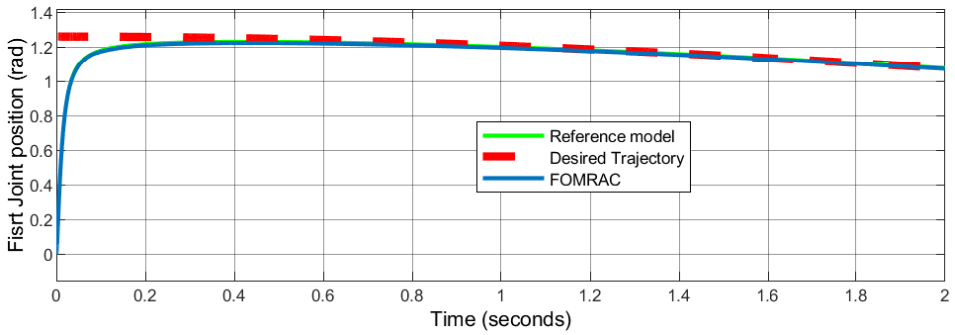


Figure 6.8: Joint angle In presence of uncertainty

Figure 6.8 shows the joint angle response in the presence of bounded uncertainty. Due to the adaptation term in the controller, the additive perturbation is estimated and compensated. This explains the good performance in terms of tracking and stability.

Next, we will examine the controller's ability to stabilize the system in the presence of nonlinear uncertain parameters. The system is then considered as:

$$D^{\alpha}\mathbf{x}(t) = \begin{bmatrix} x_2 \\ (M(x_1) + \Delta M)^{-1}(\mathbf{u} - (C(x_1, x_2) + \Delta C)x_2 - G(x_1)) \end{bmatrix} \tag{6.30}$$

Where , ΔM and ΔC are constant bounded uncertainties.

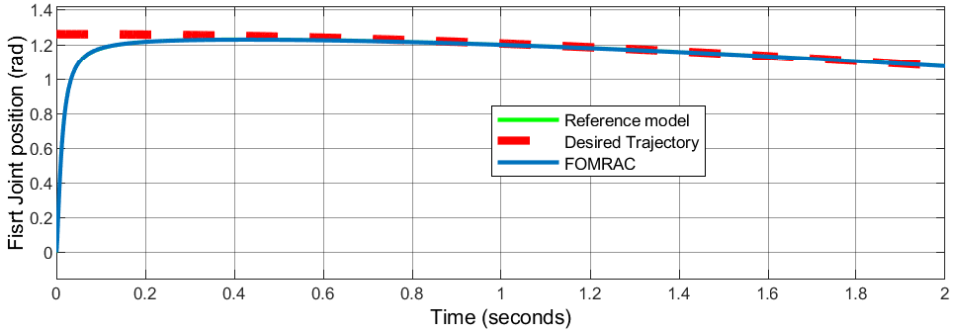


Figure 6.9: Joint angle In presence of uncertainty

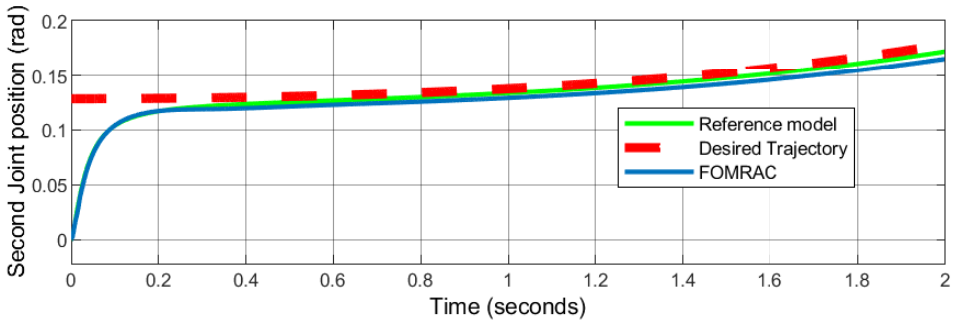


Figure 6.10: Joint angle In presence of uncertainty

The influence of nonlinear modeling errors is illustrated in Figures 6.9, 6.10 and 6.11. Despite the presence of significant model uncertainties, the system maintains stable behavior, which highlights the robustness of the proposed adaptive MRAC fractional-order controller. Although a slight degradation in tracking performance is observed—particularly in terms of precision with respect to the reference trajectory—the adaptive nature of the controller, combined with the memory effect of fractional-order dynamics, allows for effective compensation of these modeling inaccuracies, ensuring overall system reliability.

