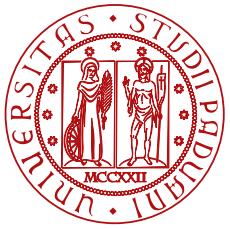


Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in Control System Engineering

Robotic Manipulator Control in the Presence of Uncertainty

Master Candidate

Mohamed Mahmoud Abdelwahab Mohamed

Student ID 2043173

Supervisor

Prof. Carli Ruggero

University of Padova

Academic Year
2022/2023

To my parents
and my wife

Abstract

This master's thesis presents a comprehensive investigation into the development and application of robust control techniques for robot manipulators operating in the presence of both known and unknown bounds of uncertainty. The primary focus is on achieving precise and robust control performance in challenging scenarios where uncertainties are inevitable. Two prominent control methodologies, namely robust feedback linearization and adaptive sliding mode control, are employed to address these challenges.

The first part of the thesis focuses on robust feedback linearization, which aims to linearize the nonlinear dynamics of the robot manipulator while considering uncertainties. The proposed approach incorporates a robust control law to handle bounded uncertainties, ensuring stability and enhanced tracking performance. Extensive simulations are conducted to evaluate the effectiveness and performance of the proposed robust feedback linearization technique.

The second part of the thesis explores adaptive sliding mode control, which is a powerful approach for achieving robustness against both parametric uncertainties and un-modeled dynamics. By integrating an adaptive law with the sliding mode control framework, the controller adapts online to unknown uncertainties, ensuring accurate tracking and disturbance rejection. The effectiveness of the proposed adaptive sliding mode control is verified through simulations and experiments on a robotic manipulator platform.

To assess the performance and compare the proposed control strategies, comprehensive evaluations are conducted using benchmark tasks and various manipulation scenarios that involve both known and unknown uncertainties. Performance metrics such as tracking accuracy, disturbance rejection, and robustness against uncertainties are quantitatively analyzed and compared.

The results demonstrate the effectiveness of both robust feedback linearization and adaptive sliding mode control in achieving robust and precise control for robot manipulators in the presence of uncertainties. The findings of this research provide valuable insights into the design and implementation of robust control strategies for real-world robotic applications.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

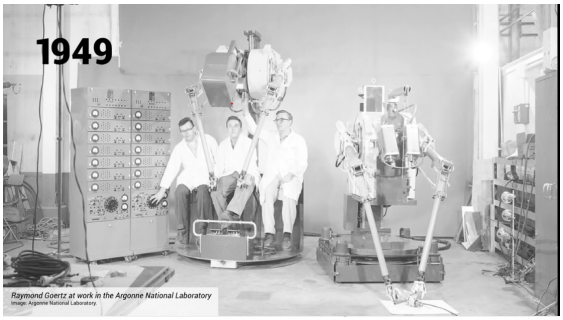


Figure 1.1: Argonne National Lab 1949

The history of robotics manipulators can be traced back to 1949 when Argonne National Lab first utilized them for assembling nuclear weapons. Nowadays ,Robotic manipulators play a pivotal role in various industrial applications, including assembly, welding, painting, and pick and place tasks. These applications are of significant importance in real-life scenarios as they offer automation, precision, and efficiency in industrial processes. The ability to achieve precise and smooth motion control is crucial for ensuring the successful execution of these tasks. However, the presence of uncertainty and external disturbances poses a formidable challenge to the control of robotic manipulators in industrial environments.

The precise and smooth motion control of robotic manipulators is essential for several reasons. In assembly tasks, it ensures accurate alignment and fitting of components, contributing to the overall quality and reliability of the final product. Welding applications require precise control to achieve high-quality welds, minimize defects, and reduce material waste. Painting tasks necessitate smooth and precise motion control to ensure even coating and prevent overspray. In pick and place operations, precise and coordinated motion control is vital for handling delicate objects without causing damage or misalignment.

Despite the exponential progress in the field of robotics, the design of robust and adaptive controllers for robotic manipulators remains an active area of research. Robust control methodologies capable of handling uncertainties and disturbances are crucial for achieving reliable and efficient performance in industrial applications. Developing robust control strategies that can adapt to uncertain and dynamic environments has a significant impact on improving industrial processes and outcomes.

The development and implementation of robust control techniques in industrial applications yield numerous benefits and have a profound impact on various sectors. The ability to handle uncertainties and disturbances enhances the reliability, repeatability, and safety of industrial processes involving robotic manipulators. This, in turn, leads to reduced production costs, improved product quality, increased production rates, and enhanced workplace safety.

The application of robust control methodologies in industrial settings ensures precise and smooth motion control, even in the presence of uncertainties and disturbances. By mitigating the adverse effects of uncertainties, these control strategies improve the accuracy and consistency of robotic manipulators, thereby reducing errors and rework. The capability to adapt to changing conditions and disturbances in real-time enables robotic manipulators to perform optimally in dynamic environments.

Moreover, the development of robust control techniques in the field of robotics has far-reaching implications beyond industrial applications. Advancements in robust control methodologies can also benefit other domains such as healthcare, service robotics, and autonomous systems. These technologies have the

potential to improve surgical procedures, facilitate assistance to people with disabilities, and enhance the reliability of autonomous vehicles.

1.2 Thesis contributions and objectives

This master's thesis aims to address the challenges posed by uncertainties and disturbances in the control of robotic manipulators through the investigation and development of robust control strategies. By enhancing the reliability and efficiency of robotic manipulator systems, the research contributes to the ongoing efforts to improve industrial processes and advance the field of robotics.

The primary objective of this thesis is to develop robust and adaptive control strategies to overcome the challenges associated with controlling robot motion under unknown uncertainty of dynamic parameters. Specifically, the thesis aims to achieve the following objectives:

1. Investigate the impact of unknown uncertainty and dynamic parameters on robot motion control in industrial applications.
2. Explore existing control methodologies and identify their limitations in handling unknown uncertainties.
3. Develop novel robust and adaptive control strategies that can effectively handle unknown uncertainties in real-time.
4. Evaluate and compare the performance of the proposed control strategies with existing approaches through simulations and experiments.
5. Demonstrate the applicability and effectiveness of the developed control strategies in representative industrial tasks

1.3 Thesis Structure

This thesis is structured into three main parts, each addressing distinct aspects of robotics modeling and control. Part I provides an introduction to the field, presenting two cases of kinematic and dynamic modeling for a planar RR

manipulator and a SCARA manipulator. These models serve as the foundation for subsequent chapters focused on simulation and controller development. Part II constitutes the core of the thesis, discussing common approaches for manipulator control and extending these controllers to handle unknown uncertainties. It also includes an analysis of stability, convergence, and adaptability under varying levels of unknown parameter variation and noise. Finally, Part III encompasses the conclusions, suggestions for future work.

Outline of Part I

Chapter 1: Introduction

- Motivation: Applications of manipulators and the necessity for precise controllers
- Thesis contributions and Objectives
- Structure of the Thesis

Chapter 2: Background Review:

- Linear Algebra review
- stability review

Chapter 3: Manipulator Modeling:

- General manipulator modeling using the Lagrangian approach
- Case 1: Kinematic and dynamic modeling of a planar RR manipulator
- Case 2: Kinematic and dynamic modeling of a SCARA manipulator
- Schemes and simple simulations

Outline of Part II

Chapter 4: Manipulator Controllers

- Open loop Control: Pros and Cons, Performance evaluation
- Feedback Linearization

- Robust Feedback Linearization
- Sliding Mode Control

Chapter 5: Implementation,Simulation and Analysis

- Types of Uncertainty
- Uncertainty modeling techniques
- Disturbances: Constant vs. time-varying
- Case 1: RR Experimental Design
- Case 2: SCARA Experimental Design
- Case 2: UR10 Experimental Design
- Results and Analysis

Outline of Part III

Chapter 6: Conclusions and Future Work

- Summary of Achievements
- Contributions to the Field
- Limitations and Challenges
- Suggestions for Future Research
- Conclusion

2

Background Review

2.1 Linear Algebra

2.1.1 Introduction

Linear algebra plays a vital role in the modeling, analysis and control of robot manipulators. It provides a powerful mathematical framework for understanding the kinematics, dynamics, and control of robotic systems. This chapter highlights the importance of linear algebra in robot manipulator analysis and control, showcasing how it enables the representation and solution of complex robotic problems [1].

2.1.2 Linear Algebra Review for Robotics

Vectors and Matrices

In robotics, vectors and matrices are fundamental entities used to represent positions, orientations, and transformations of robot manipulators. A vector represents a quantity with both magnitude and direction, while a matrix is a rectangular array of numbers. Vectors in 3D space are commonly denoted as:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

where v_1, v_2, and v_3 are the components of the vector in three orthogonal directions.

Dot and Cross Product

Dot Product

The dot product of two vectors **a** and **b** is a scalar value defined as:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|\|\mathbf{b}\| \cos(\theta)$$

where θ is the angle between the vectors, and $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ are the magnitudes of the vectors.

Cross Product

The cross product of two vectors **a** and **b** is another vector **c** defined as:

$$\mathbf{c} = \mathbf{a} \times \mathbf{b}$$

where the components of **c** are given by:

$$c_1 = a_2 b_3 - a_3 b_2$$

$$c_2 = a_3 b_1 - a_1 b_3$$

$$c_3 = a_1 b_2 - a_2 b_1$$

2.1.3 Homogeneous Transformations

Homogeneous transformations are 4x4 matrices used to represent the pose and transformation of robot manipulators. The transformation matrix from frame A to frame B is denoted as T_B^A. It consists of a 3x3 rotation matrix R_B^A representing orientation and a 3x1 translation vector $\mathbf{d}_B^A$ representing position:

$$T_B^A = \begin{bmatrix} R_B^A & \mathbf{d}_B^A \\ \mathbf{0}^T & 1 \end{bmatrix}$$

2.1.4 Image and Kernel in Robotic Analysis

Image & Kernel

The image (column space) and kernel (null space) of a matrix play a crucial role in robotic manipulator analysis. The image of a matrix A is the subspace spanned by all possible linear combinations of its columns. It represents the range of the linear transformation induced by A. The kernel of A is the set of all vectors that are mapped to the zero vector under the linear transformation of A.

Usage in Robotic Manipulator Analysis

Solving Forward Kinematics

The forward kinematics problem in robotics involves determining the end-effector pose given the joint variables [2]. The transformation matrix T_{end} representing the end-effector pose can be expressed as a function of joint variables. The image of the forward kinematics matrix provides the reachable workspace of the manipulator, while the kernel of Jacobian matrix indicates the singular configurations where the manipulator loses mobility.

Solving Inverse Kinematics

The inverse kinematics problem in robotics involves finding the joint variables given the desired end-effector pose. It is an essential task in trajectory planning and control. The kernel of the Jacobian matrix represents the set of joint velocities that yield zero end-effector velocity. This is crucial in singularity avoidance and dexterity analysis.

2.1.5 Conclusion

Linear algebra is an indispensable tool in robot manipulator analysis and control. It provides the foundation for representing positions, orientations, and transformations, as well as solving kinematic and dynamic equations. The concepts of image and kernel of the matrix are essential in understanding the manipulator's reachable workspace, singularity analysis, and trajectory planning. By leveraging linear algebra, researchers and engineers can develop robust and efficient control strategies for various robotic applications.

2.2 Stability

2.2.1 Introduction

In the field of manipulator control, stability analysis plays a crucial role in ensuring the safe and efficient operation of robotic arms and manipulators. Stability is the property of a system that characterizes its ability to return to an equilibrium state after experiencing perturbations or disturbances. Understanding the stability of manipulator systems is essential to design reliable control strategies and guarantee their robustness in real-world applications. This chapter aims to explore the importance of stability analysis in manipulator control, discuss different types of stability, and focus on Lyapunov Stability, its types, and its relation to manipulator control [3].

2.2.2 Importance of Stability Analysis

Stability analysis is a fundamental aspect of manipulator control as it provides insights into the system's behavior under various conditions. The stability of a manipulator determines whether the system will converge to a desired configuration or trajectory or deviate uncontrollably. In robotic applications, such as pick-and-place tasks, manufacturing, or autonomous vehicles, stable control is crucial for precise and accurate movements.

Without proper stability analysis, the manipulator could exhibit oscillations, vibrations, or even uncontrollable motion, leading to potential accidents, damages, or loss of efficiency. Stability analysis aids in designing control algorithms that ensure the manipulator remains within acceptable operating limits, preventing any harmful consequences.

2.2.3 Types of Stability

Lyapunov Stability

Lyapunov Stability is a widely used concept in control theory to assess the stability of dynamical systems, including manipulators. The primary idea is to determine if the system's trajectories remain bounded and converge to an equilibrium point or a desired trajectory.

Let us consider a manipulator system described by the state-space representation:

$$\dot{x} = f(x, u)$$

where x represents the state vector, and u is the control input.

A point x_e is an equilibrium point if $f(x_e, 0) = 0$.

Asymptotic Stability

A system is considered asymptotically stable if it is stable (trajectories remain bounded) and, additionally, all trajectories starting sufficiently close to an equilibrium point converge to that equilibrium point as time approaches infinity.

Mathematically, the asymptotic stability condition is expressed as follows:

For all $\epsilon > 0$, there exists a $\delta > 0$ such that
if $\|x(0) - x_e\| < \delta$, then $\lim_{t\to\infty} \|x(t) - x_e\| = 0$.

Exponential Stability

Exponential stability is a stronger form of stability. A system is exponentially stable if there exist positive constants k and λ such that the distance between the trajectory and the equilibrium point decays exponentially over time.

Mathematically, exponential stability is characterized as follows:

$\|x(t) - x_e\| \leq ke^{-\lambda t}$, where $k > 0$ and $\lambda > 0$.

BIBO Stability (Bounded Input, Bounded Output)

BIBO stability deals with the response of a system to bounded inputs. A system is BIBO stable if, for every bounded input, the output of the system remains bounded.

Mathematically, BIBO stability can be defined as follows:

For a bounded input $u(t)$ such that $\|u(t)\| \leq M_u$ for all t, there exists a constant M_y such that $\|y(t)\| \leq M_y$ for all t.

Lyapunov Stability and Manipulator Control

In manipulator control, Lyapunov Stability [4] provides a powerful framework to analyze and design stable control algorithms. The basic idea is to design

a Lyapunov function $V(x)$ that satisfies the following properties:

1. $V(x) > 0$ for all $x \neq 0$ (positive definite).
2. $V(0) = 0$ (positive semidefinite).
3. $\dot{V}(x) \leq 0$ for all x (negative definite).

If the system's state trajectories satisfy these conditions, then the system is stable. Furthermore, the type of Lyapunov function used can help determine the type of stability achieved, such as asymptotic or exponential stability.

2.2.4 Conclusion

Stability analysis is a critical aspect of manipulator control. It ensures that robotic arms and manipulators operate safely, efficiently, and within desired specifications. Lyapunov Stability, in particular, provides a powerful mathematical framework to assess the stability of manipulator systems and design control algorithms that guarantee desirable behaviors. By understanding different types of stability and leveraging Lyapunov's principles, engineers and researchers can create reliable and robust control strategies for a wide range of manipulator applications.

3

Manipulator Modelling

3.1 Forward Kinematics

In the study of robotics, understanding the kinematics and dynamics of robot manipulators is essential for analyzing their behavior and performance. One crucial aspect of kinematics is the forward kinematics problem, which involves mapping the joint variables of a manipulator $q_i \in \mathbb{R}^n$ to the pose of its end-effector, including both position and orientation. In this section, we will delve into the intricacies of the forward kinematics problem and explore various methods used to solve it, such as the Denavit-Hartenberg (DH) Convention and the screw axis representation. Additionally, we will highlight the significance of using transformations to represent rigid bodies with respect to a global frame.

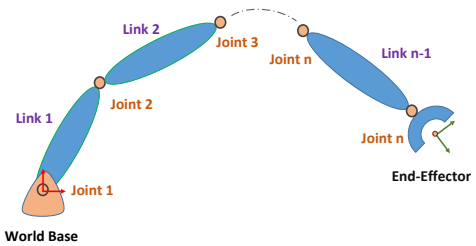


Figure 3.1: General Robot

3.1.1 Forward Kinematics Problem

The forward kinematics problem is concerned with determining the position and orientation of a manipulator's end-effector based on the joint variables, which typically correspond to the joint angles or displacements. Mathematically, this problem can be defined as finding the transformation matrix T, which represents the pose of the end-effector frame (usually denoted as frame E) with respect to a base frame (often referred to as frame W). The transformation matrix T relates the joint variables (q) to the end-effector pose (p, R), where p represents the position vector and R represents the Rotation matrix.

3.1.2 Methods for Solving Forward Kinematics

Method(1): Denavit-Hartenberg (DH) Convention The DH Convention [5] is a systematic sequence of steps commonly used to solve the forward kinematics problem for serial manipulators. It provides a framework to establish a set of reference frames for each joint of the manipulator, enabling a straightforward representation of the end-effector pose.

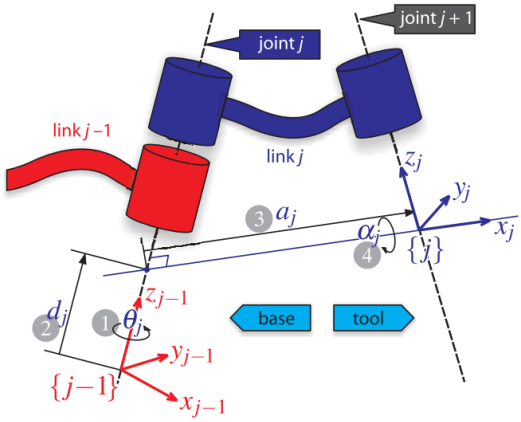


Figure 3.2: The colors red and blue denote all things associated with links $i1$ and i respectively. The numbers in circles represent the order in which the elementary transforms are applied.

The transformation from link coordinate frame $\{j-1\}$ to frame $\{j\}$ is defined in terms of elementary rotations and translations as :

$$T_{i-1}^{i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

where,

Joint Angle	$\boldsymbol{\theta_i}$	Rotation around $\boldsymbol{z_{i-1}}$ between $\boldsymbol{x_{i-1}}$ and $\boldsymbol{x_i}$	**Revolute joint variable**
Link Offset	$\boldsymbol{d_i}$	Translation along $\boldsymbol{z_{i-1}}$ from $\boldsymbol{O_{i-1}}$ to $\boldsymbol{O_{i-1}}$	**Prismatic joint variable**
Link Length	$\boldsymbol{a_i}$	Translation along $\boldsymbol{x_i}$ from $\boldsymbol{O_{i-1}}$ to $\boldsymbol{O_{i-1}}$	**Constant**
Link Twist	$\boldsymbol{\alpha_i}$	Rotation around $\boldsymbol{x_i}$ between $\boldsymbol{z_{i-1}}$ and $\boldsymbol{z_i}$	**Constant**

Figure 3.3: Denavit-Hartenberg parameters and their physical meaning

The steps involved in the DH Convention are as follows:

- **Step 1:** Assign reference frames to each joint: Typically, the joint reference frames are attached to the links of the manipulator, following a predefined convention.
- **Step 2:** Assign coordinate systems: Define coordinate systems for each joint frame, specifying the origin and orientation of the frame using appropriate parameters, namely the link length (a), link twist (α), link offset (d), and joint angle (θ).
- **Step 3:** Derive the transformation matrix: Use the DH parameters to calculate the transformation matrix between consecutive joint frames, yielding a transformation matrix T_i that represent the total change of a frame i w.r.t frame $i-1$.
- **Step 4:** Compute the forward kinematics: To obtain the transformation matrix T between the base frame W and the end-effector frame E, multiply

the individual transformation matrices in a sequential manner: $T_n^0 = T_1^0 * T_2^1 * ... * T_n^{n-1}$.

