Master Final Project

Master in Industrial Engineering

Analysis and design of a complex-valued sliding mode controller of a 2-link planar manipulator

Author: Adrián Blázquez Romañá
Director: Arnau Doria Cerezo
Summon: January, 2023

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

Abstract

Robotics and robot manipulators are some common concepts which nowadays are seen as something usual in a lot of industries. However, they are quite young fields in engineering and they include lots of different specialities such as mathematics and mechanical or electrical engineering. These lasts decades, the development of new robots and their control techniques have grown a lot, having now a wide variety of knowledge about their behavior and control algorithms that allow them to do their specific tasks with low errors and high performance.

This project presents a new strategy in control engineering for the robotics field, which consists of an extension of a sliding mode controller to a complex-valued domain. This controller allows to track the tool center position (TCP) of a 2-link planar manipulator without the direct use of inverse kinematics and working always in the complex space. Hence, the forward kinematics of the end-effector of the robot are modeled with complex variables to design the nonlinear controller and be able to analyze its performance and study the potential of this new approach in this application field. Moreover, three more different controllers (a real-valued sliding mode controller, a state-feedback and a PID control) are also designed so the main controller (Complex-valued sliding mode controller) can be analyzed and compared with other solutions to study the possible benefits and disadvantages it may have. All the controllers are analyzed and compared with MATLAB and simulated with SIMULINK and the results obtained are studied according to a qualitative and quantitative analysis based on some Key Performance Indicators (KPIs).

Finally, all the results obtained during all the development of this project are summarized, discussed and presented with the conclusions extracted.

Resum

La robòtica i els robots manipuladors són conceptes que actualment es contemplen com a temes habituals en moltes indústries. De totes maneres, es tracta de camps relativament nous en l'enginyeria i inclouen diverses especialitats com la matemàtica i l'enginyeria mecànica o elèctrica. Durant aquestes últimes dècades, el desenvolupament d'aquests robots i les seves tècniques de control han crescut considerablement, arribant a tenir un ampli coneixement sobre el seu comportament i els algorismes de control que els hi permeten realitzar les seves tasques amb errors baixos i gran rendiment.

Aquest projecte presenta una estratègia nova en l'enginyeria de control pel camp de la robòtica, la qual consisteix en una extensió del control per mode de lliscament (*sliding mode control*, en anglès) en un domini complex. Aquest controlador permet fer un seguiment de la posició final d'un robot manipulador pla de dos braços sense l'ús directe de la cinemàtica inversa i treballant sempre a l'espai complex. D'aquesta manera, la cinemàtica directa del robot s'ha modelat amb variables complexes per tal de dissenyar el controlador no lineal i poder analitzar el seu rendiment i estudiar el potencial d'aquest nou enfocament en aquest camp d'aplicació. A més, s'han dissenyat també tres controladors (un control per mode de lliscament estàndard, un PID i un control per realimentació d'estat) per a poder analitzar i comparar el controlador principal (control per mode de lliscament complex) amb altres solucions i, d'aquesta manera, estudiar les seves possibles avantatges i inconvenients. Tots els controladors s'analitzen i comparen fent servir MATLAB i se simulen amb SIMULINK, estudiant els resultats obtinguts fent una anàlisi qualitativa i quantitativa basada en uns Indicadors Clau de Rendiment (KPIs).

Finalment, tots els resultats obtinguts durant el desenvolupament del projecte es resumeixen, discuteixen i presenten amb les conclusions extretes.

Resumen

La robótica y los robots manipuladores son conceptos que actualment se contemplan como temas habituales en muchas industrias. De todas maneras, se trata de campos relativamente nuevos en la ingeniería e incluyen diversas especialidades como la matemática i la ingeniería mecánica o eléctrica. Durante estas últimas décadas, el desarrollo de estos robots y sus técnicas de control han crecido considerablemente, llegando a tener un amplio conocimiento sobre su comportamiento y los algoritmos de control que les permiten realizar sus tareas con bajo error y gran rendimiento.

Este proyecto presenta una estrategia nueva en ingeniería de control para el campo de la robótica, consistente en una extensión de un controlador en modo deslizante (*sliding mode control*, en inglés) a un dominio de valores complejos. Este controlador permite seguir la posición final de la herramienta de un manipulador plano de 2 eslabones sin el uso directo de la cinemática inversa y trabajando siempre en el espacio complejo. De esta manera, la cinemática directa del efector final del robot se modela con variables complejas para diseñar el controlador no lineal y poder analizar su rendimiento y estudiar el potencial de este nuevo enfoque en este campo de aplicación. Además, también se han disseñado tres controladors (un control en modo deslizante, un PID y un control por realimentación de estado) para poder analizar y comparar el controlador principal (un control en modo deslizante complejo) con otras soluciones y, de esta manera, estudiar sus posibles ventajas e inconvenientes. Todos los controladores se analizan y comparan usando MATLAB y se simulan con SIMULINK, estudiando los resultados obtenidos con un análisis cualitativo y cuantitativo basado en unos Indicadores Clave de Rendimiento (KPIs).

Finalmente, todos los resultados obtenidos durante el desarrollo del proyecto se resumen, discuten y presentan junto a las conclusiones extraídas.

Acknowledgments

First of all, I would like to thank my family and friends for their continuous support and patience during these years of the master's degree studies, specially during the realisation of this work.

I would also like to express my sincere gratitude to Arnau Doria for proposing me such an interesting topic and for accepting to guide my project. For these last months of complete dedication, technical support and theoretical classes and all the advises and suggestions based on his knowledge and personal experience, that have allowed me to do this work the way it has been until this final result.

Thank you all.

Contents

List of Figures

List of Tables

List of Acronyms

DOF	Degrees of freedom
TCP	Tool center position
FK	Forward kinematics
IK	Inverse kinematics
EOM	Equations of motion
CoG	Center of gravity
DH	Denavit-Hartenberg
SMC	Sliding mode control
CV-SMC	Complex-valued sliding mode control
RV-SMC	Real-valued sliding mode control
PID	Proportional-Integral-Derivative
SFC	State feedback controller

Chapter 1

Introduction

1.1 Motivation

Robotics and control theory are some wide and current topics and they are present in a lot of areas, specially in machines and industries, as they allow to perform tasks faster and with high precision. The control strategies used are really diverse [Jiang et al., 2020] [George Thuruthel et al., 2018] and depend a lot on the task that needs to be carried on and the requirements it may have.

The main motivation of this project is to introduce a different new approach of a sliding mode controller [Utkin, 1993] in the robotics and automatics field based on the complex-valued description [Dòria-Cerezo et al., 2020]. There are some areas or topics in robotics that are an extension of complex numbers, such as quaternions [Goldman, 2011], or that can be simplified considering the complex domain, such it could be the Cartesian position of a planar robot, that could be benefited in terms of control strategies by considering a complex-valued approach when designing the controllers. This work introduces this new approach in this area so it can be analyzed and it can be considered whether it can bring benefits to the current control algorithms or not.

1.2 State of the art

All the industries and sectors have been changing for the last past decades as new improvements have been reached. One of the most important changes that has been introduced is the use of robots to help the workers do some tasks, whether it is because of its difficulty or because any other requirement such as precision, performance, and so on. All these robots can be modeled using several techniques and can use different control methods depending on the areas or robotic fields for which these algorithms are needed.

The modeling methods, and emphasizing on robotic manipulators, can be divided between kinematics and dynamics models, prioritizing each one of them according to its future use. For the kinematic modeling, in terms of getting the forward kinematics equations, one of the most used method is the Denavit Hartenberg convention [Denavit and Hartenberg, 1955]. On the other hand, the equations of motion of a dynamic model can be obtained through different methods, of which it is worth noting the Newton-Euler equations [Ardema, 2004], the Lagrange-Euler formulation [Lagrange, 1853] or the generalized d'Alembert equation of motion formulation [Lee et al., 1983].

Regarding the control strategies used, the manipulators are usually controlled directly

with the torque applied to each of their joints and they typically use the joint angles as state variables. The target position is transformed through inverse kinematics to the joint space so it can be controlled according to the angles inputs. The most typical controllers used are based on the feedback-linearization technique, which allows to design linear controllers like PID [Johnson and Moradi, 2005], sliding mode controllers (that can also be complemented with other controllers like PID or one of its variants) and passivity-based controllers [Khan et al., 2012].

On the other hand, although the use of complex variables to model dynamic systems is not new, the control theory tools for complex-valued systems are not extensive at all. Some works that rely on this approach include mainly the control of electrical systems like power converters or electrical machines [Dòria-Cerezo et al., 2020] or complex-valued neural networks [Hirose, 2003].

1.3 Objectives

The main objective of this project is to introduce and design a sliding mode controller for a 2-link planar manipulator based on a complex-valued approach to describe the variables of the system. To achieve this goal, the following specific objectives must be met:

- Explore the state of the art of the control of robotic manipulators.
- Study the kinematic and dynamic behavior of a robot.
- Introduce and develop a proper nonlinear model of a 2-link manipulator.
- Design different controllers that control the TCP properly and can be used to compare the results obtained.
- Define the benefits and disadvantages of the main controller designed according to an analysis and comparison of the response of the other controllers presented.
- Give a basis in the robotics framework about complex-valued based controllers.

1.4 Outline of the project

This work is divided and organised with the following structure:

Chapter 2

In Chapter 2, a brief introduction to sliding mode control is presented, describing its basis and defining some important concepts that will be used during the design of the controllers. Moreover, the introduction of the SMC is extended to a class of complex-valued dynamic systems, so it can be used to present the different concepts before designing the main controller.

Chapter 3

This chapter introduces the kinematics and dynamics of a robot manipulator in order to get a proper model that can be used to design the different controllers proposed. Moreover, the kinematics of the TCP are also modeled in the complex domain so it can be used in Chapter 3 to design the complex-valued sliding mode controller. Once the robotic arm is modeled, it will be implemented with the use of MATLAB and SIMULINK and simulated to get the different responses and analyze them.

Chapter 4

In this chapter, the proposed controllers are designed. First, the three controllers that will be used to analyze and compare the results obtained with the complex-valued sliding mode controller are introduced and described. One of them consists of a standard (real-valued) sliding mode controller whereas the other two are linear controls (PID and state-feedback controls) based on a feedback linearization technique. Finally, it introduces a complex-valued sliding mode controller to track the final position of the end-effector of a 2-link planar manipulator. All of them are designed and implemented with MATLAB and simulated via SIMULINK.

Chapter 5

In Chapter 5, the different tests conducted to analyze the response of the controllers, as well as the different indicators used to compare them are presented. The controllers are simulated and compared between them to see the relative benefits and drawbacks they have.

Chapter 6

This chapter introduces the economic analysis of all the project, divided between personal cost and material cost.

Chapter 7

In this chapter, the environmental impact of all the work is studied and analyzed.

Chapter 8

Chapter 8 introduces a brief analysis of the social impact and gender equality.

Chapter 9

The final chapter presents the conclusions of the work and the results obtained during its realisation. The future work based on what has been presented in this work is also exposed in Chapter 9.

1.5 Project planning

The developing of any project must be adjusted properly to the calendar and the deadlines established. Then, the first step has been defining the different tasks required to achieve the objectives mentioned before. The project has been divided according to the following activities:

- **Objective definition:** The first step should always be the complete definition of the objective that must be accomplished during the realization of the project.
- **Research and previous study:** Analyze the state of the art of complex-valued controllers, as well as for the other strategies proposed during this work. The different theoretical background of all of them and the modeling methods are also included in this task.
- **Kinematic modeling:** Model the kinematic behaviour of a 2-link planar manipulator.
- **Dynamic modeling:** Model the dynamic behaviour of the robot and get its equations of motion.
- **Design of the linear controllers:** Design the linear controllers based on a previous

feedback linearization.

- **Design of the standard (real-valued) SMC:** Design of the real-valued sliding mode controller.
- **Design of the complex-valued SMC:** Design of the complex-valued sliding mode controller.
- **Simulation and comparison:** Simulation and comparison of the different control strategies proposed.
- **Impact analysis:** Economical, environmental and social analysis related to this work.
- **Conclusions:** The final step is to sum up all the conclusions extracted and analyze the future work to do next.

The entire task breakdown and the distribution of time dedicated to each task is graphically represented in the Gantt chart [Gantt, 1974] shown in Figure 1.1, in where all the durations have been defined in weeks.

TASKS	Duration	September 2022				October 2022				November 2022				Desember 2022				January 2022			
		W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Initialization	6																				
Objective definition	0,25																				
Reasearch and previous study	6																				
Modeling	1,5																				
Kinematic modeling	0,5																				
Dynamic modeling	1																				
Design	8																				
Design of the linear controllers	2																				
Design of the standard (real-valued) SMC	2																				
Design of the complex-valued SMC	4																				
Results and Conclusions	2																				
Simulation and comparison	1,5																				
Impact analysis	0,5																				
Conclusions	0,5																				

Figure 1.1: Gantt diagram for the detailed activities of the project.

Chapter 2

Sliding mode control

2.1 Sliding mode control basis

Previous to starting the project, a brief introduction to the sliding mode control (SMC) for real-valued systems is introduced.

Given the nonlinear system affine in the control [1]

$$\dot{x} = f(x) + g(x)u, \tag{2.1}$$

where $\dot{x}$ is the evolution of the state variables, $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the input vector and $f : \mathbb{R}^{n_x} \longmapsto \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \longmapsto \mathbb{R}^{n_x}$ are the state and input mapping. The SMC is a nonlinear control method that modifies the dynamics of the system by applying a discontinuous control signal that forces the system state to reach (reaching face) and to remain (sliding face) on a specified sliding surface (σ). Then, the main components to design the proper SMC are:

- The switching function is defined as the function ($\sigma(x)$) so that the system on the sliding surface (switching manifold) $\hat{\Sigma} \subseteq \Sigma$ evolves in the desired way, being $\Sigma := \{x \in \mathbb{R}^n; \sigma(x) = 0\}$ and $\hat{\Sigma}$ the sliding domain.
- A discontinuous control law

$$u = \begin{cases} u^+, & \sigma(x) > 0 \\ u^-, & \sigma(x) < 0 \end{cases} \tag{2.2}$$

  which has to be chosen in order to enforce a sliding mode.

On the other hand, it is important to consider the existence of sliding modes. Considering the positive-definite Lyapunov function candidate

$$V(x) = \frac{1}{2}\sigma^T(x)\sigma(x) \tag{2.3}$$

and the system defined in (3.1), a sufficient condition for the existence of a sliding mode is

[1] During the realisation of the work, the time dependency of all the variables will be omitted to simplify notation and all the equations presented.

$$\dot{V}(x) = \sigma^T(x)\dot{\sigma}(x) < 0, \tag{2.4}$$

where $\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x}\dot{x} = \frac{\partial \sigma}{\partial x}(f(x) + g(x)u)$. Hence, the discontinuous control action defined in (3.2) can be complemented as

$$u = \begin{cases} u^+, & \sigma(x) > 0 \Rightarrow \dot{\sigma}(x) < 0 \\ u^-, & \sigma(x) < 0 \Rightarrow \dot{\sigma}(x) > 0. \end{cases} \tag{2.5}$$

Finally, when the system is confined to the sliding surface ($\sigma(x) = \dot{\sigma}(x) = 0$), it behaves as a reduced system driven by an equivalent control (u_{eq}). This equivalent control is the continuous control law that makes Σ an invariant manifold with respect to (3.1) and can be defined as

$$u_{eq} = -\left(\frac{\partial \sigma}{\partial x} g(x)\right)^{-1} \frac{\partial \sigma}{\partial x} f(x). \tag{2.6}$$

It is important to notice how $\frac{\partial \sigma}{\partial x} g(x) \neq 0$, known as reachability condition, must be guaranteed for all x in a neighborhood of $\hat{\Sigma} \subseteq \Sigma$. For practicality, the control law is usually defined as $u = u_{eq} + u_d$, being u_{eq} the equivalent control (continuous) and u_d the discontinuous part ensuring a finite time convergence to the surface. Hence, the control law is fitted according to

$$u^+ < u_{eq} < u^-. \tag{2.7}$$

Finally, ideal sliding mode controllers require of infinite switching frequency, which in real applications is not possible to achieve. It is also a behavior that may produce low control accuracy, heat loses in electrical systems or even excite undesired high frequency dynamics. One of the most common solutions to solve this chattering problem is changing the discontinuous control law, which depends on the sign of the sliding manifold, with the use of a saturation function as the following:

$$u = \begin{cases} \text{sign}(\sigma), & |\sigma| > \epsilon \\ \kappa\sigma, & |\sigma| \leq \epsilon, \end{cases} \tag{2.8}$$

where κ is a gain, $\epsilon \in \mathbb{R}^+$ is the thickness of the boundary layer and sign() stands for the sign function. However, it is also worth noting that chattering might also be caused because of the switching behaviour of some actuators used in the system, as it could be an electronic power converter.

2.2 Complex-Valued SMC

Sliding modes can also be extended to a class of complex-valued nonlinear dynamic systems [Dòria-Cerezo et al., 2020]. In this section, a brief introduction to complex-valued sliding mode control (CV-SMC) is presented. Then, the first step is to introduce some notation that will be used during the work.

Let $j = \sqrt{-1}$ be the imaginary number and let $\mathbb{C}^n$ denote the complex nth-dimensional space. Then, defining $z = |z|e^{-j\varphi_z} \in \mathbb{C}$ as a complex variable represented in exponential

form, where $|z|$ and φ_z are its modulus and argument, respectively, its real and imaginary parts can be written as $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$. Moreover, notation $\bar{z}$ will be used to represent the complex conjugate of z. Finally, $\Omega \subset \mathbb{C}^n$ denotes an open subset of $\mathbb{C}^n$ and $H(\Omega, \mathbb{C}^n)$ the set of holomorphic maps from Ω to $\mathbb{C}^n$.

Let's consider the nonlinear system affine in the control

$$\dot{z} = f(z) + g(z)u, \tag{2.9}$$

where $z \in \mathbb{C}^{n_z}$ is the state vector, $u \in \mathbb{C}$ is the input and $f, g \in H(\Omega, \mathbb{C}^n)$ are the state and input mapping, respectively. Then, the complex switching function is defined as the function $(\sigma(z))$ so that the system on the sliding surface (switching manifold) $\hat{\Sigma} \subseteq \Sigma$ evolves in the desired way, being $\Sigma := \{z \in \mathbb{C}^n; \sigma(z) = 0\}$ and $\hat{\Sigma}$ the sliding domain.

The equivalent control definition presented for real-valued systems can also be extended to the complex domain

$$u_{eq} = -\left(\frac{\partial \sigma}{\partial z} g(z)\right)^{-1} \frac{\partial \sigma}{\partial z} f(z), \tag{2.10}$$

where the reachability condition $\frac{\partial \sigma}{\partial z} g(z) \neq 0$ is also fulfilled.

On the other hand, the same Lyapunov candidate function as for real-valued sliding mode controllers can be considered when working in the complex domain. Hence,

$$V = \frac{1}{2}\bar{\sigma}(z)\sigma(z) \Rightarrow \dot{V} = \frac{1}{2}(\bar{\sigma}\dot{\sigma} + \dot{\bar{\sigma}}\sigma) = \mathrm{Re}(\bar{\sigma}\dot{\sigma}),$$
$$\dot{V} = \mathrm{Re}\left(\bar{\sigma}\frac{\partial \sigma}{\partial z}(f(z) + g(z)u)\right).$$

From here, the control action $u = u_{eq} + u_d$, with u_{eq} defined in (2.10), can be chosen so that it induces sliding motion, as

$$\dot{V} = \mathrm{Re}\left(\bar{\sigma}\frac{\partial \sigma}{\partial z}(g(z)u_d)\right) < 0. \tag{2.11}$$

Moreover, starting from (2.11), the reachability in finite-time of the sliding manifold can be studied. Finally, as it happens with the standard sliding mode controllers, complex-valued solutions to the chattering problem must be considered. The most similar approach to the saturation function presented previously, is the consideration of a boundary layer, which can be defined as

$$u = \begin{cases} -\kappa\frac{\sigma}{|\sigma|}, & |\sigma| > \epsilon \\ -\kappa\frac{\sigma}{\epsilon}, & |\sigma| \leq \epsilon, \end{cases} \tag{2.12}$$

where κ is a gain and $\epsilon \in \mathbb{R}^+$ is the thickness of the boundary layer. It is noticeable how the expression $\frac{z}{|z|} = e^{j\varphi_z}$, $z \neq 0$, is the representation of the sign function for complex variables. Apart from this strategy, some other alternatives are also presented in [Dòria-Cerezo et al., 2020], such as some hysteretic approximations.

2.3 Summary

In this chapter, a brief introduction to sliding mode controllers has been presented, describing its basis and defining terms such as the equivalent control action or the importance of a boundary layer in real applications. Then, sliding modes have been briefly extended to complex-valued dynamic systems, presenting a short introduction of this approach and the notation that will be used during the realisation of the project.

Chapter 3

Manipulator modeling

As it has been commented in Chapter 1, the main objective of this project is to design and analyze a complex-valued sliding mode control of a 2-link planar manipulator. In order to design a suitable controller, the first step is to understand how a 2-DOF robotic arm behaves and to get a proper mathematical model from it.

In this chapter, a brief introduction to robotics modeling and which parameters should be taking into account is discussed.

3.1 Kinematic model

In general terms, kinematics in robotics is referred to the study or representation of the movement of a robot according to a reference frame, regardless of its generation forces [Craig, 2005]. It describes in an analytical form the position, velocity, acceleration (an all of their time derivatives) as a function of time and their geometrical properties.

3.1.1 Denavit-Hartenberg Convention

A robotic manipulator is composed of a set of links connected together with joints. In the case studied, it will be considered that each link is connected by the use of an angular joint, which allows a relative rotation between them. This means each joint provides one degree of freedom to the robot. In order to build the kinematic model according to the reference frame (base of the robot), the relation (rotation and position) of each frame relative to its predecessor must be determined. This allows to set a relation between all the different links of the manipulator and to obtain the relative position and orientation of each point by using homogeneous transformations. Concatenating the corresponding transformation matrices of each link (containing the rotation component and the translation component) has as a result a global transformation matrix that links the Tool Center Position (TCP or end-effector) according to its base (reference frame).

These analysis and calculations to obtain the transformation matrix between frames and choosing each link's frame can be done in different ways, which makes it usual to use the Denavit-Hartenberg (DH) convention to do so [Denavit and Hartenberg, 1955]. Although there are more different methods that lead to the same result, this project only considers the DH methodology. In this convention, each homogeneous transformation (A) is calculated as a product of four basic transformations, which depend on four different parameters θ, α, a, d, explained later in the chapter.

In case of a manipulator, an orthonormal coordinate system is established for each joint (X_i, Y_i, Z_i), where $i = 1, 2, ..., n$, being n the degrees of freedom of the robot, following three specific rules:

1. The Z_{i-1} axis matches the joint axis, being its positive direction according to the positive value of its rotation θ_i.
2. The X_i axis is perpendicular to the Z_{i-1} axis.
3. The Y_i axis is defined to complete the orthonormal coordinates system.

Then, the four parameters needed to obtain the transformation matrix are defined as follows

1. θ_i is the angle in Z_{i-1} needed for X_{i-1} to reach X_i.
2. d_i is the distance between X_{i-1} and X_i in the Z_{i-1} direction.
3. a_i is the distance between Z_{i-1} and Z_i in the X_i direction.
4. α_i is the angle rotated in the X_i direction that makes Z_{i-1} reach Z_i.

This notation is represented in Figure 3.1.