6.7 Conclusion

In contrast, the Fractional-Order Model Reference Adaptive Control (FO-MRAC) strategy exhibited strong performance even under completely unknown system dynamics. The adaptive mechanism enables real-time adjustment of control parameters, ensuring accurate trajectory tracking and robustness to a wide range of uncertainties. However, this improved responsiveness

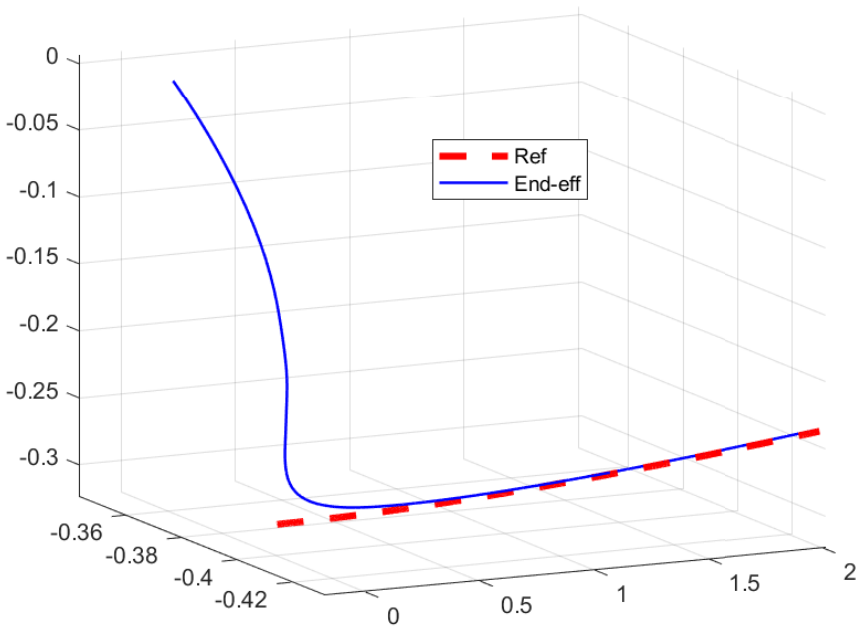


Figure 6.11: Joint angle In presence of uncertainty

comes at the cost of a more abrupt control signal, which may lead to increased actuator effort or mechanical stress in practical applications. Therefore, while FO-MRAC is highly effective in terms of adaptability and precision, careful consideration must be given to the trade-off between control smoothness and responsiveness when deploying it in real-world systems.

Chapter 7

Comparative Analysis of the Proposed Control Strategies

7.1 Introduction

This chapter presents a comparative analysis of three advanced control strategies Model Reference Adaptive Control (MRAC), Backstepping, and Sliding Mode Control (SMC) implemented for the same dynamic system. Each method offers distinct advantages in handling system uncertainties, external disturbances, and nonlinearities, which are critical challenges in modern control applications.

The objective of this chapter is to evaluate and compare the performance, robustness, and implementation complexity of these strategies using consistent metrics and simulation conditions. Through this analysis, the strengths and limitations of each controller will be highlighted, guiding the selection of the most appropriate technique for a given application context.

7.2 Stability and performances

This section provides an in-depth comparative evaluation of the stability and performance characteristics exhibited by three advanced control strategies: Model Reference Adaptive Control (MRAC), Backstepping, and Sliding Mode Control (SMC). The comparison is conducted using a combination of qualitative insights and quantitative performance metrics to ensure a comprehensive understanding of each controller's behavior and effectiveness.

The assessment criteria include several critical aspects of control system performance. First, the trajectory tracking accuracy is examined to determine how closely each control approach enables the system to follow a desired reference path over time. This reflects the precision and responsiveness of the controllers under nominal conditions.

Second, the convergence behavior of each strategy is analyzed. This involves evaluating the rate at which the system's states approach their desired values and the smoothness or abruptness of this convergence. Fast and smooth convergence is generally indicative of a well-tuned and stable control law.