Method(2):Screw Axis Representation Another method commonly used for solving the forward kinematics problem is the screw axis representation [6]. This approach characterizes the rigid body motion of a joint by a twist, defined as a pair of a screw axis and a pitch. The twist represents the direction and magnitude of the joint's instantaneous velocity. By recursively propagating the twists from the base to the end-effector, the end-effector pose can be determined. Utilizing Transformations for Rigid Body Representation.

3.2 Velocity Kinematics

The forward kinematic equations establish a functional relationship between the space of Cartesian positions and orientations and the space of joint positions in robot manipulators. In turn, the velocity relationships are derived from the Jacobian matrix associated with this function. The Jacobian matrix can be considered as the vector representation of the ordinary derivative of a scalar function. In the analysis and control of robot motion, the Jacobian matrix holds paramount importance, as it plays a crucial role in various aspects of robotic manipulation.

Firstly, the Jacobian matrix allows for the determination of the velocity in the workspace based on the given joint velocities. By relating the joint velocities to the end-effector velocities, it enables the understanding and prediction of the overall motion and behavior of the manipulator.

Secondly, the Jacobian matrix facilitates the control of the velocity in the workspace. By manipulating the joint velocities, one can precisely and efficiently control the motion of the end-effector, which is crucial for successful object manipulation and interaction with the environment.

Thirdly, the Jacobian matrix aids in identifying "singular configurations" where small workspace velocities result in large joint velocities. These configurations are of significant concern as they can limit the manipulator's dexterity and stability. By detecting and avoiding singular configurations, more reliable

and robust robot motions can be achieved.

Furthermore, the Jacobian matrix is employed in calculating the inertia matrix of the robot. This matrix assists in transforming forces and torques from the end-effector to the manipulator joints, thereby facilitating the analysis and control of the manipulator's dynamic behavior.

3.2.1 Jacobian calculation

The relation between joint velocities $\dot{q}$ and end-effector velocity can be written as :

$$v = J\left(q\right)\dot{q} \tag{3.2}$$

where,

$$J = \begin{bmatrix} J_{v_1} & ..... & J_{v_n} \\ J_{w_1} & ..... & J_{w_n} \end{bmatrix} \qquad and \qquad v = \begin{bmatrix} v \\ w \end{bmatrix}$$

- v : end-effector linear velocity
- w : end-effector angular velocity
- J_{v_i} : describe the effect of the joint on translational velocity
- J_{w_i} : describe the effect of the joint on rotational velocity

we can evaluate J_v & J_w for any joint based on its type as follows:

$$J_v = \begin{bmatrix} J_{v_1} & J_{v_2} & \cdots & J_{v_n} \end{bmatrix} \tag{3.3}$$

where the i-th column J_{v_i} is defined as:

$$J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & \text{for revolute joint } i \\ z_{i-1} & \text{for prismatic joint } i \end{cases} \tag{3.4}$$

The lower half of the Jacobian is given as:

$$J_\omega = \begin{bmatrix} J_{\omega_1} & J_{\omega_2} & \cdots & J_{\omega_n} \end{bmatrix} \tag{3.5}$$

where the i-th column J_{ω_i} is defined as:

$$J_{\omega_i} = \begin{cases} z_{i-1} & \text{for prismatic joint } i \\ 0 & \text{for revolute joint } i \end{cases} \tag{3.6}$$

Combining the upper and lower halves of the Jacobian, the Jacobian matrix for an n-link manipulator can be expressed in matrix form as: $J = \begin{bmatrix} J_1 & J_2 & \cdots & J_n \end{bmatrix}$
where the i-th column J_{v_i} is defined as:

$$J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & \text{for revolute joint } i \\ z_{i-1} & \text{for prismatic joint } i \end{cases} \tag{3.7}$$

The lower half of the Jacobian is given as:

$$J_\omega = \begin{bmatrix} J_{\omega_1} & J_{\omega_2} & \cdots & J_{\omega_n} \end{bmatrix} \tag{3.8}$$

where the $i^t h$ column J_ω is defined as:

$$J_{\omega i} = \begin{cases} z_{i-1} & \text{for prismatic joint } i \\ 0 & \text{for revolute joint } i \end{cases} \tag{3.9}$$

Combining the upper and lower halves of the Jacobian, the Jacobian matrix for an n-link manipulator can be expressed as:

$J = \begin{bmatrix} J_1 & J_2 & \cdots & J_n \end{bmatrix}$ The i-th column J_i is defined as follows:

If joint i is revolute:

$$J = \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{3.10}$$

If joint i is prismatic:

$$J = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} \tag{3.11}$$

3.3 Dynamics

The dynamics of robotic manipulators play a vital role in understanding and analyzing their motion and performance. While the kinematic model provides information on motion between poses, dynamics provides insights into various aspects such as the influence of robot material, the impact of object weight on joint torques, the effect of external forces, and the determination of collaboration or resistance from these forces. By employing approaches such as the Newtonian and Lagrangian methods, dynamic equations can be derived to describe the manipulator's behavior. Additionally, understanding the different sources of actuator torques is crucial for ensuring the stability and proper functioning of the manipulator. Advancements in manipulator dynamics contribute to enhanced manipulator control, simulation, and overall performance in a wide range of applications.

3.3.1 Approaches to Derive Dynamic Equations

To derive the dynamic equations for a robotic manipulator, two common approaches are widely employed: the Newtonian approach and the Lagrangian approach.

1. Newtonian Approach: The Newtonian approach applies Newton's laws of motion to each link and joint of the manipulator. It involves determining the forces and torques acting on each element and subsequently solving the equations of motion. This approach can be computationally efficient if applied in a recursive approach [7].

2. Lagrangian Approach: The Lagrangian approach [8] utilizes the concept of the Lagrangian function, which is defined as the difference between kinetic and potential energies. It provides a systematic and efficient method for deriving the equations of motion for robotic manipulators.

Newtonian Approach	Lagrangian Approach
- Uses **Newton & Euler** equations to derive body dynamics**.**	- Uses **Lagrange equations** for energy analysis of the body to derive dynamics.
- Depends on the analysis of each body's **forces** and **moments.**	- Depends on calculating **energies** of the robot.
- **Less common** in **Robotics research.**	- **More common** in **Robotics research** as it provides symbolic representation that is useful for control.
- **More common** in **Robotics Industry** as it needs much less computational power.	- **Less common** in **Robotics Industry**.

Figure 3.4: Comparison between Newtonian Approach Vs Lagrangian Approach

The steps involved in calculating the equation of motion using the Lagrangian approach can be summarized as follows:

Step 1: Define the Lagrangian function: The Lagrangian function ($\mathcal{L}$) is defined as the difference between the kinetic energy (K) and the potential energy (P) of the manipulator system: $\mathcal{L} = K - P$.

Step 2 Determine the generalized coordinates: Identify a set of generalized coordinates that describe the configuration of the manipulator. These coordinates capture the essential degrees of freedom required to specify the system's state.

Step 3: Derive the Euler-Lagrange equations: Apply the Euler-Lagrange equations, which are derived from the Lagrangian function, to obtain a set of coupled differential equations that describe the system's dynamics.

Step 4: Solve the equations of motion: Solve the resulting differential equations to determine the joint torques or forces required to achieve the desired motion of the manipulator.

3.3.2 Actuator Torques and Sources

To ensure the proper functioning of a robotic manipulator, the actuator must balance torques from various sources:

1) Dynamic Torques: Dynamic torques are generated due to the motion of the manipulator and consist of three components:

- **Inertial torques**: Proportional to joint acceleration, in accordance with Newton's law.
- **Centripetal torques**: Proportional to the square of joint velocity, directed toward the center of circular motion.
- **Coriolis torques**: Arise from the interaction between two rotating links, resulting in vertical forces.

2) Static Torques: Static torques arise due to friction within the manipulator's joints.

3) Gravity Torques: Gravity torques are caused by the gravitational forces acting on the manipulator's links.

4) External Torques: External torques are exerted on the manipulator's end-effector, typically resulting from the task being performed.

3.3.3 Equation of Motion

In this section, we focus on the specialization of the Euler-Lagrange equations to a specific case where two conditions are satisfied. Firstly, the kinetic energy (K) is expressed as a quadratic function of the vector $\dot{q}$ in the form: Consider a manipulator consisting of n links. As discussed in the Velocity Kinematics section, the linear and angular velocities of any point on any link can be represented using the Jacobian matrix and the derivative of the joint variables. In our case, since the joint variables correspond to the generalized coordinates, we can express the velocities as:

$$v_i = J_{v_i}(q)\dot{q}, \quad \omega_i = J_{\omega_i}(q)\dot{q} \tag{3.12}$$

Suppose the mass of link i is m_i and the inertia matrix of link i, evaluated around a coordinate frame parallel to frame i but with its origin at the center of mass, is denoted as I_i. In such a scenario, the overall kinetic energy of the manipulator can be computed as:

$$K = \frac{1}{2}\dot{q}^T \left(\sum_{i=1}^{n} m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q) \right) \dot{q} \tag{3.13}$$

Here, $R_i(q)$ represents the rotation matrix associated with frame i.

$$K = \frac{1}{2} \sum_{i,j} d_{ij}(q) \dot{q}_i \dot{q}_j := \frac{1}{2} \dot{q}^T D(q) \dot{q} \tag{3.14}$$

Here, the $n \times n$ "inertia matrix" $D(q)$ is symmetric and positive definite for each $q \in \mathbb{R}^n$. Secondly, the potential energy (P) is independent of $\dot{q}$. It is worth mentioning that these conditions are satisfied by robotic manipulators.

Consider the potential energy term. In the case of rigid dynamics, the sole source of potential energy is gravity. The potential energy associated with the i-th link can be computed by assuming that the entire object's mass is concentrated at its center of mass. This potential energy is given by:

$$P_i = g^T l_{ci} m_i \tag{3.15}$$

Here, g is a vector representing the direction of gravity in the inertial frame, l_{ci} denotes the coordinates of the center of mass of link i, and m_i represents the mass of link i.

Consequently, the total potential energy of the n-link robot can be expressed as:

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} g^T l_{ci} m_i$$

The Euler-Lagrange equations for such a system can be derived as follows. Considering that:

$$\mathcal{L} = K - P = \frac{1}{2} \sum_{i,j} d_{ij}(q) \dot{q}_i \dot{q}_j - P(q) \tag{3.16}$$

The Euler Formulation can be then written as:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} = \tau_j \tag{3.17}$$

where $\frac{d}{dt}$ represents the derivative with respect to time, $\frac{\partial \mathcal{L}}{\partial \dot{q}_j}$ denotes the partial derivative of the Lagrangian ($\mathcal{L}$) with respect to $\dot{q}_j$, $\frac{\partial \mathcal{L}}{\partial q_j}$ represents the partial derivative of the Lagrangian with respect to q_j, and τ_j represents the generalized force or torque acting on the j-th degree of freedom of the system.

The matrix form of the Euler-Lagrange equations can be expressed as:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{3.18}$$

where $D(q)$ represents the inertia matrix, $\ddot{q}$ denotes the second derivative of q with respect to time, $C(q, \dot{q})$ represents the Coriolis matrix, $\dot{q}$ is the first derivative of q with respect to time, $g(q)$ represents the gravity vector, and τ represents the vector of external torques or forces applied to the system.

3.4 Case Studies

we will comprehensively investigate the kinematics and dynamics of two distinct types of robot manipulators, namely the RR planar manipulator and the SCARA manipulator.The analysis and understanding of these models will serve as a foundation for the subsequent control section, where they will be extensively utilized. By studying the kinematic and dynamic properties of these manipulators, we aim to enhance our comprehension of their behavior and enable effective control strategies for achieving desired motion and performance.

3.4.1 RR Planar Manipulator

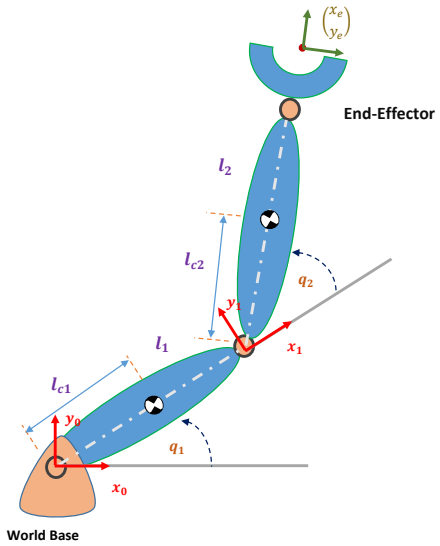


Figure 3.5: RR PLanar Manipulator

Forward Kinematics

Link	θ	d	a	α
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0

Table 3.1: DH-table for 2-link RR planar manipulator

From (3.1), we have the following transformation matrices:

$$A_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The T-matrices are then given by:

$$T_1^0 = A_1$$

$$T_2^0 = A_1A_2 = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that the first two entries of the last column of T_2^0 represent the end-effector position x_e and y_e w.r.t the base frame, given by:

$$x_e = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) \tag{3.19}$$

$$y_e = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) \tag{3.20}$$

Velocity Kinematics

Since both joints are revolute, the Jacobian matrix, denoted as $J(q)$, has dimensions of 6×2 and can be represented as:

$$J(q) = \begin{bmatrix} z_0 \times (O_2 - O_0) & z_1 \times (O_2 - O_1) \end{bmatrix} \tag{3.21}$$

The various quantities in the equation are as follows:

$$O_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad O_1 = \begin{bmatrix} l_1\cos(\theta_1) \\ l_1\sin(\theta_1) \\ 0 \end{bmatrix} \qquad O_2 = \begin{bmatrix} l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2) \\ 0 \end{bmatrix}$$

$$z_0 = z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

By performing the necessary calculations as described in Section 3.2, we obtain the Jacobian matrix J given by:

$$J = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.22}$$

It is evident that the first two rows of Eq. (3.22) correspond precisely to the 2×2 Jacobian obtained by differentiation of Eq. (3.19) and (3.20), representing the linear velocity of the origin O_2 relative to the base. The third row in Eq. (3.22) denotes the linear velocity in the direction of z_0, which is constantly zero since we are in planar case. The final three rows indicate the angular velocity of the end frame, which is simply a rotation about the vertical axis at a rate of $\dot{\theta}_1 + \dot{\theta}_2$.

Dynamics

Inertia Matrix

Following the Lagrangian approach mentioned in Section (3.3.3), the inertia matrix can be written as:

$$D(q) = \begin{bmatrix} d_{11} & d_{21} \\ d_{12} & d_{22} \end{bmatrix} \tag{3.23}$$

where,

$$\begin{aligned} d_{11} &= m_1 \dot{q}_1^2 c_1 + m_2(\dot{q}_2^2 + \dot{q}_1^2 c_2 + 2\dot{q}_1 \dot{q}_2 c_2 + 2\dot{q}_1 \dot{c}_2 \cos q_2) + I_1 + I_2 \\ d_{12} &= d_{21} = m_2(\dot{q}_2^2 c_2 + \dot{q}_1 \dot{c}_2 \cos q_2) + I_2 \\ d_{22} &= m_2 \dot{q}_2^2 c_2 + I_2 \end{aligned}$$

Coriolis and centrifugal matrix

$$C(q,\dot{q}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{3.24}$$

Where,

$$h = -m_2\dot{q}_1\dot{c}_2 \sin q_2$$
$$c_{11} = h\dot{\theta}_2$$
$$c_{12} = h\dot{\theta}_1 + \dot{\theta}_2$$
$$c_{21} = -h\dot{\theta}_1$$
$$c_{22} = 0$$

Gravitational forces

$$g\,(q) = \begin{bmatrix} (m_1\dot{c}_1 + m_2\dot{q}_1)g\cos q_1 + m_2\dot{c}_2 g\cos(q_1+q_2) \\ m_2\dot{c}_2\cos(q_1+q_2) \end{bmatrix} \tag{3.25}$$

3.4.2 SCARA Manipulator

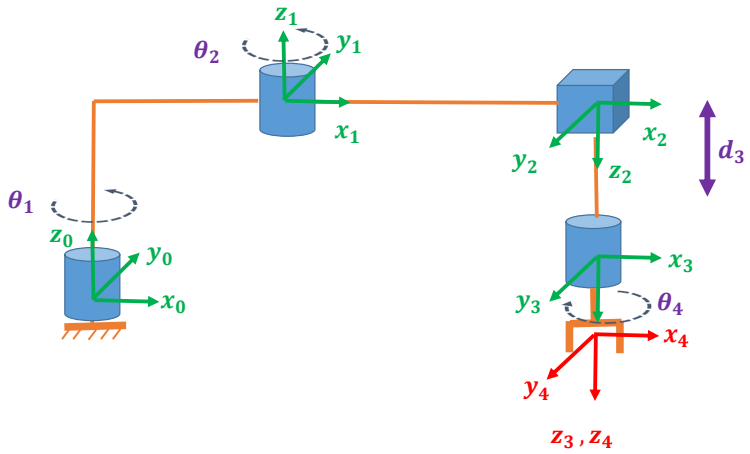


Figure 3.6: SCARA Manipulator

Forward Kinematics

Link	θ	d	a	α
1	${\theta_1}^*$	0	l_1	0
2	${\theta_2}^*$	0	l_2	π
3	0	${d_3}^*$	0	0
4	${\theta_4}^*$	d_4	0	0

Table 3.2: DH-table for SCARA manipulator

Velocity Kinematics

We will now derive the Jacobian of the SCARA [] . The SCARA manipulator has four degrees of freedom, thus resulting in a 6×4 Jacobian matrix. To compute the Jacobian, we need to calculate the transformation matrices $T_j^0 = A_1 \dots A_j$, where the A-matrices are determined by Eq. (3.1).