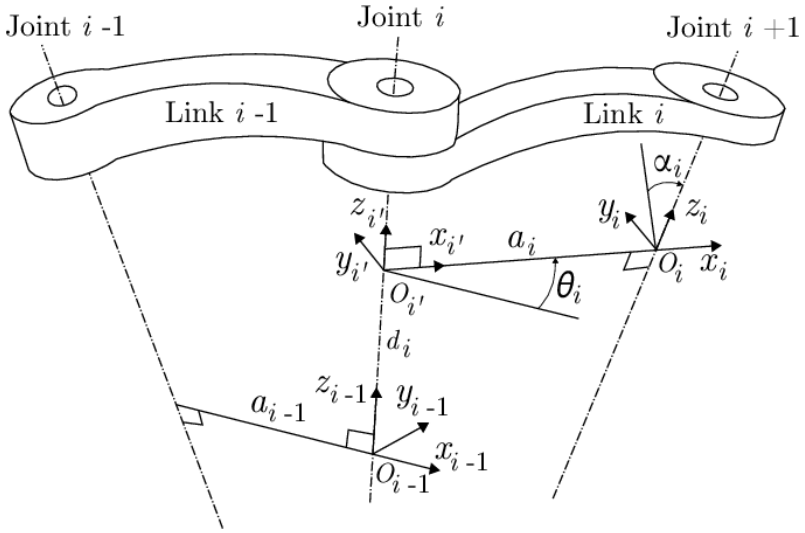

Figure 3.1: Denavit Hartenberg parameters definition [Huynh et al., 2018].

With all the parameters defined, the homogeneous transformation matrix ${}^{i-1}A_i$ between two links can be obtained as

$$ {}^{i-1}A_i = T_{(Z_{i-1},\theta_i)} T_{(0,0,d_i)} T_{(a_i,0,0)} T_{(X_i,\alpha_i)}, \tag{3.1} $$

where T is the transformation corresponding to the DH parameters (2 rotations and 2 translations). Developing the previous equation, a general matrix can be obtained as follows:

$$
{}^{i-1}A_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}.
$$

In the same way, the inverse of this transformation can be computed, getting as a result the following matrix

$$
{}^{i-1}A_i^{-1} = \begin{pmatrix} \cos\theta_i & \sin\theta_i & 0 & -a_i \\ -\cos\alpha_i\sin\theta_i & \cos\alpha_i\cos\theta_i & \sin\alpha_i & -d_i\sin\alpha_i \\ \sin\alpha_i\sin\theta_i & -\sin\alpha_i\cos\theta_i & \cos\alpha_i & -d_i\cos\alpha_i \\ 0 & 0 & 0 & 1 \end{pmatrix}.
$$

3.1.2 Forward kinematics

Forward kinematics (FK) refers to the use of kinematic equations to study the position and orientation of the end-effector of the manipulator relative to its base (reference frame) as a function of the variables of its previous joints [Paul, 1981]. This means moving from a joint space to a Cartesian space to get the transformation of the TCP coordinates.

To obtain these matrices that allow such transformation, the Denavit-Hartenberg convention is applied to the robotic arm studied, represented in Figure 3.2, in which the different frames are represented. It can be seen how there is one frame per joint and TCP ($\{O_1\}, \{O_2\}$) and the reference frame $\{O_0\}$ assigned to the base of the robot.

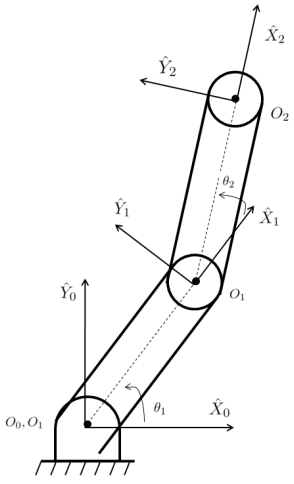

Figure 3.2: 2-DOF planar manipulator scheme and DH frame assignment.

As mentioned before, the manipulator is composed of two links of length l_1 and l_2, respectively. The angles rotated from link i with respect the previous frame are named as θ_i.

In order to get the transformation matrix from frame 0 to frame 2, the four DH parameters must be defined for each link. Table 3.1 summarizes this information below.

Table 3.1: Denavit Hartenberg parameters.

Link (i)	$\theta_\mathbf{i}$	$\alpha_\mathbf{i}$	$\mathbf{a_i}$	$\mathbf{d_i}$
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0

Now, the homogeneous transforms ${}^0A_1$ and ${}^1A_2$ can be obtained as

$$ {}^0A_1 = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.2} $$

$$ {}^1A_2 = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.3} $$

Concatenating these matrices the global homogeneous transform is obtained:

$$ {}^0T_2 = {}^0A_1\,{}^1A_2 = \begin{pmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.4} $$

Notice how the first 3×3 components of the matrix refer to the rotation of the end-effector from the reference frame and how the last column is the translation vector with respect the base of the robot; that is,

$$ x = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2), \tag{3.5} $$

$$ y = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2), \tag{3.6} $$

$$ z = 0. \tag{3.7} $$

3.1.3 Inverse kinematics

Inverse kinematics (IK) is referred to the opposite problem of forward kinematics, that is, obtaining the kinematic equations that, given the TCP coordinates, allow to get the different joint configurations to reach them [D'Souza et al., 2001]. This is a more complex problem to solve than forward kinematics and does not always have a valid solution, as there can be a configuration for the end-effector that requires of non-valid joint configurations to reach it. Apart from having to consider the working and joint space to determine the existence of a solution, it is also important to notice that one final configuration might have multiple solutions, which depend on the number of joints of the robot.

The methodology used to solve the inverse kinematics problem is by matching the components of the global transformation matrix ${}^0T_n$ to find the different joints configurations. In general terms, the algorithm used is the following:

$$
\begin{aligned}
&{}^0T_n = {}^0A_1 \ {}^1A_2 \ \dots \ {}^{i-1}A_i \ \dots \ {}^{n-1}A_n, \\
&{}^0A_1^{-1} \ {}^0T_n = {}^1A_2 \ \dots \ {}^{i-1}A_i \ \dots \ {}^{n-1}A_n, \\
&{}^1A_2^{-1} \ {}^0A_1^{-1} \ {}^0T_n = {}^2A_3 \ \dots \ {}^{i-1}A_i \ \dots \ {}^{n-1}A_n, \\
&\dots \\
&{}^{n-2}A_{n-1}^{-1} \ \dots \ {}^1A_2^{-1} \ {}^0A_1^{-1} \ {}^0T_n = {}^{n-1}A_n.
\end{aligned}
$$

Applying this inverse kinematics procedure presented and some algebra calculations, the joints configurations that allow to get the desired final position of the end-effector (x_2, y_2) can be obtained with the following equations:

$$\theta_2 = \arccos\left(\frac{x_2^2 + y_2^2 - l_1^2 - l_2^2}{2l_1 l_2}\right), \tag{3.8}$$

$$\theta_1 = \arctan\left(\frac{y_2}{x_2}\right) - \arctan\left(\frac{l_2 \sin(\theta_2)}{l_1 + l_2 \cos(\theta_2)}\right). \tag{3.9}$$

3.1.4 Jacobian matrix

The jacobian is a matrix of partial derivatives that allows to get a relationship between the linear and angular velocities of the TCP and the speeds of the rest of the joints. This means it is a way to move from a static problem to a dynamic one, as shown in (3.10).

$${}^0V_n = \begin{pmatrix} {}^0\dot{P}_n \\ {}^0\dot{W}_n \end{pmatrix} = J_n \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ . \\ . \\ . \\ \dot{\theta}_n \end{pmatrix} = J_n \dot{\Theta}_n, \tag{3.10}$$

where $J_n \in M_{6\times n}(\mathbb{R})$ is the jacobian matrix, ${}^0\dot{P}_n \in M_{3\times n}(\mathbb{R})$ is the linear velocity vector and ${}^0\dot{W}_n \in M_{3\times n}(\mathbb{R})$ is the angular speed vector with respect the reference frame $\{\hat{X}_0, \hat{Y}_0, \hat{Z}_0\}$. For the specific case of a 2-link planar manipulator, the jacobian matrix can be obtained as follows

$$J_2 = \begin{pmatrix} J_{21} & J_{22} \end{pmatrix} = \begin{pmatrix} {}^0z_0 \times {}^0P_2 & {}^0z_1 \times {}^1P_2 \\ {}^0z_0 & {}^0z_1 \end{pmatrix}, \tag{3.11}$$

being

$$\begin{aligned} {}^0z_0 &= [0, 0, 1], \\ {}^0z_1 &= [0, 0, 1], \\ {}^0P_2 &= [l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2), l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2), 0], \\ {}^1P_2 &= {}^0P_2 - {}^0P_1 = [l_2 \cos(\theta_1 + \theta_2), l_2 \sin(\theta_1 + \theta_2), 0], \end{aligned}$$

which can be obtained with the third and fourth column of ${}^0A_0$, ${}^0A_1$ and ${}^0A_2$. The resultant jacobian matrix is

$$J_2 = \begin{pmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}. \tag{3.12}$$

Then, the non-zero terms of the velocity of the TCP, according to (3.10), are

$$v_x = -l_1 \sin\theta_1 \dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2), \tag{3.13}$$

$$v_y = l_1 \cos\theta_1 \dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2), \tag{3.14}$$

$$\omega_z = \dot{\theta}_1 + \dot{\theta}_2. \tag{3.15}$$

Equation (3.10) shows that there is a possibility to get the joint configuration needed to reach some specific values of velocity, which means that a velocity control can be done by calculating the jacobian matrix for each sample time. Then, it is important to notice that this is not possible if the jacobian matrix is not invertible. This is called a singularity and represent those combinations of joint states that do not allow the movement in certain directions or areas [Donelan, 2010]. All manipulators have singularities in their workspace limits but there can also be some within the workspace scenario. In these configurations, at least one degree of freedom is lost.

3.2 Dynamic model

Up until now, only the static formulation of a manipulator has been discussed without taking into consideration the forces required to produce the movement. It is clear that, in order to control a manipulator, these forces cannot be neglected. Although there are several methods to model the dynamic behaviour of a robot, this project is not focused on modeling but in controlling the robotic manipulator. This is why only the Lagrange-Euler formulation will be discussed as it will be the methodology used to model the robot studied [Lagrange, 1853].

3.2.1 Lagrange-Euler formulation

It is known that the dynamic model of a robot can always be written as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda, \tag{3.16}$$

where

- $q \in \mathbb{R}^{n_q}$ is the vector of configurations of the robot and $\dot{q}$ and $\ddot{q}$ its derivatives.
- $M(q) \in M_{n_q \times n_q}(\mathbb{R})$ is its inertial matrix.
- $C(q,\dot{q}) \in M_{n_q \times n_q}(\mathbb{R})$ is the Coriolis matrix (or centrifugal forces matrix).
- $F(\dot{q}) \in \mathbb{R}^{n_q}$ is the friction term of the system.
- $G(q) \in \mathbb{R}^{n_q}$ is the gravity component.
- $\tau_d \in \mathbb{R}^{n_q}$ is used to model the possible disturbances in the system.
- $\tau \in \mathbb{R}^{n_u}$ is the joint torques vector.
- $B(q) \in M_{n_u \times n_u}(\mathbb{R})$ refers to the input.
- $A^T(q) \in M_{n_u \times n_u}(\mathbb{R})$ and $\lambda \in \mathbb{R}^{n_u}$ are the kinematic constraints and a vector of lagrangian multipliers, respectively.

Notice how the n_u and n_q dimensions must be the same, as they are related to the number of links of the robot ($n_q = n_u = n$). For this project, some of these terms will be neglected and a simpler dynamic model will be used:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = \tau. \tag{3.17}$$

Unlike other methods, the Lagrange-Euler formulation is based on establishing an energy balance to get the equations of motion (EOM) and obtain the matrices that form the model in (3.16). The first step is to define the Lagrangian, which is a scalar function obtained as the difference between the kinetic energy and the potential energy.

$$\mathcal{L}(q,\dot{q}) = E_c(q,\dot{q}) - E_p(q). \tag{3.18}$$

From here, the equation of motion can be computed with

$$\tau = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q}, \tag{3.19}$$

$$\tau = \frac{d}{dt}\left(\frac{\partial E_c}{\partial \dot{q}}\right) - \frac{\partial E_c}{\partial q} + \frac{\partial E_p}{\partial q}. \tag{3.20}$$

Finally, and before starting the corresponding calculations, some considerations about the planar manipulator studied are defined:

- Each link of the robot will be considered as an homogeneous bar, so its inertia can be calculated as $I_i = \frac{1}{12} m_i (l_i)^2$, being m_i the mass.
- The center of gravity (CoG) of each link is positioned in the middle ($l_{CoG} = l_i/2$).

Kinematic energy

The kinetic energy can be defined as the energy the manipulator has as a consequence of its movement. It can be calculated as superposition of the kinetic energy of each link and it is important to notice that two different terms must be taking into account (kinetic energy due to the linear velocity and kinetic energy due to the angular speed).

$$E_c = \sum_{i=1}^{n} \frac{1}{2} m_i (v_i)^2 + \frac{1}{2} I_i (\omega_i)^2. \tag{3.21}$$

For each link, its energy is calculated as follows:

$$E_{c1} = \frac{1}{2} m_1 (v_1)^2 + \frac{1}{2} I_1 (\omega_1)^2 = \frac{1}{2}\left(m_1 \frac{l_1^2}{4} + I_1\right)\dot{\theta}_1^2,$$

$$E_{c2} = \frac{1}{2} m_2 (v_2)^2 + \frac{1}{2} I_2 (\omega_1)^2 =$$

$$= \frac{1}{2} m_2 \left(l_1^2 \dot{\theta}_1^2 + \frac{l_2^2}{4}(\dot{\theta}_1 + \dot{\theta}_2)^2 + 2 l_1 \frac{l_2}{2} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_2)\right) + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2.$$

Expressing these equations in a matrix form, the inertial matrix from (3.16) can be obtained.

$$E_c = \frac{1}{2}\dot{\theta}^T M \dot{\theta}, \text{ being } M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix},$$

where

$$m_{11} = m_1 \frac{l_1^2}{4} + I_1 + m_2\left(l_1^2 + \frac{l_2^2}{4} + l_1 l_2 \dot{\theta}_1 \cos(\theta_2)\right) + I_2,$$

$$m_{12} = m_{21} = m_2 \frac{l_2^2}{4} + m_2 l_1 \frac{l_2}{2}\cos(\theta_2) + I_2,$$

$$m_{22} = m_2 \frac{l_2^2}{4} + I_2.$$

Potential energy

On the other hand, the robot also has a potential energy as a consequence of its weight. As it has been done with the kinetic energy, the robot's potential energy is calculated as the sum of the energy of each link.

$$E_p = E_{p1} + E_{p2} = m_1 g \frac{l_1}{2} \sin\theta_1 + m_2 g \left(\frac{l_2}{2} \sin\left(\theta_1 + \theta_2\right) + l_1 \sin\theta_1 \right). \tag{3.22}$$

In the previous equation, g is used for the gravity acceleration vector. Then, the gravity component of the model (3.16) can be obtained considering the partial derivatives, which are calculated as follows:

$$G_1 = \frac{\partial E_p}{\partial \theta_1} = m_1 g \frac{l_1}{2} \cos\theta_1 + m_2 g \left(\frac{l_2}{2} \cos\left(\theta_1 + \theta_2\right) + l_1 \cos\theta_1 \right), \tag{3.23}$$

$$G_2 = \frac{\partial E_p}{\partial \theta_2} = m_2 g \frac{l_2}{2} \cos\left(\theta_1 + \theta_2\right). \tag{3.24}$$

Equations of motion

The last two terms to completely define the dynamic model of the planar manipulator are the friction and the Coriolis matrices. For the first one, a constant parameter proportional to the rotational speed will be considered whereas determining the Coriolis matrix requires of some more complex calculations. In a general form, this matrix can be written as

$$C = \begin{pmatrix} c_{111}\dot{\theta}_1 + c_{121}\dot{\theta}_2 & c_{211}\dot{\theta}_1 + c_{221}\dot{\theta}_2 \\ c_{112}\dot{\theta}_1 + c_{122}\dot{\theta}_2 & c_{212}\dot{\theta}_1 + c_{222}\dot{\theta}_2 \end{pmatrix}, \tag{3.25}$$

being $c_{ijk} = c_{jik} = \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial \theta_i} + \frac{\partial m_{ki}}{\partial \theta_j} - \frac{\partial m_{ij}}{\partial \theta_k} \right)$. The results of each variable are

$$\begin{aligned} c_{111} &= 0, & c_{211} &= -m_2 l_1 \frac{l_2}{2} \sin\theta_2, \\ c_{121} &= -m_2 l_1 \frac{l_2}{2} \sin\theta_2, & c_{221} &= -m_2 l_1 \frac{l_2}{2} \sin\theta_2, \\ c_{112} &= m_2 l_1 \frac{l_2}{2} \sin\theta_2, & c_{212} &= 0, \\ c_{122} &= 0, & c_{222} &= 0. \end{aligned} \tag{3.26}$$

Now that all the components have been defined, the full dynamic model can be expressed in its matrix form

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \tau = M \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + C \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} + F \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} + G. \tag{3.27}$$

The equations of motion can be obtained by developing the previous matrix system into two different equations, one per joint torque τ_i.

$$\tau_1 = \left(\frac{m_1 l_1^2}{3} + \frac{m_2 l_2^2}{3} + m_2 l_1^2 + m_2 l_1 l_2 \cos(\theta_2)\right)\ddot{\theta}_1 + \left(\frac{m_2 l_2^2}{3} + \frac{m_2 l_1 l_2}{2}\cos(\theta_2)\right)\ddot{\theta}_2 -$$
$$- \frac{m_2 l_1 l_2}{2}\sin(\theta_2)\dot{\theta}_2^2 - m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 + \frac{m_1 g l_1}{2}\cos(\theta_1) + \frac{m_2 g l_2}{2}\cos(\theta_1+\theta_2) +$$
$$+ m_2 g l_1 \cos(\theta_1),$$
$$\tau_2 = \left(\frac{m_2 l_2^2}{3} + \frac{m_2 l_1 l_2}{2}\cos(\theta_2)\right)\ddot{\theta}_1 + \frac{m_2 l_2^2}{3}\ddot{\theta}_2 + \frac{m_2 l_1 l_2}{2}\sin(\theta_2)\dot{\theta}_1^2 + \frac{m_2 g l_2}{2}\cos(\theta_1+\theta_2).$$

Finally, some model disturbances τ_d will be added as Gaussian noise, with zero mean and non-zero deviation.

3.3 Complex-valued kinematic model

The forward kinematics model of a robotic manipulator can be extended to the complex domain, thing that will allow to design the proper complex-valued controller. Then, the position of the TCP of a n-link robot, $z_n \in \mathbb{C}$, can be obtained as

$$z_n = \sum_{i=1}^{n} l_i e^{j\sum_{p=1}^{i}\theta_p}, \tag{3.28}$$

which, for the particular case of this project (2-link manipulator), derives in the following equation:

$$z_2 = l_1 e^{j\theta_1} + l_2 e^{j(\theta_1+\theta_2)}, \tag{3.29}$$

whose time derivatives are

$$\dot{z}_2 = l_1 j\dot{\theta}_1 e^{j\theta_1} + l_2 j(\dot{\theta}_1+\dot{\theta}_2)e^{j(\theta_1+\theta_2)}, \tag{3.30}$$
$$\ddot{z}_2 = l_1 j\ddot{\theta}_1 e^{j\theta_1} - l_1 j\dot{\theta}_1^2 e^{j\theta_1} + l_2 j(\ddot{\theta}_1+\ddot{\theta}_2)e^{j(\theta_1+\theta_2)} - l_2 j(\dot{\theta}_1+\dot{\theta}_2)^2 e^{j(\theta_1+\theta_2)}. \tag{3.31}$$

The use of the inverse kinematics to get a relation between the joint variables and the position of any part of the links of a robot is no longer needed when using this formulation and, in terms of getting a control model, it can be combined it with its corresponding dynamic equations of motions, presented in (3.17). It is also important to mention that the complex representation reduces the dimension of the states studied, as it merges two states (both, vertical and horizontal, positions) in a single expression.

3.4 Summary

In this chapter, a brief introduction to robotics modeling was exposed. Both, the kinematic and dynamic models have been presented as well as the simplifications considered in this project. With the final equations of motion, the 2-link planar manipulator can be modeled by using the MATLAB/SIMULINK software [MATLAB, 2022] so the different controllers can be tested and analyzed. Moreover, the forward kinematics model has been extended to the complex domain so it can be used later on the design of the CV-SMC.

Chapter 4

Control of a 2-link planar manipulator

The main purpose of this project is to design and analyse a complex-valued sliding mode controller to track the tool-center position (TCP) of the manipulator to follow a reference signal. In this chapter, the main controller is designed, as well as three alternative strategies that will be used to compare and analyze the results obtained.

4.1 Feedback linearization

The first two controllers proposed are based on linear techniques. Then, the first step is to linearize the nonlinear model of the plant using a feedback linearization strategy [Mustafa, 2014]. This approach implies transforming the nonlinear system into an equivalent linear system through a change of variables ($z = \Phi(x)$) and a suitable control input ($u = \alpha(x) + \beta(x)v$). The feedback linearization control input proposed to make this change is

$$u = C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + M(\theta)v. \tag{4.1}$$

Applying these input signals, the system transforms from the nonlinear model (4.1) to the following linear system

$$\dot{z} = Az + Bv = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v, \tag{4.2}$$

$$y = Cz + Dv = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}. \tag{4.3}$$

This transformation allows to design a linear controller to track the position of the TCP of the robot to follow the desired trajectory. The system modelling and control scheme can be seen in Figure 4.1.

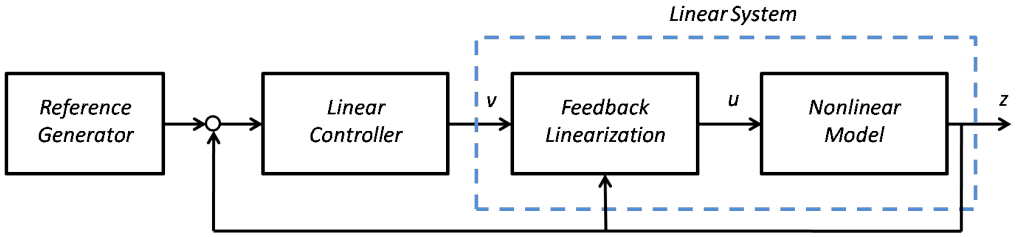

Figure 4.1: General system scheme.

4.1.1 State feedback control

The first controller proposed is a state feedback control. In order to directly track the position of the end-effector, this project proposes to change the state vector by adding the final Cartesian coordinates of this configuration instead of the joint angles. Combining equations (3.5) and (3.6) with equations (3.13) and (3.14), the time derivatives of the position (linear speed) can be computed as a function of the other states as follows:

$$\dot{x}_{TCP} = -y_{TCP}\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)\dot{\theta}_2,$$
$$\dot{y}_{TCP} = x_{TCP}\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2.$$

Then, the resulting nonlinear model is given by

$$\dot{z} = \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{x}_{TCP} \\ \dot{y}_{TCP} \end{pmatrix} = f(x,u) + \omega = \begin{pmatrix} v_1 \\ v_2 \\ -y_{TCP}\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)\dot{\theta}_2, \\ x_{TCP}\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2. \end{pmatrix}, \tag{4.4}$$

which can be linearized around its equilibrium point with the Jacobian matrices:

$$A = \left.\frac{\partial f_i}{\partial z_i}\right|_{(z_{eq},v_{eq})} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -y & -l_2 \sin(\theta_1 + \theta_2) & 0 & -\dot{\theta}_1 \\ x & l_2 \cos(\theta_1 + \theta_2) & \dot{\theta}_1 & 0 \end{pmatrix}_{(z_{eq},v_{eq})},$$

$$B = \left.\frac{\partial f_i}{\partial v_i}\right|_{(z_{eq},v_{eq})} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}^T_{(z_{eq},v_{eq})}.$$

Finally, the matrix $C = \frac{\partial h_i}{\partial z_i}$ stands for the direct measurements of the system. From the analysis around the desired position (x_d, y_d), the equilibrium point obtained is $z_{eq} = [\dot{\theta}_{1e}, \dot{\theta}_{2e}, x_e, y_e] = [0, 0, x_d, y_d]$ and $v_{eq} = [v_{1e}, v_{2e}] = [0, 0]$. It can be noticed how the relation between the desired coordinates of the end-effector and the equivalent joint angles

comes given by the inverse kinematics equations mentioned in Chapter 2. The final linear approximation results in

$$\Delta\dot{z} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -y_d & -l_2\sin(\theta_{1d}+\theta_{2d}) & 0 & 0 \\ x_d & l_2\cos(\theta_{1d}+\theta_{2d}) & 0 & 0 \end{pmatrix} \Delta z + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \Delta v,$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \Delta z + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \Delta v,$$

where $\Delta\epsilon$ stands for the difference between the state or input and the equilibrium point ($\Delta\epsilon = \epsilon - \epsilon_{eq}$). Now, the controllability can be checked with the A and B matrices:

$$\mathcal{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -y_d & -l_2\sin(\theta_{1d}+\theta_{2d}) & 0 & 0 & 0 & 0 \\ 0 & 0 & x_d & l_2\cos(\theta_{1d}+\theta_{2d}) & 0 & 0 & 0 & 0 \end{pmatrix},$$

which has full rank when $y_d\cos(\theta_{1d}+\theta_{2d}) \neq x_d\sin(\theta_{1d}+\theta_{2d})$. Geometrically, this corresponds to both links being aligned.