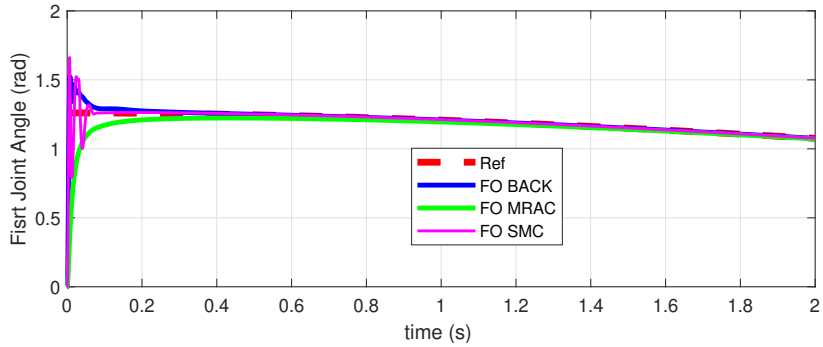


Figure 7.1: First Joint position

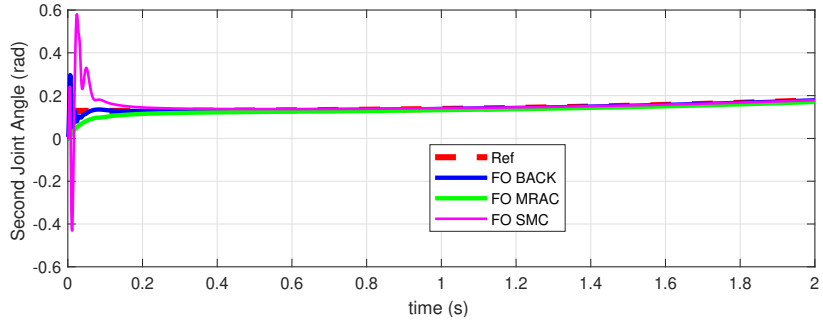


Figure 7.2: Second Joint position

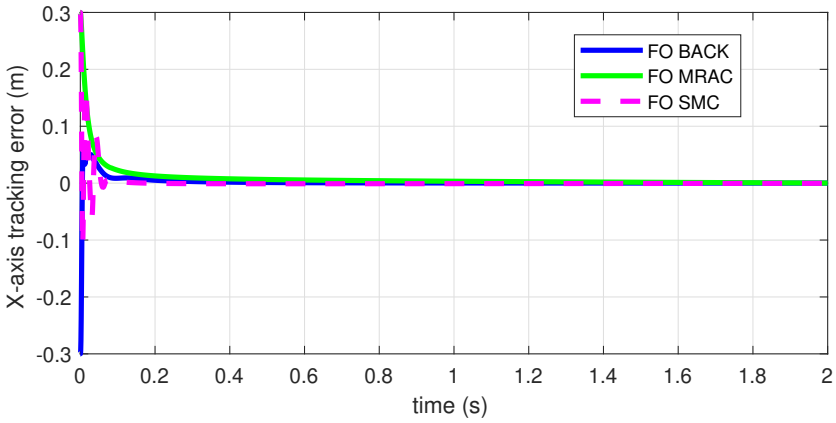


Figure 7.3: X-axis tracking error

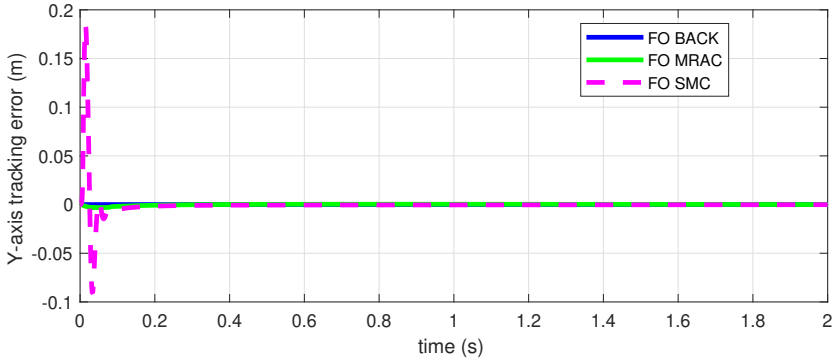


Figure 7.4: Y-axis tracking error

- Figures 7.1 and 7.2 illustrate the joint angular position trajectories. The Fractional-Order Backstepping (FO-Backstepping) controller exhibited an excellent tracking performance,

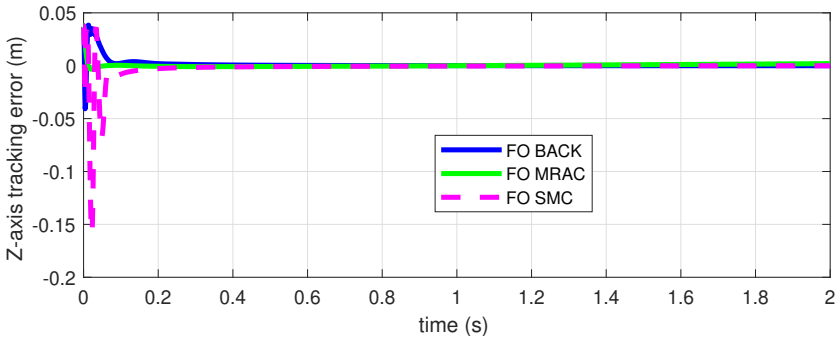


Figure 7.5: Z-axis tracking error

Table 7.1: RMSE values

Controller	RMSE (x-axis)	RMSE (y-axis)	RMSE (z-axis)
FO Backstepping	1.8×10^{-4}	3.2×10^{-10}	6.4×10^{-5}
FO Sliding mode	9.96×10^{-4}	2.24×10^{-10}	1.02×10^{-5}
FO MRAC	7.86×10^{-3}	7.75×10^{-3}	4.3×10^{-3}

characterized by a fast response time of approximately 0.1 s and a small overshoot. In comparison, the Fractional-Order Sliding Mode Control (FO-SMC) also achieved accurate tracking, but with more noticeable overshoots—particularly in certain states—and a slightly slower response time of around 0.2 s.