Since joints 1, 2, and 4 are revolute, and joint 3 is prismatic, with $o_4 - o_3$ being parallel to z_3 (and hence $z_3 \times (o_4 - o_3) = 0$), the Jacobian takes the following form:

$$J = \begin{bmatrix} z_0 \times (o_4 - o_0) & z_1 \times (o_4 - o_1) & z_2 & z_0 \\ z_0 & z_1 & 0 & z_3 \end{bmatrix}$$

Performing the calculations, we obtain the following expressions:

$$o_1 = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix}, \quad o_2 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \\ 0 \end{bmatrix}, \quad o_4 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_2 + l_2 s_{12} \\ d_3 - d_4 \end{bmatrix}$$

Similarly, we have $z_0 = z_1 = k$ and $z_2 = z_3 = -k$. where, $k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. Therefore, the Jacobian of the SCARA manipulator is given by:

$$J = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \tag{3.26}$$

4

Manipulator Controllers

4.1 Introduction

Robot control plays a vital role in enabling robotic systems to perform various tasks with accuracy and precision. The primary goal of robot control is to determine the time history of torques that need to be sent to the actuators, allowing the robot to execute desired tasks. There are two types of tasks: regulation, where a constant reference is maintained, and tracking, where the reference trajectory varies with time.

The manipulator system constitutes a intricate multi-input multi-output framework characterized by strong interconnections, uncertain behaviors, and nonlinear dynamics. The prevailing uncertainty can be broadly classified into two distinct categories: structural uncertainty, encompassing perturbations in system parameters, unaccounted dynamics, frictional components both static and dynamic, and elasticity; and non-structural uncertainty, arising from external disturbances, actuator saturation, sampling delays, measurement errors, and other external influences.

In the realm of industrial applications, various established control approaches for manipulators have been explored, spanning from traditional methodologies to advanced techniques. These include PID control [9]-[11], feedback linearization [12],[13], adaptive backstepping control [14], discrete control [15], adaptive control [16]-[18], robust control [19],[20], neural network control [21] , fuzzy

control [22], iterative learning control [23], and variable structure control [24].

PID control, suited for linear systems, proves effective for relatively straightforward scenarios but lacks the capacity to govern systems characterized by intricacy, significant inertia, and pronounced lag [25]. Discrete control, operating through step-wise sampling, excels in noise reduction and showcases commendable resistance to interference [26]. Adaptive backstepping control introduces a virtual control concept, decomposing the system into simpler low-order entities for effective regulation [27]. Feedback linearization involves linearizing both state and output equations, presenting a common technique to address the limitations of conventional linearization which worsens with expanding operational ranges. Nevertheless, this method isn't universally applicable to all nonlinear systems and requires complete state measurement. In cases of parameter perturbations, system robustness cannot be assured.

Robust control, resilient against external perturbations and parameter variations, suits systems with widely varying ranges of uncertainty, yet often compromises steady-state accuracy due to non-optimality . Neural network and fuzzy control emulate human intelligence to tackle intricate nonlinear challenges. By designing suitable network architectures, optimal control system parameters can be approximated to ensure control precision .Iterative learning control proves valuable for repetitive motion control tasks, particularly for complex nonlinear systems with strong coupling and intricate modeling. Similar to robust control, it accommodates system uncertainties and promotes high steady-state accuracy, achieving comprehensive tracking .

The combination of these control methodologies spawns various hybrid approaches, enhancing the potential to proficiently and accurately address trajectory tracking challenges in manipulator control. Adaptive controllers strive to update unknown dynamic parameters using suitable updating laws. An advantageous trait of adaptive controllers is their ability to function without prior knowledge of constants like payload masses or friction coefficients. However, these controllers demand substantial real-time computation and lack robustness against bounded additive disturbances.

In this research, we address the control of robots when their dynamical

model is uncertain. Uncertainty in the model may arise due to factors such as the robot carrying an unknown load or the high cost associated with accurately evaluating the robot's dynamics. To overcome these challenges, robust control strategies are employed.

Control of uncertain systems is typically achieved through two main approaches: adaptive control and robust control. The adaptive approach involves designing a controller that aims to "learn" the uncertain parameters of the system and eventually become the optimal controller for the given system. On the other hand, the robust approach employs a fixed controller structure that ensures acceptable performance for a class of plants, which includes the uncertain plant under consideration. Although the adaptive approach is applicable to a wider range of uncertainties, robust controllers are simpler to implement and require no tuning specific to the plant.

In this chapter, we review various control designs used in controlling the motion of robots. Additionally, we introduce a novel approach that enables the control law to effectively handle unknown uncertainties. By developing robust control strategies, we aim to enhance the performance and accuracy of manipulator control in the presence of uncertainty, thereby expanding the capabilities of robotic systems in real-world applications.

4.2 Inverse Dynamics Control

Considering that dynamics of general manipulator can be written as follows:

$$D(q)\ddot{q} + \underbrace{C(q,\dot{q})\dot{q} + F\dot{q} + g(q)}_{n(q,\dot{q})} = \tau \tag{4.1}$$

Idea: Using our knowledge of robot dynamics to compute the required torque to follow desired trajectory of $(q_d, \dot{q}_d)$

Let control law:

$$\tau = D(q_d)\ddot{q}_d + n(q_d, \dot{q}_d) \tag{4.2}$$

Assume: exact initialisation $q(0) = q_d(0), \dot{q}(0) = \dot{q}_d(0)$

Then system dynamics dynamics after substituting with (4.2) in (4.1) will be:

$$\ddot{q}(t) = \ddot{q}_d(t)$$

This approach can't handle the following:

- Initial state not matched to the desired trajectory q.
- External disturbance.
- Inaccurate model parameters.
- Unknown payload.
- Unmodeled dynamics.

4.3 Feedbak Linearization

4.3.1 Introduction

In the realm of robot manipulator control, feedback linearization emerges as a valuable nonlinear design methodology [28],[29] . Its fundamental concept revolves around transforming a nonlinear system into a (fully or partially) linear system, enabling the application of well-established linear design techniques for control design. Feedback linearization has proven to be effective in addressing various practical nonlinear control challenges. It is particularly applicable to important classes of nonlinear systems known as input-state linearizable or minimum-phase systems, albeit with the requirement of full state measurement. However, it is important to note that while feedback linearization offers valuable advantages in terms of system linearity and control design, it does not inherently guarantee robustness in the presence of parameter uncertainty or disturbances.

Idea:

The concept of feedback linearization in manipulator control involves the application of state feedback to cancel non-linearity followed by the use of linear control to stabilize the tracking error to zero. State feedback is utilized to mitigate the impact of nonlinearities by measuring and incorporating appropriate feedback gains based on the manipulator's state variables. Subsequently, linear control strategies, such as PID or LQR controllers, are employed to regulate the system and minimize tracking errors, ensuring accurate trajectory tracking. Considering robot dynamics model in Eq.(4.1)

Let: $e_q = q_d - q, \quad \dot{e}_q = \dot{q}_d - \dot{q}$

Then: closed loop dynamics will be

$$\ddot{e}_q + K_d \dot{e}_q + K_p e_q = 0 \tag{4.3}$$

where , K_p and K_d are $(n \times n)$ symmetric and positive definite matrices

Let control law:

$$\tau = D(q_d) \underbrace{[\ddot{q} + K_d \dot{e}_q + K_p e_q]}_{a} + n(q_d, \dot{q}_d) \tag{4.4}$$

Control Scheme

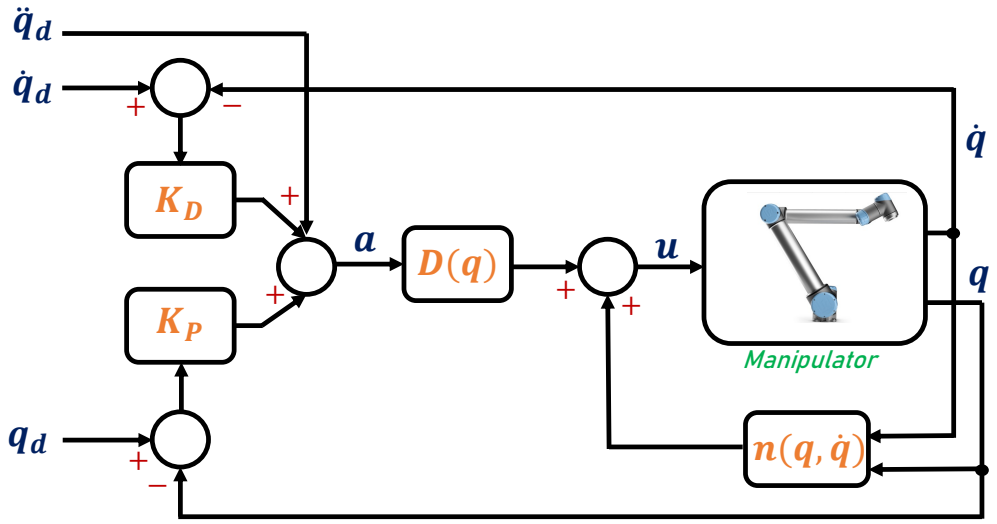


Figure 4.1: Block scheme of feedback linearization control

This approach can't handle the following:

- Inaccurate model parameters .
- Unmodeled dynamics.

4.4 Robust Feedbak Linearization

In the case of imperfect compensation, it is reasonable to assume that a control vector expressed by

$$u = \widehat{D}(q)a + \widehat{n}(q, \dot{q}) \tag{4.5}$$

where $\widehat{D}(q)$ and $\widehat{n}(q, \dot{q})$ represent the computational model adopted in terms of estimates of the terms in the dynamic model. The error on the estimates, i.e., the uncertainty, is represented by

$$\widetilde{D} = \widehat{D} - D(q), \quad \widetilde{n} = \widehat{n} - n \tag{4.6}$$

and is due to imperfect model compensation as well as intentional simplification in inverse dynamics computation. Note that by setting $D(q) = \overline{D}$ (where $\overline{D}$ is the

diagonal matrix of average inertia at the joint axes) and $\widehat{n} = 0$, the decentralized control scheme is recovered, where the control action a can be of the general PID type computed on the error.

Using (4.5) as a nonlinear control law gives

$$D\ddot{q} + n = \widehat{D}a + \widehat{n} \tag{4.7}$$

where the functional dependence has been omitted. Since the inertia matrix D is invertible, it follows that

$$\ddot{q} = a + (D^{-1}\widehat{D} - I)a + D^{-1}\widetilde{n} = a - \eta$$

Where

$$\eta = (I - D^{-1}\widehat{D})a - D^{-1}\widetilde{n} \tag{4.8}$$

Assuming

$$a = \ddot{q}_d + K_D\dot{e}_q + K_P e_q \tag{4.9}$$

we obtain,

$$\ddot{q} + K_D\dot{e}_q + K_P e_q = \eta \tag{4.10}$$

The system described by (4.10) is still nonlinear and coupled, since η is a nonlinear function of q and $\dot{q}$. The first term in (4.8) represents the error in acceleration due to an incorrect model of $D(q)$, while the second term accounts for the error caused by an incorrect model of $n(q, \dot{q})$. The controller alone does not guarantee convergence of the error to zero.

Idea:

In order to compensate for model uncertainty and ensure error convergence to zero while tracking a trajectory in the presence of uncertainties, a linear PD

control is no longer adequate. Therefore, the control law needs to be modified [30]−[32].

we need to modify (4.9) to compensate for uncertainty η. The Lyapunov direct method can be employed to design an outer feedback loop on the error , which should be robust to the uncertainty η.

$$a = \ddot{q}_d + K_D \dot{e_q} + K_P e_q + w \tag{4.11}$$

where w will be designed based on knowledge of uncertainty bounds.

Consider the error state vector :

$$\xi = \begin{bmatrix} e_q \\ \dot{e}_q \end{bmatrix} \tag{4.12}$$

It's time evolution is described by the following state-space representation:

$$\dot{\xi} = H\xi + G(\ddot{e}_q - a + n) \tag{4.13}$$

where H and G are block matrices of dimensions $(2n \times 2n)$ and $(2n \times n)$, respectively:

$$H = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{4.14}$$

In order to analyze stability of (4.13) , we consider the following Lyapunov function :

$$V(\xi) = \xi^T Q \xi > 0 \quad \forall \xi \neq 0 \tag{4.15}$$

where $Q = Q^T > 0$

It's time derivative :

$$\begin{aligned}\dot{V} &= \dot{\xi}^T Q\xi + \xi^T Q\dot{\xi} \\ &= \xi^T(\widetilde{H}^T Q + Q\widetilde{H})\xi + 2\xi^T QG(\eta - w)\end{aligned} \tag{4.16}$$

where,

$$\widetilde{H} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}$$

Since $\widetilde{H}$ has eigenvalues with all negative real parts, it is well-known that for any symmetric positive definite matrix P, the equation

$$\widetilde{H}^T Q + Q\widetilde{H} = -P \tag{4.17}$$

gives a unique solution Q which is symmetric positive definite as well. In view of this, equation (4.16) becomes

$$\dot{V} = \underbrace{-\xi^T P\xi}_{<0} + 2\xi^T QG(\eta - w) \tag{4.18}$$

The asymptotic stability of $\xi = 0$ is guaranteed iff $\dot{V} < 0 \quad \forall \xi \neq 0$. since $-\xi^T P\xi$ is negative definite $\forall \xi \neq 0$, we need to find w s.t $\quad 2\xi^T QG(\eta - w) < 0$. To do so, we need to make some assumptions on the uncertainty bounds of the model.

$$\left\| I - D^{-1}\left(q\right)\widehat{D}\left(q\right)\right\| \leq \alpha \in (0,1) \quad \forall q \tag{4.19}$$

$$\left\|\widetilde{n}\left(q,\dot{q}\right)\right\| \leq \Phi \leq \infty \quad \forall q, \dot{q} \tag{4.20}$$

$$\left\|\ddot{q}_d(t)\right\| \leq \varphi_M \leq \infty \quad \forall \ddot{q}_d \tag{4.21}$$

It is known that matrix D is positive definite with bounded upper and lower norms, the following inequality is valid :

$$0 < D_m \leq \left\|D^{-1}(q)\right\| \leq D_M < \infty \quad \forall q \tag{4.22}$$

Additionally, there always exists a choice for $\widehat{D}$ that satisfies equation 4.19. In fact, the following expression for $\widehat{D}$ can be set:

$$\widehat{D} = \frac{2}{D_m + D_m} I \tag{4.23}$$

this leads to

$$\left\|D^{-1}\widehat{D} - I\right\| \leq \frac{D_M - D_m}{D_M + D_m} = \alpha < 1 \tag{4.24}$$

It can be seen that a more accurate estimation of D will lead to a smaller value of α

Now, we can impose ,

$$z = G^T Q \xi,$$

By using the following control law to compensate for the uncertainty

$$w = \frac{\rho}{\|z\|} z \qquad \rho > 0 \tag{4.25}$$

In this way we have :

$$\begin{aligned} z^T(\eta - w) &= z^T\eta - \frac{\rho}{\|z\|} z^T z \\ &\leq \|z\|\|\eta\| - \rho\|z\| \\ &= \|z\|(\|\eta\| - \rho) \end{aligned} \tag{4.26}$$

Imposing $\rho > \|\eta\|$ in Eq.(4.25) will lead to $\dot{V}$ less than zero along all error system trajectories.

To fulfill Eq.(4.26), it is important to observe that considering the definition of as presented in Eq.(4.8) and the assumptions Eq.(4.19)-(4.20), with the condition that $\|w\| = \rho$, the following relationship can be established:

$$\begin{aligned} \|\eta\| &\leq \left\|I - D^{-1}\widehat{D}\right\| \left\|\ddot{q}_d\right\| + \|K\| \|\xi\| + \|w\| + \left\|D^{-1}\right\| \|\widetilde{n}\| \\ &\leq \alpha\varphi_M + \alpha \|K\| \|\xi\| + \alpha\rho + D_M\Phi \end{aligned} \tag{4.27}$$

By setting

$$\rho > \frac{1}{1-\alpha}(\alpha\varphi_M + \alpha \|K\| \|\xi\| + D_M\Phi) \tag{4.28}$$

where $K = [K_p \quad K_d]$

In the special case where $P = \overline{P}I$, solving for Q in Eq.(4.17), we have

$$\begin{aligned} q_1 &= \frac{\overline{p}(k_p + k_p^2 + k_d)}{2k_p k_d} \\ q_2 = q_3 &= \frac{\overline{p}}{2k_p} \\ q_4 &= \frac{\overline{p}(k_p + 1)}{2k_p k_d} \end{aligned} \tag{4.29}$$

Substituting of Q into Z , we obtain:

$$z = q_2 I e_q + q_4 I \dot{e}_q \tag{4.30}$$

and

$$w = \frac{\rho}{\|z\|} q_2 I e_q + \frac{\rho}{\|z\|} q_4 I \dot{e}_q \tag{4.31}$$

Finally , Substituting of Q into a , we have:

$$a = \ddot{q}_d + \underbrace{\left(k_p + \frac{\rho}{\|z\|} q_2 I\right)}_{\tilde{K}_P} e_q + \underbrace{\left(k_d + \frac{\rho}{\|z\|} q_4 I\right)}_{\tilde{K}_D} \dot{e}_q \tag{4.32}$$

Based on equation 4.32, it can be observed that the inclusion of the supplementary robust term leads to an adjustment of the linear proportional-derivative

(PD) control gains, thereby providing compensation for uncertainties. As a consequence, the feedback linearization methodology demonstrates a capacity to effectively manage uncertainty, especially when high gain matrices are employed.

Note:

This control law will drive the error to the sliding plane $z = 0$. At this point, the control is switched at infinite frequency (chattering) , which is not feasible in practice. In order to mitigate the presence of high-frequency components (chattering), it is possible to employ a robust control strategy that, while not guaranteeing error convergence to zero, guarantees the boundedness of the error with respect to its norm. Such a control law can be expressed as follows:

$$w = \begin{cases} \rho \dfrac{z}{\|z\|} & \text{if} \quad \|z\| \geq \varepsilon \\ \\ \rho \dfrac{z}{\varepsilon} & \text{if} \quad \|z\| < \varepsilon \end{cases} \tag{4.33}$$

This control law effectively regulates the control input based on the norm of the error signal, where represents a scaling factor, z denotes the error signal, and ϵ is a predetermined threshold value.

Control Scheme

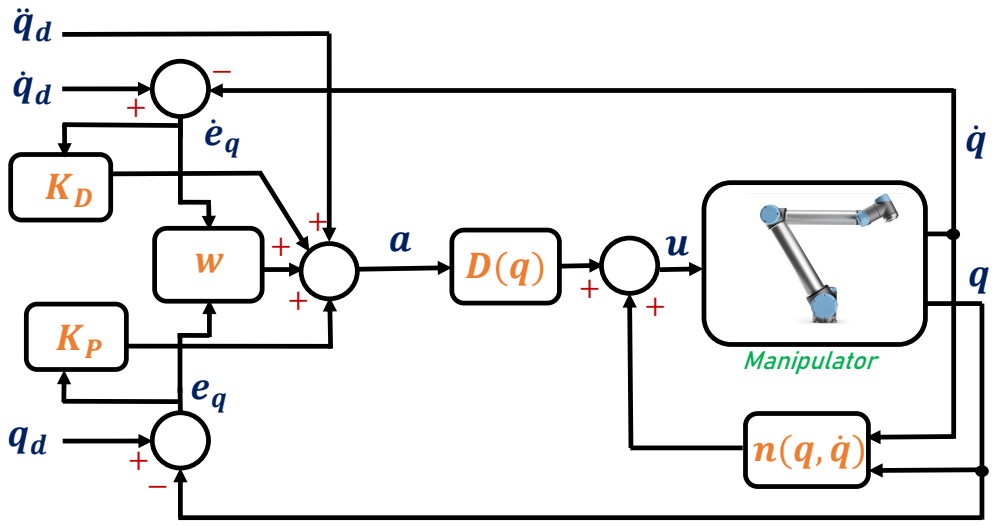


Figure 4.2: Block scheme of robust feedback linearization control

4.4.1 Adaptive ρ

One of the key contributions of this thesis is the proposed method aimed at addressing uncertain certainty bounds associated with robot dynamic parameters. This approach involves utilizing an update law to adjust the parameter ρ based on the value of $\dot{V}$ as defined in equation (4.18). In the subsequent simulation chapter, it will be demonstrated that this method exhibits the capability to handle varying degrees of uncertainty and achieve a response comparable to that of robust feedback linearization.