The equations of the state feedback controller (SFC) chosen for this system can be expressed as

$$v_1 = -\begin{pmatrix} K_{11} & K_{12} \end{pmatrix} \begin{pmatrix} \Delta\dot{\theta}_1 \\ \Delta x \end{pmatrix}, \tag{4.5}$$

$$v_2 = -\begin{pmatrix} K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \Delta\dot{\theta}_2 \\ \Delta y \end{pmatrix}, \tag{4.6}$$

which has been chosen this way to decouple the dynamics of both joints. Finally, the values of each of the gains can be obtained with the pole placement technique. It is important to remark that, for this project, the poles of the controllers will be selected to obtain a desired response that can be compared with the results of the CV-SMC, although this might imply not being the optimal tuning of the controllers.

Having to linearize the system around its equilibrium point is not a problem in case of the regulation problem, in which the desired position (and angles) is constant through all the time. This will not be the case of the tracking problem, as the desired position changes through the time and so does the jacobian matrices used to get the state-feedback controller. To solve this issue, a typical solution is to expand the model by adding an integral control component. The final scheme of the feedback-state controller is shown in Figure 4.2.

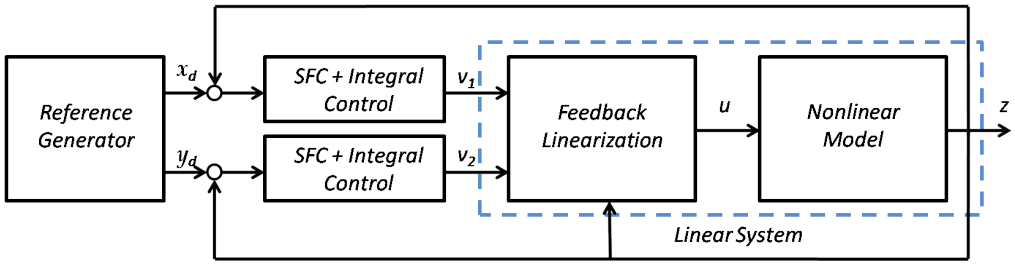

Figure 4.2: State-feedback controller scheme.

4.1.2 PID control

The system can also be controlled considering the double integrator model in (4.2). In this case, the linearized model is already decoupled and equivalent to 2 SISO systems, which makes a PID controller a suitable option. The overall control signal has the form

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}, \tag{4.7}$$

where K_p, K_i and K_d are the proportional, integral and derivative gains, respectively, and $e(t)$ is the error between the desired value and the current one. A brief explanation and discussion of each component is presented to evaluate the need of each part in the controller.

- Proportional: The control action is proportional to the current error. Increasing its gain tends to make the system react faster but it also causes overshoot.
- Integral: Helps reduce the steady-state error. Increasing its gain may produce oscillations in the system.
- Derivative: Anticipates the error, as it takes into account the rate of change of the error. It decreases the overshoot present in the system as it has a damping effect.

The qualitative effect of each action in the closed-loop response is summarized in Table 4.1.

Table 4.1: Effect of each action in the system's response.

Gain	Rise time	Overshoot	Steady-state	Settling time
K_p	Decrease	Increase	Decrease	Small change
K_i	Decrease	Increase	Decrease	Increase
K_d	Small change	Decrease	No change	Decrease

Then, taking into account the case studied, it is clear that the controller must have a proportional and integral action to set the tool center point of the manipulator in the desired position and maintain it there. However, it also seems proper to add a derivative component to avoid having a big overshoot in the system.

As it has been done previously, the gains are chosen to accomplish the desired response to be able to compare this controller with the others. The representation of this system in

the Laplace domain (transfer functions) is the following

$$\begin{pmatrix}\theta_1(s)\\ \theta_2(s)\end{pmatrix} = \begin{pmatrix}\frac{1}{s^2} & 0\\ 0 & \frac{1}{s^2}\end{pmatrix}\begin{pmatrix}V_1(s)\\ V_2(s)\end{pmatrix}. \tag{4.8}$$

The characteristic polynomial of the close-loop dynamics is the numerator resulting of $1 + C_i(s)\theta_i(s)/V_i(s)$, being $C_i(s)$ the controller of each joint. For the first link, a PD controller is proposed to avoid any overshoot in the response, whereas for the final link, a PID controller is assumed. Figure 4.3 shows the control scheme representation.

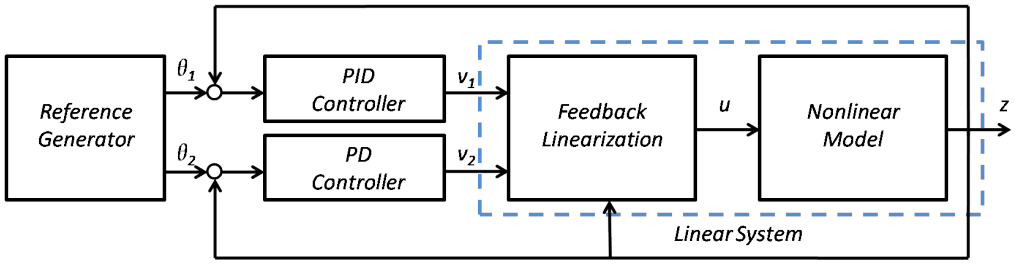

Figure 4.3: General system scheme.

The different values of the constants of the PID and PD controllers can be obtained using the pole placement method. In case of the PID controller, the linear system is a third order system and three poles have to be defined in order to get the different constants of the controller. On the other hand, for the PD controller, it is a two order system and only two poles must be defined. To simplify the tuning of the controller, this project proposes choosing a dominant pole (p_1), which would make the selection of the other ones automatic (i.e. $p_2 > 5p_1$ and $p_3 > 7p_1$). However, in order to get a proper response that can be compared to the other controllers, some other tools for tuning the parameters might be considered, as it could be the *PID Tuning Toolbox* of MATLAB.

Finally, in order to use the feedback linearization strategy effectively, it is important to remark the importance and need of knowing properly the system. The change of variables and the control input applied to transform the nonlinear model into an equivalent linear system are strictly related to the dynamic model of the system that wants to be controlled. In this particular case, it is mandatory a good knowledge of matrices M, C and G, as seen in (4.1). Then, if the knowledge or estimation of these matrices is not good enough, the resultant system will not be lineal and both strategies proposed will lead to undesired results. Moreover, it can be noted how the PID and the state-feedback controllers are based on an approximation around the working point. All of these considerations will be taken into account for the qualitative analysis.

4.2 Sliding mode controller

Apart from the linear strategies presented, a nonlinear sliding mode controller is proposed in order to deal with the nonlinearities of the robotic manipulator dynamics. Hence, the second controller designed to be used to compare the response of the CV-SMC is a real-valued sliding mode controller (RV-SMC). From equation (3.17), which contains all the dynamics of the system considered, the control model used to design the controllers can be extracted. To do so, the disturbance torques added (τ_d) will be neglected as they are

considered unknown variables. Then, the equations of motion used to design all the control algorithms are the following:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau. \tag{4.9}$$

4.2.1 Joint angles control

The control problem is defined such that the pointer position tracks a certain reference trajectory (z_{2d}). Then, the switching surface σ considered will be defined as a function of the error between the desired value and the current one. It is necessary to take into account that the dimension of this function will not be the same as for the CV-SMC, as both angles have to be considered as separate states. Then, the switching function is defined as

$$\sigma = e + \lambda\dot{e}, \tag{4.10}$$

where e stands for the error between both angles of links of the manipulator and its desired values

$$e = (q - q_d) = \begin{pmatrix} \theta_1 - \theta_{1d} \\ \theta_2 - \theta_{2d} \end{pmatrix}, \tag{4.11}$$

and its time derivative as

$$\dot{\sigma} = \dot{e} + \lambda\ddot{e}. \tag{4.12}$$

Then, defining the same Lyapunov candidate function as presented in Chapter 2,

$$V = \frac{1}{2}\sigma^T\sigma, \tag{4.13}$$

a control law that allows the existence of a sliding mode can be obtained ensuring that

$$\dot{V} = \sigma^T\dot{\sigma} < 0. \tag{4.14}$$

Replacing (4.11) in (4.14)

$$\dot{V} = \sigma^T(\dot{e} + \lambda\ddot{e}) = \sigma^T(\dot{e} - \lambda\ddot{q}_d + \lambda\ddot{q}),$$

which considering (4.9) can be extended to

$$\dot{V} = \sigma^T(\dot{e} - \lambda\ddot{q}_d) + \sigma^T\lambda M^{-1}(u - C\dot{q} - G).$$

Then, that expression can be reorganized as follows:

$$\dot{V} = \sigma^T(\dot{e} - \lambda\ddot{q}_d - \lambda M^{-1}(C\dot{q} + G)) + \sigma^T\lambda M^{-1}u.$$

For this analysis, the use of an equivalent control u_{eq} is considered. Hence, the control law obtained will have the form of $u = u_{eq} + u_d$. Considering its definition in (2.6)

$$\dot{\sigma} = 0 \Rightarrow e - \lambda\ddot{q}_d + \lambda M^{-1}(u_{eq} - C\dot{q} - G) = 0, \tag{4.15}$$

$$u_{eq} = M(\ddot{q}_d - \frac{1}{\lambda}\dot{e}) + C\dot{q} + G. \tag{4.16}$$

Substituting the full control law

$$\dot{V} = \sigma^T \lambda M^{-1} u_d < 0 \Rightarrow u_d = -\frac{k}{\lambda} M \text{sign}(\sigma), \tag{4.17}$$

where the sign() function stands for the sign of each of its components. Then, the full expression of the derivative of the Lyapunov function is

$$\dot{V} = -k|\sigma|,$$

which ensures being negative for $k > 0$.

As explained in Chapter 3, it is important to consider that for a real implementation, the use of a saturation function as in (2.8) might be necessary. Although the scope of this project only considers simulation, it is an important fact that must be taken into account when implementing this control algorithm in a robot manipulator. It is also important to notice that the inverse kinematics transformation may be used to pass from the Cartesian coordinates (x, y) to the joint space (q_1, q_2).

The diagram scheme of the real-valued sliding mode controller (RV-SMC) implementation is represented in Figure 4.4.

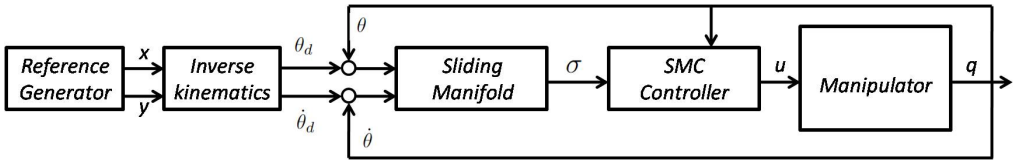

Figure 4.4: Real-Valued sliding mode controller scheme.

4.2.2 Coordinates position control

However, although the approach of controlling the different angles is the one most commonly used, this project presents an alternative to control directly the position of the end-effector of the manipulator. By doing this, the inverse kinematic calculation requirement is no longer needed and only the use of forward kinematics is used. It is also noticeable how this approach allows a more reliable comparative with the CV-SMC, as both control directly the position of the TCP instead of the joint angles.

In Chapter 3, the position of the tool of the manipulator is obtained in (3.5) and (3.6) from its kinematic model. Starting from this point, its first and second time derivatives, which will be needed to design this new controller, can be obtained as

$$\dot{x}_{TCP} = -y_{TCP}\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)\dot{\theta}_2,$$
$$\dot{y}_{TCP} = x_{TCP}\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2,$$
$$\ddot{x}_{TCP} = -\dot{y}_{TCP}\dot{\theta}_1 - y_{TCP}\ddot{\theta}_1 - l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 - l_2 \sin(\theta_1 + \theta_2)\ddot{\theta}_2,$$
$$\ddot{y}_{TCP} = \dot{x}_{TCP}\dot{\theta}_1 + x_{TCP}\ddot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 + l_2 \cos(\theta_1 + \theta_2)\ddot{\theta}_2.$$

The same procedure presented before will be followed to design this new approach. Then, the new manifold function in terms of the error (e) and its time derivative can be computed as follows:

$$\sigma = e + \lambda\dot{e} = \begin{pmatrix} x - x_d \\ y - y_d \end{pmatrix} + \lambda \begin{pmatrix} \dot{x} - \dot{x}_d \\ \dot{y} - \dot{y}_d \end{pmatrix} \tag{4.18}$$

$$\dot{\sigma} = \dot{e} + \lambda\ddot{e} = \begin{pmatrix} \dot{x} - \dot{x}_d \\ \dot{y} - \dot{y}_d \end{pmatrix} + \lambda \begin{pmatrix} \ddot{x} - \ddot{x}_d \\ \ddot{y} - \ddot{y}_d \end{pmatrix}, \tag{4.19}$$

being from now on (x, y) the coordinates of the TCP of the planar manipulator and (x_d, y_d) its desired target position. Using the time derivatives of this position the ones presented previously, the derivative of the switching surface can be expressed as $\dot{\sigma} = \Psi + \Gamma\ddot{q}$, where

$$\Psi = \begin{pmatrix} \dot{x} - \dot{x}_d - \lambda(\ddot{x}_d + \dot{y}\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2) \\ \dot{y} - \dot{y}_d - \lambda(\ddot{y}_d - \dot{x}\dot{\theta}_1 + l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2) \end{pmatrix} \tag{4.20}$$

$$\Gamma = \lambda \begin{pmatrix} -y & -l_2 \sin(\theta_1 + \theta_2) \\ x & l_2 \cos(\theta_1 + \theta_2) \end{pmatrix}. \tag{4.21}$$

Substituting the equation of motion modeled in (4.9), the equivalent control signal can be calculated as

$$u_{eq} = -M\Gamma^{-1}\Psi + C\dot{q} + G. \tag{4.22}$$

Now, considering the same Lyapunov function

$$V = \frac{1}{2}\sigma^T\sigma \Rightarrow \dot{V} = \sigma^T\dot{\sigma} = \sigma^T(\Psi + \Gamma\ddot{q}), \tag{4.23}$$

and substituting dynamic model of the system presented in (4.9)

$$\dot{V} = \sigma^T(\Psi + \Gamma M^{-1}(u - C\dot{q} + G)), \tag{4.24}$$

a control signal ($u = u_{eq} + u_d$) that fulfils the condition of existence of a sliding mode can be obtained if

$$\dot{V} = \sigma^T\Gamma M^{-1}u_d < 0. \tag{4.25}$$

By considering the discontinuous control signal as

$$u_d = -kM\Gamma^{-1}\text{sign}(\sigma), \tag{4.26}$$

the condition of existence of a sliding mode gets fulfilled for $k > 0$, as $\dot{V} = -k|\sigma|$.

4.3 Complex-valued SMC

4.3.1 Equivalent control design

Once all the real-valued controllers have been introduced, the control problem can be extended to the complex space to track the pointer position of the robotic manipulator. It can be seen how, working in the complex domain, allow to control the position in two axis as one single state (i.e, $z = x + yj$), but this is not possible when working in a real-valued space. The control problem will also use the same model introduced in (4.9). The output, given by (3.29), has relative degree two with the control input in (3.12). Then, the following second-order complex switching manifold is proposed

$$\sigma = (z_2 - z_{2d}) + \lambda(\dot{z}_2 - \dot{z}_{2d}), \tag{4.27}$$

whose time derivative can be expressed as

$$\dot{\sigma} = (\dot{z}_2 - \dot{z}_{2d}) + \lambda(\ddot{z}_2 - \ddot{z}_{2d}) = \Psi + \xi^T\ddot{q}, \tag{4.28}$$

being

$$\Psi = l_1 j\dot{\theta}_1 e^{j\theta_1} + l_2 j(\dot{\theta}_1 + \dot{\theta}_2)e^{j(\theta_1+\theta_2)} - \dot{z}_{2d} - \lambda(\ddot{z}_{2d} + l_1 j\dot{\theta}_1^2 e^{j\theta_1} + l_2 j(\dot{\theta}_1 + \dot{\theta}_2)^2 e^{j(\theta_1+\theta_2)}),$$

$$\xi = j\begin{pmatrix} l_1 e^{j\theta_1} + l_2 e^{j(\theta_1+\theta_2)} \\ l_2 e^{j(\theta_1+\theta_2)} \end{pmatrix},$$

where $\Psi \in \mathbb{C}$ and $\xi \in \mathbb{C}^2$. Using the Lyapunov candidate presented in Chapter 2, its time derivative in the complex domain can be written as

$$\dot{V} = \frac{1}{2}(\bar{\sigma}\dot{\sigma} + \dot{\bar{\sigma}}\sigma) = \text{Re}(\bar{\sigma}\dot{\sigma}). \tag{4.29}$$

Substituting (3.30) and (3.31) in (4.29) and doing some basic algebra, the following results are obtained:

$$\begin{aligned}\dot{V} &= \text{Re}(\bar{\sigma}\dot{\sigma}) = \text{Re}(\bar{\sigma}(\Psi + \xi^T\ddot{q})) = \text{Re}(\bar{\sigma}(\Psi + \xi^T M^{-1}(u - C\dot{q} - G))) \\ &= \text{Re}(\bar{\sigma}(\Psi - \xi^T M^{-1}(C\dot{q} + G))) + \text{Re}(\bar{\sigma}\xi^T M^{-1}u).\end{aligned}$$

Breaking down the complex sliding surface into its modulus and argument,

$$\dot{V} = |\sigma|\phi + |\sigma|\gamma^T M^{-1}u, \tag{4.30}$$

being

$$\gamma = \mathrm{Re}(e^{-j\varphi_\sigma}\xi),$$
$$\phi = \mathrm{Re}(e^{-j\varphi_\sigma}(\Psi - \xi^T M^{-1}(C\dot{q} + G))).$$

According to its previous definition, the expression of the equivalent control can be obtained as

$$\dot{\sigma} = 0 \Rightarrow \Psi + \xi^T M^{-1}(u_{eq} - C\dot{q} - G) = 0 \Rightarrow u_{eq} = (\xi^T M^{-1})^{-1}(\xi^T M^{-1}(C\dot{q} + G) - \Psi). \quad (4.31)$$

It can be noticed how, this calculation must be done with a pseudo-inverse method, as it is not a square matrix. Then, defining the control law as $u = u_{eq} + u_d$,

$$\dot{V} = |\sigma|\phi + |\sigma|\gamma^T M^{-1}(u_{eq} + u_d)$$

and using u_{eq} from (4.31)

$$\dot{V} = |\sigma|\gamma^T M^{-1} u_d.$$

From here, the discontinuous control signal can be chosen as

$$u_d = -k\gamma, \quad (4.32)$$

with $k > 0$, that ensures $\dot{V} < 0$ if $\gamma^T\gamma \neq 0$.

4.3.2 Simplified control design

However, in order to avoid the use of the equivalent control and the pseudo-inverse matrix calculation required, a second approach for the control law is studied in this project. Instead of considering the equivalent control term, only the discontinuous part obtained before will be used ($u = -k\gamma$). Then, the constant k must be bounded to guarantee the existence of a sliding mode.

Starting from the Lyapunov function defined

$$\dot{V} = |\sigma|\phi - k|\sigma|\gamma^T M^{-1}\gamma,$$

the existence of a sliding mode is ensured for

$$k > \frac{\phi}{\gamma^T M^{-1}\gamma}.$$

Being M a symmetric matrix implies that its inverse will also be symmetric. By definition, it can be proven that a real square and symmetric matrix can be diagonalized in an orthogonal base as

$$M^{-1} = QDQ^{-1}, \tag{4.33}$$

where $Q^T = Q^{-1}$ and D is diagonal matrix with eigenvalues λ_1 and λ_2, being $|\lambda_1| \leq |\lambda_2|$. Now, the previous condition to guarantee the existence of a sliding mode can be bounded as

$$\dot{V} = |\sigma|(\phi - k\gamma^T M^{-1}\gamma) \leq |\sigma|(\phi - k\lambda_1\gamma^T\gamma). \tag{4.34}$$

Then, $\dot{V} < 0$ can be achieved with

$$k > \frac{\phi}{\lambda_1\gamma^T\gamma}. \tag{4.35}$$

Finally, the vector γ is a function of the link rotation angles (θ_1 and θ_2) and the argument of the complex manifold. Considering the bounds of the trigonometric functions, the following bounds are obtained:

$$\gamma_{min} \leq \gamma^T\gamma \leq \gamma_{max},$$

$$0 \leq \gamma^T\gamma \leq (l_1^2 + 2l_1l_2 - 2l_2^2).$$

Now, these limits can be added to expression (4.35) in order to choose an admissible gain of the sliding mode controller as

$$k > \frac{\phi}{\lambda_1\gamma_{min}}. \tag{4.36}$$

The condition to ensure the existence of a sliding mode is guaranteed with $\dot{V} < 0$. On the other hand, the finite-time convergence can be proved as follows.

From

$$\dot{V} = |\sigma|(\phi - k\gamma^T M^{-1}\gamma),$$

and defining $\mu = (\phi - k\lambda_1\gamma_{min})$, the time derivative of the Lyapunov function can be bounded as

$$\dot{V} \leq -\mu|\sigma|.$$

Then, considering its definition introduced in Chapter 2,

$$\dot{V} \leq -\mu\sqrt{2}\sqrt{V},$$

which leads to

$$\sqrt{V(t)} \leq \frac{-\mu\sqrt{2}}{2}t + \sqrt{V(0)}.$$

Then, Σ is reached ($V(t) = 0$) in finite time $t - t_0 \leq t_r$, where $t_r = \frac{\sqrt{2}\sqrt{V(0)}}{\mu}$ and $\mu > 0$. The final control scheme designed is represented in Figure 4.5.

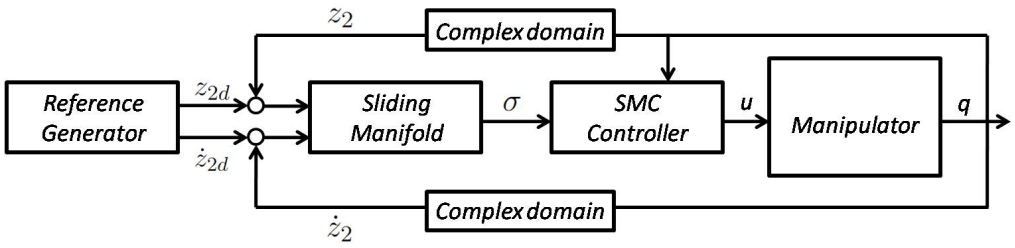

Figure 4.5: General system scheme.

It is also important to take into account that an implementation of this controller in a real robotic manipulator may require the use of a boundary layer as defined in (3.8). The use of it would avoid discontinuities in the control action and a better response of the controller in real-life applications. However, as this project is based mainly in simulation, this term is not strictly mandatory, as the chattering effect will not cause any problem during the simulations.

4.4 Summary

In this chapter, four different controllers have been proposed. On one hand, two linear controllers (state feedback control and PID control) have been designed based on a feedback linearization strategy; on the other hand, two nonlinear sliding mode controllers have been presented to track the TCP of the planar manipulator: the first one with real-based variables and the second one with a complex-valued scheme. All the controllers' gains will be chosen to get the same time response so they can be compared between them. In the next chapter, the controllers will be modelled and implemented in MATLAB and SIMULINK so they can be simulated to get the proper results to analyze them.