- The Fractional-Order Model Reference Adaptive Controller (FO-MRAC) provided a smoother response with no overshoot. However, this came at the expense of a slower convergence to the desired trajectory. The overall response time was approximately 0.2 s, with a rise time of about 0.05 s, which is attributed to the fractional-order integration that improves smoothness but delays rapid convergence.

- The tracking error in Cartesian space reached significant values during the transient phase, especially along the x-axis Figure 7.3 . Nonetheless, all controllers were able to rapidly stabilize the error to near-zero levels, with convergence to the origin occurring at approximately 0.1 s.

- For the y and z axes Figures 7.4 and 7.5, the FO-SMC controller produced larger tracking errors compared to FO-Backstepping and FO-MRAC. Despite this, the errors were also quickly stabilized, with the system settling around 0.2 s.

- The Root Absolute Mean Square Error (RAMSE) values demonstrate outstanding performance across all tested controllers. Notably, the backstepping controller achieves the lowest RAMSE, highlighting its superior accuracy and robustness in tracking the desired trajectory compared to the other methods.

7.3 Optimality and Feasibility Analysis

Beyond evaluating controller performance and stability , it is essential to consider the aspects of *optimality* and *feasibility*, especially when transitioning from simulation environments to real-world implementations. This section provides a critical assessment of each control strategy ,

Model Reference Adaptive Control (MRAC), Backstepping, and Sliding Mode Control (SMC) in terms of their optimal behavior and practical applicability.