Our research approach centers on Adaptive Robust Feedback Linearization(ARFBL), a technique grounded in Lyapunov analysis. This method hinges on the fundamental principle that to achieve stability and convergence within the system, the derivative of the Lyapunov function in Equation (4.16) should exhibit negativity.

Instead of using constant ρ , we will use the following update law to make it adaptive.

Our initial strategy involves devising an update law to augment the parameter ρ . This adjustment aims to counteract uncertainties when the rate of change of $\dot{V}$ is positive. Nevertheless, empirical experimentation revealed a limitation in this approach it tends to overestimate ρ actual value due to its singular upward adjustment.

$$\dot{\rho} = \begin{cases} k_\rho & \text{if} \quad \dot{V} > 0 \\ 0 & \text{if} \quad \dot{V} < 0 \end{cases} \tag{4.34}$$

where $k_\rho > 0$

To address this limitation, we introduce a modified update law in Eq. (4.35). Once again, our methodology capitalizes on Lyapunov analysis. This modified approach affords the adaptive scheme greater flexibility in modifying rho. The degree of freedom in ρ adjustment is extended to both incrementation and decrementation. The basis for this modification lies in the $\dot{V}_{id}$, which serves as a reference point. This $\dot{V}_{id}$ is derived from the initial term in Equation (4.18), a

scenario where uncertainties are absent. Notably, in this context, we assume the identity matrix P.

$$\dot{\rho} = \begin{cases} k_\rho & \text{if} \quad \dot{V} > \dot{V}_{id}/2 \\ 0 & \text{if} \quad \dot{V}_{id} < \dot{V} < \dot{V}_{id}/2 \\ -k_a & \text{if} \quad \dot{V} < \dot{V}_{id} \end{cases} \tag{4.35}$$

where,

$$k_a > 0 \qquad and \qquad \dot{V}_{id} = \underbrace{-\xi^T P \xi}_{<0}$$

4.5 Sliding Mode Control

4.5.1 Introduction

In the realm of robust control for robot manipulators, sliding mode control (SMC) has emerged as a prominent technique. SMC offers a powerful framework to address uncertainties, disturbances, and nonlinearities present in robotic systems [33]–[37]. Its main concept revolves around the notion of a sliding surface, which plays a crucial role in achieving robust control performance.

The essence of sliding mode control lies in the ability to guide the system state onto a predefined sliding surface, where subsequent motion remains confined. By doing so, the control system operates in a switching mode, rapidly switching between different control laws or dynamics, leading to a robust response. This switching mechanism serves as a distinctive feature of sliding mode control.

Sliding mode motion encompasses two distinct phases: the reaching stage and the sliding stage. In the reaching stage, the system state undergoes a finite-time convergence towards the sliding surface from any initial state, adhering to predefined reaching conditions until the sliding surface is attained. The subsequent sliding stage involves the system state maintaining a sliding mode trajectory along the sliding surface due to the influence of the applied control law.

The central challenge in the development of a sliding mode controller resides

in guaranteeing perpetual convergence of the system state to the sliding mode surface, while simultaneously constraining its movement to the sliding hyperplane. This task also involves determining the existence and compliance with the accessibility condition of the sliding hyperplane, as elaborated in reference [31]. Distinct sliding mode controllers can be formulated using a reaching law to satisfy the dynamic performance requirements of the control system and to ensure the overall stability of the control system.

4.5.2 Design of Sliding Surface

The sliding surface can be categorically delineated into two classifications: one pertains to the linear sliding surface, while the other pertains to the nonlinear sliding surface. A prominent illustration of a nonlinear sliding surface is the terminal sliding surface. Specifically, when the system state is distantly removed from the equilibrium point, the rate of convergence of the linear sliding surface exceeds that of the terminal sliding surface. Conversely, when the system state is proximate to the equilibrium point, the convergence rate of the terminal sliding surface surpasses that of the linear sliding surface. An instance of a linear sliding mode surface is formulated as:

$$s = \lambda e_q + \dot{e}_q \qquad \lambda \in \mathbb{R}^{n \times n} \tag{4.36}$$

An exemplar of a nonlinear sliding surface was introduced in reference [38]:

$$s = \dot{e}_q + \mu_1 {e_q}^{m/p} + \mu_2 {e_q}^{k/l} \tag{4.37}$$

For the parameters in the expressions:

$$\mu_1 > 0, \quad \mu_2 > 0, \quad m > p > 0,$$

where m and p are positive odd numbers.

Similarly, the parameters are constrained as:

$$l > k > 0,$$

where l and k are positive odd numbers.

4.5.3 Design of Reaching Law

To enhance the dynamic performance of sliding mode control during the reaching phase, Academician Gao Weibing introduced the concept of a reaching law and devised several representative reaching laws [39]. By appropriately tuning the parameters within the reaching law formulation, the dynamic performance of sliding mode control can be effectively enhanced. Among the reaching laws commonly employed in sliding mode control are the constant reaching law, power reaching law, exponential reaching law [40],[41], and generalized reaching law. The constant reaching law inadequately attenuates chattering phenomena. While the power reaching law mitigates chattering, the speed of attaining the sliding surface is suboptimal, potentially resulting in inadequate tracking performance.

A prototypical expression for the exponential reaching law is as follows:

$$\dot{s} = -K_1 s - K_2 \text{sign}(s) \tag{4.38}$$

Here, $K_1 = [k_{11}, k_{12}, \ldots, k_{1n}]$ represents the parameter associated with the exponential reaching component, where $k_{1i} > 0$ for $i = 1, 2, \ldots, n$. Similarly, $K_2 = [k_{21}, k_{22}, \ldots, k_{2n}]$ denotes the parameter linked to the constant reaching component, with $k_{2i} > 0$ for $i = 1, 2, \ldots, n$. The term $\dot{s} = -K_1 s$ represents the exponential reaching component. This component ensures that when the system state closely approaches the sliding surface s, the system state rapidly reaches the sliding surface. Naturally, as the system state advances towards the sliding surface, its reaching speed gradually diminishes to zero. This not only minimizes the time taken to reach the sliding surface but also significantly reduces the speed of the system state's approach.

The process of the system state reaching the sliding surface transpires as a gradual evolution governed solely by the exponential reaching component. However, this sole reliance on the exponential reaching component does not guarantee a finite-time attainment of the sliding surface. Consequently, to ensure a finite-time arrival at the sliding surface, an additional constant reaching component $-K_2 \text{sign}(s)$ is incorporated based on the foundation of the exponential reaching term. The incorporation of this component presents the advantage of enabling a reaching speed K_2 when the system state approximates the slid-

ing surface. This modification guarantees the system's finite-time arrival at the sliding surface .

The steps involved in sliding mode control can be summarized as follows:

1. Define an intermediate variable, denoted as s.
   - $\dot{s}$ must contains the control signal u
   - $(s \to 0) \implies (\mathbf{e_q}(t) \to 0)$
2. Determine the rate of convergence for the sliding mode control based on the Sliding Condition.
3. Establish the control law for the sliding mode control.

The resultant control law, utilizing the linear sliding surface as expressed in Eq. (1) and employing the constant reaching law (SMC-CRL), is given by:

$$\tau = \widehat{n}(q, \dot{q}) + \widehat{D}(q)\ddot{q} + \widehat{D}(q)\lambda\dot{e}_q + \rho\text{sign}(s) \qquad , \rho > 0 \tag{4.39}$$

This control law is well-suited for straightforward manipulators such as the RR planar configuration. However, for more intricate robots like the 6-DOF UR10 manipulator, modifications to the reaching law are essential to achieve satisfactory performance. In the subsequent discussion, we introduce a control law employing the linear sliding surface from Eq.(1) and incorporating the exponential reaching law (SMC-ERL):

$$\tau = \widehat{D}(q)\left(\ddot{q}_d + K_1 s + K_2\text{sign}(s) + K_3\dot{e}_q\right) + \widehat{n}(q, \dot{q}) \tag{4.40}$$

Furthermore, an adapted version of the prior control law can be defined, employing a nonlinear sliding surface as described in Eq. (3) and integrating the exponential reaching law (SMC-MERL):

$$\tau = D(q)\left(\ddot{q}_d + a\dot{e}_q + K_1 s + K_2\text{sign}(s)\right) + \widehat{n}(q, \dot{q}) \tag{4.41}$$

where ,

$$a = \mu_1(m/p)e^{(m/p)-1} + \mu_2(k/l)e^{(k/l)-1}$$

4.5.4 Chattering Phenomenon

The aforementioned controllers will encounter a well-known issue in Sliding Mode Control which is the chattering phenomenon, which occurs due to the discontinuous nature of the control input. Chattering refers to the high-frequency oscillations of the control signal, which can lead to undesirable wear and tear of actuators and produce undesirable noise in the system.

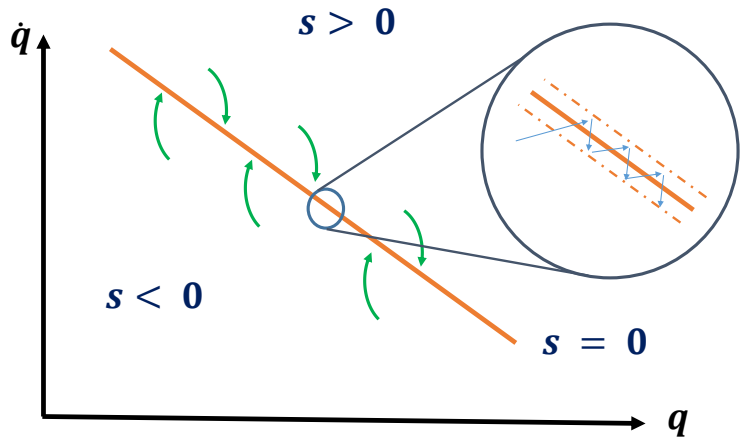


Figure 4.3: Control chattering problem

4.5.5 Sliding Mode Control with Boundary Layer

To mitigate the chattering problem, a boundary layer can be introduced around the sliding surface [42]. The boundary layer is a small region where the control input is smoothed to reduce the chattering effect while preserving the robustness of the SMC.

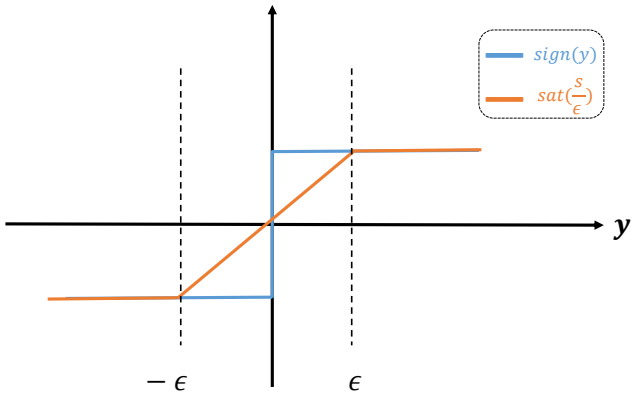


Figure 4.4: smooth sat function Vs non-smooth $sign$ function

This can be achieved by replacing **sign** function in control laws in Eq.(4.39-4.41) with **sat** function .

where the saturation function is defined as:

$$\text{sat}(s) = \begin{cases} \dfrac{s}{\epsilon} & \text{if } |s| < \epsilon \\ \text{sign}(s) & \text{if } |s| > \epsilon \end{cases} \tag{4.42}$$

The saturation function, denoted as sat(s), plays a role in limiting the control input when the sliding surface s exceeds a certain threshold determined by ϵ. When $|s|$ is less than ϵ, the control input is scaled by $\dfrac{1}{\epsilon}$ to ensure its boundedness. On the other hand, if $|s|$ exceeds ϵ, the sign of s is preserved to maintain the control input's direction.

Note:

The update law in section (4.4.1) can be used to update ρ based on the derivative of Lyapunov function.

Control Scheme

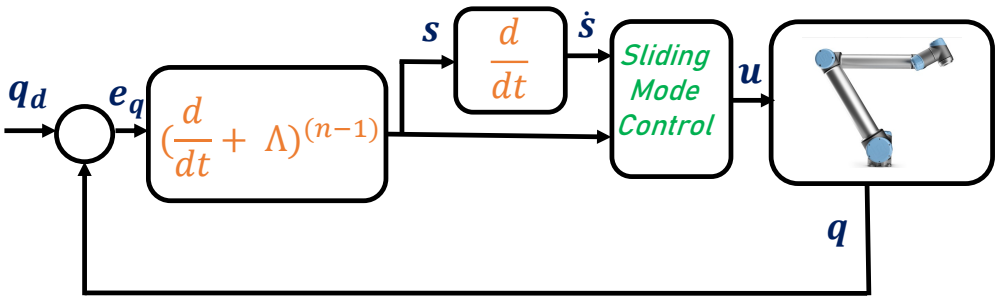


Figure 4.5: Sliding Mode Control Scheme

5

Implementation,Simulation and Analysis

5.1 RR Manipulator

This section focuses on investigating the behavior and conducting an analysis of three case studies, namely the RR manipulator, SCARA and UR10, in relation to the controllers discussed in Chapter 4. The objective is to perform a comparative analysis that highlights the strengths and weaknesses of each controller method. Through this analysis, we aim to gain insights into the performance and effectiveness of these controllers in practical scenarios.

5.1.1 Kinematic and Dynamic Parameters

Link	θ	d	a	α
1	${\theta_1}^*$	0	0.3	0
2	${\theta_2}^*$	0	0.15	0

Table 5.1: DH-table for 2-link RR planar manipulator

Inertia Matrix

$$I = \begin{bmatrix} 0.176 & 0 \\ 0 & 0.0411 \end{bmatrix}$$

Link	m_i	l_{c_i}
1	7.848	.1554
2	4.49	0.0411

Table 5.2: Dynamic parameters for 2-link RR planar manipulator

5.1.2 Setup

The desired input trajectory is defined as follows:

$$q_{d1} = \frac{\pi}{4}\cos(t + \frac{\pi}{2}), \quad q_{d2} = \frac{\pi}{6}\cos(2t + \frac{\pi}{4})$$

To simulate parameter uncertainty, random constant values were added to the original dynamical parameters. This allows us to estimate the lower and upper bounds of the parameters.

5.1.3 Feedback Linearization Control Simulation

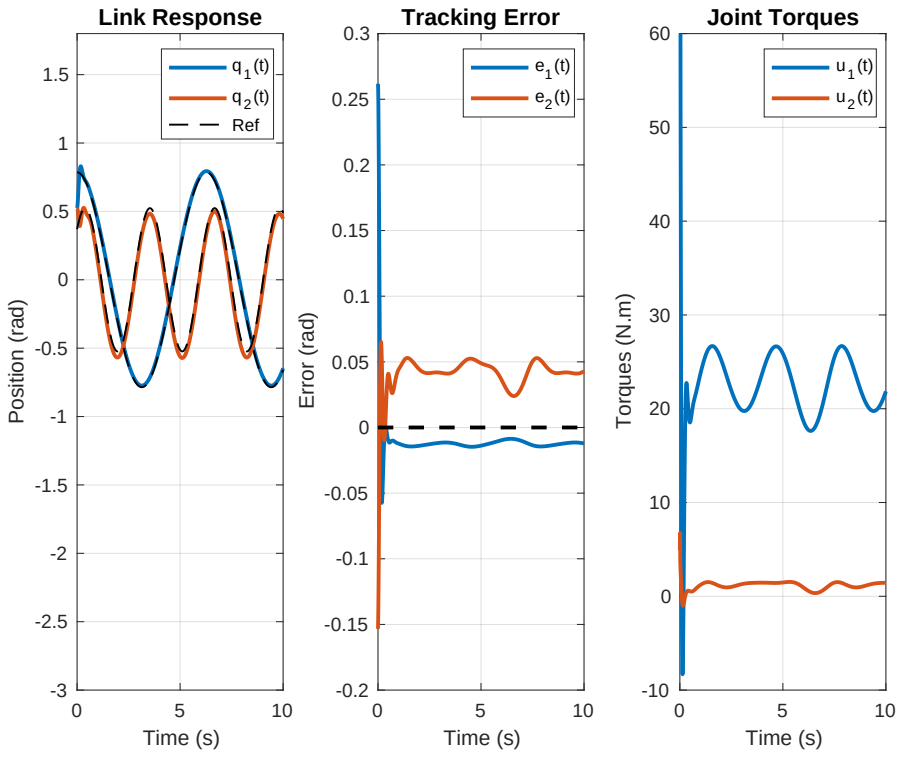


Figure 5.1: Feedback linearization control

In Figure 5.1, feedback linearization was employed to control the robot's behavior. The control gains used were

$$K_p = \begin{bmatrix} 120 & 0 \\ 0 & 120 \end{bmatrix}$$

and

$$K_d = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$$

Increasing the gains demonstrates that feedback linearization can effectively handle uncertainty. A detailed analysis of robust feedback linearization will be discussed to verify this claim. The tracking performance exhibits bounded tracking error, and the joint torques exhibit smooth behavior.