Chapter 5

Simulation and comparison

The final step once all the controllers have been designed is to simulate them to evaluate their performance and be able to compare them and extract some proper conclusions. During this chapter, all the different tests carried on are defined and discussed and a comparison between all the controllers is done according to the results obtained. All the implementation and simulations are done with MATLAB and SIMULINK, as well as the post-analysis.

5.1 Tests description

The first thing to define are the different tests that will be used to evaluate the controllers designed during this work. According to the reference signal proposed, the control problems that exist in manipulators are mainly the regulation and tracking problems:

- Regulation problem: The reference signal is constant, which basically means that the time derivatives of this signal are zero during all the test.
- Tracking problem: The reference signal to be followed is time-dependant and bounded, and so are its time derivatives.

In this case, the reference signal is the TCP of the manipulator (and consequently, also the joint angles), the first time derivative is its velocity and the second derivative is the acceleration. It is important to take into account how all the reference signals must be continuous during all the time, as well as their derivatives. To design them properly, a commonly used method is base them on splines to ensure they are continuous and derivable during all the duration of the test [Egerstedt and Martin, 2001]. However, this project will not get into detail with this technique and will simply use different signals that fulfil these requirements.

Moreover, for each test problem, some different variables will be added to test the efficiency and robustness of the controllers:

- Model disturbance: As mentioned during Chapter 3, the model can be complemented with an unknown input disturbance (τ_d) that represents the uncertainty of the model. Here, it will be modelled as a constant disturbance.
- Parametric uncertainties: The parameters used to model all the controllers might differ from the real ones. In this case, the uncertainty will be added to the masses of each link.

- Sensor noise: The sensor used to measure the input variables of the controllers will also be complemented with a small white noise. It will be introduced to the system as a Gaussian noise, with zero mean and non-zero deviation.
- Initial position: The response of the controller might vary depending on the initial coordinates of the manipulator. To take this fact into account, different initial conditions will be used for each test.

Then, the tests done to analyze the controllers and compare them are the following:

1. Regulation: Regulation test for a reference position of $(x, y) = (0.0866, 0.25)$ m (which corresponds to the angles $(\theta_1, \theta_2) = (\frac{\pi}{6}, \frac{\pi}{3})$ rad).
2. Tracking: Tracking test for a reference signal according to the one shown in Figure 5.1.

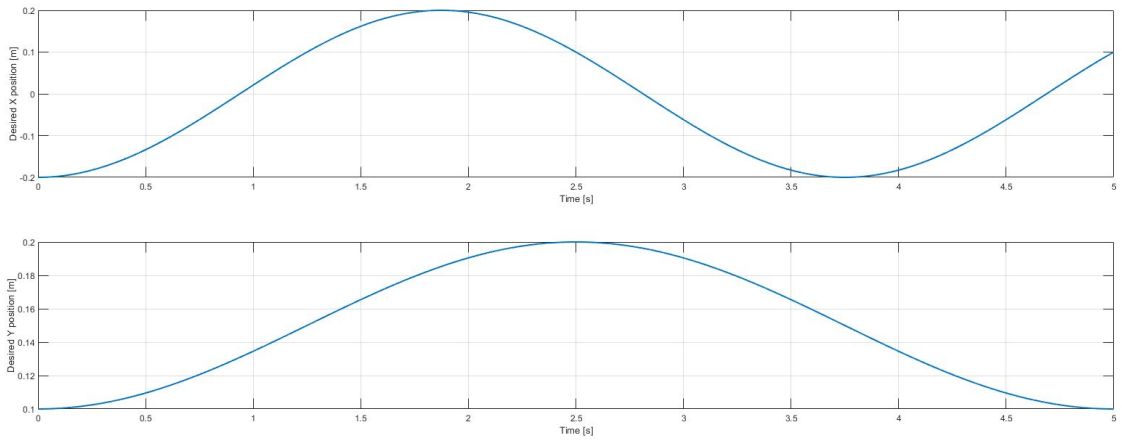

Figure 5.1: Desired position for the tracking test.

Each test of the ones described above will be simulated for an ideal behavior, for input disturbance (model disturbance will be added), for parametric uncertainty, for noise sensitivity (sensor noise will be added) and for another different initial condition. However, the first step will be tuning all the gains of each strategy to have approximately the same settling time for the worst initial condition, so all the controllers are equally adjusted in order to be fairly compared.

To sum up, the different control strategies simulated are summarized below.

1. Complex-valued sliding mode controller (CV-SMC): SMC with the control law $u = -k\gamma$ according to (4.32).
2. Real-valued sliding mode controller (RV-SMC): SMC with the control law $u = u_{eq} + u_d$, being $u_{eq} = -M\Gamma^{-1}\Psi + C\dot{q} + G$ and $u_d = -kM\Gamma^{-1}\text{sign}(\sigma)$, according to (4.22) and (4.26).
3. State feedback controller (SFC): Control strategy based on (4.5) and (4.6) and a previous feedback linearization to deal with the non-linearities of the model.
4. PID strategy: A PID and a PD control law according to (4.7) and a feedback linearization.

Finally, it is also important to mention the kinematic and dynamic parameters of the robot modeled and simulated to test the controllers, which are presented in Table 5.1

Table 5.1: Robot parameters.

Link (i)	Mass [kg]	Length [m]	Inertia [kg m^2]
1	1.5	0.1	0.0013
2	2	0.2	0.0067

5.2 Key Performance Indicators

Once all the tests have been defined and before starting the simulations to compare the four controllers proposed it is important to describe the Key Performance Indicators (KPIs). These parameters and variables will allow a fair comparison and see the advantages and drawbacks of the CV-SMC against the other control strategies used. Although all the indicators will provide a quantitative analysis, it is important to take into account that some qualitative variables will be also used. The quantitative KPIs are the following:

- Average of the errors: Defining the error between the current value and the desired one as $e_i(t) = x_i(t) - x_{id}(t)$, the average of the errors is calculated as $\frac{1}{N}\sum_{i=1}^{N}|e_i(t)|$.
- Area of the infinity norm of the error vector: The second indicator related to the error against the target position value will be calculated as $\int_0^T ||e(t)||_\infty dt$.
- Infinity norm of the control input vector: Apart from the error, the control signal required to achieve the desired value is an important indicator to evaluate each control strategy. Thus, the third KPI will be the infinity norm of the control input vector and it will be calculated according to $||u(t)||_\infty = \max_i |u_i(t)|$.
- Area of the infinity norm of the control input vector: The last indicator will be the area of the third indicator and it will be calculated as $\int_0^T ||u(t)||_\infty dt$.

These KPIs will be used to compare the response of the four controllers studied in this project. However, for each simulation related to the SMC, the sliding surface will also be plotted so a better analysis can be executed.

5.3 Regulation problem results

Now that all the considerations related to the tests and comparison factors have been completely discussed, the four controllers can be simulated and analyzed (the CV-SMC, RV-SMC, PID and SFC). The first test, as mentioned in section 5.1, is the regulation problem.

The desired position for the first regulation test is the one obtained from setting the joint angles to $(\theta_1, \theta_2) = \left(\frac{\pi}{6}, \frac{\pi}{3}\right)$ rad, which corresponds to the TCP of $(x, y) = (0.0866, 0.25)$ m. The gains and tuning parameters have been adjusted so all of them have approximately the same settling time of $t_s \simeq 0.3$ s and they are the following:

1. CV-SMC: Control parameters: $\lambda = 0.05$ and $k = 750$.
2. RV-SMC: Control parameters: $\lambda = 0.05$ and $k = 25$.
3. PID: Control gains: $K_{pPID} = 500$, $K_{iPID} = 10$, $K_{dPID} = 50$, $K_{pPD} = 500$, $K_{dPD} = 50$.
4. SFC: Control gains: $K_{11} = 255.5$, $K_{12} = -46352$, $K_{21} = 196.7$, $K_{22} = -44838$.

However, having two different states (X and Y coordinates) makes difficult to tell the time in which they reach the target. As this is the key parameter to tune the controllers, both states have been reduced to a single state, being this last the modulus of the position ($\sqrt{x^2 + y^2}$). The test has been simulated during 1 second and using a solver step-size of 1×10^{-4} s, beginning in the initial condition $(x_0, y_0) = (-0.2932, 0.0518)$ m. Figure 5.2 shows the evolution of the position modulus in time, during all the test. It can be seen how all the four controllers reach the target value in almost the same time ($t_s \simeq 0.3$) s and it can be said that all controllers can be fairly compared.

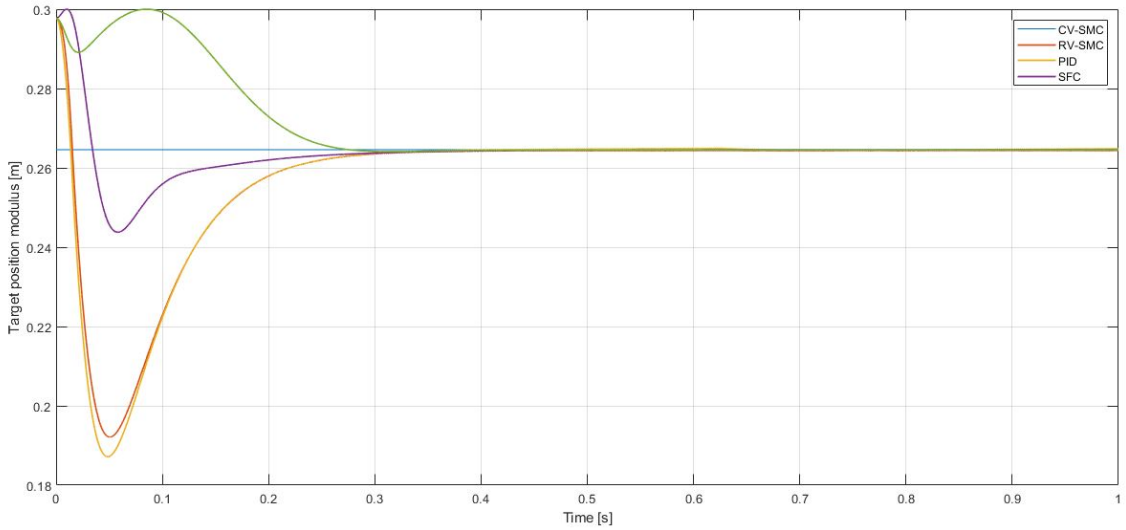

Figure 5.2: Time response of the modulus of the position.

In Figure 5.3, the evolution of the vertical position of the end-effector is represented. It can be seen how all the four controllers are able to maintain the desired position once it has been reached. In order to achieve this settling time requirement, both the PID and the SFC present a small overshoot, being the absolute value of the error higher for the SFC.

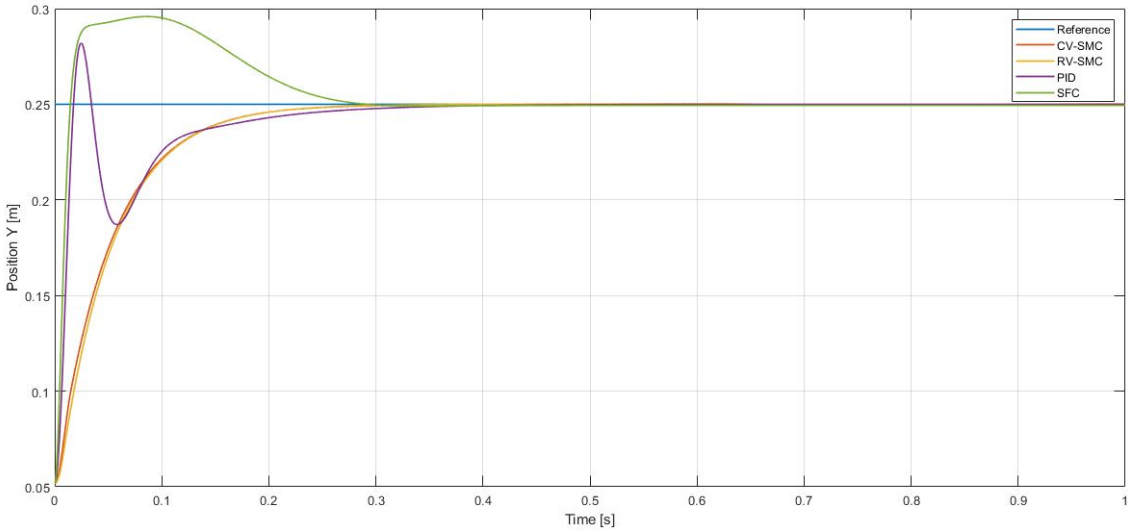

Figure 5.3: Time response of Y position.

In the same way, Figure 5.4 shows the time evolution of the position in the horizontal axis. For this variable, the only strategy that presents an overshoot is the PID controller. The difference between the SMCs is quite small and the exact values of the error will be analyzed properly in section 5.5.

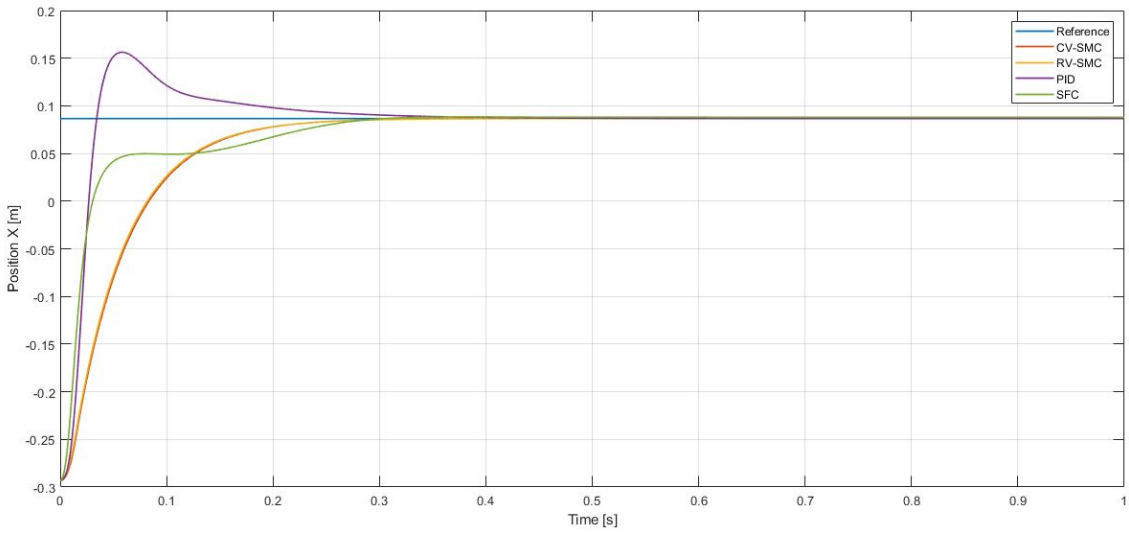

Figure 5.4: Time response of X position.

It is also important to analyze the control input signal required to achieve this performance. In Figure 5.5, all of the control actions for both joint actuators are represented. It can be seen how the control action related to the actuator of the second joint is always lower than the one related to the first joint. The higher values are also obtained for the state-feedback controller, whereas the lower absolute values appear to be the SMCs signals.

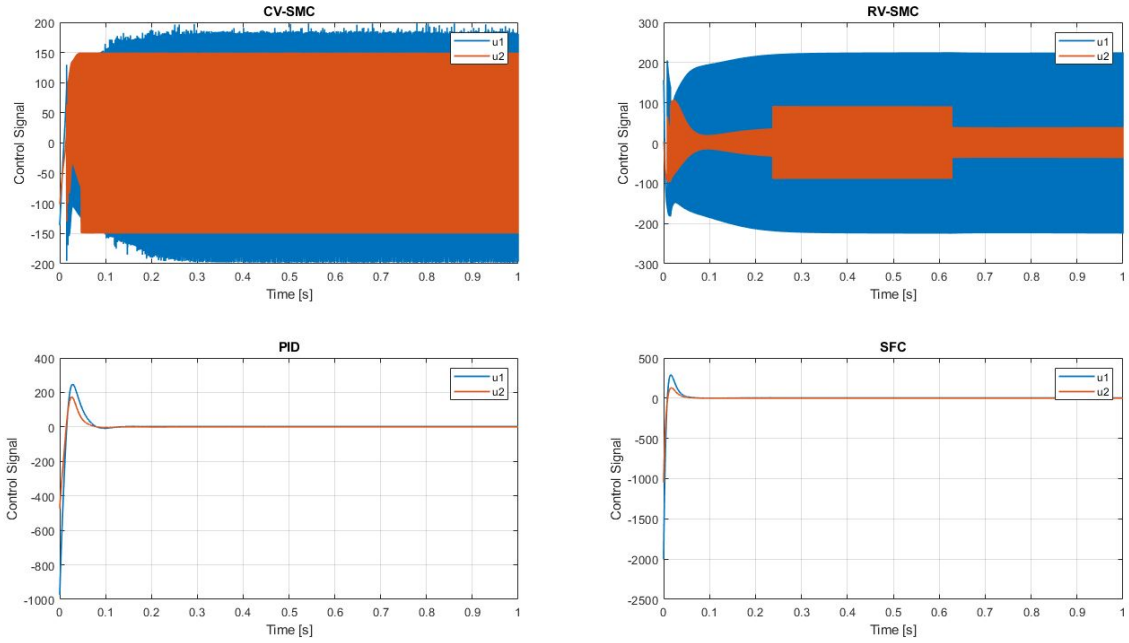

Figure 5.5: Time response of the control signals.

However, comparing the profiles of the input signals of the PID and the SFC with the inputs of the SMCs in a regulation problem might not be adequate, their maximum and minimum values can give some interesting information when designing the actuators. Finally, for the case of the two sliding mode controllers (CV-SMC and RV-SMC), the evolution of the switching manifold function is shown in Figures 5.6 and 5.7, respectively. The error related to the horizontal variables (position and velocity) is included in the first component of the RV-SMC manifold and in the real part of the complex-valued surface for the CV-SMC, whereas the error for the vertical components is represented in the imaginary term and second term of the switching manifold function. The modulus of both components is also represented to have a better understanding of their behaviour.

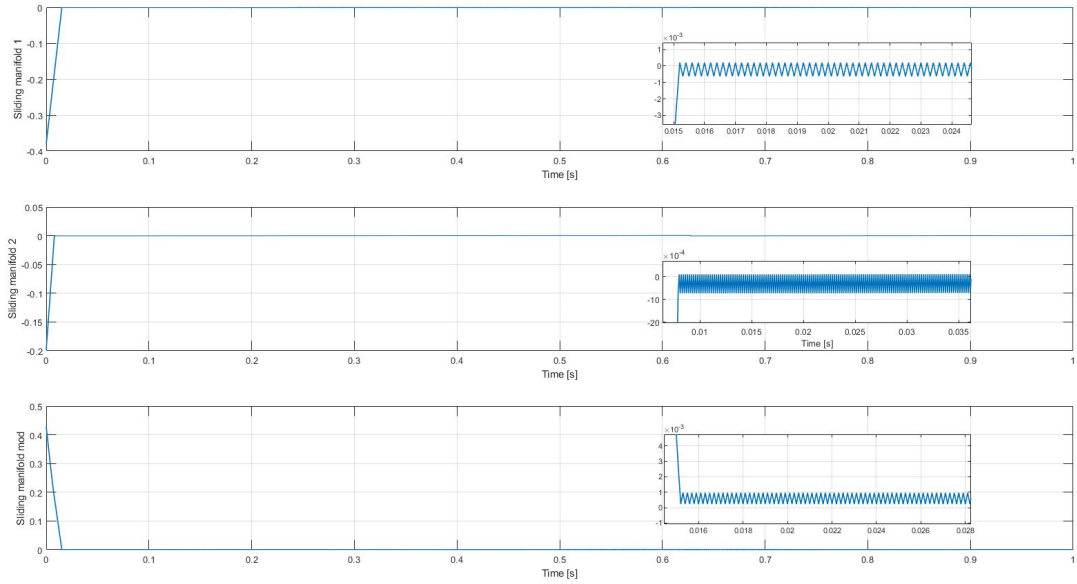

Figure 5.6: Time evolution of the switching manifold for the RV-SMC.

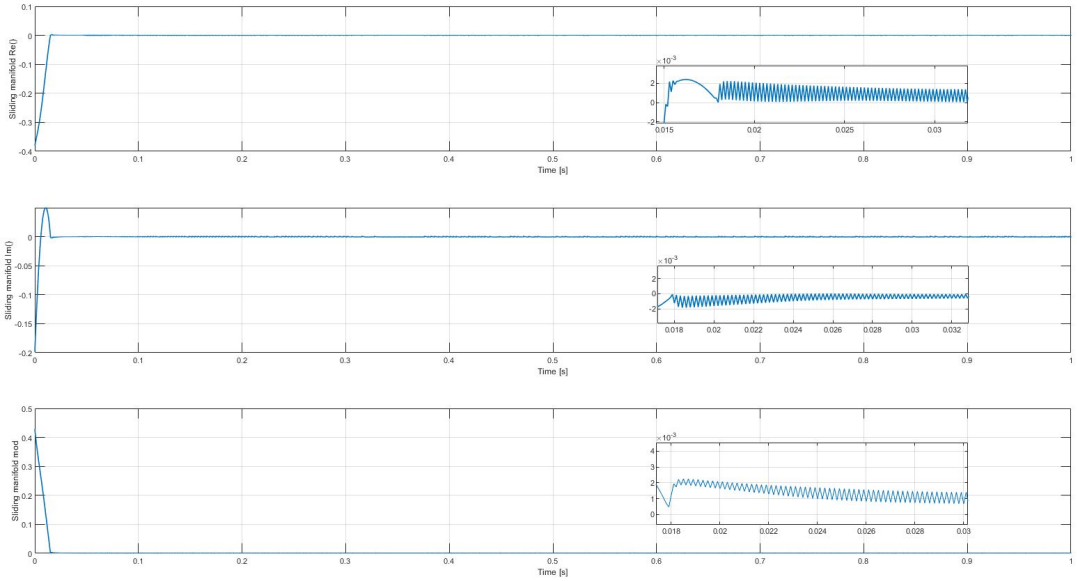

Figure 5.7: Time evolution of the switching manifold for the CV-SMC.

It is important to notice how each position in a plane can be achieved by two different configurations of the robot. When directly controlling the Cartesian coordinates of the end-effector instead of the angles of each joint, the trajectory followed by each of the controllers cannot be directly controlled, which results on very different configurations followed to go from the initial conditions to their targets. Figure 5.8 shows the difference between the planar trajectories followed with each control strategy. Although the final position is the same for all of them, the path to get there is different. This is an important behaviour to take into account when defining the control problem, especially when working in a space with obstacles.

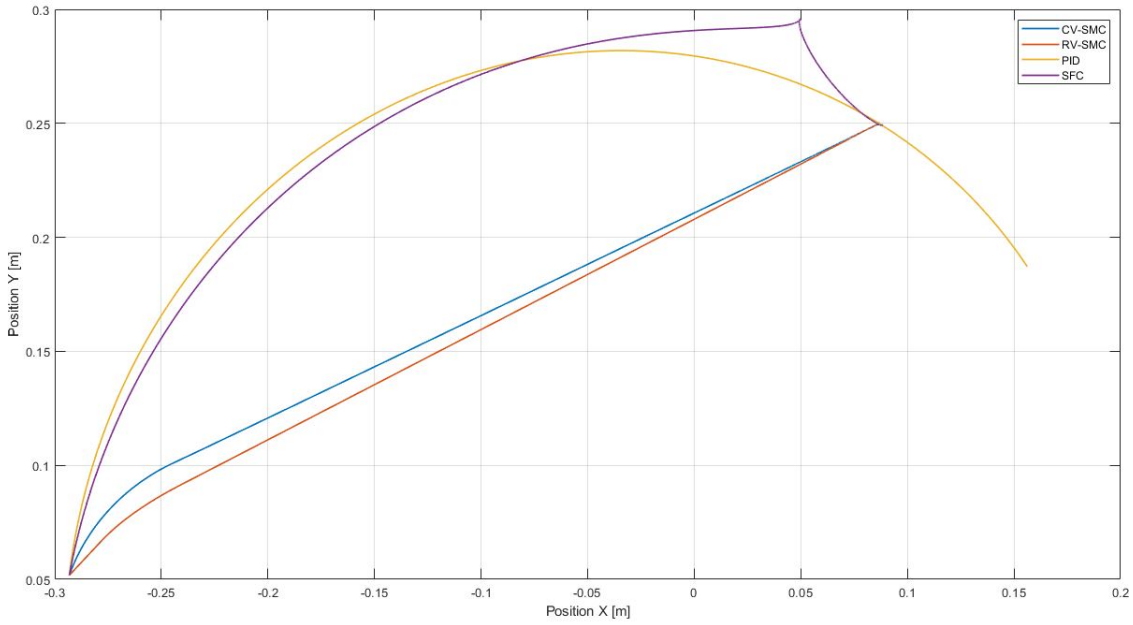

Figure 5.8: Planar trajectory followed with each control strategy.