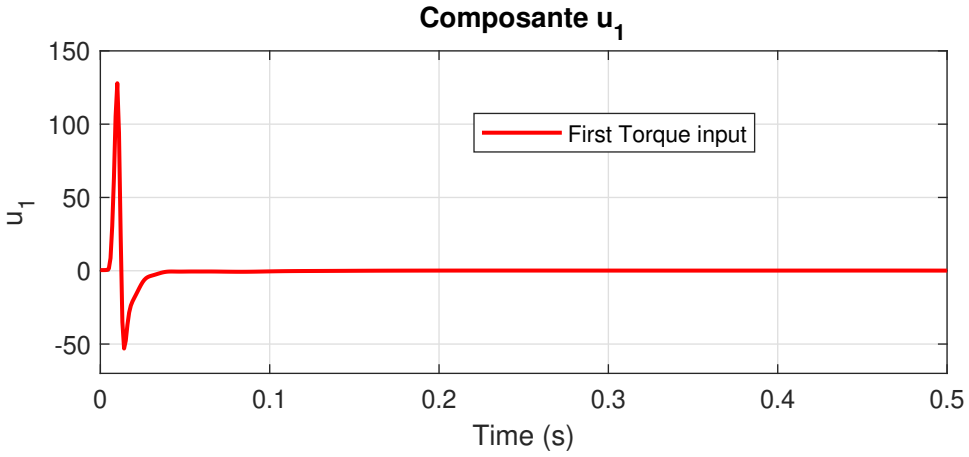


Figure 7.6: Induced Control Signal FO Backstepping

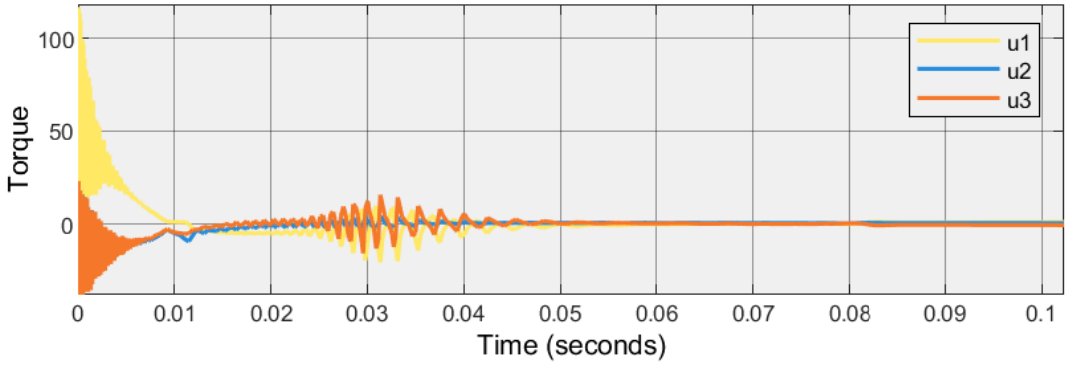


Figure 7.7: Induced Control Signal FO MRAC

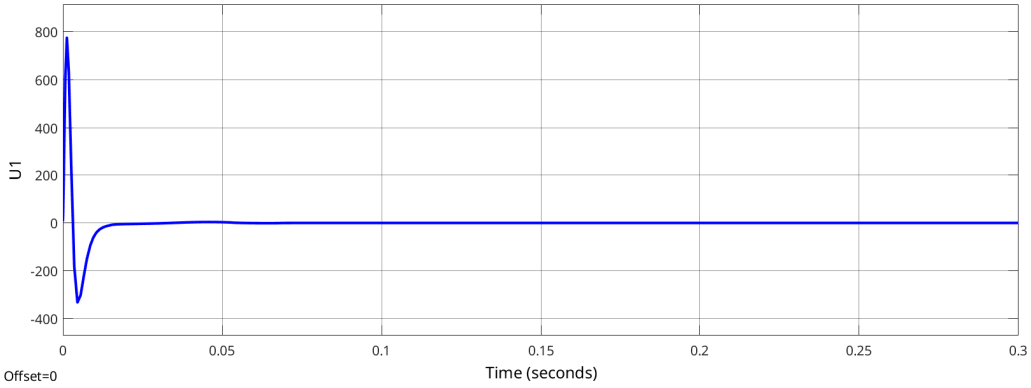


Figure 7.8: Induced Control Signal FO SMC

7.3.1 Optimality Considerations

Optimality, in the context of control design, refers to the extent to which a controller minimizes a specific performance index, such as tracking error, control energy, or convergence time. Although none of the controllers studied in this chapter are inherently optimal in the classical sense , their behavior can be assessed with respect to commonly used performance metrics:

- **Tracking Accuracy:** Quantified using the Root Absolute Mean Square Error (RAMSE), Backstepping exhibited the lowest tracking error, indicating near-optimal performance in trajectory following.

- **Control Effort:** An analysis of the control signal magnitude over time can provide insight into energy efficiency. Excessive or highly oscillatory control signals, often observed in SMC, may indicate sub optimality in terms of actuator wear and energy consumption.

- **Response Time and Overshoot:** Time-domain characteristics such as rise time, settling time, and overshoot contribute to the practical optimality of a control method, particularly in dynamic or safety-critical systems.

7.3.2 Feasibility Considerations

Feasibility encompasses the practicality of implementing a controller in real-world systems, considering aspects such as computational complexity, ease of tuning, robustness to model uncertainties, and hardware limitations.

- **Implementation Complexity:** MRAC [49] requires the identification and continuous adaptation of parameters, which can demand high computational resources and careful initialization. In contrast, Backstepping and SMC may offer more straightforward implementations once the system model is well defined.

- **Robustness to Uncertainty:** SMC is renowned for its robustness against bounded disturbances and modeling errors, making it highly feasible in uncertain environments despite its chattering drawback.

- **Tuning and Parameter Sensitivity:** Feasibility also depends on how sensitive a controller is to its parameters. Adaptive schemes (MRAC) may alleviate this through online adjustment, while Backstepping and SMC typically require careful offline tuning.

7.4 Conclusion

In summary, the choice between Backstepping, Sliding Mode Control (SMC), and Model Reference Adaptive Control (MRAC) largely depends on the specific requirements and constraints of the application. Backstepping control offers strong performance in terms of both optimality and practical implementation.

On the other hand, Sliding Mode Control is renowned for its robustness against model uncertainties and external disturbances. It achieves this by enforcing system trajectories to "slide" along a predefined surface, thus ensuring predictable behavior even in the presence of nonlinearities and parameter variations. Despite its strengths, SMC often suffers from the phenomenon known as control chattering—high-frequency oscillations in the control signal—which has been overcome by the use of fractional calculus and the super-twisting algorithm.

Model Reference Adaptive Control (MRAC) introduces adaptability into the control loop, enabling the system to adjust in real time to changes in system dynamics. This makes MRAC a versatile option that strikes a balance between robustness and performance. However, the adaptive nature of MRAC comes at the cost of increased design and computational complexity.

Careful tuning and real-time estimation are required, which may necessitate more powerful hardware or advanced software infrastructure.

Ultimately, these trade-offs must be carefully considered in light of the intended application. In scenarios where hardware limitations, energy efficiency, or real-time processing constraints are critical, the complexity and implementation overhead of each method become decisive factors. Therefore, the control strategy should be selected not only based on theoretical performance but also on practical feasibility.