5.1.4 Robust Feedback Linearization Control Simulation

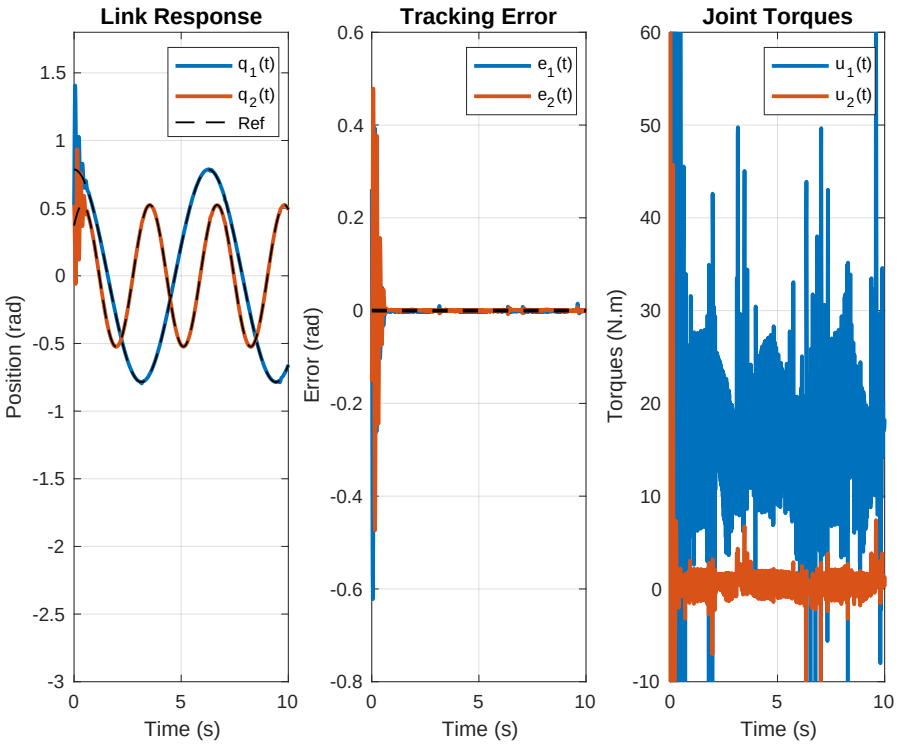


Figure 5.2: Robust feedback linearization control (chattering)

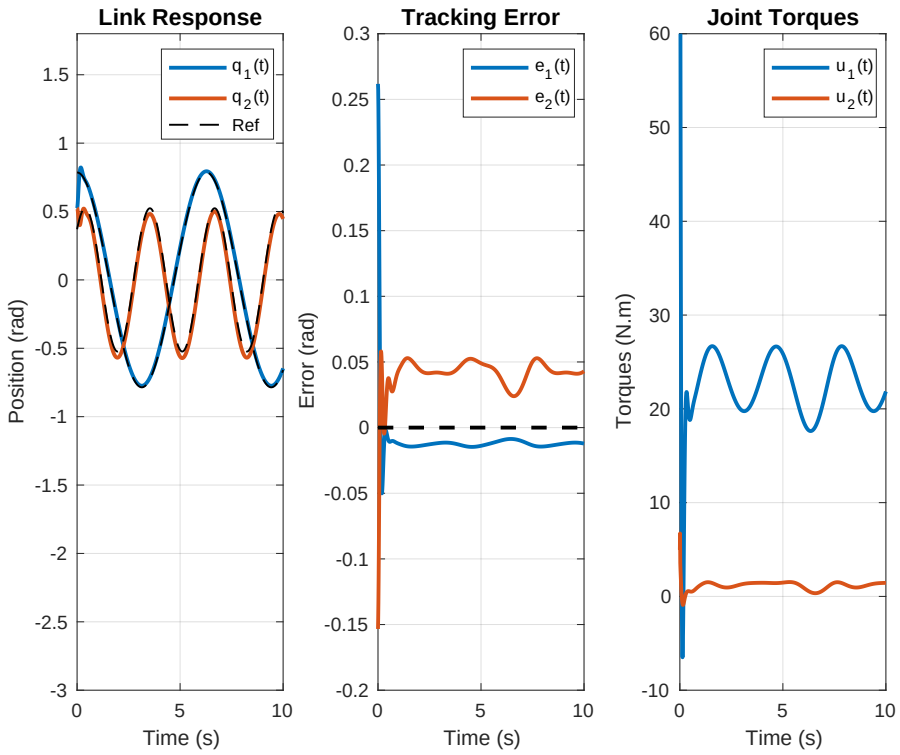


Figure 5.3: Robust feedback linearization control (without chattering)

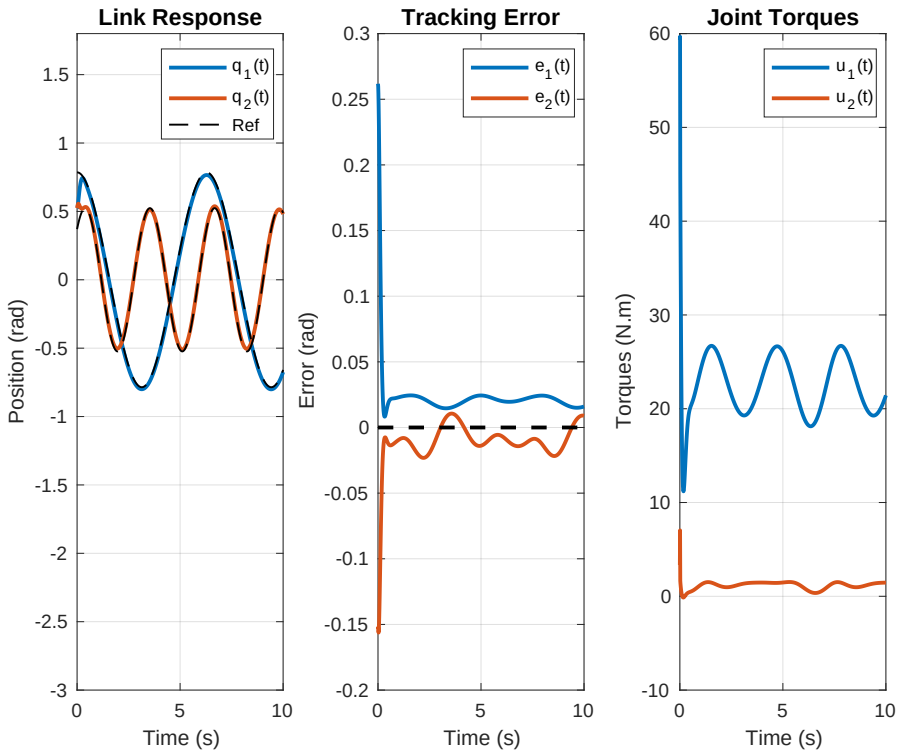


Figure 5.4: Robust feedback linearization control (adaptive ρ)

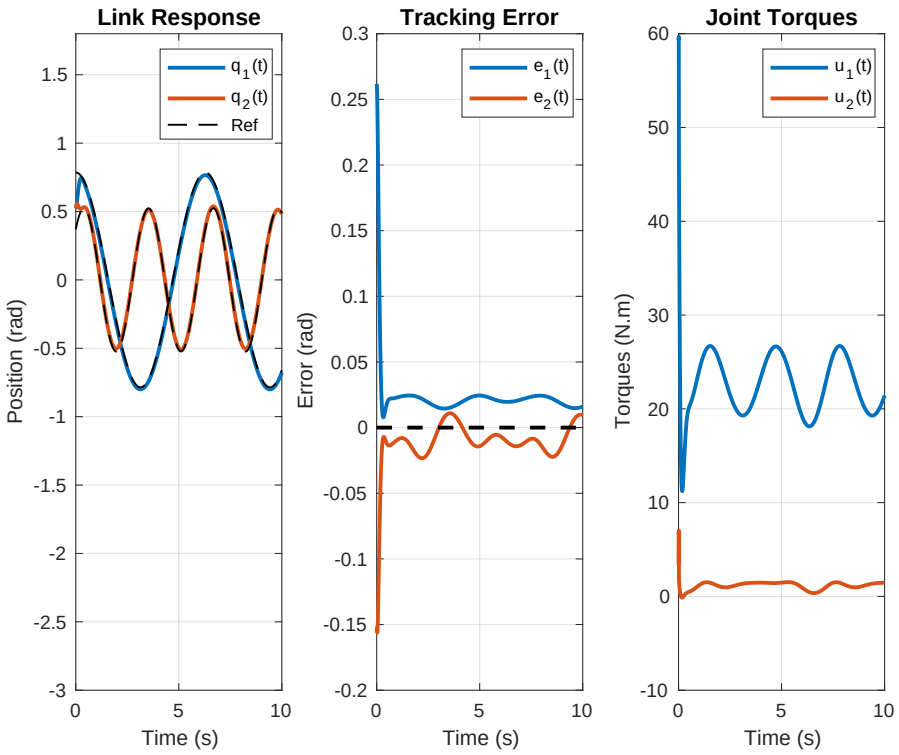
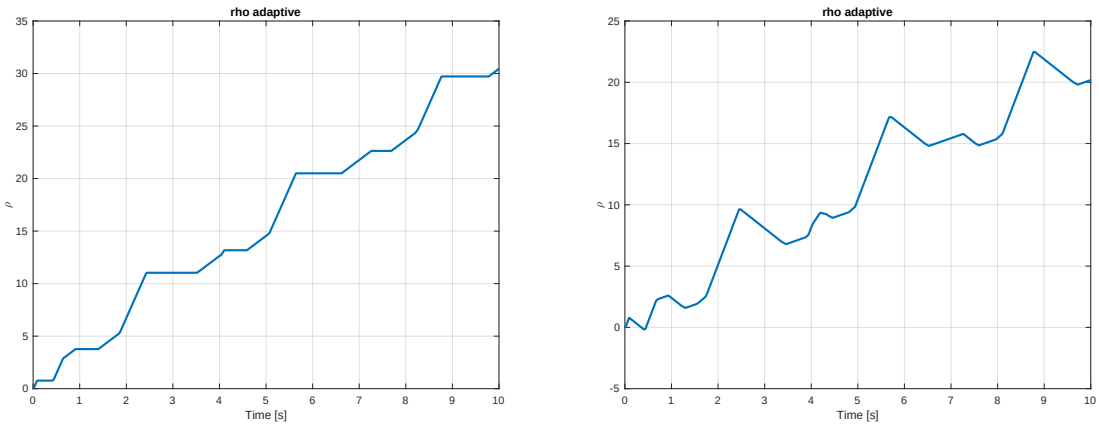


Figure 5.5: Robust feedback linearization control (modified adaptive ρ)



(a) adaptive ρ evolution (b) modified adaptive ρ evolution

Figure 5.6: ρ evolution based on two diffrent update laws

In the following, we analyze the response of the robust feedback linearization technique as implemented in Section 4.4 of our study.

Figure 5.2 illustrates two key observations. Firstly, the utilization of the sign function introduces chattering behavior, resulting in high-frequency switching

dynamics. This behavior is impractical in real-world applications. Secondly, we note that the error converges to zero for both joints.

On the contrary, Figures (5.3) and (5.4) demonstrate smoother joint torque inputs, which are more suitable for actuator operation. However, this comes at the expense of allowing a bounded tracking error. The magnitude of this boundary layer of error is governed by the parameter ϵ in Eq.(4.33). It is worth mentioning that we need a priori knowledge of the bounds of the dynamic parameters.

The recently introduced adaptive robust feedback linearization (ARFBL) methodology depicted in Figures (5.4) and (5.5) demonstrates a notable advancement, as it operates independently of a priori knowledge concerning the associated bounds. This technique successfully attains commendable performance outcomes.

Upon careful examination, as evidenced in Figure (5.6), a comparative analysis is conducted concerning the evolution of parameter ρ for two distinct approaches: adaptive feedback linearization (AFBL) and modified adaptive feedback linearization (MARFBL). It is noteworthy that MARFBL exhibits the capability to regulate parameter ρ effectively, thereby maintaining its value within a proximity to the one derived from a comprehensive understanding of dynamic bounds. In contrast, the ARFBL technique exhibits an escalating trend in parameter ρ over time.

5.1.5 Sliding Mode Control Simulation

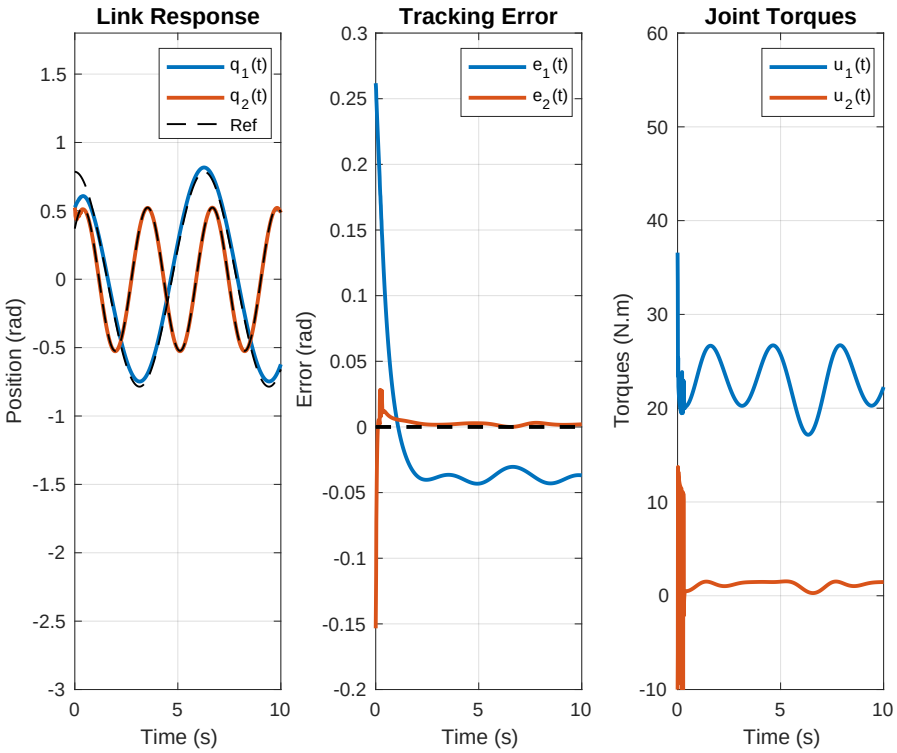


Figure 5.7: Sliding mode Control ($\epsilon = 8$)

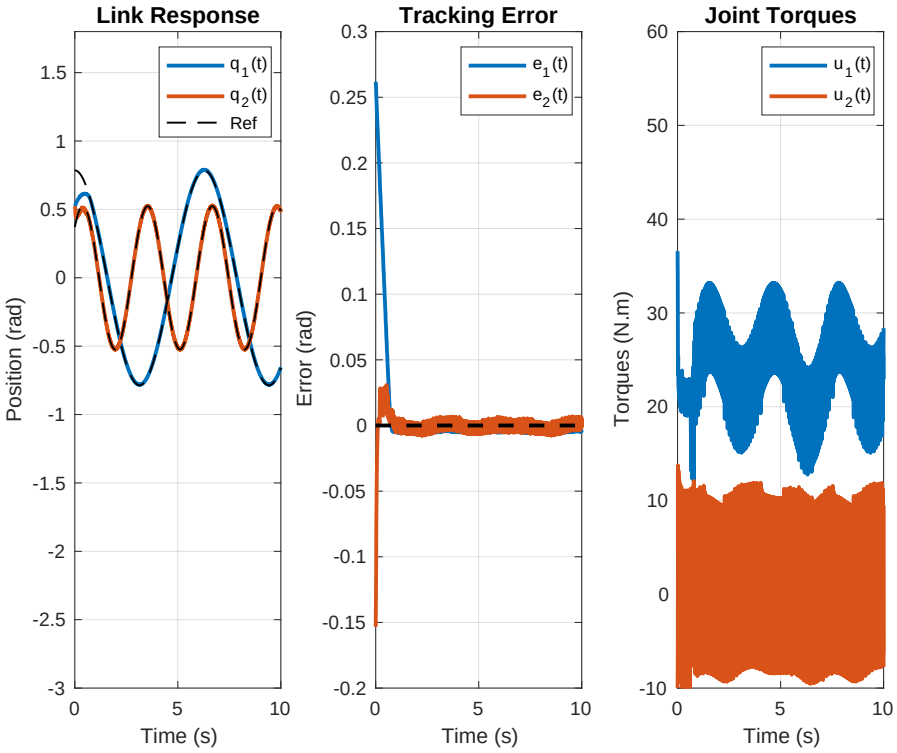


Figure 5.8: Sliding mode Control ($\epsilon = 1$)

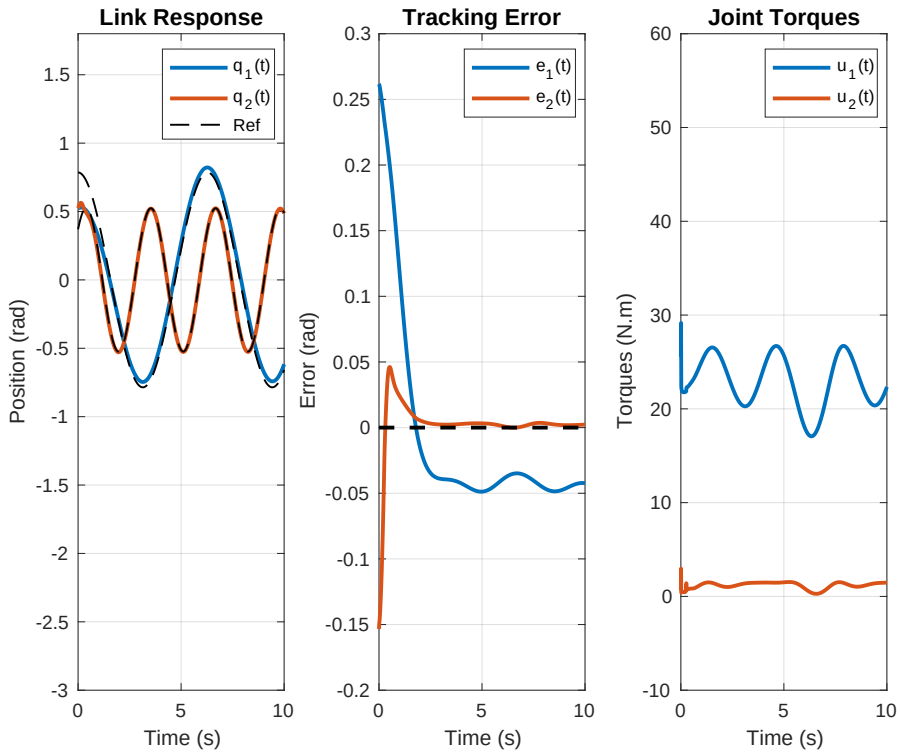


Figure 5.9: Sliding mode Control (adaptive ρ)

In this study, we proceed to investigate the sliding mode controller. Specifically, we compare the responses presented in Figure (5.7) and Figure (5.8), where we set $\epsilon = 8$ and $\epsilon = 1$, respectively. It is noteworthy that a decrease in ϵ leads to the reemergence of the chattering problem.
To address this issue, we introduce an update law in Figure 5.9, which aims to increase the value of ρ only when necessary. This approach offers the advantage of maintaining chattering-free behavior even when we decrease ϵ.

The performance of various controllers is assessed based on root mean square error (RMSE) values, as shown in Table 5.3 and Fig(5.10) The table presents RMSE values for each controller across three distinct scenarios of external disturbance: No Error, Constant Error, and Time-varying Error.

In the case of "No Error," the controllers' behavior is compared. Among the controllers, the Adaptive Robust Feedback Linearization (ARFBL) and Modified Adaptive Feedback Linearization (MARFBL) exhibit the lowest RMSE values, both at approximately 0.0494 and 0.0495, respectively. These two controllers demonstrate superior performance in accurately tracking the desired trajectory despite the absence of errors.