Moreover, the variations of the test used to check and analyze their behaviour with input disturbances, parametric uncertainties, sensor noise and change of initial conditions are also simulated. The control signals are quite similar to the ideal behaviour; hence, only the error for each position variable is represented in these cases. The time response of the position of the end-effector is represented in Figures 5.9 and 5.10 for an input disturbance of $\tau_d = [0.5, -0.5]$ Nm. It can be seen how the sliding mode controllers maintain a similar response than for the ideal case. However, the linear controllers are not able to follow the real target coordinates. The PID controller presents the biggest error of all the strategies.

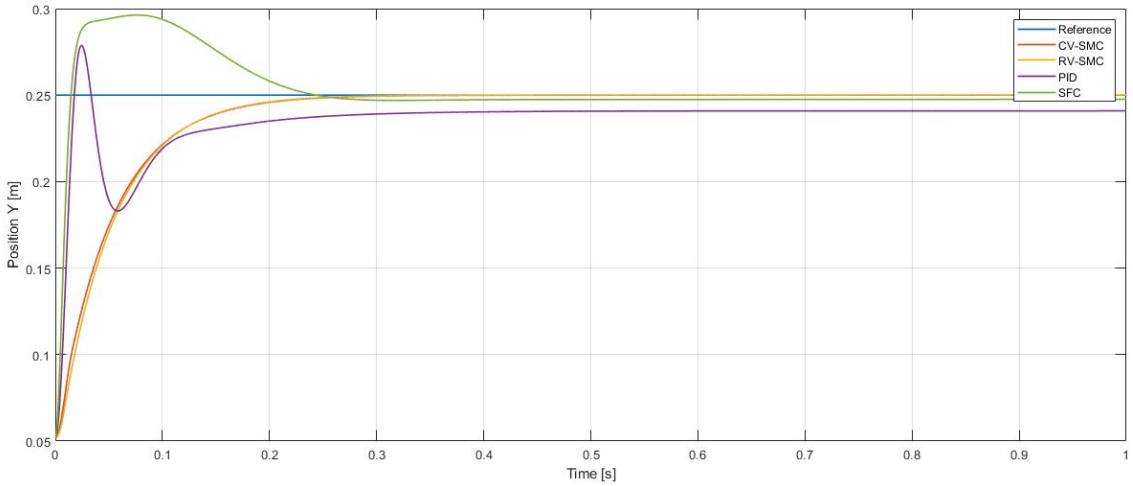

Figure 5.9: Time response of Y position with the input disturbance $\tau_d = [0.5, -0.5]$ Nm.

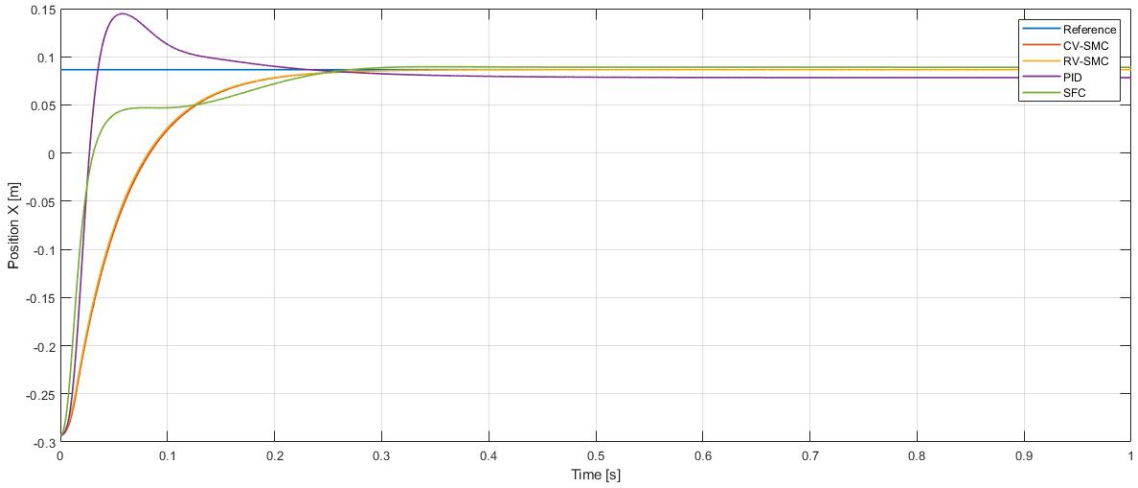

Figure 5.10: Time response of X position with the input disturbance $\tau_d = [0.5, -0.5]$ Nm.

The second test conducted corresponds to the parametric uncertainties. As it has been mentioned before, the difference to the real parameters will be added in terms of both masses. The controllers have been designed considering the masses presented in Table 5.1, but the values implemented have been set to $m_1 = 2.25$ kg and $m_2 = 1$ kg. The results obtained, which are represented in Figures 5.11 and 5.12, show the time response for both coordinates. It can be seen how the error for the linear controllers increase significantly, as it could be expected according to the design strategy followed (feedback linearization). However, being a regulation problem, the target position remains constant and both controllers are able to converge to a solution. The PID controller is the one that presents the highest error.

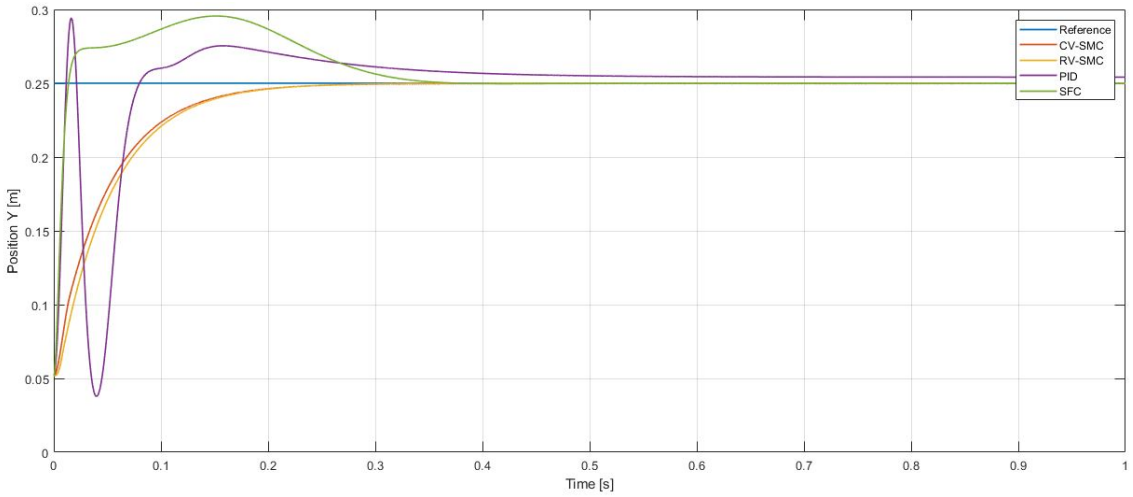

Figure 5.11: Time response of Y position with parametric uncertainties.

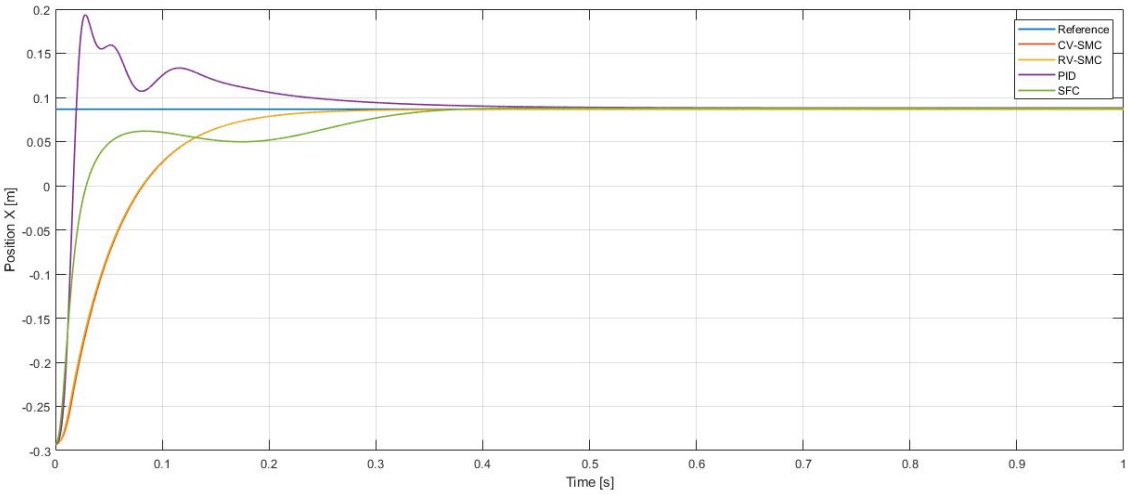

Figure 5.12: Time response of X position with the parametric uncertainties.

The results obtained for the noise sensitivity test are represented in Figures 5.13 and 5.14. In this third test, the two joint angle sensors have been modeled to follow a normal distribution of zero mean and a standard deviation of 0.0286 deg. Similar to the previous test, all the strategies are able to follow the desired path and overcome the white noise present in the sensors when working on a regulation problem.

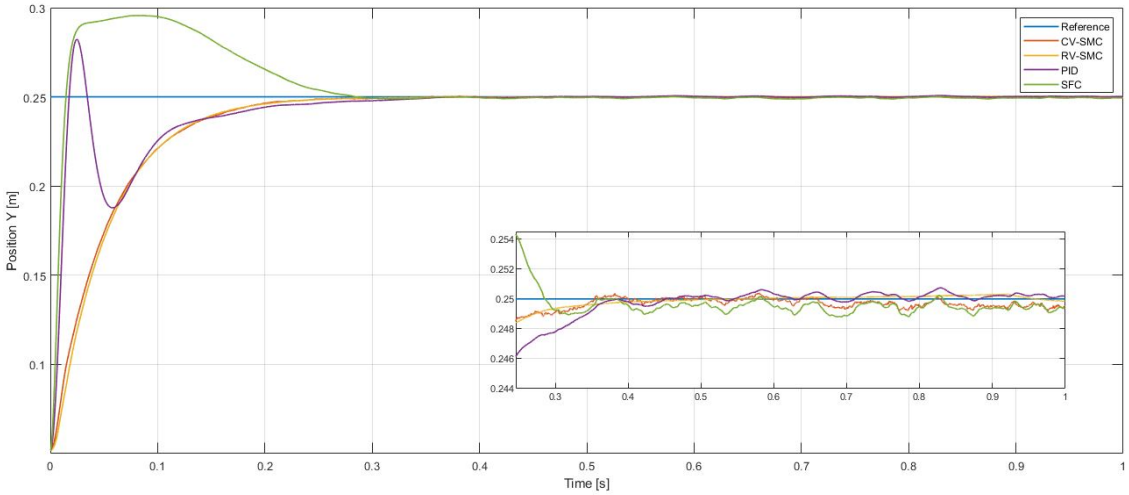

Figure 5.13: Time response of Y position for the noise sensitivity test.

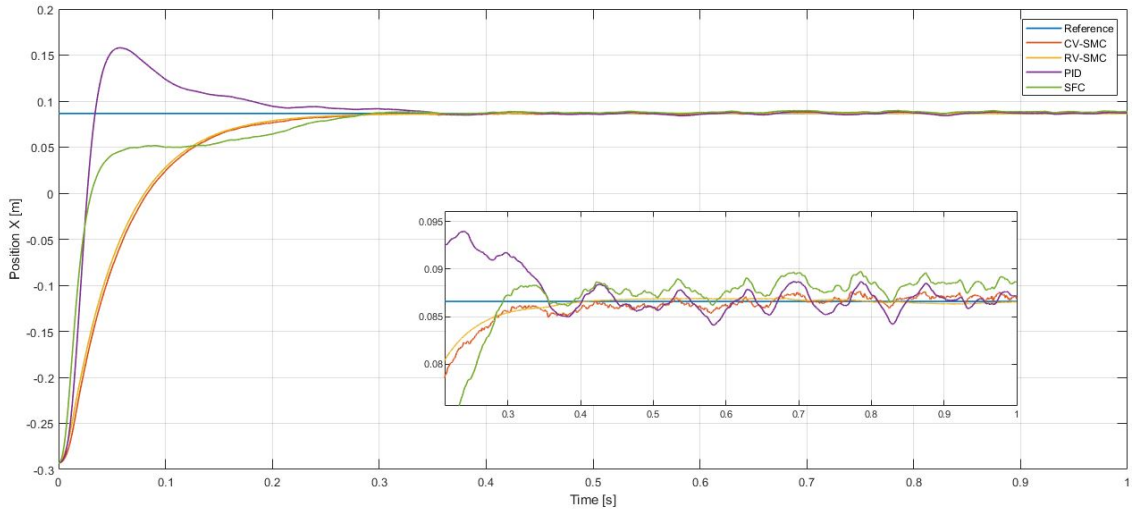

Figure 5.14: Time response of X position for the noise sensitivity test.

The final test is executed by changing the initial conditions of the manipulator. In this particular case, the robot begins in the position $(x_0, y_0) = (-0.2639, 0.1225)$ m, which corresponds to the initial angles of $(\theta_1, \theta_2) = (\frac{3\pi}{4}, \frac{\pi}{6})$ rad. Now, the robot begins the regulation problem closer to its target position, and, as a consequence, the absolute error is always lower than in the previous test. The time response of the TCP can is represented in Figures 5.15 and 5.16.

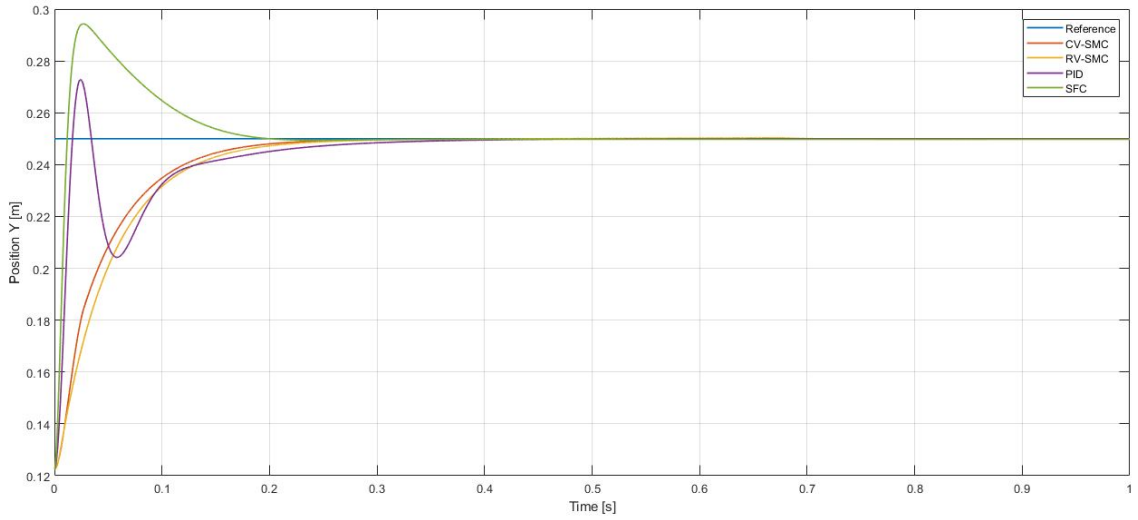

Figure 5.15: Time response of Y position with $(x_0, y_0) = (-0.2639, 0.1225)$ m.

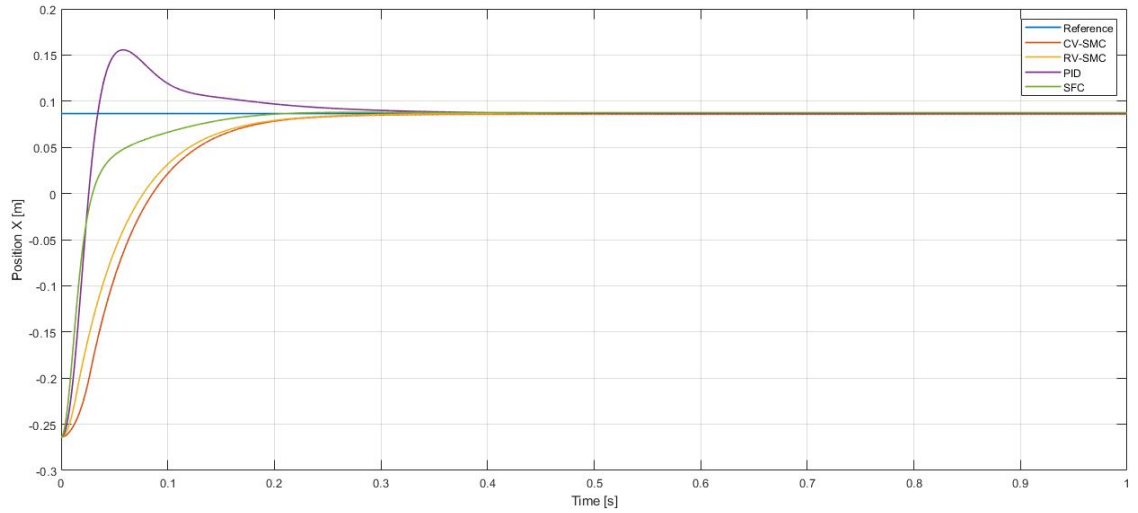

Figure 5.16: Time response of X position with $(x_0, y_0) = (-0.2639, 0.1225)$ m.

With these new initial conditions, the path followed with each of the controllers can be seen in Figure 5.17. The profile followed by the PID controller is similar that for the other cases, but, for the SMCs and the SFC, some significant differences can be noticed. However, all the numerical results will be analyzed more precisely in the end of this chapter.

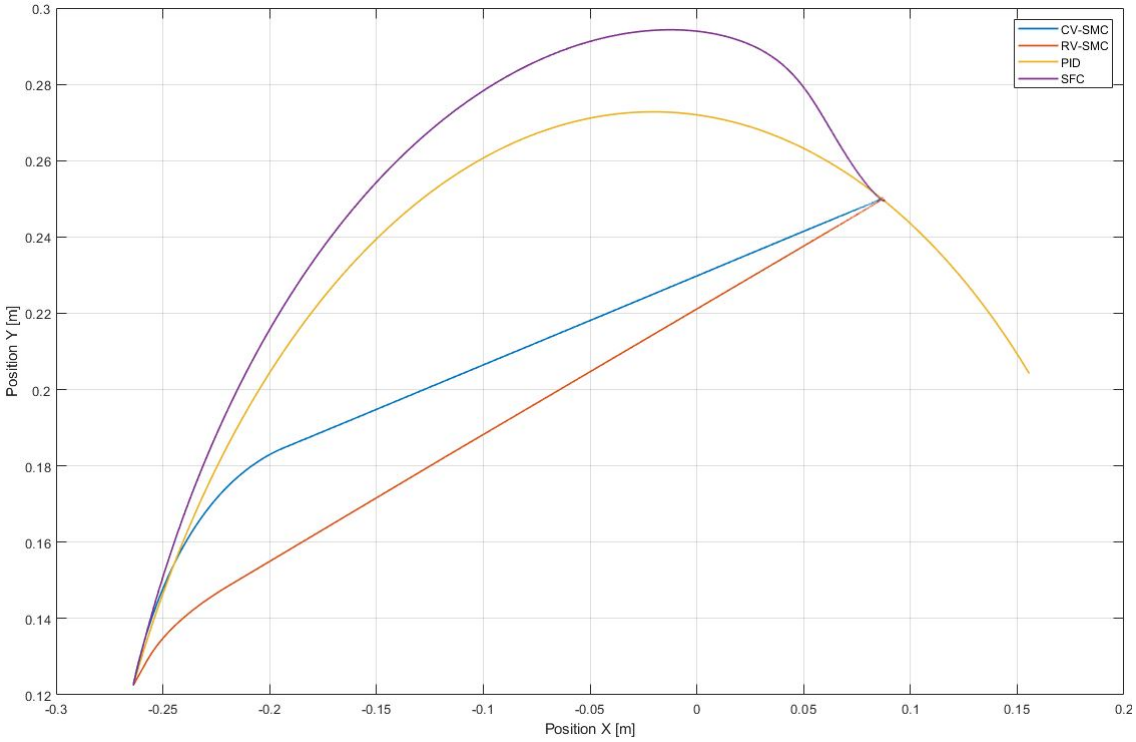

Figure 5.17: TCP path followed with each strategy.

It can be seen how all the four control strategies proposed allow to reach the desired position in proper conditions. Although the paths followed by all of them are quite different, both SMC appear to have a similar behaviour with similar errors. However, a more precise analysis with its corresponding numerical results and according to the KPIs presented is done in Section 5.5. The PID and the state-feedback controllers present some overshoot when reaching the desired position but they can achieve the target value with a low error.

5.4 Tracking problem results

The second test used to analyze the controllers designed is a tracking problem. The target path is the one presented previously in Figure 5.1 and will also be simulated during 7.5 s and with the same step size of 1×10^{-4} s. In this case, the initial position is chosen to be $(x_0, y_0) = (0.1, 0.2)$ m and all the controllers are tuned to reach a settling time of $t_s \simeq 0.25$ s. The final parameters are the following:

1. CV-SMC: Control parameters: $\lambda = 0.045$ and $k = 250$.
2. RV-SMC: Control parameters: $\lambda = 0.05$ and $k = 100$.
3. PID: Control gains: $K_{pPID} = 500$, $K_{iPID} = 50$, $K_{dPID} = 40$, $K_{pPD} = 500$, $K_{dPD} = 40$.
4. SFC: Control gains: $K_{11} = 75$, $K_{12} = -12500$, $K_{21} = 75$, $K_{22} = -6250$. The integral gains are set to $K_{i1} = K_{i2} = -5000$.

In Figure 5.18, the time response of the vertical position is represented. It can be seen how the SFC and the PID present a big overshoot that implies a higher absolute error than the SMCs. However, all of the controllers can track the target position properly once they

have converge enough. The exact quantitative results will be analyzed in the end of the chapter to get a proper analysis of the four strategies studied. At time $t = 4.7$ s, the linear controllers significantly increase their error against the desired value. This happens mainly because of two causes: the first one is related to the fact that both controllers are designed based on an approach around the working point, which is constantly changing its value, whereas the second cause is because they are in a point near to the non-controllability condition. However, this condition is not fully achieved and hence, the SFC is able to keep tracking the desired value.