General Conclusion

This thesis has focused on the modeling and advanced control of Delta parallel robots with a particular emphasis on trajectory tracking in the presence of system uncertainties and nonlinearities. The Delta robot, a type of parallel manipulator known for its lightweight structure, high-speed operation, and mechanical precision, presents significant challenges in terms of control due to its complex dynamic behavior. The nonlinearity of the model, the strong coupling of joints, and susceptibility to parameter variation require advanced techniques that go beyond classical linear control theory. As such, this work has explored the potential of fractional-order adaptive control as a modern and effective strategy for enhancing robot performance.

Throughout the thesis, a comprehensive approach was adopted. First, a thorough review of the structure and advantages of parallel manipulators was presented, along with a detailed introduction to fractional calculus and its growing relevance in control theory. The modeling phase included both kinematic and dynamic representations of the Delta robot, serving as a foundation for the design and implementation of advanced controllers. A clear emphasis was placed on the integration of fractional-order dynamics within well-established nonlinear adaptive frameworks to exploit the benefits of memory and hereditary properties offered by fractional-order systems.

The main contribution of this research lies in the synthesis and evaluation of three distinct but complementary control strategies: fractional-order adaptive backstepping, fractional-order sliding mode control based on the super-twisting algorithm, and fractional-order model reference adaptive control (FO-MRAC). These control architectures were rigorously designed to accommodate the specific dynamic characteristics of the Delta robot. The fractional-order backstepping controller provided a recursive and systematic method for stabilizing the system, while the super-twisting sliding mode controller introduced robustness against matched disturbances and minimized chattering effects. The FO-MRAC approach enabled the tracking of reference trajectories under uncertain model dynamics by adjusting its parameters adaptively.

All controllers were validated through numerical simulations. The results demonstrated excellent performance in terms of tracking accuracy, robustness, and convergence time. In particular, fractional-order dynamics allowed for a more flexible tuning of the system's transient and steady-state responses, proving their added value compared to integer-order counterparts. The comparative evaluation between the controllers revealed their respective strengths and trade-offs: the backstepping approach ensured smooth convergence with moderate computational complexity; the sliding mode controller excelled in disturbance rejection but required careful design to mitigate chattering; and the FO-MRAC exhibited strong adaptability to dynamic changes with the added complexity of reference model design.

Despite the encouraging results obtained, it is important to acknowledge certain limitations of this work. All evaluations were conducted in a simulated environment, which, while effective for initial validation, cannot fully capture the nonlinearities, delays, noise, and actuator limitations present in real-world systems. Moreover, the design of fractional-order controllers often

involves the selection of multiple parameters—such as fractional orders, adaptation gains, and boundary layers—whose tuning remains nontrivial and problem-dependent. Therefore, future investigations should aim at automating this tuning process and extending the framework to experimental setups.

Looking toward future work, the integration of artificial intelligence (AI) and data-driven control methods presents a highly promising direction. One potential avenue is the adoption of reinforcement learning (RL), where the robot learns optimal control strategies by interacting with its environment and improving performance over time. Such methods are particularly attractive for dynamic and uncertain environments where modeling is difficult or incomplete. Another direction involves neural network-based adaptive controllers, which can approximate unknown nonlinear functions in real-time and adjust their structure and parameters on the fly. These approaches could be used in combination with fractional-order control frameworks, yielding hybrid controllers that combine the theoretical strength of model-based design with the adaptability of AI.

Furthermore, the use of online learning and adaptive observers could allow the controller to identify system changes in real time and reconfigure itself accordingly. This would be especially valuable in applications where the robot's physical conditions evolve (e.g., payload variation, joint wear, or sensor degradation). Integration with vision-based feedback and sensor fusion would also enhance the system's autonomy and reliability.

Bibliography

[1] M. Bouri, "The linear delta: Developments and applications," 2015, available at: [URL if available].

[2] L. Clavel, "Conception d'un robot parallèle rapide à 04 ddl," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, 1999.

[3] M. BEN RABAH and A. BENALI, "Modélisation et commande d'ordre fractionnaire du robot parallèle delta isir 88. simulation et expérimentation," 2021.

[4] A. Gholami, T. Homayouni, R. Ehsani, and J.-Q. Sun, "Inverse kinematic control of a delta robot using neural networks in real-time," *Robotics*, vol. 10, no. 4, p. 115, 2021.

[5] H. Liu, Y. Pan, S. Li, and Y. Chen, "Adaptive fuzzy backstepping control of fractional-order nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1005–1014, 2015.

[6] P. J. Torvik and R. L. Bagley, "Fractional dynamics in human motion control and robotic systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 2, pp. 123–129, 1984.