Transitioning to the scenario of "Constant Error," a similar trend emerges. Once again, ARFBL and MARFBL controllers demonstrate their robustness by achieving the lowest RMSE values of around 0.1281 and 0.1282, respectively. This signifies their ability to handle and mitigate the influence of a constant external error on the system.

In the context of "Time-varying Error," ARFBL and MARFBL controllers remain the most effective choices, maintaining their performance consistency with RMSE values of approximately 0.0606 and 0.0607, respectively. These controllers showcase their adeptness at adapting to varying disturbances over time.

On the other hand, the Sliding Mode Control (SMC) and Adaptive Sliding Mode Control (ASMC) controllers exhibit comparatively higher RMSE values as well as chattering across all disturbance scenarios . Notably, ASMC displays the highest RMSE value of 0.1722 in the case of "Constant Error," signifying its relative susceptibility to consistent disturbances.

In conclusion, the Adaptive Robust Feedback Linearization (ARFBL) and Modified Adaptive Feedback Linearization (MARFBL) controllers consistently emerge as the best performers across all examined disturbance scenarios, effectively minimizing RMSE values and thereby demonstrating superior control robustness and accuracy. In contrast, the Adaptive Sliding Mode Control (ASMC) controller generally presents less favorable results.

	FBL	**RFBL**	**ARFBL**	**MARFBL**	**SMC**	**ASMC**
No Error	0.0775	0.0499	0.0494	0.0495	0.0591	0.1171
Constant Error	0.2078	0.1274	0.1281	0.1282	0.3695	0.1722
Time-varying Error	0.0968	0.0602	0.0606	0.0607	0.0831	0.1119

Table 5.3: RMSE of different controllers under different external disturbances

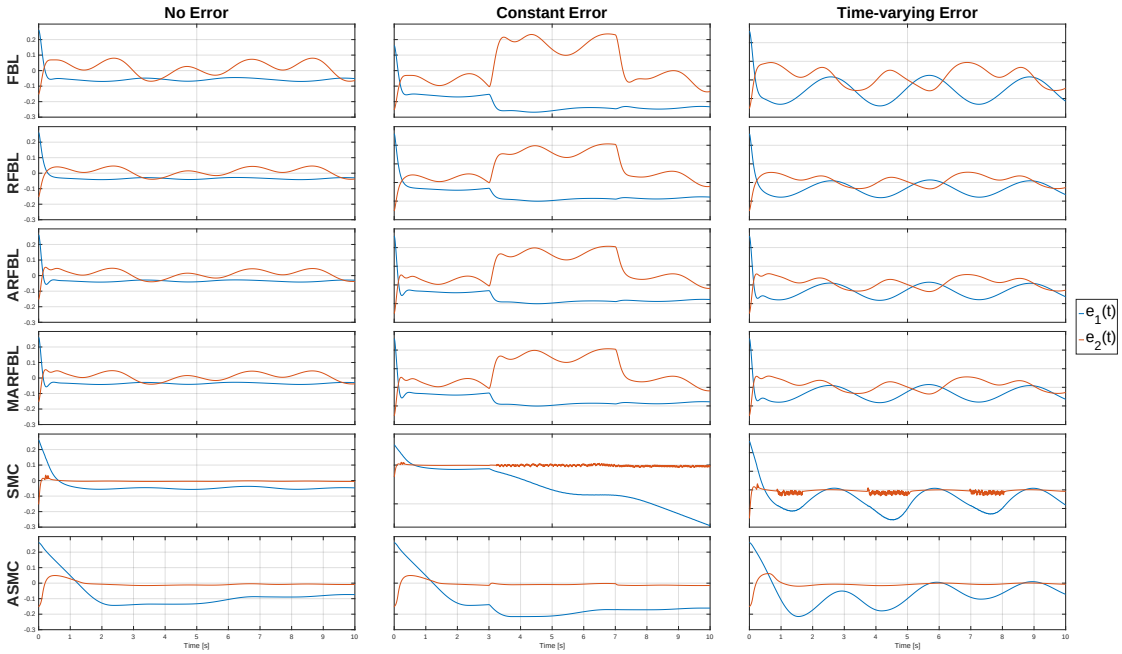


Figure 5.10: RMSE of different controllers under different levels of external disturbances)

5.2 SCARA Manipulator

5.2.1 Kinematic and Dynamic Parameters

Link	θ	d	a	α
1	${\theta_1}^*$	0.5	0.5	0
2	${\theta_2}^*$	−0.005	0.4	0
3	0	${d_3}^* - 0.15$	0	0
4	${\theta_4}^*$	−0.005	0.2	0

Table 5.4: DH-table for SCARA manipulator

In the table 5.5, we reported the parameters defining the dynamics, where p_x, p_y, and p_z are the three components of p_c, expressed w.r.t. the correspondent reference frame of the Denavit-Hartenberg convention. Each link is assumed to be a cylinder, with radius r_i and length l_{c_i}. For simplicity, we will neglect the contributions due to motors .

Link	$m_i[kg]$	$p_x[m]$	$p_y[m]$	$p_z[m]$	$r_i[m]$	$l_i[m]$
0	25	0.0	0.0	0.25	0.1	0.5
1	20	−0.25	0.0	0.0	0.1	0.5
2	15	−0.2	0.0	0.0	0.1	0.4
3	10	0.0	0.0	0.15	0.1	0.3
4	5	−0.1	0.0	0.0	0.1	0.2

Table 5.5: Dynamic parameters for SCARA manipulator

Inertia Matrix

$$I_i = \begin{bmatrix} \frac{m(3{r_i}^2 + {l_i}^2)}{12} & 0 & 0 \\ 0 & \frac{m(3{r_i}^2 + {l_i}^2)}{12} & 0 \\ 0 & 0 & \frac{m{r_i}^2}{2} \end{bmatrix}$$

5.2.2 Setup

The desired input trajectory is defined as follows:

$$q_{d_1} = \frac{\pi}{4}\cos(t+\frac{\pi}{2}), \quad q_{d_2} = \frac{\pi}{6}\cos(2t+\frac{\pi}{4}), \quad q_{d_3} = \frac{\pi}{4}\cos(t+\frac{\pi}{2}), \quad q_{d_4} = \frac{\pi}{6}\cos(2t+\frac{\pi}{4})$$

5.2.3 Feedback Linearization Simulation

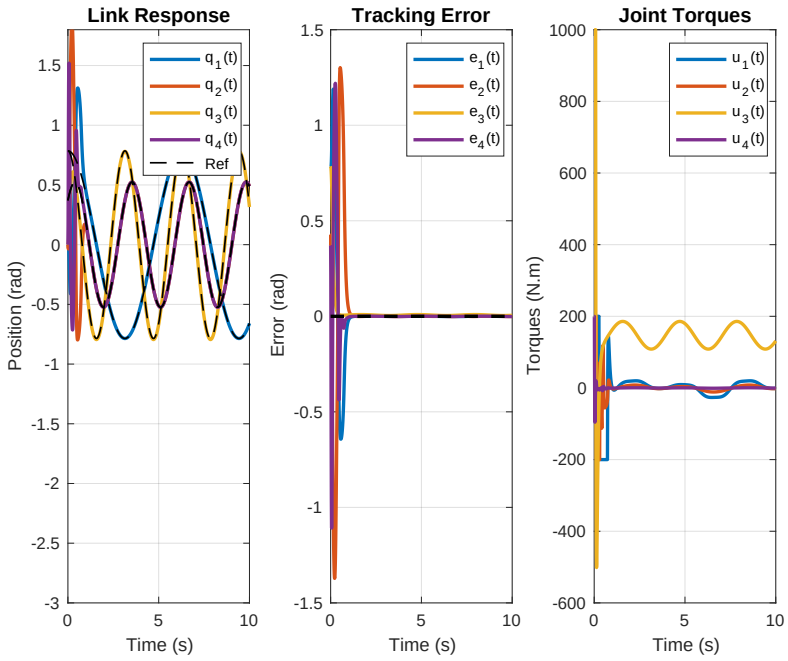


Figure 5.11: Feedback linearization control

In Figure 5.11, feedback linearization was employed to control the robot's behavior. The control gains used were

$$K_p = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1000 \end{bmatrix}$$

and

$$K_d = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$$

As expected, by increasing the gains, feedback linearization can effectively handle uncertainty while maintaining smooth input torque.

5.2.4 Robust Feedback Linearization Simulation

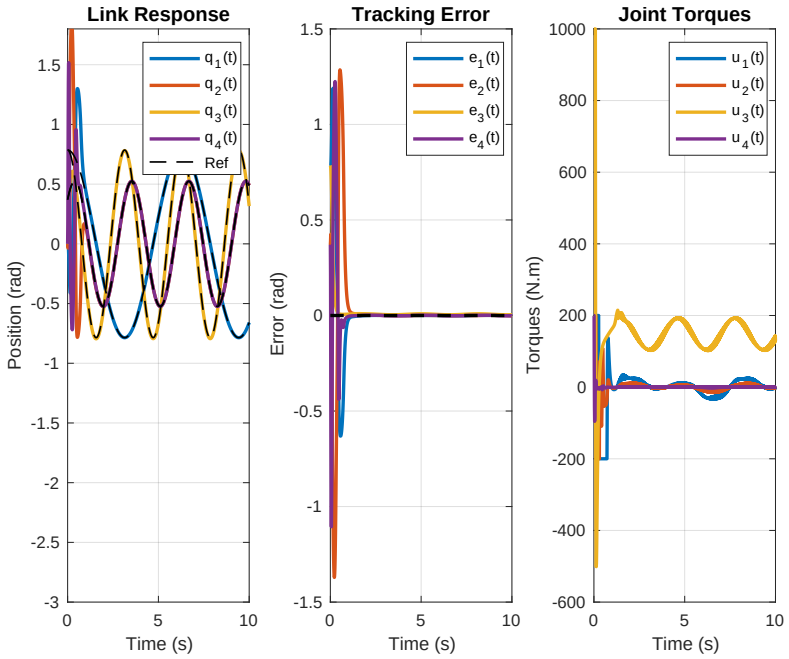


Figure 5.12: Robust Feedback linearization control(chattering)

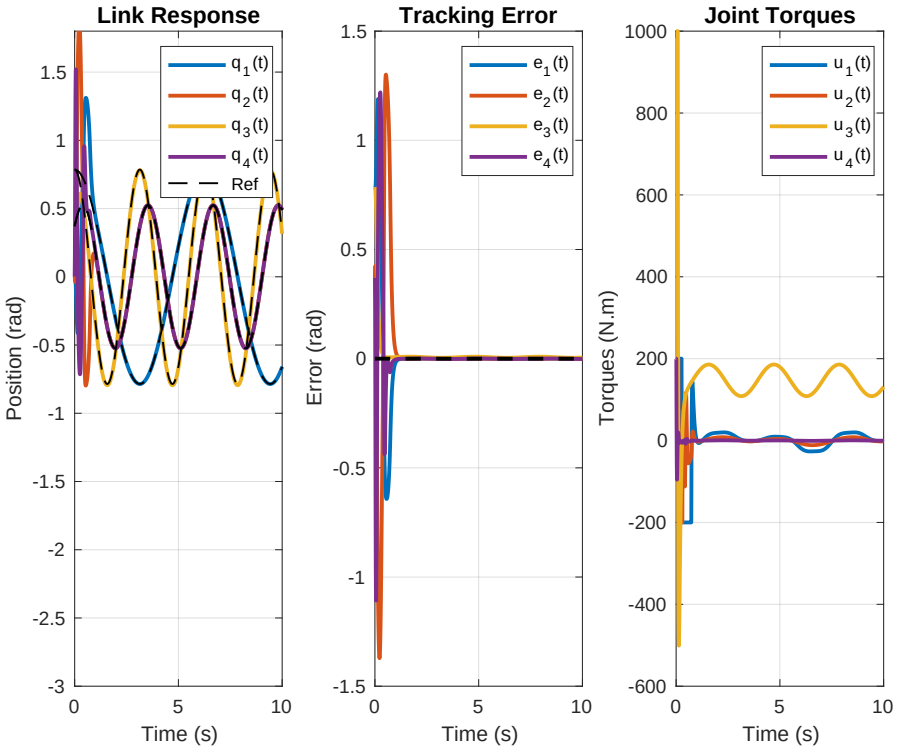


Figure 5.13: Robust Feedback linearization control(without chattering)

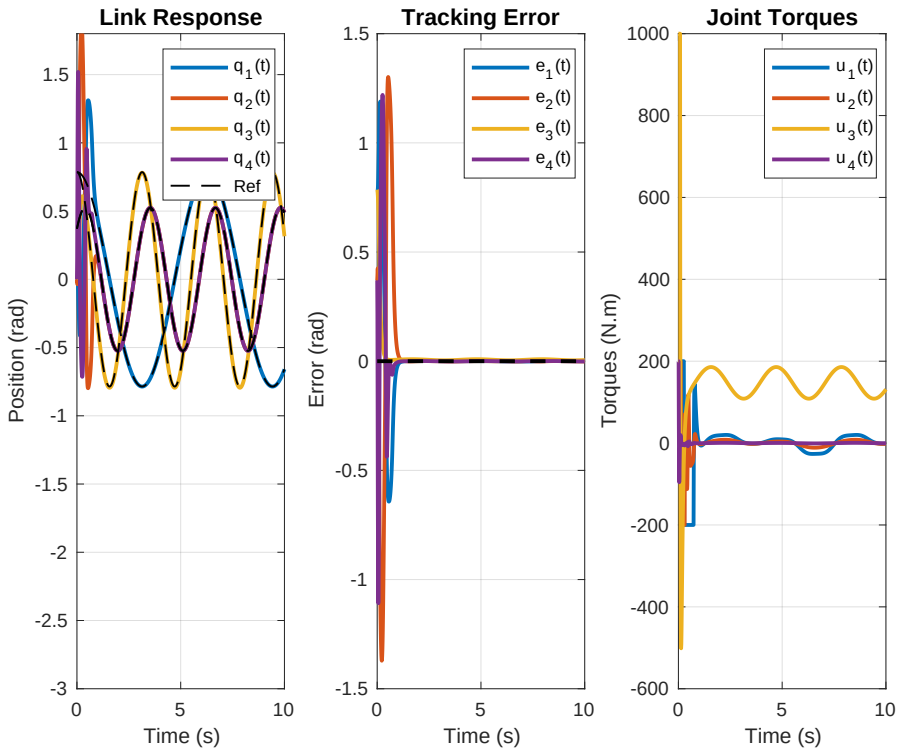


Figure 5.14: Robust Feedback linearization control(adaptive ρ)

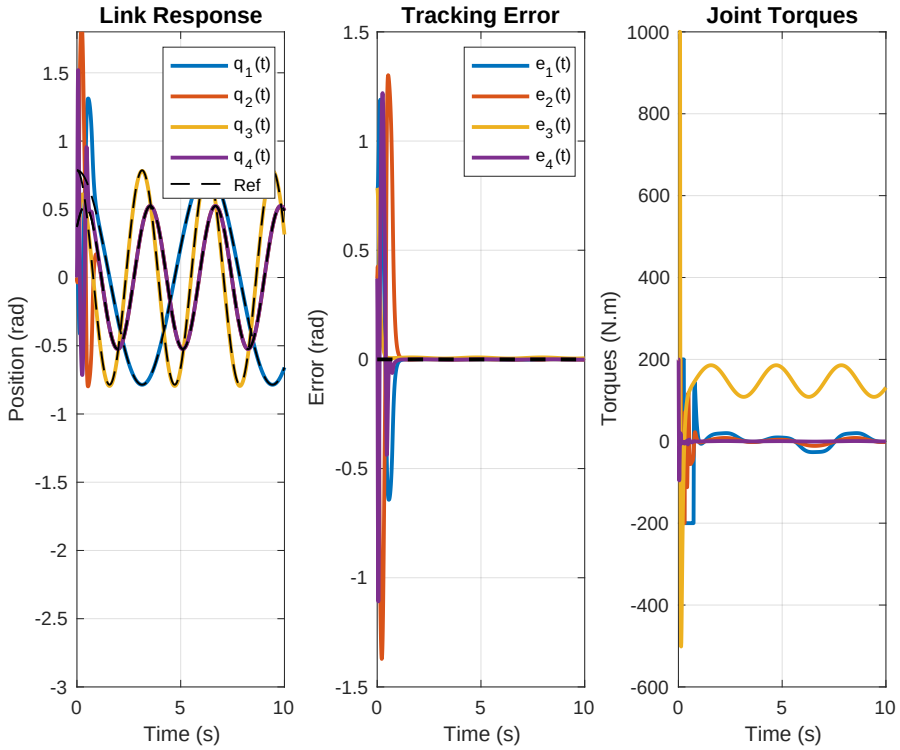
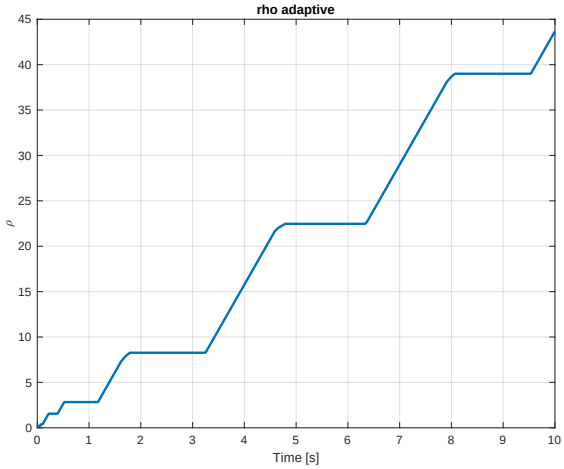
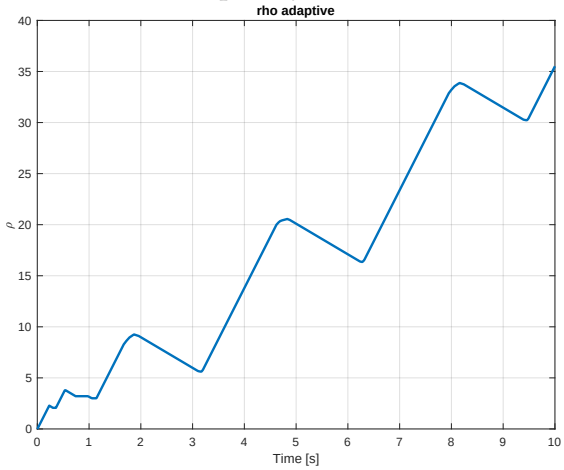


Figure 5.15: Robust Feedback linearization control(modified adaptive ρ)



(a) adaptive ρ evolution



(b) modified adaptive ρ evolution

Figure 5.16: ρ evolution based on derivative of $\dot{V}$

Similar to the analysis in Section 5.1.4, we investigated the response of the robust feedback linearization method, as implemented in Section 4.4, on a SCARA robot to understand the differences in controlling the prismatic joint.