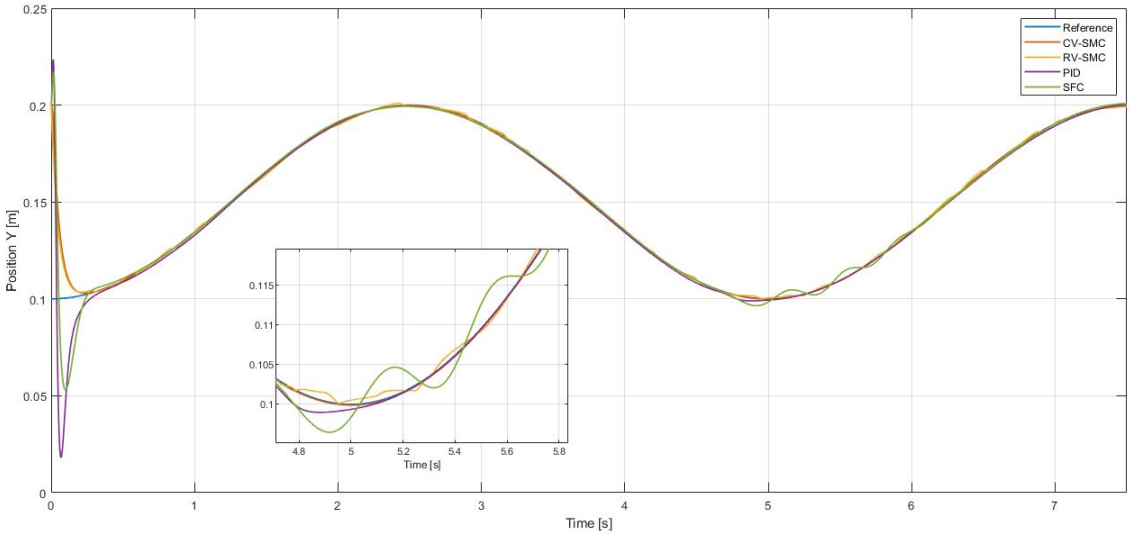

Figure 5.18: Vertical position followed with each strategy.

The horizontal position obtained for the different controllers is shown in Figure 5.19. All four strategies achieve the desired value and are capable of following the path with low error. As it happened in the regulation problem, the PID controller needs a bit of overshoot to reach the target position within the same time than the other options. On the other hand, the effect seen for the Y position is not clearly noticeable for the X position.

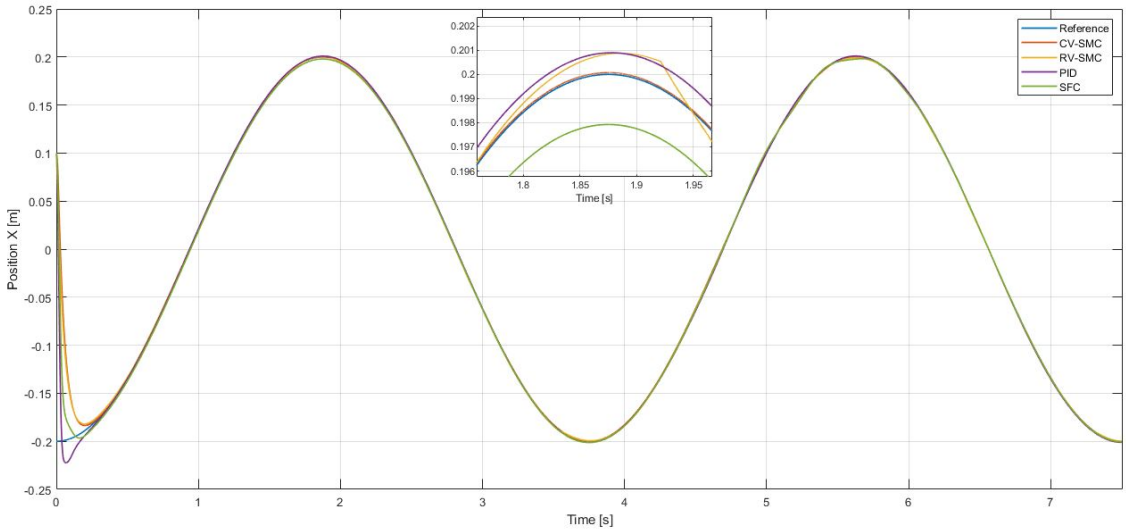

Figure 5.19: Horizontal position followed with each strategy.

The control input required to achieve this performance during the test is plotted in Figure 5.20. It is worth pointing that the SMCs present a noticeable difference in terms of absolute values of their control signals, being them way higher for the RV-SMC than for the CV-SMC.

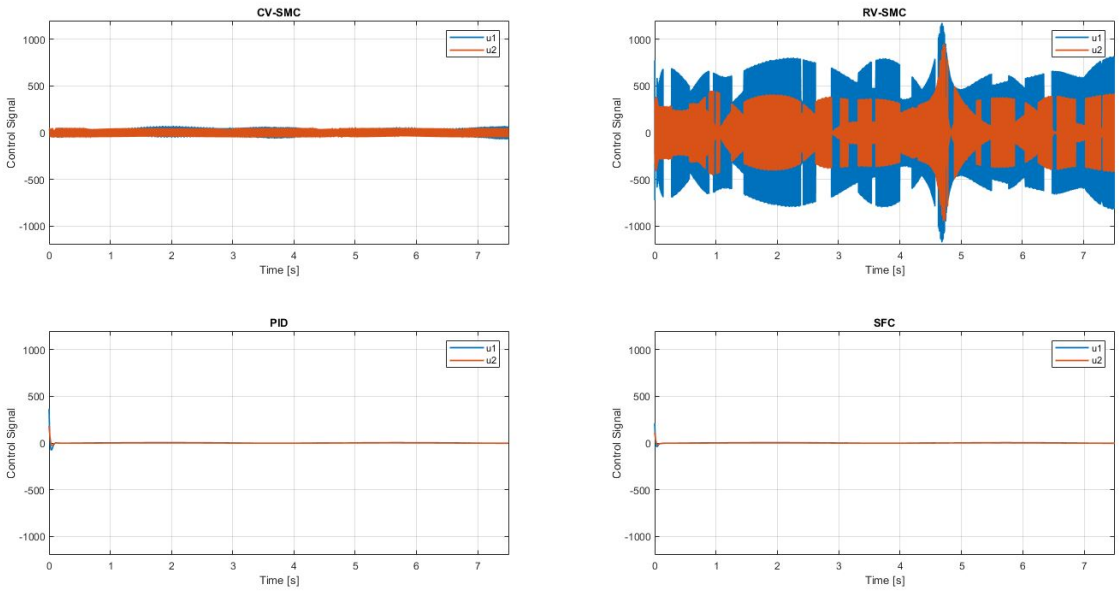

Figure 5.20: Control inputs for all the controllers.

Finally, for the sliding mode controllers, both components of the sliding surface are represented in Figures 5.21 and 5.21.

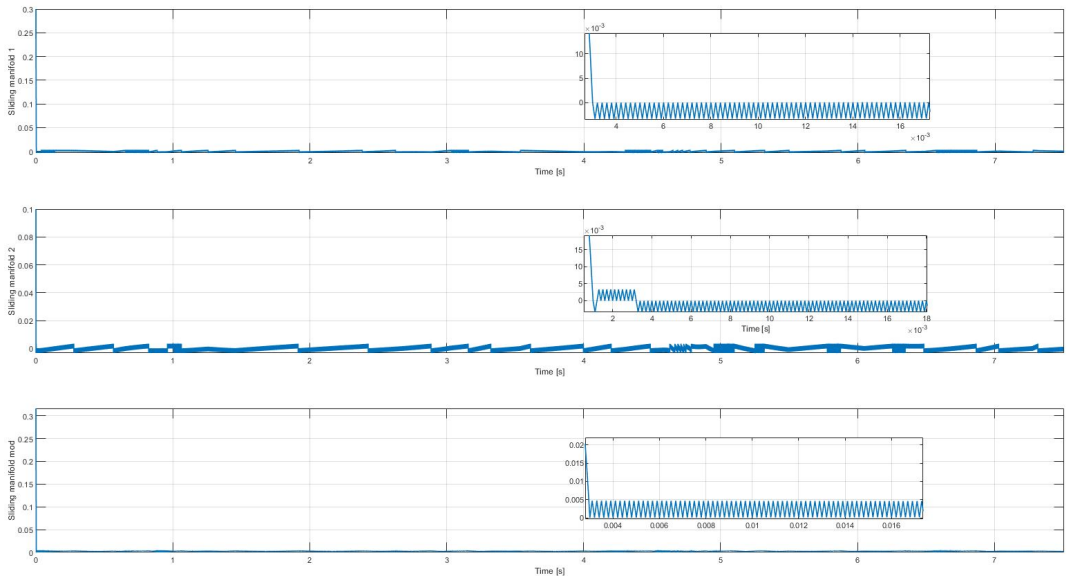

Figure 5.21: Sliding surface for the RV-SMC.

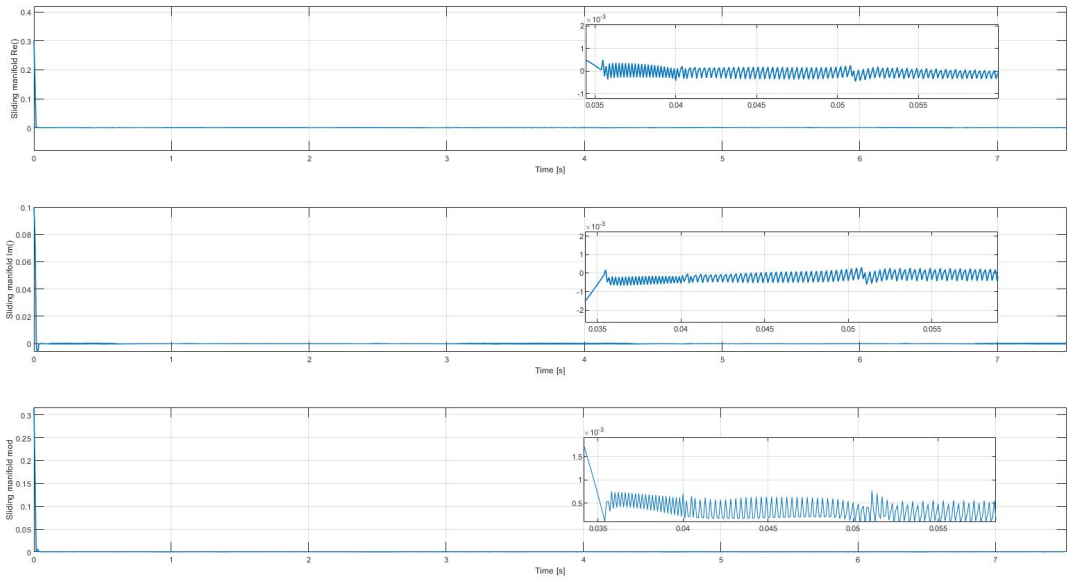

Figure 5.22: Sliding surface for the CV-SMC.

It can be seen how none of the SMCs lose the sliding condition during all the test. This is an important fact to consider when comparing them with the behaviour of the linear controllers, specially at $t = 4.7$ s, in which both SMCs are still able to track the target position. A more detailed analysis of this test will be done in section 5.5, according to the KPIs indicated previously.

Once the ideal behaviour has been simulated, the four test variants are presented. The first one is the input disturbance test, in which the same disturbance torques than in the regulation problem are implemented to the model. Figures 5.23 and 5.24 show the TCP coordinates obtained for the first 7.5 seconds of the tracking test. It can be clearly seen how the SMCs are robust to these disturbances added. For the PID controller, the error increases significantly, specially to track the Y position. Finally, the SFC presents a response between the two behaviours seen. The error is not as big as for the PID but it is still higher than for the SMCs. Then, it can be said that the SMCs are the most robust strategies in terms of overcome input disturbances.

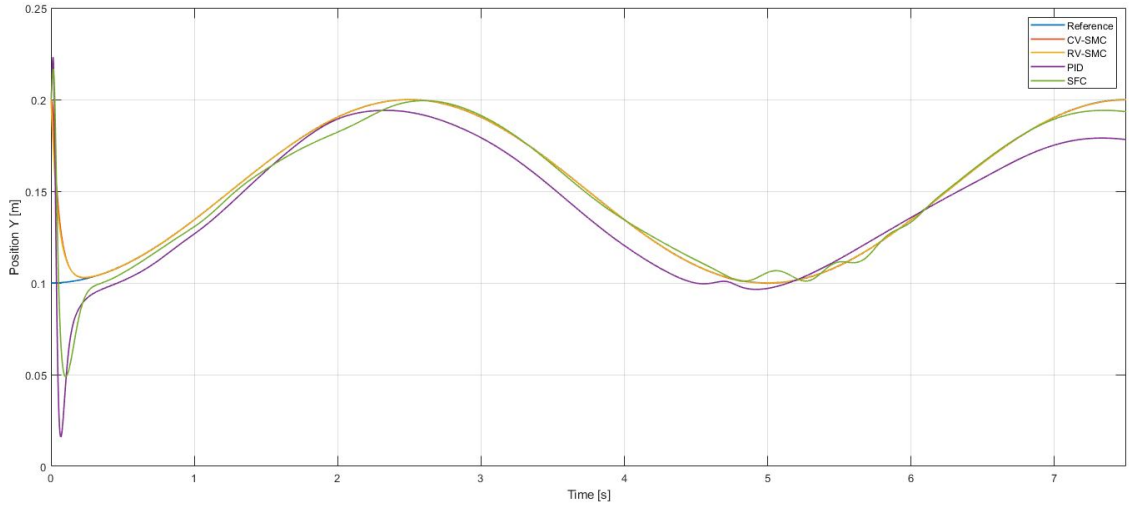

Figure 5.23: Horizontal position followed with each strategy with torque disturbance.

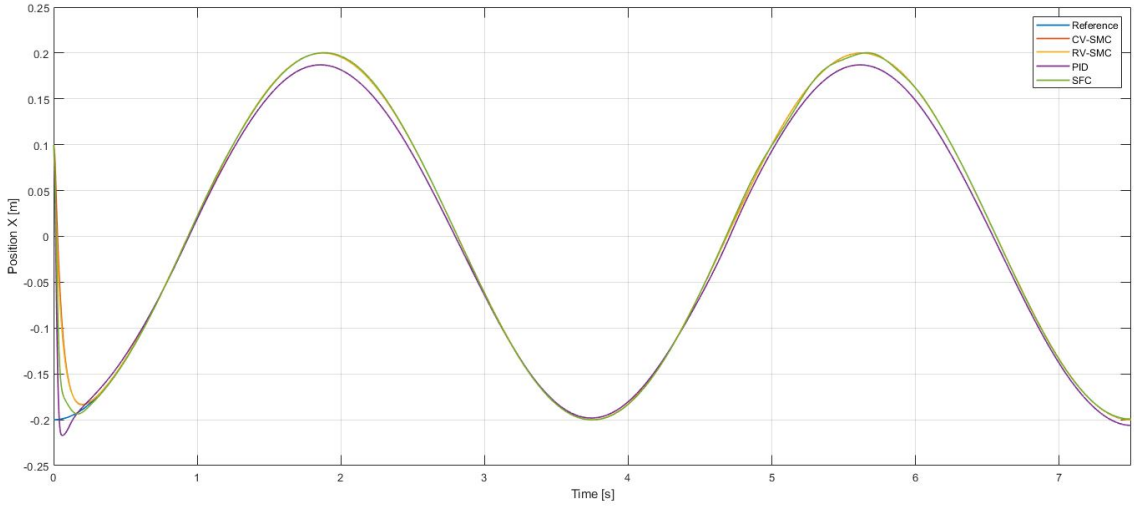

Figure 5.24: Vertical position followed with each strategy with torque disturbance.

The second test is for the parametric uncertainties. As it has been done in the regulation analysis, the uncertainty is added in the masses and in the same amount. Figures 5.25 and 5.26 show the time response obtained during for this case. It can be noticed how the error for the linear controllers is much higher than for the nonlinear controllers. Although they remain stable during the simulation, the followed trajectory is not close to the target, which can be seen specially for the Y position.

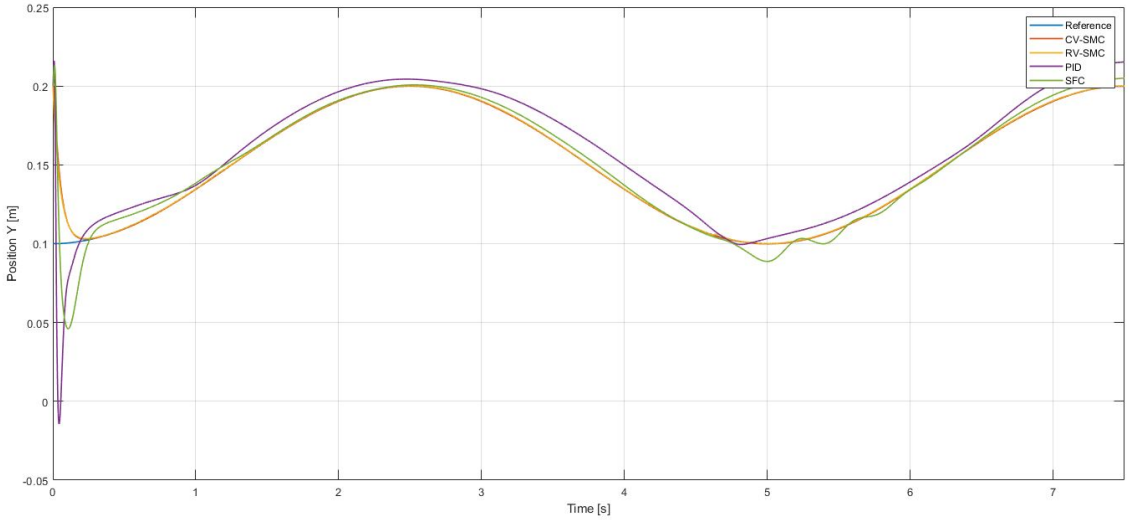

Figure 5.25: Horizontal position followed with each strategy with parametric uncertainties.

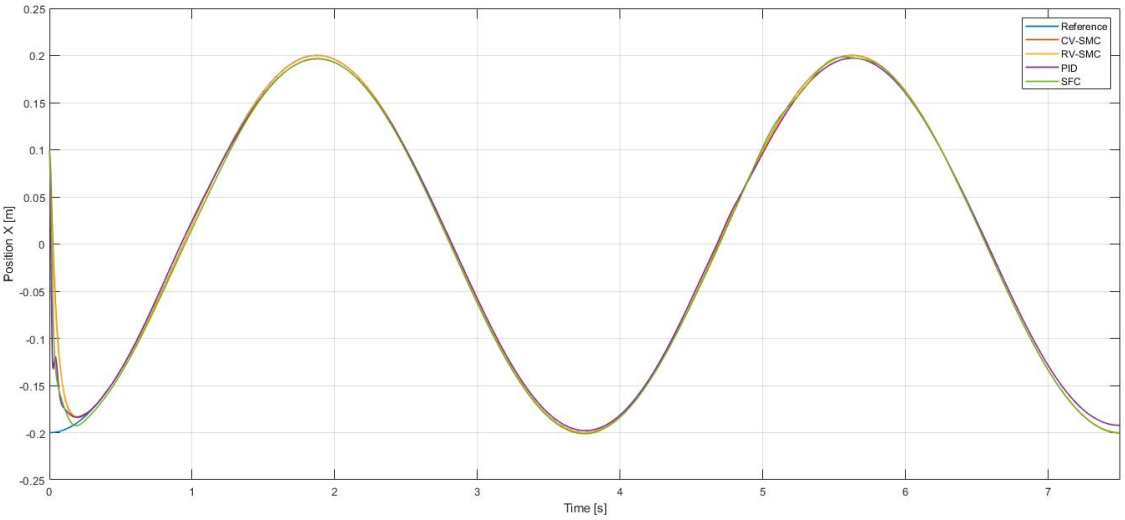

Figure 5.26: Vertical position followed with each strategy with parametric uncertainties.

The third analysis done is to test the sensor noise sensitivity. The joint angle sensors are modeled following the same white noise distribution presented previously. The time response of the TCP coordinates obtained is represented in Figures 5.27 and 5.28, in where it can be seen again the robustness of the two SMCs designed. On the other hand, the SFC and the PID controllers have a bit of a higher error but they are able to achieve a similar time response than for the ideal case. It can be said, then, that all four controllers are robust to a random sensor noise.

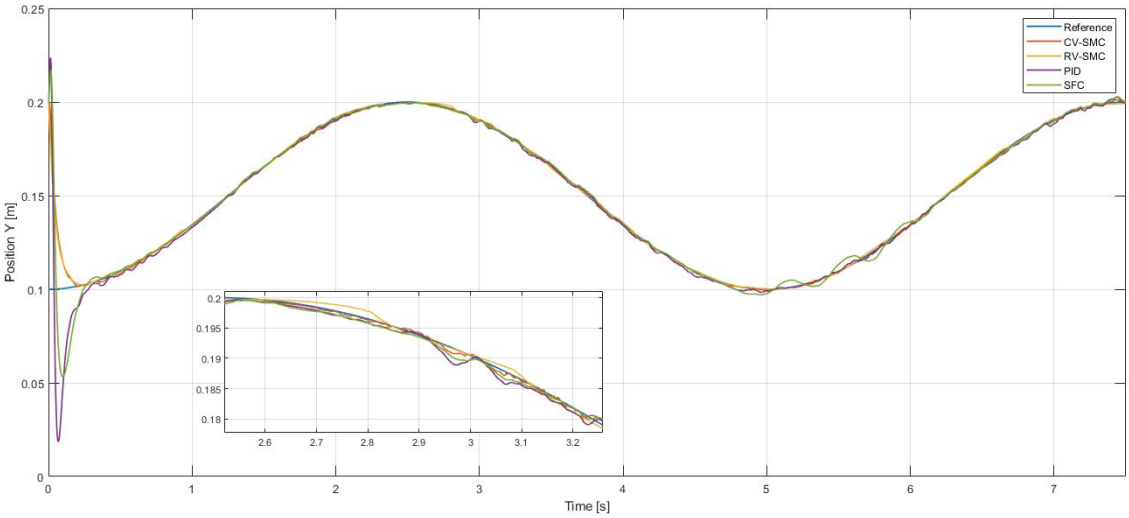

Figure 5.27: Horizontal position followed with each strategy adding sensor noise.

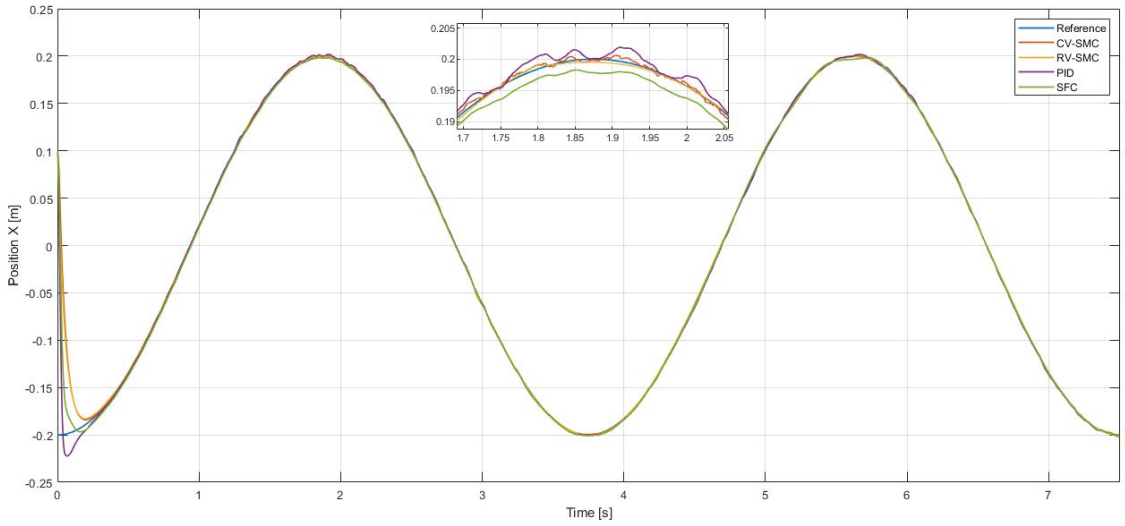

Figure 5.28: Vertical position followed with each strategy adding sensor noise.

The last test done consists of changing the initial coordinates of the planar manipulator. The new starting position simulated is $(x_0, y_0) = (-0.2639, 0.1225)$ m. Being these initial conditions closer to the target position allow all the controllers to reach faster the desired path and with lower overshoot errors for the case of the PID and SFC. However, for the Y coordinate, the SFC present a high absolute error, most likely caused because of the linearization done when designing the controller. The results obtained for this last scenario are presented in Figures 5.29 and 5.30.