[7] O. Khelifi and L. Azzoug, "Modélisation et commande d'un robot parallèle de type delta," 2010, ecole Nationale Polytechnique, Département d'Automatique, Projet de fin d'étude.

[8] A. Tepljakov, *Fractional-order Modeling and Control of Dynamic Systems*. Springer, 2017.

[9] L. Angel and J. Viola, "Fractional order pid for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, 2018.

[10] N. Aguila-Camacho, M. A. Duarte-Mermoud, and J. A. Gallegos, "Lyapunov functions for fractional order systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 2951–2957, 2014.

[11] Y. Wei, Y. Chen, S. Liang, and Y. Wang, "A novel algorithm on adaptive backstepping control of fractional order systems," *Neurocomputing*, 2015.

[12] M. A. Duarte-Mermouda, N. Aguila-Camacho, J. A. Gallegosa, and R. Castro-Linares, "Using general quadratic lyapunov functions to prove lyapunov uniform stability for fractional order systems," *Nonlinear Analysis: Hybrid Systems*, vol. 39, p. 100851, 2021.

[13] S. Ladaci, S. Khelas, A. Ynineb, D. Copot, and C. Ionescu, "Fractional order mrac control design for a lightning system based on a fractional order second degree model," *IFAC-PapersOnLine*, vol. 58, pp. 95–100, 2024.

[14] I. Podlubny, *Fractional Differential Equations*. Academic Press, 1999.

[15] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-order Systems and Controls: Fundamentals and Applications*. Springer, 2010.

[16] S. Li, Y. Xu, and Y. Chen, "Fractional-order impedance control for stable human-robot interaction," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 599–610, 2017.

[17] P. J. Torvik and R. L. Bagley, "Fractional dynamics in human motion control and robotic systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 2, pp. 123–129, 1984.

[18] R. L. Magin, "Fractional calculus in bioengineering," *Critical Reviews in Biomedical Engineering*, vol. 32, no. 1, pp. 1–104, 2004.

[19] A. C. Luo and N. H. Ibragimov, *Nonlinear Physical Science*. Springer, 2008.

[20] S. Ladaci, "Contribution à la commande adaptative d'ordre fractionnaire," Ph.D. dissertation, École Nationale Polytechnique, Alger, 2007.

[21] M. A. Duarte-Mermouda, N. Aguila-Camachoa, J. A. Gallegosa, and R. Castro-Linaresc, "Using general quadratic lyapunov functions to prove lyapunov uniform stability for fractional order systems," *Nonlinear Analysis: Hybrid Systems*, vol. 39, p. 100851, 2021.

[22] N. Aguila-Camacho, M. A. Duarte-Mermoud, and J. A. Gallegos, "Lyapunov functions for fractional order systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 2951–2957, 2014.

[23] H. Liu, Y. Pan, S. Li, and Y. Chen, "Adaptive fuzzy backstepping control of fractional-order nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1005–1014, 2015.

[24] I. Podlubny, "Fractional-order systems and piλdμ controllers," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 208–214, 2002.

[25] A. Researcher and B. Scholar, "Advanced fopid control techniques," *Control Systems Journal*, 2020.

[26] C. Engineer and D. Analyst, "Applications of fractional controllers in robotics," *IEEE Transactions on Control Systems*, 2022.

[27] J. A. Moreno and I. Osorio, "A lyapunov approach to second-order sliding modes," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 7, pp. 712–737, 2008.

[28] Z. Man and D. Habibi, "A robust adaptive sliding-mode control for rigid robotic manipulators with arbitrary bounded input disturbances," *Journal of Intelligent and Robotic Systems*, vol. 17, no. 4, pp. 371–386, 1996.

[29] M. Rachedi, "Modélisation et commande robuste de robots parallèles: Application au robot delta," Ph.D. dissertation, Ecole Nationale Polytechnique, Département d'Automatique, 2016.

[30] 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.

[31] A. Mujumdar, B. Tamhane, and S. Kurode, "Fractional order modeling and control of a flexible manipulator using sliding modes," in *Proceedings of the American Control Conference (ACC)*, 2014, pp. 1–6.

[32] K. Bingi, B. R. Prusty, and A. P. Singh, "A review on fractional-order modelling and control of robotic manipulators," *Robotics*, vol. 12, no. 1, p. 20, 2023.

[33] L. Angel and J. Viola, "Fractional order pid for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, 2018.

[34] A. Rebai, K. Guesmi, and boualem hemici, "Design of an optimized fractional order fuzzy pid controller for a piezoelectric actuator," *Control Engineering and Applied Informatics*, 2015.

[35] D. Zhu, Y. He, and F. Li, "Trajectory tracking of delta parallel robot via adaptive backstepping fractional-order non-singular sliding mode control," *Mathematics*, vol. 12, no. 14, p. 2236, 2024.