As expected, in Figures (5.12) - (5.15), the system converges to the desired trajectory. However, the chattering problem arises in this case as well, which we addressed by employing a smooth saturation function instead of a sign function. It is noteworthy how effective the new approach based on adaptive ρ is in handling uncertainty without requiring prior knowledge of its bounds.

Figure 5.16 compares the evolution of ρ in using the proposed update law in section 4.4.1.

5.2.5 Sliding Mode Control Simulation

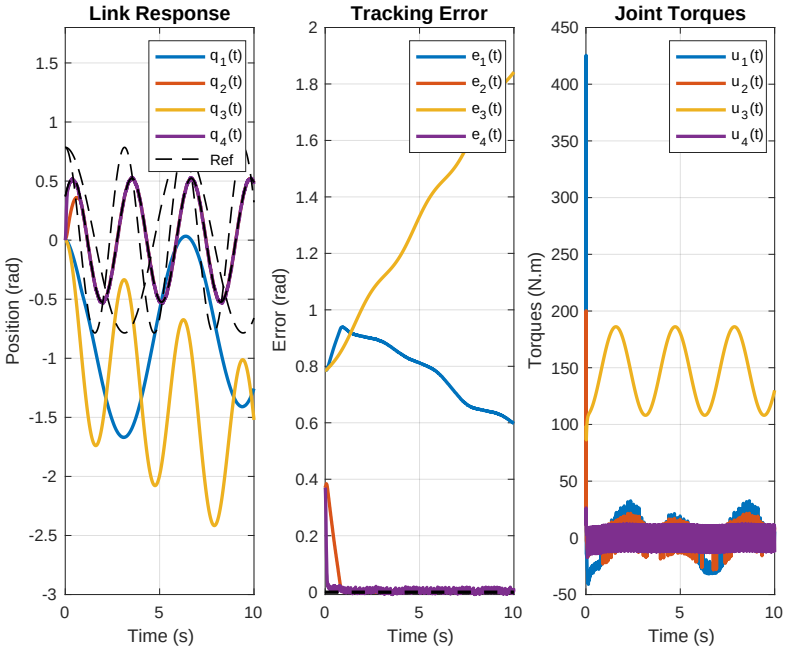


Figure 5.17: Sliding mode control(scalar ρ)

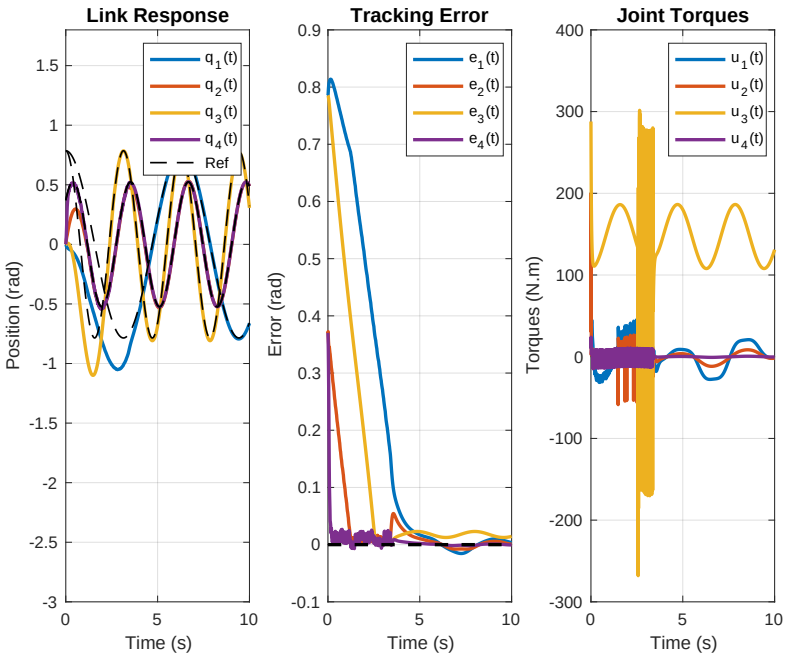


Figure 5.18: Sliding mode control(matrix ρ)

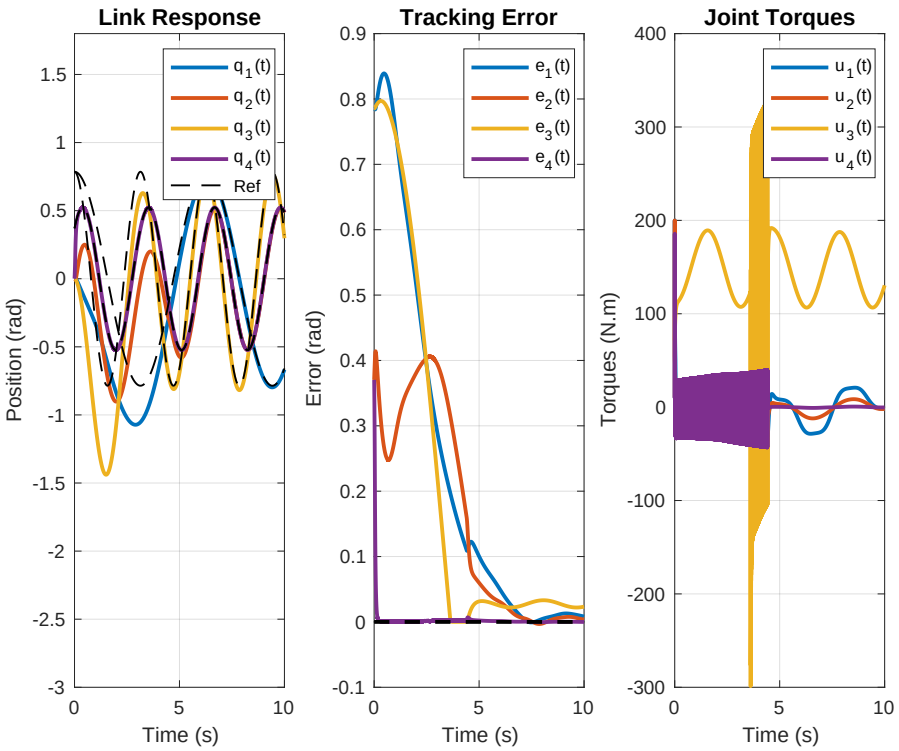
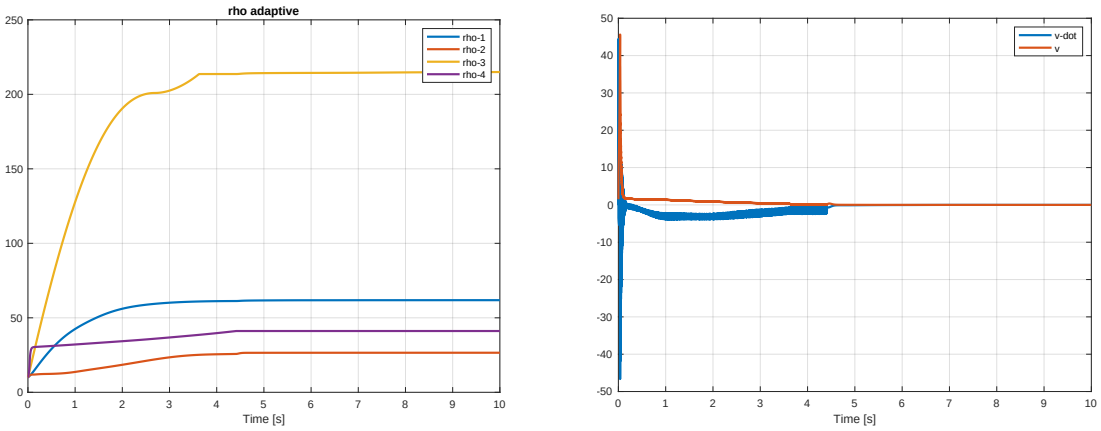


Figure 5.19: Sliding mode control(adaptive matrix ρ)



(a) adaptive ρ evolution

(b) energy function and its derivative

Figure 5.20: Adaptive parameters evolution based on lyapunov function

In this section, we aim to evaluate the sliding mode controller developed in Section 4.5. The behavior of the joints when ρ is a scalar is depicted in Figure 5.17. However, the observed behavior does not align with our expectations. The reason for this discrepancy is that ρ is intended to compensate for the uncertainty in the system, and each link possesses its own uncertainty. Hence, it becomes

necessary to select a distinct ρ value for each input. To address this, we introduce ρ as an $(n \times n)$ matrix. By setting the control parameters as follows:

$$\rho = \begin{bmatrix} 70 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad \epsilon = 3, \quad \lambda = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 30 \end{bmatrix}$$

we achieve the results depicted in Figure 5.18 . In this figure, all joint variables converge to their desired values, and the tracking error converges to zero. Since selecting the parameters of the ρ matrix is nontrivial, an update law was introduced to learn the parameters based on the error dynamics. By setting the control parameters as follows:

$$\rho_0 = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}, \quad \epsilon = 3, \quad \lambda = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 30 \end{bmatrix}$$

where ρ_0 represents the initialization of the ρ matrix, we can observe the evolution of ρ_i and the Lyapunov energy function in Figure 5.20(a) and 5.20(b), respectively. It is evident that the tracking error eventually approaches zero. The convergence speed depends on the initialization of ρ.

The previously mentioned sliding mode control (SMC) and adaptive sliding mode control (ASMC) strategies employ the linear sliding surface as defined in Eq.(4.36) coupled with a constant reaching law. An examination of both scenarios reveals a common tendency towards gradual convergence, marked by a subdued convergence rate. Furthermore, it is noteworthy that both instances are characterized by the occurrence of oscillations in joint torques.

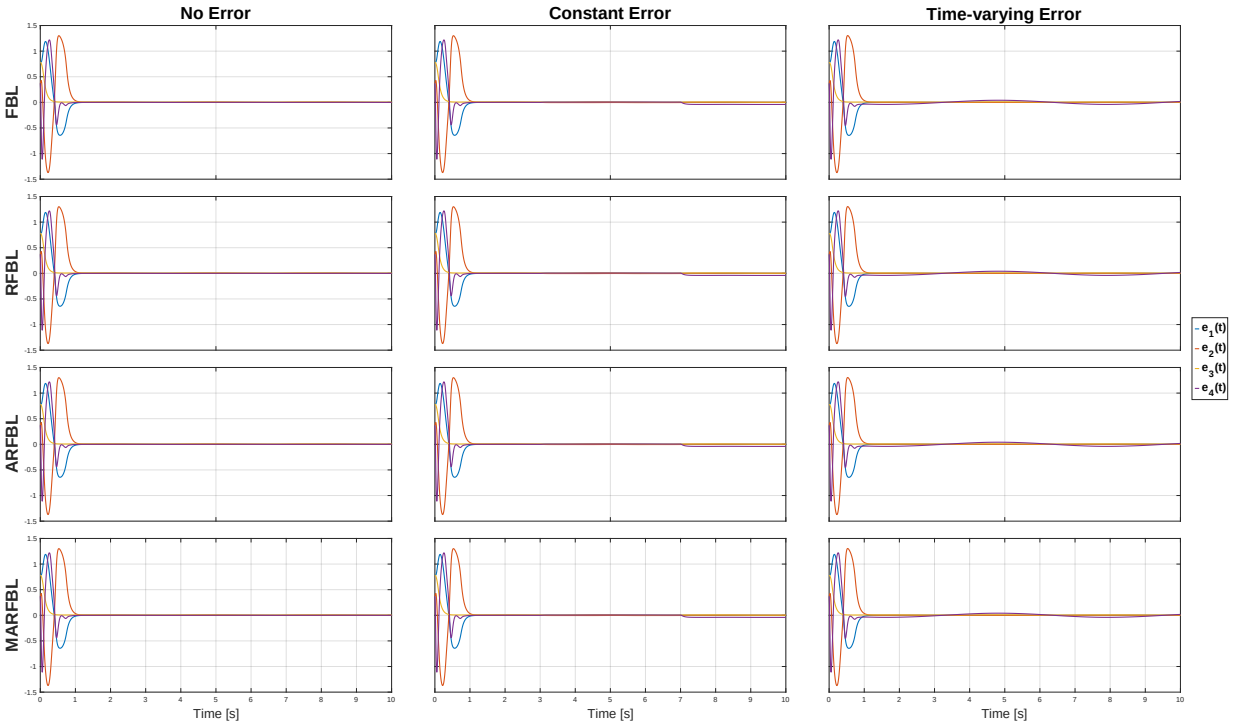


Figure 5.21: RMSE under diffrent levels of external disturbances

	FBL	RFBL	ARFBL	MARFBL
No Error	0.1125	0.1125	0.1125	0.1125
Constant Error	0.1234	0.1233	0.1234	0.1234
Time-varying Error	0.1307	0.1307	0.1307	0.1307

Table 5.6: RMSE of different robust feedback linearization controllers under different external disturbances

Examining Table 5.6 and Fig.(5.21), which presents a comparative evaluation of various robust feedback linearization laws and their variants in the presence of distinct magnitudes of external disturbance, it becomes evident that the root mean square error (RMSE) values across all controllers exhibit a notable degree of similarity. This phenomenon finds its explanation in the underlying principles outlined in Eq.(4.32), wherein the parameter ρ assumes the responsibility of amplifying the proportional-derivative (PD) control gains proportionate to the extent of uncertainty and the discrepancy between the actual and desired trajectories. In cases where the PD gains become exceedingly elevated, the influence exerted by ρ tends to diminish significantly in comparison to these dominant gains.

5.3 UR10 Manipulator

5.3.1 Kinematic and Dynamic Parameters

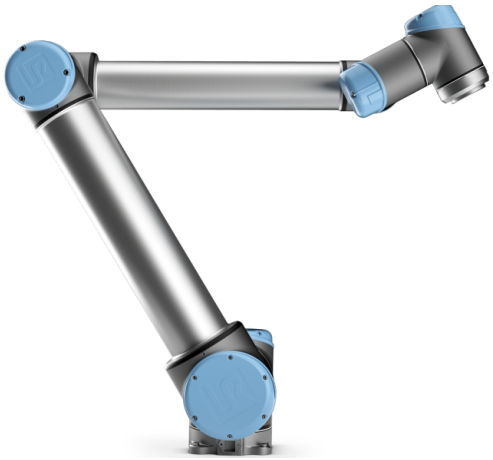


Figure 5.22: UR10 Manipulator

The Universal Robots UR10 in Fig (5.22) is a versatile and widely used robotic manipulator known for its flexibility and ease of integration into various industrial applications. As with any robot manipulator, understanding its kinematics and dynamics is crucial for precise control and effective task planning.

Forward Kinematics

Link	θ	d	a	α
1	${\theta_1}^*$	0.1273	0	$\pi/2$
2	${\theta_2}^*$	0	-0.612	0
3	${\theta_3}^*$	0	-0.5723	0
4	${\theta_4}^*$	0.163941	0	$\pi/2$
5	${\theta_5}^*$	0.163941	0	$-\pi/2$
6	${\theta_6}^*$	0.0922	0	0

Table 5.7: DH-table for UR10 manipulator

where the kinematic parameter can be illustrated from Fig.(5.23)

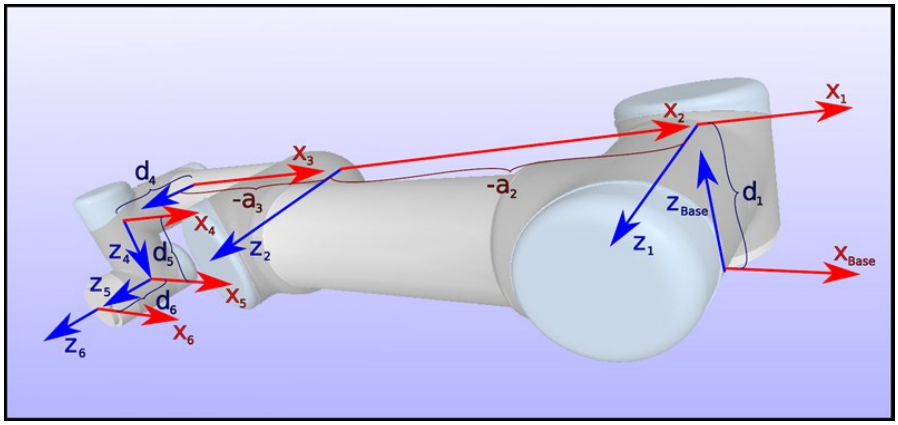


Figure 5.23: UR10 Manipulator Kinematic Parameters

Dynamics

In the Table 5.8 ,we reported the parameters defining the dynamics, where p_x, p_y, and p_z are the three components of p_c, expressed w.r.t. the correspondent reference frame of the Denavit-Hartenberg convention. Each link is assumed to be a cylinder, with radius r_i and length l_{c_i}.