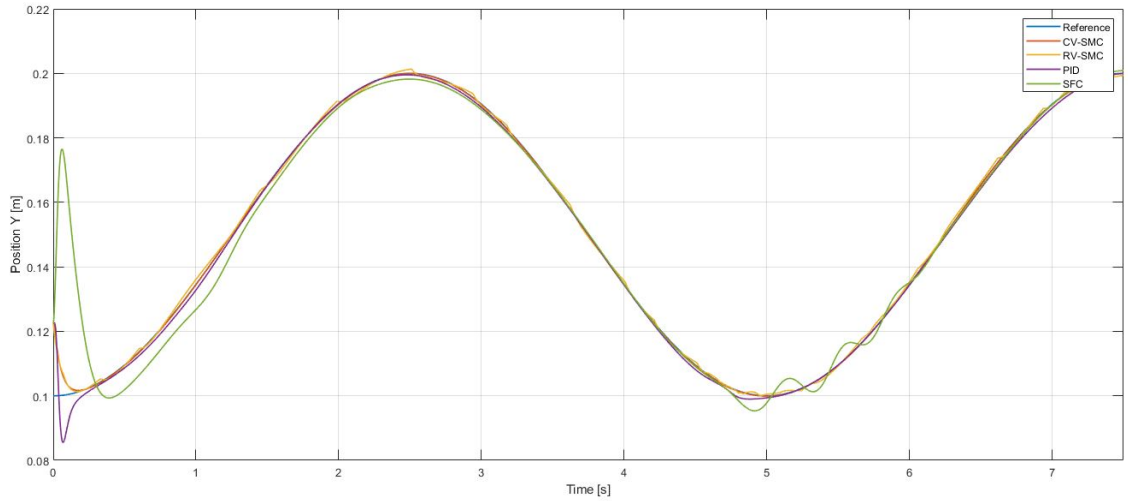

Figure 5.29: Horizontal position followed for $(x_0, y_0) = (-0.2639, 0.1225)$ m.

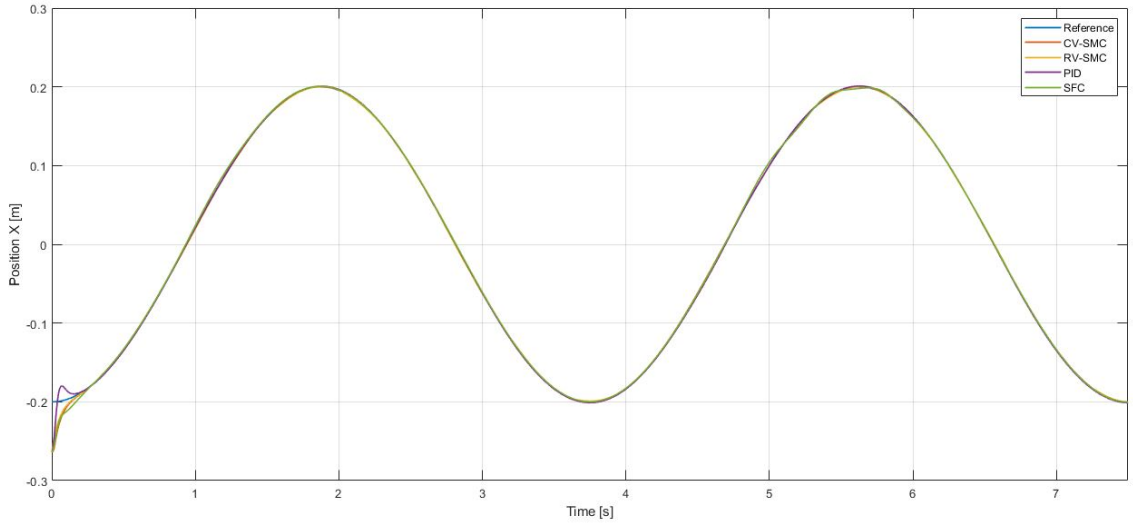

Figure 5.30: Vertical position followed for $(x_0, y_0) = (-0.2639, 0.1225)$ m.

5.5 Controllers comparison

Once all the tests have been simulated, the controllers are compared between them according to the KPIs defined previously. However, it is also important to consider some qualitative concepts regarding each estimator and relative to some responses observed. Finally, it has been giving more importance to the tracking problem as the regulation can be considered as an specific case of the tracking and allows and is a less generic solution.

5.5.1 Regulation analysis

The first test analyzed through the KPIs is the regulation problem. Figure 5.31 shows the evolution through time of the average error obtained for each strategy. Although the linear controllers are based on an approach around the working point, this is not a problem when doing a regulation test, as the working point remains constant through the time. However, it is also important to notice the high overshoot they presented during this test. In terms

of absolute error, this does not look as a problem; however, normally this is not an ideal behaviour in the robotic manipulators field.

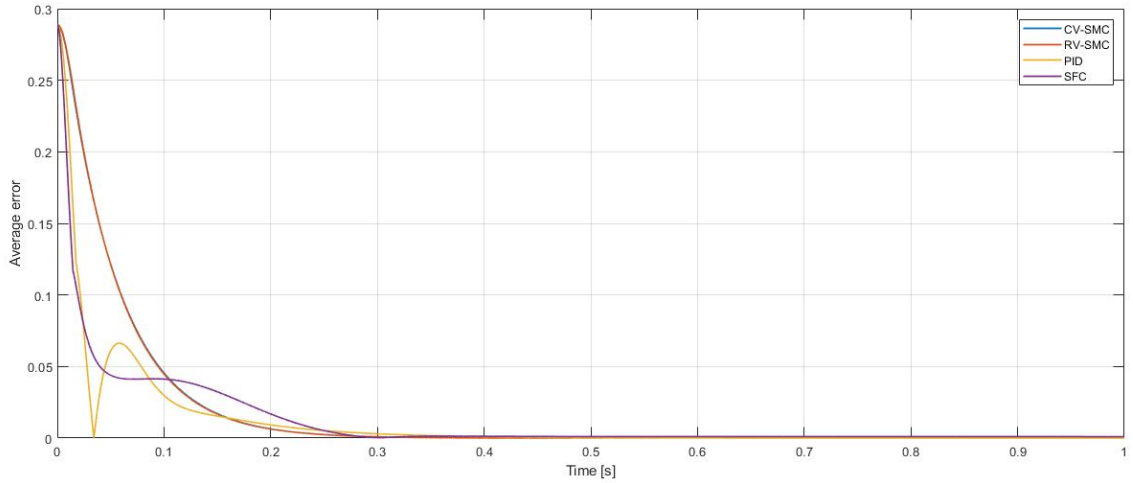

Figure 5.31: Average error for an ideal regulation problem.

Both SMCs appear to have similar results. The difference between them can be more easily compared with Figure 5.32, in which it can be seen the area of the infinity norm of the errors. It can be seen how the RV-SMC has a slightly lower error than the CV-SMC. However, this difference can be reduced when decreasing the step size defined for the simulations. It can also be seen how for the SFC, this error keeps increasing with time.

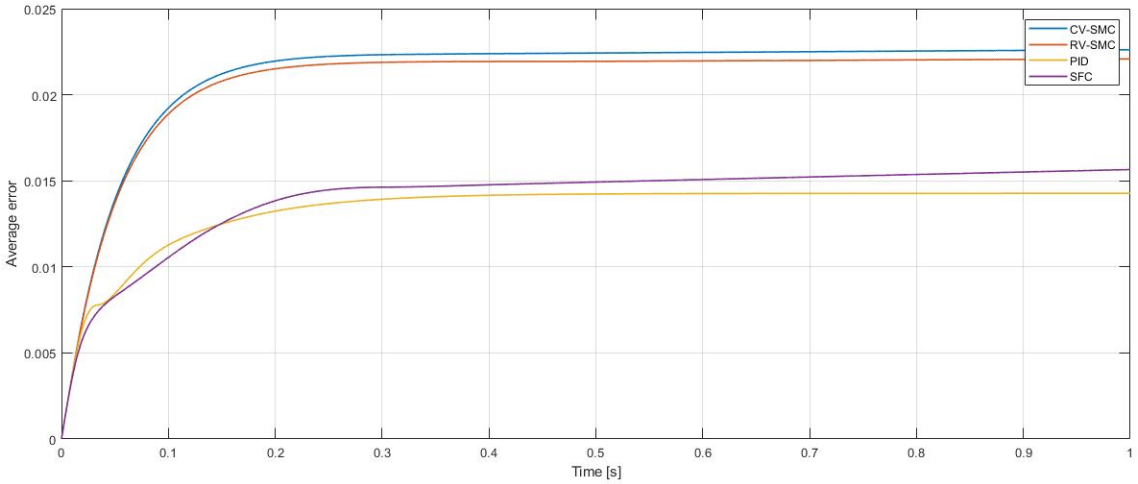

Figure 5.32: Area of the average error for an ideal regulation problem.

The input required to achieve these results is also an important factor to take into account. However, comparing the input signals for a SMC and a feedback controller is not consistent when analyzing a regulation problem. By definition, the SMC input will always remain switching whereas for the feedback controllers will reach a constant value. On the other hand, the maximum value can be compared. Figure 5.33 shows the infinity norm of these signals. This maximum input value is obtained for the SFC, being halved by the PID. The SMC have some initial values of a tenth of both linear controllers.

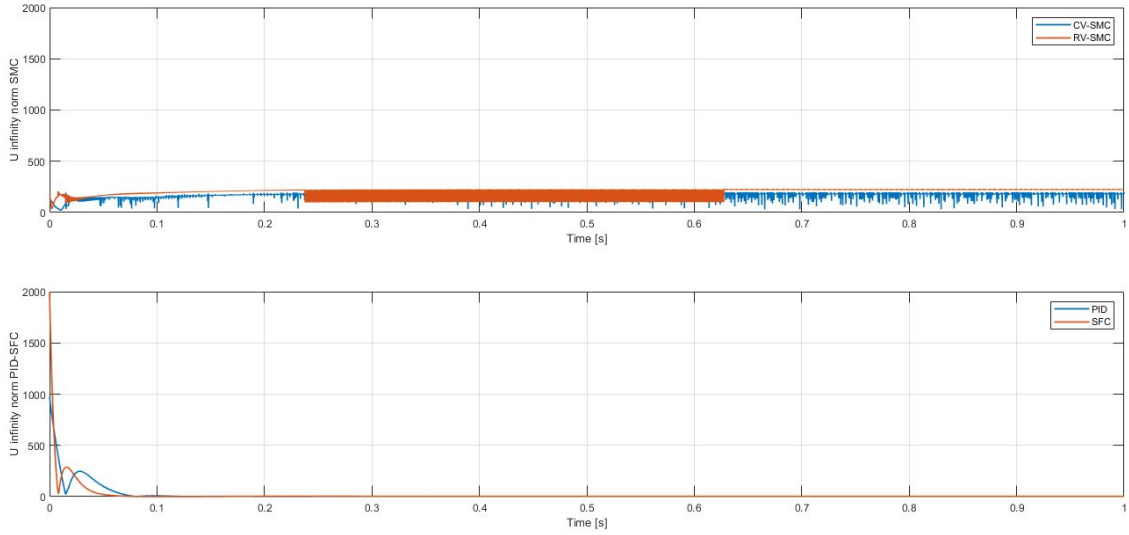

Figure 5.33: Results of $||u(t)||_\infty$ for an ideal regulation problem.

In Figure 5.34, the area of these values is represented. Comparing the SMCs, it is seen how the CV-SMC presents a lower value than the RV-SMC. Regarding the linear controllers, the SFC requires less effort than the PID controller to achieve the desired response.

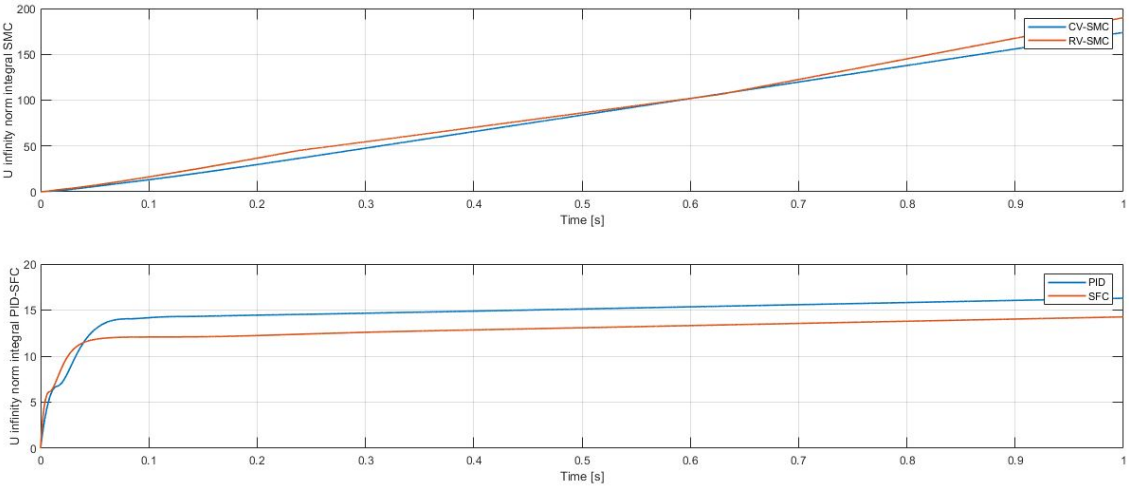

Figure 5.34: Area of $||u(t)||_\infty$ for an ideal regulation problem.

Regarding the test variations done for the regulation problem, some relevant conclusions can be extracted without the need of analyzing these KPIs. First of all, the SMCs present the most robust behaviour of all the controllers studied. On the other hand, it can be said that all the controllers are robust to sensor noise sensitivity, although the error for the linear controllers increases, specially for the PID. Finally, in terms of adding parametric uncertainties or input disturbances, the PID controller is not able to reach the desired value. The SFC has less error than the PID but it does not completely reach the target. Then, both linear controllers are not robust against disturbances and model uncertainty, as it could be expected because of the feedback linearization procedure needed to design them.

5.5.2 Tracking analysis

The results obtained for the tracking problem are also analyzed through the KPIs proposed. The tracking problem is considered to be more significant than the regulation as it is a more generic test.

Figure 5.35 shows the time evolution of the average of the errors. It can also be seen the existence of the high overshoot needed for the linear controllers. One important thing to notice here is the increase of the error at time $t = 4.7$ s, specially for the SFC. As it has been commented previously, this happens when it is close to a singular configuration. However, the controller is still able to overcome this problem and keep tracking properly the position.

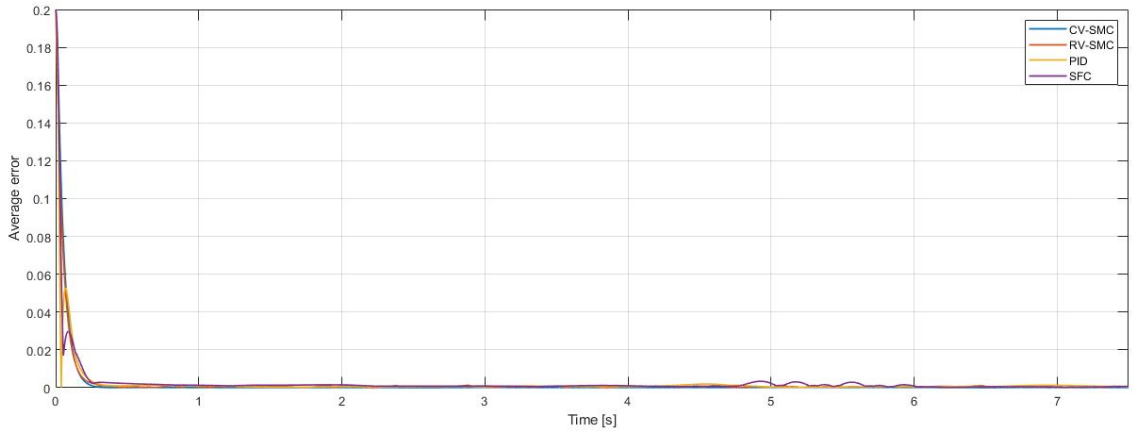

Figure 5.35: Average error for an ideal tracking problem.

To get a better understanding of these errors, their area is represented in Figure 5.36. It can be seen how the CV-SMC begins with a higher error than the other strategies but, once it has reached the target value, it is able to track the position with an error almost null. All the other controllers have a higher error during the test, specially for the PID controller. Then, the most adequate response here is accomplished by the CV-SMC, as the tracking error once it has reached the target is lower than for the other controllers.

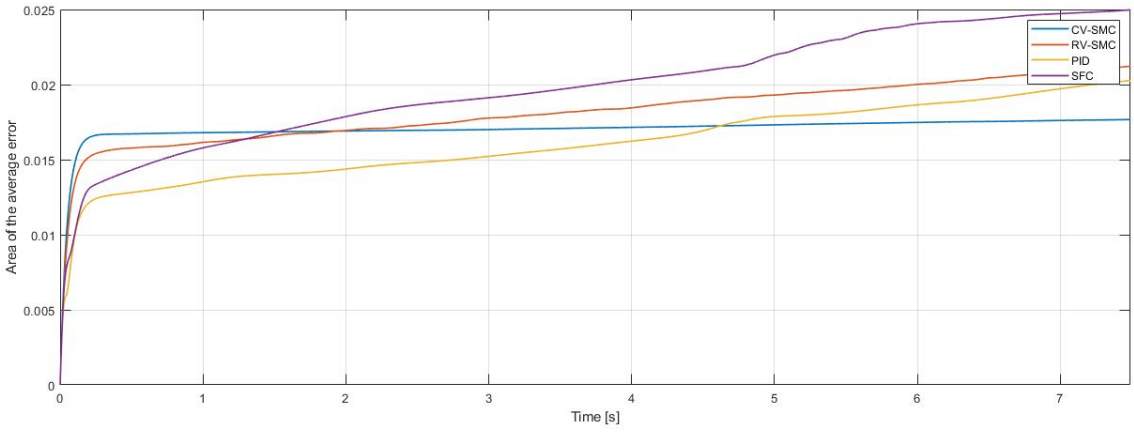

Figure 5.36: Area of the average error for an ideal tracking problem.

As it has been done previously, the input signals are analyzed in Figure 5.37. Comparing

the SMCs, it is clear that the CV-SMC is able to get a better result (lower error) with less control effort. Regarding the linear controllers, the PID presents a higher value than the SFC.

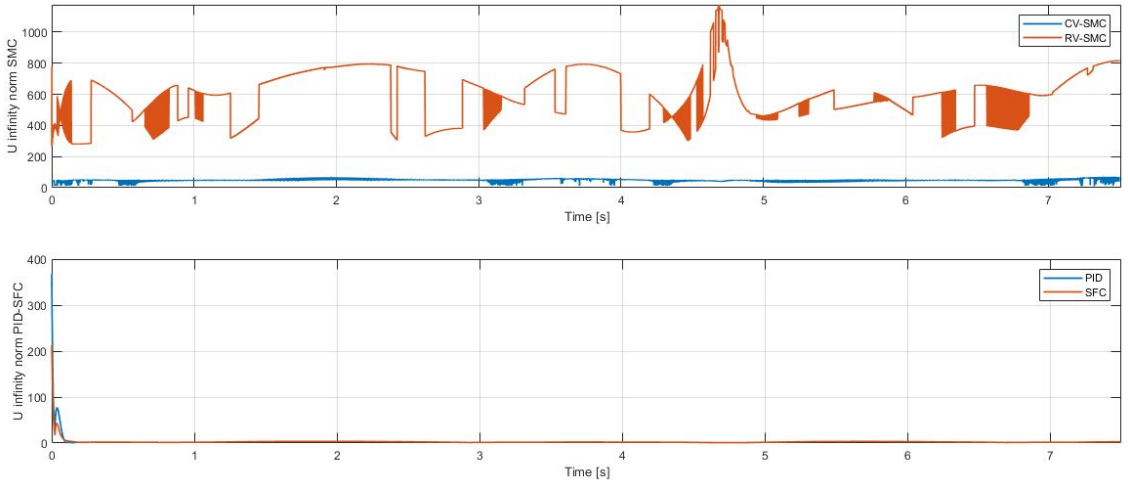

Figure 5.37: Results of $||u(t)||_\infty$ for an ideal tracking problem.

The integral of these infinite norms is also represented in Figure 5.38. Both behaviours mentioned are also noticeable in this image.

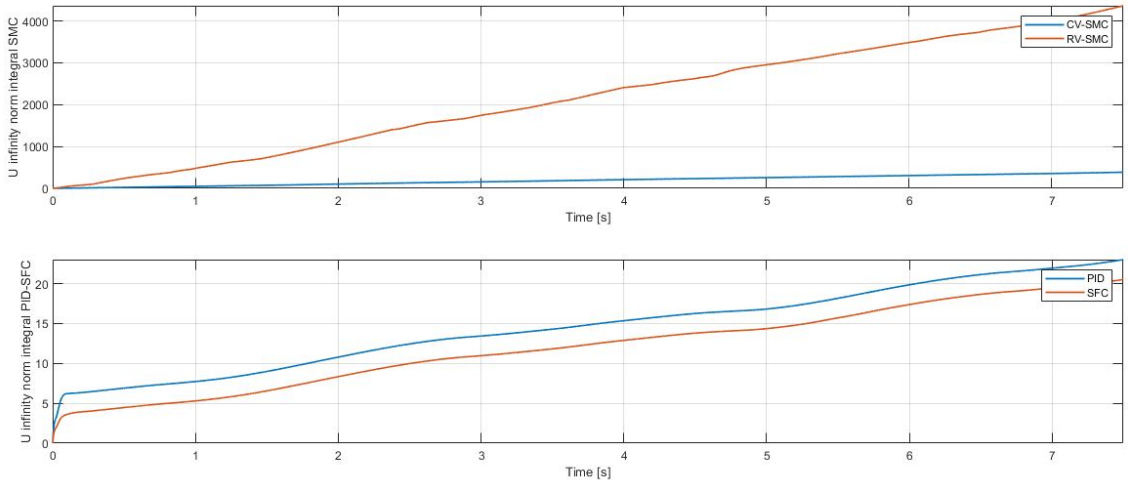

Figure 5.38: Area of $||u(t)||_\infty$ for an ideal tracking problem.

The KPIs have also been analyzed for the test's variants. The average of the errors and its area are summarized in Figures 5.39 and 5.40. The same behaviour as before can be seen, as the CV-SMC presents the lowest error once it has reached the target position. From here, it is clear that the linear controllers are not able to track the desired position properly with disturbances and without the adequate knowledge of the system (parametric uncertainties), specially the PID controller. The SFC has also a bad behaviour when adding sensor noise to the system. To sum up, in terms of the average error, the CV-SMC is the most suitable solution of the ones presented.

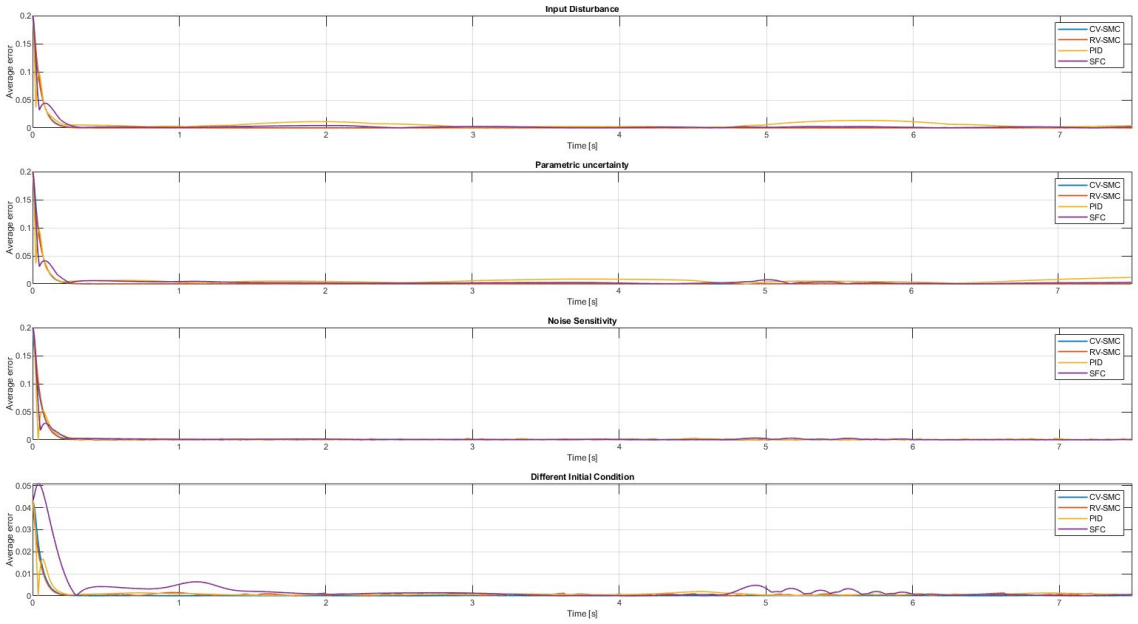

Figure 5.39: Average error for the tracking problem variants.