[36] Z. Zhengsheng and L. Yuxin, "Adaptive super-twisting algorithm-based fractional-order sliding mode control of redundantly actuated cable driving parallel robot with uncertainty and disturbance estimation," *IET Control Theory & Applications*, vol. 15, no. 18, pp. 2301–2312, 2021.

[37] Y. Bensafia, 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, no. 2, pp. 138–156, 2018.

[38] A. Gholami, T. Homayouni, R. Ehsani, and J.-Q. Sun, "Inverse kinematic control of a delta robot using neural networks in real-time," *Robotics*, vol. 10, no. 4, p. 115, 2021.

[39] R. Chittillapilly and D. Hepsiba, "Tube model reference adaptive control for a cylindrical tank system," *International Journal of Innovative Research in Technology*, vol. 1, no. 10, pp. 252–255, 2014.

[40] K. Bingi, B. R. Prusty, and A. P. Singh, "A review on fractional-order modelling and control of robotic manipulators," *Robotics*, vol. 12, no. 1, p. 20, 2023.

[41] Y. Wei, Y. Chen, S. Liang, and Y. Wang, "A novel algorithm on adaptive backstepping control of fractional order systems," *Neurocomputing*, 2015.

[42] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *International Journal of Control*, vol. 76, no. 9, pp. 924–941, 2003.

[43] C. Edwards and S. K. Spurgeon, *Sliding Mode Control: Theory and Applications*. CRC Press, 1998.

[44] S.-H. Han, M. S. Tran, and D.-T. Tran, "Adaptive sliding mode control for a robotic manipulator with unknown friction and unknown control direction," *Applied Sciences*, vol. 11, no. 9, p. 3919, 2021.

[45] B. Li, J. Zhu, R. Zhou, and G. Wen, "Adaptive neural network sliding mode control for a class of siso nonlinear systems," *Mathematics*, vol. 10, no. 7, p. 1182, 2022.

[46] Y.-H. Chang, C.-Y. Yang, and H.-W. Lin, "Robust adaptive-sliding-mode control for teleoperation systems with time-varying delays and uncertainties," *Robotics*, vol. 13, no. 6, p. 89, 2024.

[47] H. Khaled, D. Boukhetala, and F. Boudjema, "A new robust model reference adaptive control for induction motor drives using a hybrid controller," in *Proceedings of the International Conference on Power Engineering, Energy and Electrical Drives (SPEEDHAM)*, 2008, pp. 1109–1113.

[48] A. R. Ynineb and S. Ladaci, "Mrac adaptive control design for an f15 aircraft pitch angular motion using dynamics inversion and fractional-order filtering," *International Journal of Robotics and Control Systems (IJRCS)*, vol. 2, no. 2, pp. 240–252, 2022.

[49] K. H. Krishna *et al.*, "Design and development of model based controller for a spherical tank," *International Journal of Current Engineering and Technology*, vol. 2, no. 4, pp. 374–376, 2012.

Appendix

Business Model Canvas

Fractional Adaptive Control D-Delta

<table>
<tr><td colspan="2">1. Customer Segments
• Robotics and mechatronics companies
• Control systems research laboratories
• Tech startups
• Robotic biomedical sector
• Safety-critical embedded systems</td><td>2. Value Proposition
• Improved precision and stability through fractional control
• Automatic adaptation to external disturbances
• Easy integration with ROS, Arduino, etc.
• Enhanced energy efficiency and reduced wear</td></tr>
<tr><td>3. Channels
• Scientific conferences (IEEE, ICRA, MED)
• Open-source platforms (GitHub, ROS)
• Scientific publications
• Professional networks and LinkedIn</td><td>4. Customer Relationships
• Personalized technical support
• Co-development with partners
• Specialized training
• Clear and interactive documentation</td><td>5. Revenue Streams
• Licensing of the developed algorithm
• Consulting services
• Industrial collaborative projects
• Technological grants or scholarships</td></tr>
<tr><td>6. Key Resources
• Expertise in adaptive and fractional control
• Prototyping platforms (ESP32, MATLAB)
• Experimental data
• Academic supervision</td><td>7. Key Activities
• Development of the D-Delta algorithm
• Validation on real systems
• Embedded integration (Arduino, microcontrollers)
• Scientific dissemination</td><td>8. Cost Structure
• Electronic components for testing
• Software tools (MATLAB licenses, etc.)
• Travel expenses for conferences
• Development, testing, and documentation time
• Potential patent filing costs</td></tr>
<tr><td colspan="3">9. Key Partnerships
• Academic research institutions
• Robotics equipment suppliers
• Industry partners for pilot testing
• Open-source and scientific communities</td></tr>
</table>