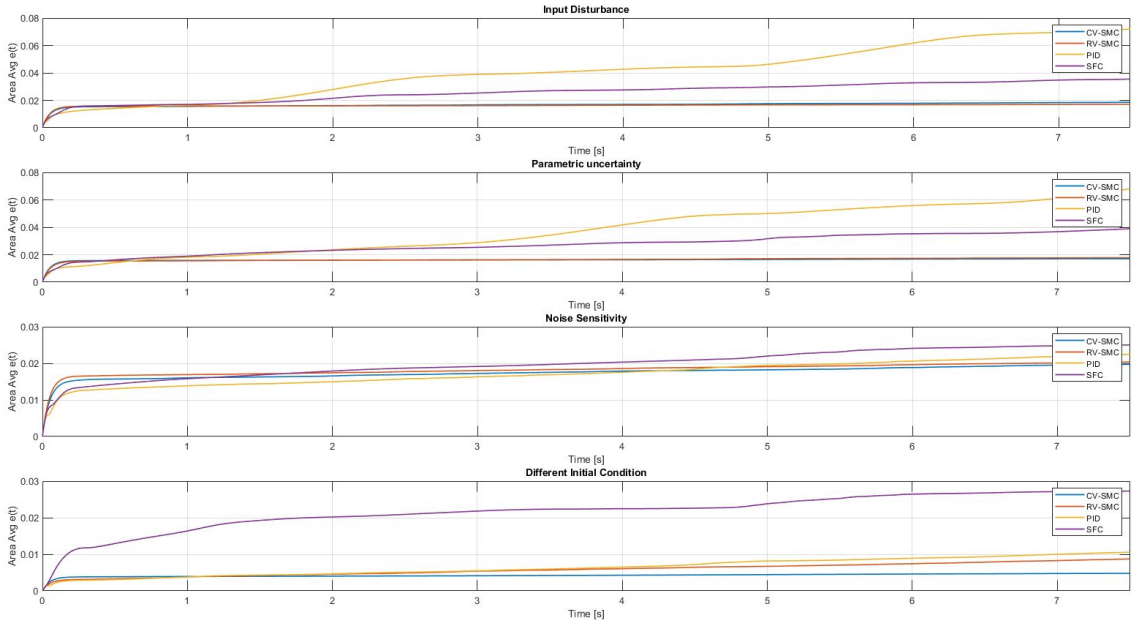

Figure 5.40: Area of the average error for the tracking problem variants.

On the other hand, although the differences between the test variants and the ideal behaviour is not that significant for the control inputs, the results obtained are also presented. Figures 5.41 and 5.42 show the infinite norm of the control signal for the linear controllers and SMCs, respectively. The highest values remain in all situations for the RV-SMC and the PID.

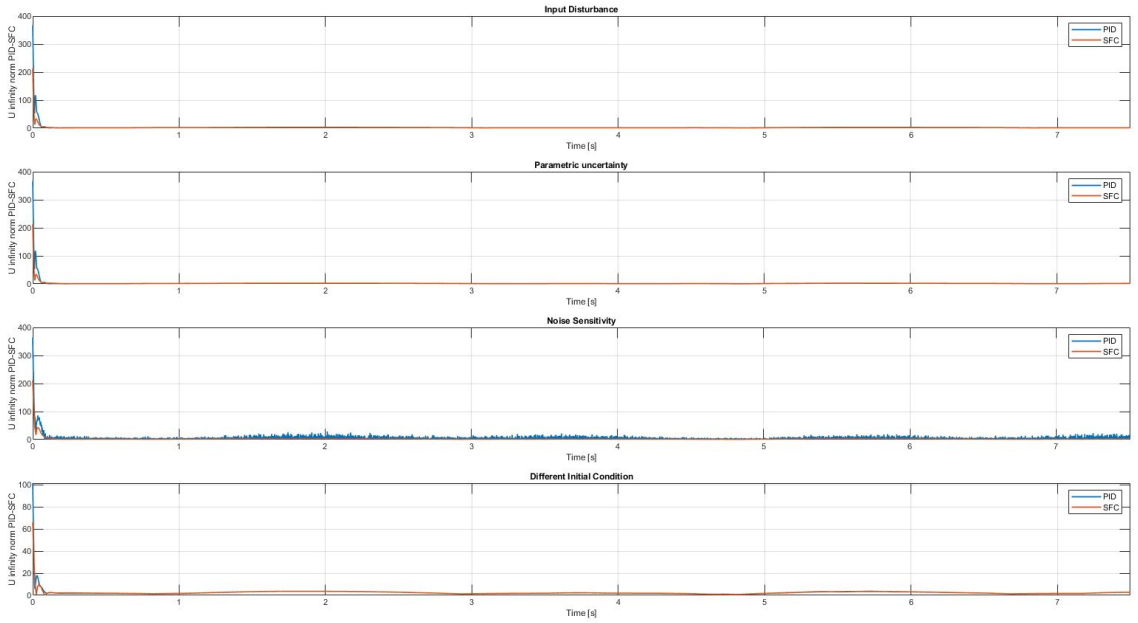

Figure 5.41: Results of $||u(t)||_\infty$ for the tracking problem variants.

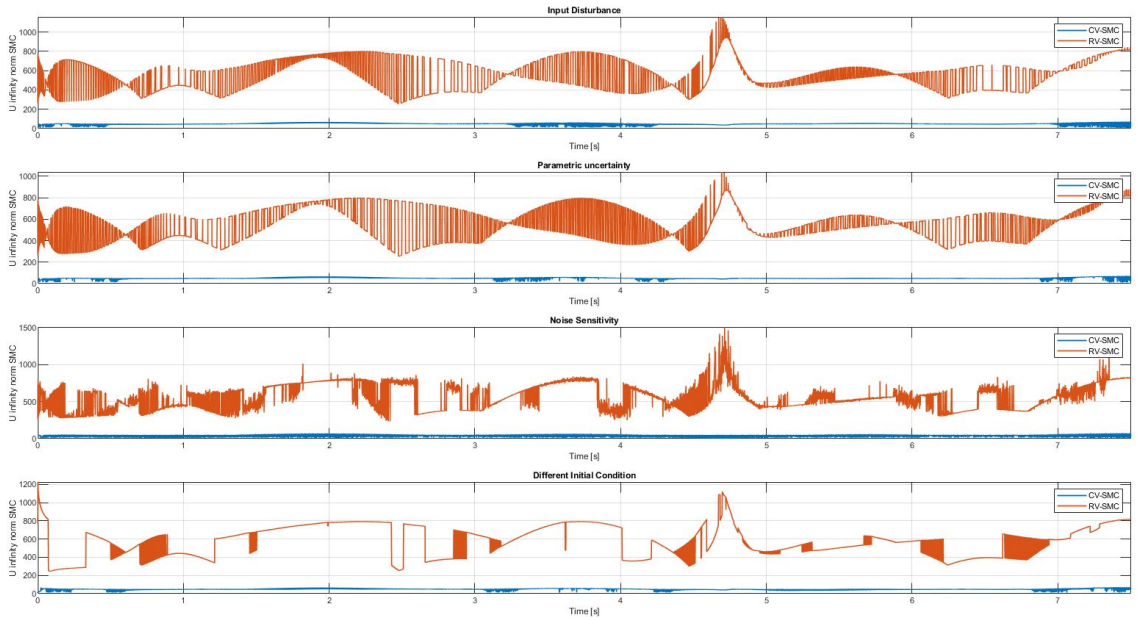

Figure 5.42: Results of $||u(t)||_\infty$ for the tracking problem variants.

The area of the infinite norm of the inputs can be analyzed in Figures 5.43 and 5.44. Apart from the conclusions extracted previously, it is noticeable how the PID has an increasing rate higher when adding white noise to the sensor.

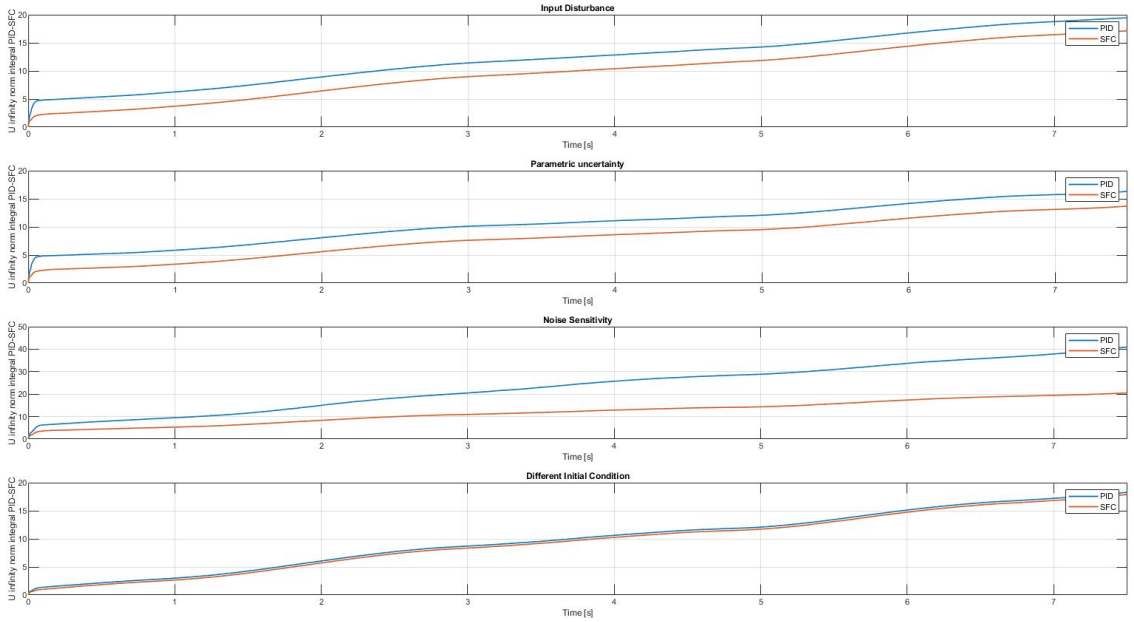

Figure 5.43: Area of $||u(t)||_\infty$ for the tracking problem variants.

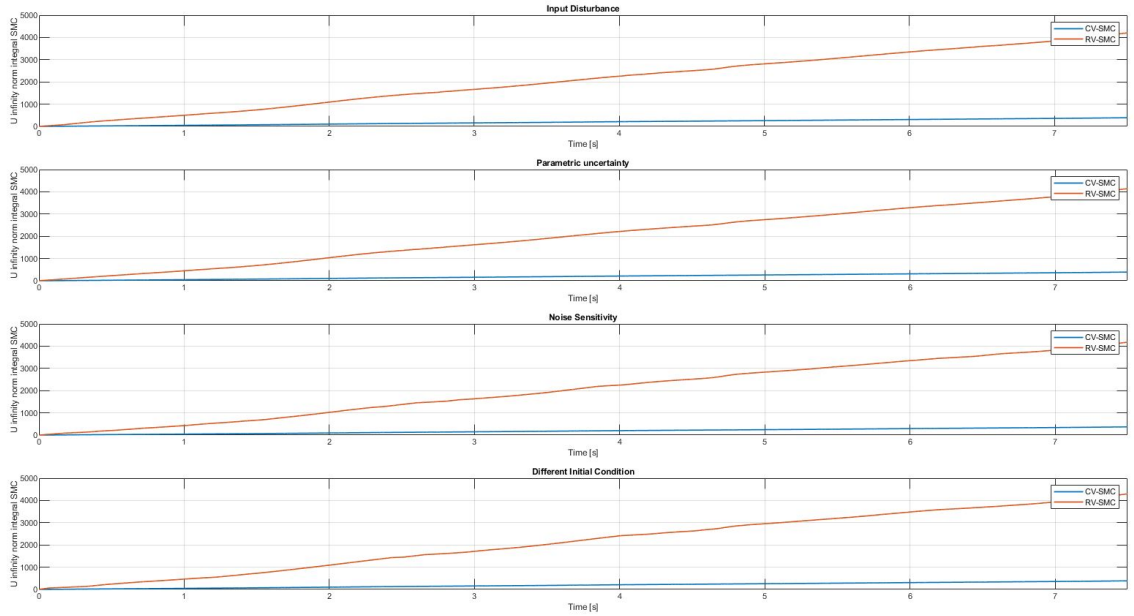

Figure 5.44: Area of $||u(t)||_\infty$ for the tracking problem variants.

To sum up, a brief summary of all the conclusions extracted from this comparison is presented. Regarding the regulation problem, it has been seen how for an ideal behaviour, all the controllers are able to reach the desired position and maintain that value through time. However, it is worth noting that, in order to reach a fast response, the linear controllers present a significant overshoot. Moreover, when adding input disturbances and parametric uncertainties, both SFC and PID are no longer able to reach the target. This is due to the necessity of knowing the model when applying a feedback linearization and because they have been designed around the working point. Regarding the noise sensitivity,

all of them have a proper performance, being better for the SMCs. Finally, the difference between the CV-SMC and the RV-SMC is low in the regulation problem.

On the other hand, in the tracking analysis it has been clearly seen how the most adequate response is accomplished with the CV-SMC. Although in terms of the error, both SMC present a similar response, the control action needed for the RV-SMC is way higher than the one required for the CV-SMC. In this test, the linear controllers present a highest error, specially for the variant test. It can be said that none of the linear solutions are robust to disturbances and model uncertainty. Finally, it is also important to mention how the SFC is so sensitive to positions near a singular position, where it loses controllability.

To sum up, it can be seen how the new control strategy presented (CV-SMC), presents a proper behaviour in this robotic application field. It can be said that it has some significant benefits compared to the linear approaches in terms of robustness and tracking problems and assures an adequate error in any problem of the ones simulated in this work. Moreover, it is worth noting how this alternative allows to reduce the states of the problem, as it is able to merge two of them using the complex domain.

5.6 Summary

In this chapter, all the controllers proposed have been simulated for two different main tests (regulation and tracking) to analyze their response according to some KPIs. Moreover, some test variant has been introduced to check their robustness against disturbances, parametric uncertainties and sensor noise. It has been concluded that the CV-SMC presented has some interesting benefits compared to the other controllers, specially against the linear solutions introduced.

Chapter 6

Economic analysis

This chapter presents an economical analysis and the budget of this project. The work carried out is an analysis and design of a complex-valued sliding mode controller of a 2-link planar manipulator based on MATLAB and SIMULINK. It is a work that can be defined as developing of control algorithms and simulation, and for this reason, all the topics outside this scope or future works to be done will not be considered in the economical analysis. Finally, the cost corresponding to external expenses such as the space and energetic consumption (light or water) used during the project will not be included in the economical study. The budget is broken down into parts in Tables 6.1 and 6.2, which correspond to the personal cost and material cost, respectively. It is also important to add some comments when analyzing the economical expenses of the project: the cost referred to software (Table 6.2) is null because of the student status licenses and that a future work related with controlling a robotic manipulator with one of the strategies presented in the project would require less hours of learning and previous research (being their cost lower than the one presented here).

Table 6.1: Economical analysis of the different tasks of the project.

Personal costs	Hours spent	Cost per hour	Total [€]
Research	240	25	6000
Kinematic and dynamic modeling	24	25	600
Design and tuning parameters for the SFC	40	25	1000
Design and tuning parameters for the PID	25	25	625
Design and tuning parameters for the RV-SMC	60	25	1500
Design and tuning parameters for the CV-SMC	140	25	3500
Controllers simulation and comparison	40	25	1000
Total	569	-	14225

Table 6.2: Economical analysis of the material and software required for the project.

Material costs	Cost per unit	Total [€]
MATLAB student license	-	0
Computer cost	800	800
Total	-	800

Then, the total budget of the project is 15025€.

Chapter 7

Environmental impact

Evaluating the environmental impact is always a necessary and important topic in any project. However, the work exposed during this work is the design and simulation of a control algorithm for a robotic manipulator. For this reason, the environmental analysis is not considered as it is a purely software based project and does not have any direct impact.

The most important aspect related to the environmental footprint would be the one produced by the robotic manipulator itself. Without considering the manufacturing process, the main energy consumption would be the electrical power required to do its tasks. Although this value can be directly related to the actuators' torques and hence, the controller could have an impact on this analysis, it is considered beyond the scope of the project. In the same direction, it is worth mentioning that the consume of electrical energy needed to develop and simulate the algorithms (mainly due to the computer) could also be studied, but it will not be considered as it is an indirect minor impact.

Chapter 8

Social impact and gender equality

In this chapter, the social impact of this work is analyzed. The project presented is a software work in which a new controller approach is designed and simulated to see its benefits and drawbacks. Being a project only based on software and simulation, it can be said that the social impact does not apply. On one hand, the performance and efficiency of this new control algorithm does not have any direct social consequence, and neither does its design and simulation. Moreover, it is a new strategy to be implemented in a robotic manipulator in the future, which will not make any significant impact on any social aspect. Then, it can be said that there is not a gender equality and social impact to be analyzed.

Chapter 9

Conclusions

All the work done during the realisation of this project and the accomplishment of the objectives defined are analyzed in this chapter, including its main contribution and the possibilities of future work from now on. Then, the conclusions are divided in three sections: Main contribution, conclusions on the project and further work and proposals.

9.1 Main Contribution

As mentioned in Chapter 1, this work is an introduction of a new approach, consisting of a complex-valued sliding mode controller, in the robotics field to see the possible benefits and drawbacks of this solution. This project introduces this strategy successfully and shows that it is a solution that can bring some benefits when having a tracking problem for manipulators. It is an alternative that reduces the state-space (by merging two states in the complex domain) and presents a lower error than the current solutions in terms of simulation.

It also gives a basis on this technique for any robotic application, specially for those that combine different spatial states (as it could be the position (x, y) of a planar manipulator) and other interesting areas that use an extension of complex numbers such as quaternions. From this work, this control strategy can be extended to different robotic areas to explore new control alternatives and techniques, for example, the ones based on quaternions.

9.2 Conclusions on the project

The main objective of this thesis has been accomplished as a 2-link planar manipulator has been designed and it has been successfully analyzed and compared with some other existing solutions. It has been modeled with a nonlinear dynamic model and a complex-valued forward kinematics model of its TCP.

Three more controllers, using different control strategies, have been designed to use them as a comparison reference for the new CV-SMC proposed. Defining some KPIs and the tests that should be used, the main controller has been simulated and compared, obtaining the most robust performance of all the alternatives for the tracking analysis. It assures convergence in a finite time and is able to track the target position with the lowest error from all the alternatives. Regarding the regulation problem, the CV-SMC presents a similar response as the RV-SMC. Although in ideal conditions, the PID and SFC look a suitable option, when adding disturbances, uncertainties and sensor noise they are no longer able

to achieve the same performance as the sliding mode controllers.

On the other hand, the CV-SMC presents some other benefits that the other controllers do not have. First of all, it is a nonlinear controller that merges two states into a single one and does not need a precise knowledge of the system. Secondly, it does not require of any direct inverse kinematic calculations, as it is able compute the position from the joint state through the complex domain. Finally, it has also been seen how the control action needed to achieve the desired response is significantly lower than for the standard SMC.

9.3 Further work and proposals

This work leaves some aspects to be done in the future, starting from the implementation of the algorithm in a manipulator. The controller should be integrated, tuned and validated to obtain the best performance possible. A more detailed study with real application can also be done to see the real impact and behaviour of this strategy and compare it with the simulation results presented in this work.

This work also proposes to extend this control strategy to other applications in robotics such as the ones mentioned previously. It could be interesting to design a complex-valued sliding mode controller using quaternions as states variables to describe systems and rotations in the three-dimensional space or extend the problem presented in this project for robots with higher DOF.

Bibliography

[Ardema, 2004] Ardema, M. D. (2004). *Newton-Euler Dynamics*. Springer Science & Business Media.

[Craig, 2005] Craig, J. J. (2005). *Introduction to robotics: mechanics and control*. Pearson Education.

[Denavit and Hartenberg, 1955] Denavit, J. and Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *American Society of Mechanical Engineers*, pages 215–221.

[Donelan, 2010] Donelan, P. (2010). Kinematic singularities of robot manipulators. In *Advances in Robot Manipulators*. InTechopen.

[Dòria-Cerezo et al., 2020] Dòria-Cerezo, A., Olm, J. M., Biel, D., and Fossas, E. (2020). Sliding modes in a class of complex-valued nonlinear systems. *IEEE Transactions on Automatic Control*, 66(7):3355–3362.

[D'Souza et al., 2001] D'Souza, A., Vijayakumar, S., and Schaal, S. (2001). Learning inverse kinematics. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 1, pages 298–303. IEEE.

[Egerstedt and Martin, 2001] Egerstedt, M. and Martin, C. F. (2001). Optimal trajectory planning and smoothing splines. *Automatica*, 37(7):1057–1064.

[Gantt, 1974] Gantt, H. L. (1974). *Work, wages, and profits*. Easton [Pa.] : Hive Pub. Co.

[George Thuruthel et al., 2018] George Thuruthel, T., Ansari, Y., Falotico, E., and Laschi, C. (2018). Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5(2):149–163.

[Goldman, 2011] Goldman, R. (2011). Understanding quaternions. *Graphical models*, 73(2):21–49.

[Hirose, 2003] Hirose, A. (2003). *Complex-valued neural networks: theories and applications*, volume 5. World Scientific.

[Huynh et al., 2018] Huynh, H. N., Riviere-Lorphevre, E., and Verlinden, O. (2018). Multibody modelling of a flexible 6-axis robot dedicated to robotic machining. In *the 5th joint international conference on multibody system Dynamics*, pages 1–18.

[Jiang et al., 2020] Jiang, J., Huang, Z., Bi, Z., Ma, X., and Yu, G. (2020). State-of-the-art control strategies for robotic pih assembly. *Robotics and Computer-Integrated Manufacturing*, 65:101894.

[Johnson and Moradi, 2005] Johnson, M. A. and Moradi, M. H. (2005). *PID control*. Springer.

[Khan et al., 2012] Khan, M. F., ul Islam, R., and Iqbal, J. (2012). Control strategies for robotic manipulators. In *2012 International Conference of Robotics and Artificial Intelligence*, pages 26–33. IEEE.

[Lagrange, 1853] Lagrange, J. L. (1853). *Mécanique analytique*, volume 1. Mallet-Bachelier.

[Lee et al., 1983] Lee, C., Lee, B., and Nigam, R. (1983). Development of the generalized d'Alembert equations of motion for mechanical manipulators. In *The 22nd IEEE Conference on Decision and Control*, pages 1205–1210.

[MATLAB, 2022] MATLAB (2022). *R2022b*. The MathWorks Inc., Natick, Massachusetts.

[Mustafa, 2014] Mustafa, A. M. (2014). Modeling, simulation and control of 2-r robot. *Global Journals of Research in Engineering*, 14(H1):49–54.

[Paul, 1981] Paul, R. P. (1981). *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*.

[Utkin, 1993] Utkin, V. I. (1993). Sliding mode control design principles and applications to electric drives. *IEEE Transactions on Industrial Electronics*, 40(1):23–36.