Link	$m_i[kg]$	$p_x[m]$	$p_y[m]$	$p_z[m]$	$r_i[m]$	$l_i[m]$
1	7.1	0.021	0.0	0.027	0.2	0.1273
2	12.7	0.382	0.0	0.158	0.15	0.612
3	4.27	0.24	0.0	0.068	0.15	0.5723
4	2	0.01	0.007	0.018	0.1	0.163941
5	2	−0.0	0.007	0.018	0.1	0.1157
6	0.365	0.0	0.0	−0.026	0.1	0.0922

Table 5.8: Dynamic parameters for UR10 manipulator

5.3.2 Setup

The desired input trajectory is defined as follows:

$$q_{d_1} = \frac{\pi}{4}\cos(t + \frac{\pi}{2}), \quad q_{d_2} = \frac{\pi}{6}\cos(2t + \frac{\pi}{4}), \quad q_{d_3} = \frac{\pi}{4}\cos(2t + \frac{\pi}{2})$$

$$q_{d_4} = \frac{\pi}{6}\cos(2t + \frac{\pi}{4}), \quad q_{d_5} = \frac{\pi}{4}\cos(2t + \frac{\pi}{6}), \quad q_{d_6} = \frac{\pi}{6}\cos(2t + \frac{\pi}{4})$$

5.3.3 Feedback Linearization Simulation

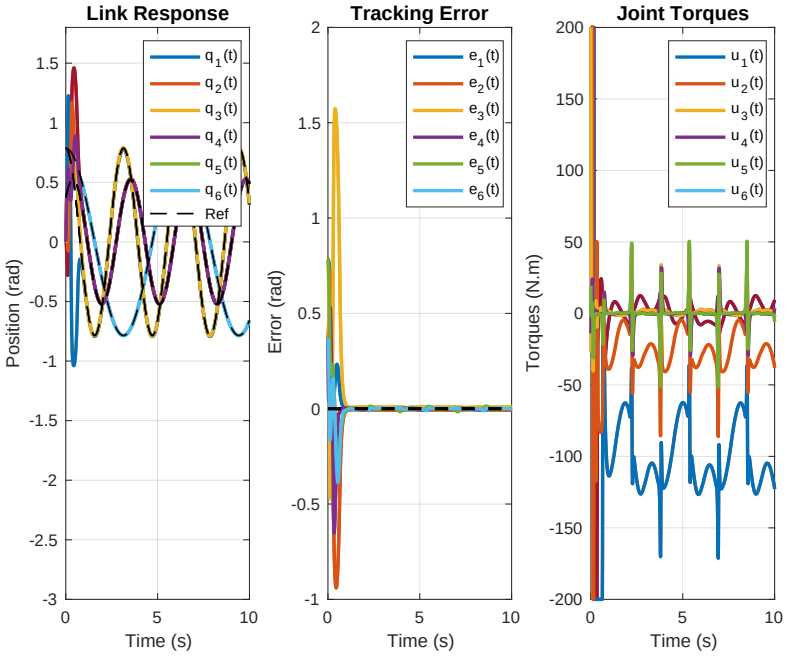


Figure 5.24: Feedback linearization control

In Figure 5.24, feedback linearization was employed to control the robot's behavior. The control gains used were

$$K_p = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix}$$

and

$$K_d = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$$

5.3.4 Robust Feedback Linearization Simulation

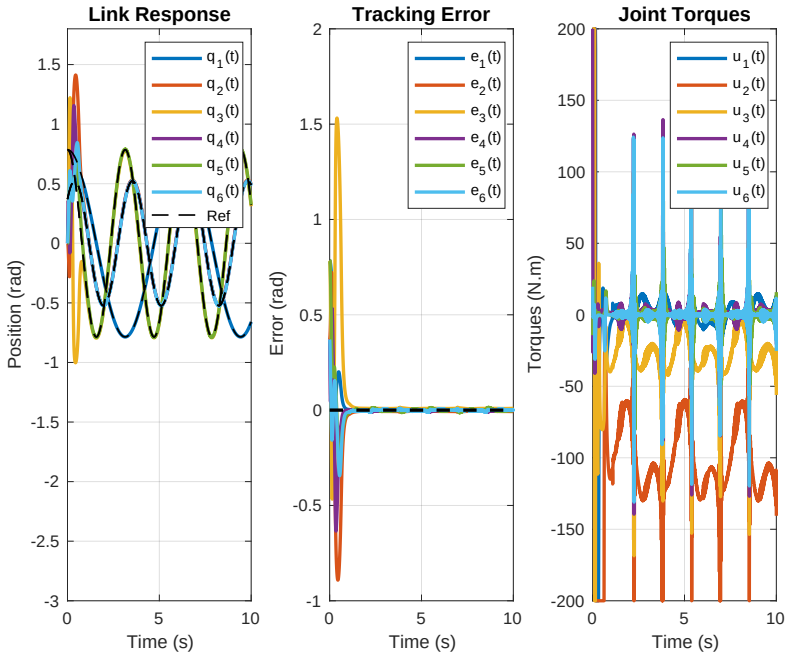


Figure 5.25: Robust Feedback linearization control(chattering)

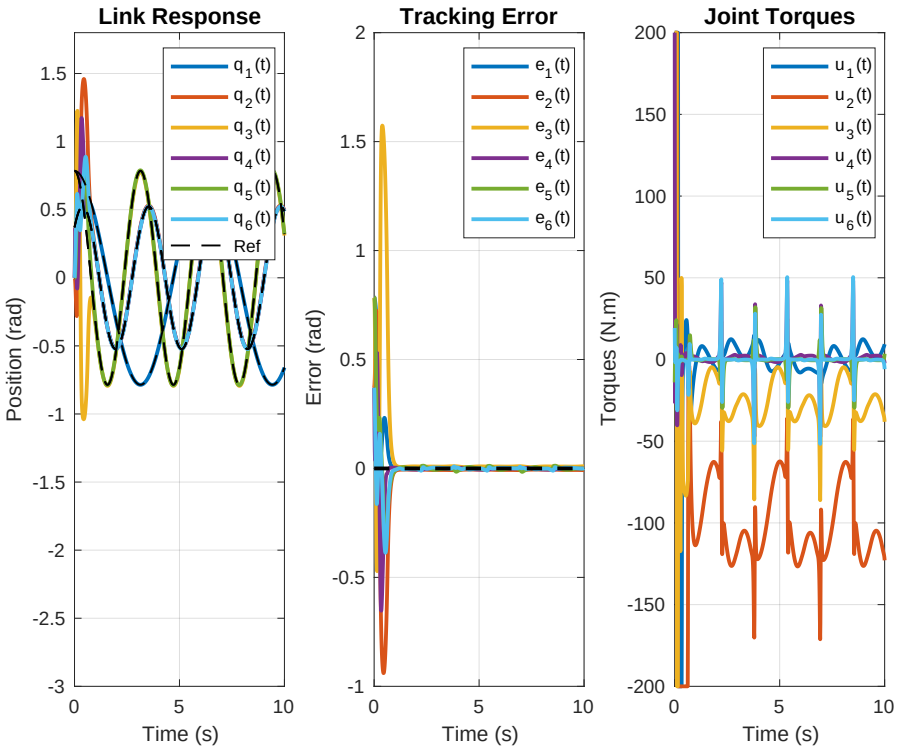


Figure 5.26: Robust Feedback linearization control(without chattering)

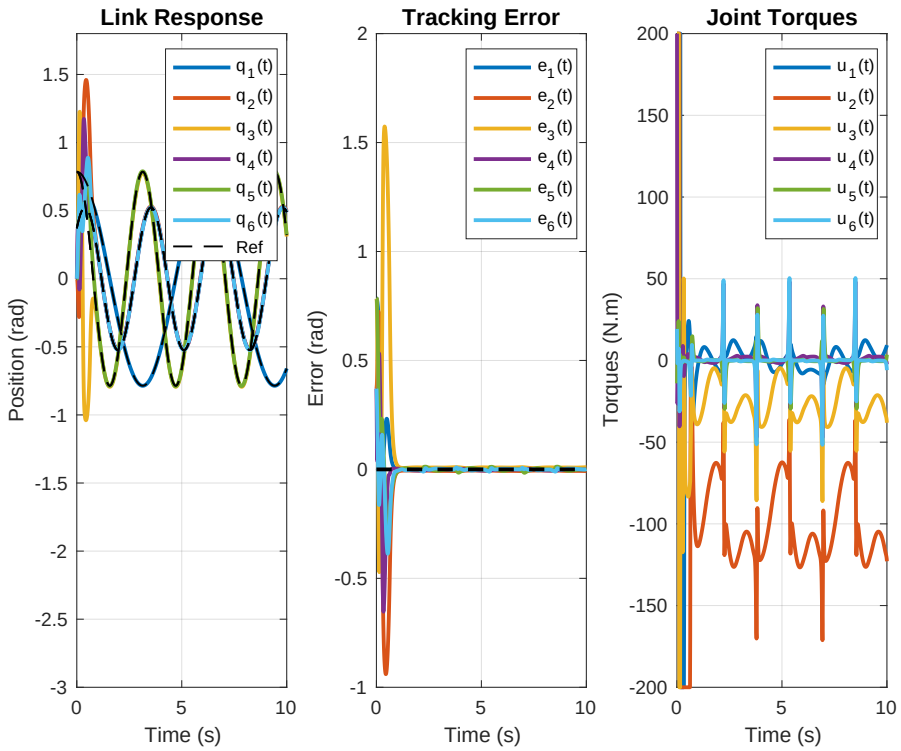


Figure 5.27: Robust Feedback linearization control(adaptive ρ)

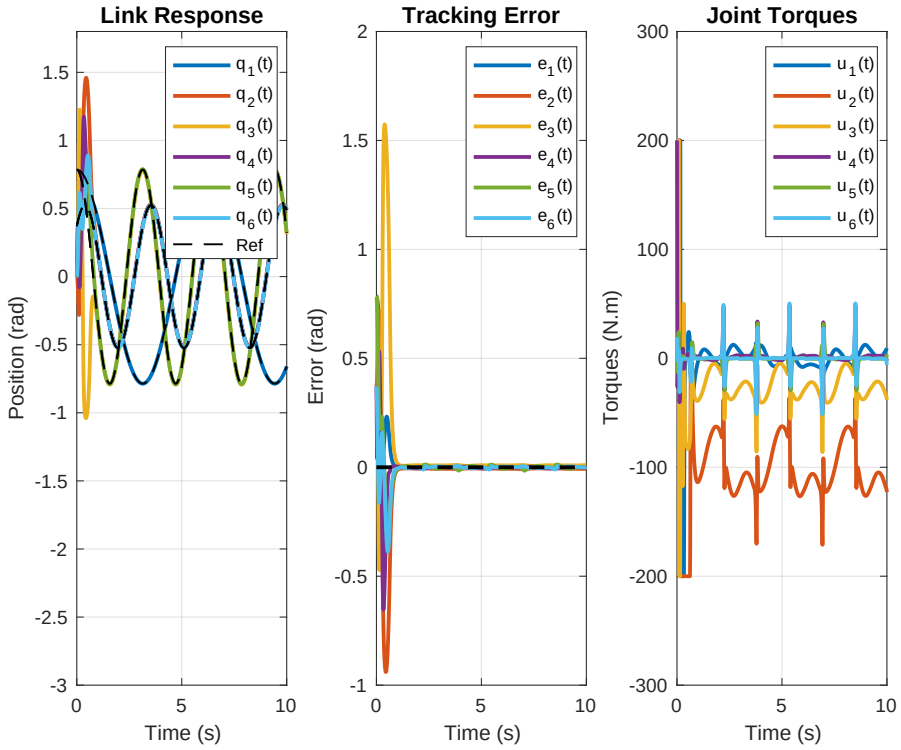
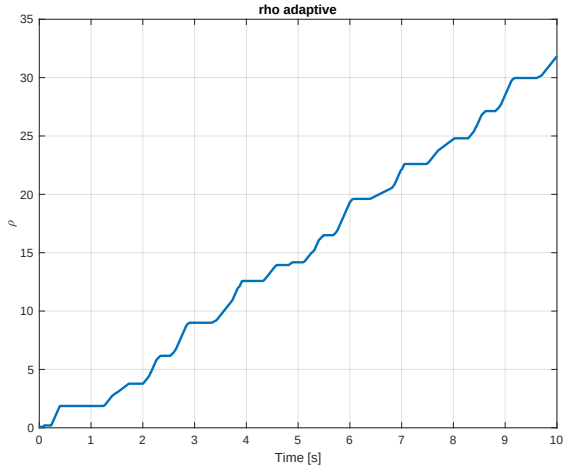
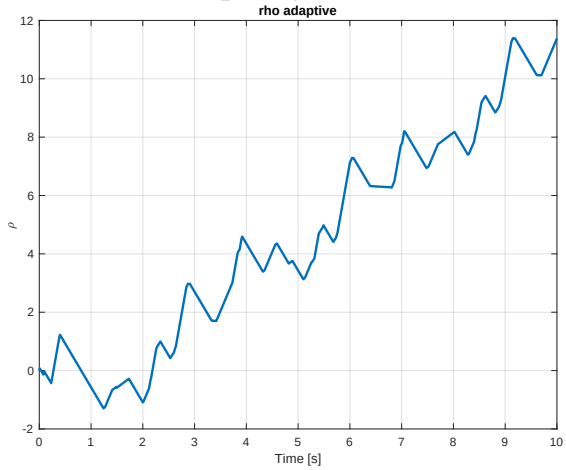


Figure 5.28: Robust Feedback linearization control(modified adaptive ρ)



(a) adaptive ρ evolution



(b) modified adaptive ρ evolution

Figure 5.29: ρ evolution based on derivative of $\dot{V}$

Illustrated within Figures (5.25-5.28) is the dynamic behavior exhibited by distinct feedback linearization approaches. Analogous to the response patterns observed in RR and SCARA manipulators, the Robust Feedback Linearization (RFBL) and Adaptive Robust Feedback Linearization (ARFBL) methodologies deliver notable performance outcomes in the presence of heightened uncertainty. Their operational efficacy is underscored by their rapid convergence dynamics and the generation of seamless input torque profiles.
However, when implementing the RFBL technique, as depicted in Figure (5.25), employing the sign function introduces undesirable chattering phenomena and eventual saturation of joint torques. To address these concerns, a novel adaptive law is proposed. The application of this refined approach yields a dual advantage: fast convergence coupled with a smooth transition profile.

5.3.5 Sliding Mode Control Simulation

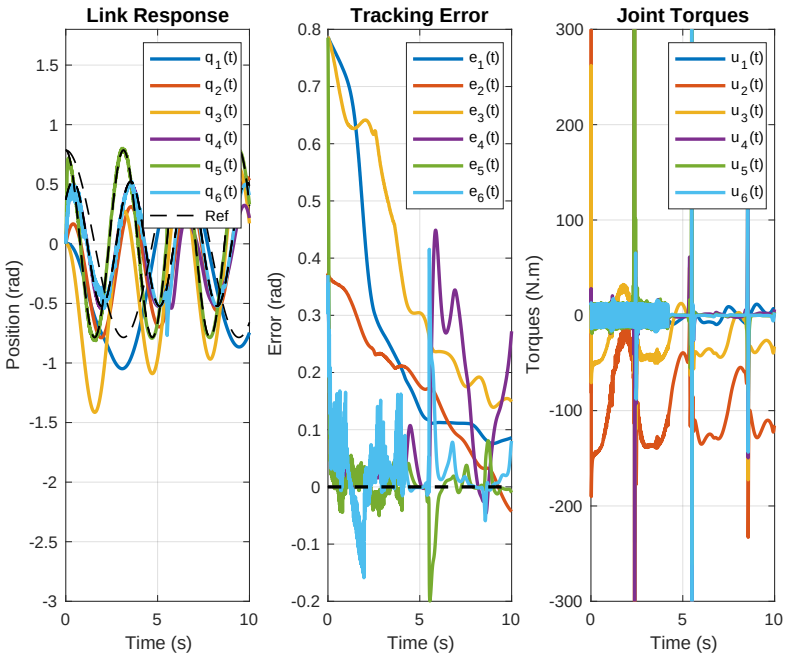


Figure 5.30: Sliding mode control(matrix ρ)

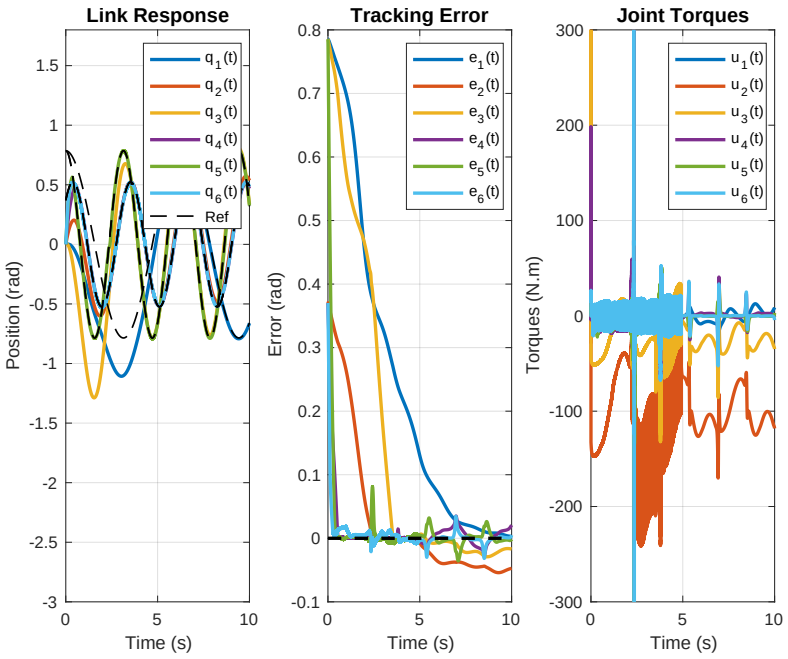


Figure 5.31: Sliding mode control(adaptive matrix ρ)

$$\rho_0 = \begin{bmatrix} 30 \\ 60 \\ 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}, \quad \epsilon = 7, \quad \lambda = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 & 0 & 30 \end{bmatrix}$$

In Figures 5.30 and 5.31, the Sliding Mode Control (SMC) and Adaptive Sliding Mode Control (ASMC) methodologies were employed, respectively. These approaches utilized a constant reaching law, characterized by the parameter ρ_0, which serves as the initialization for the ρ matrix. Notably, it is observed that the convergence of the ASMC approach exhibits enhanced performance.

It is apparent that the tracking error progressively diminishes over time, ultimately converging to zero. The rate of convergence is influenced by the initial value of ρ. This observation underscores the necessity for the integration of more intricate reaching laws to further enhance performance.

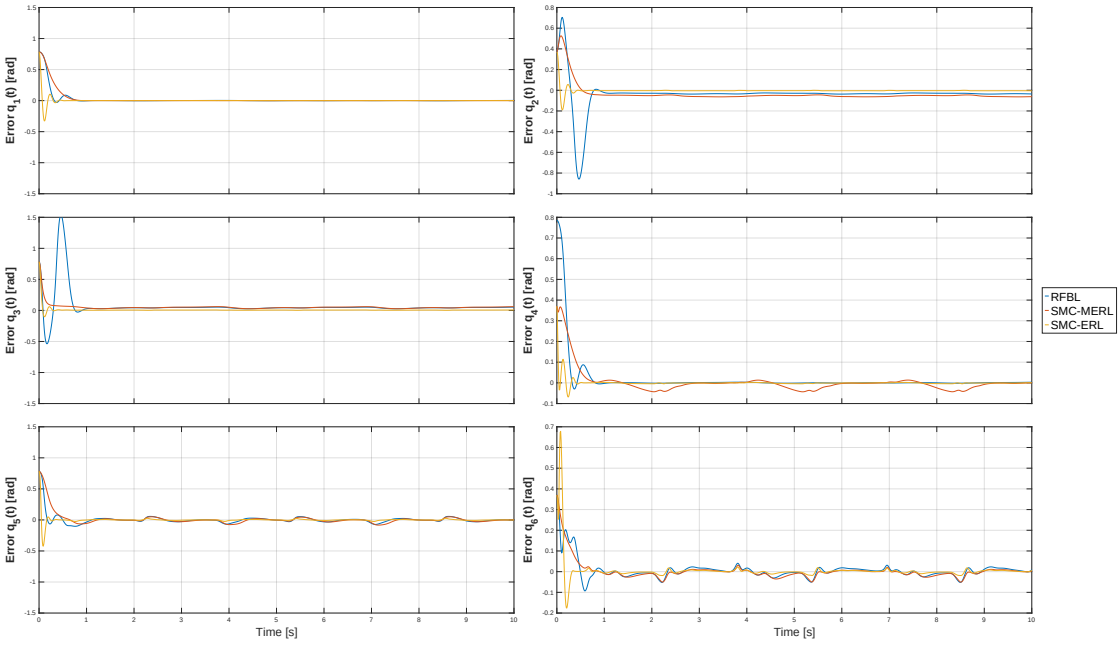


Figure 5.32: Joint error at $\epsilon = 0.5$

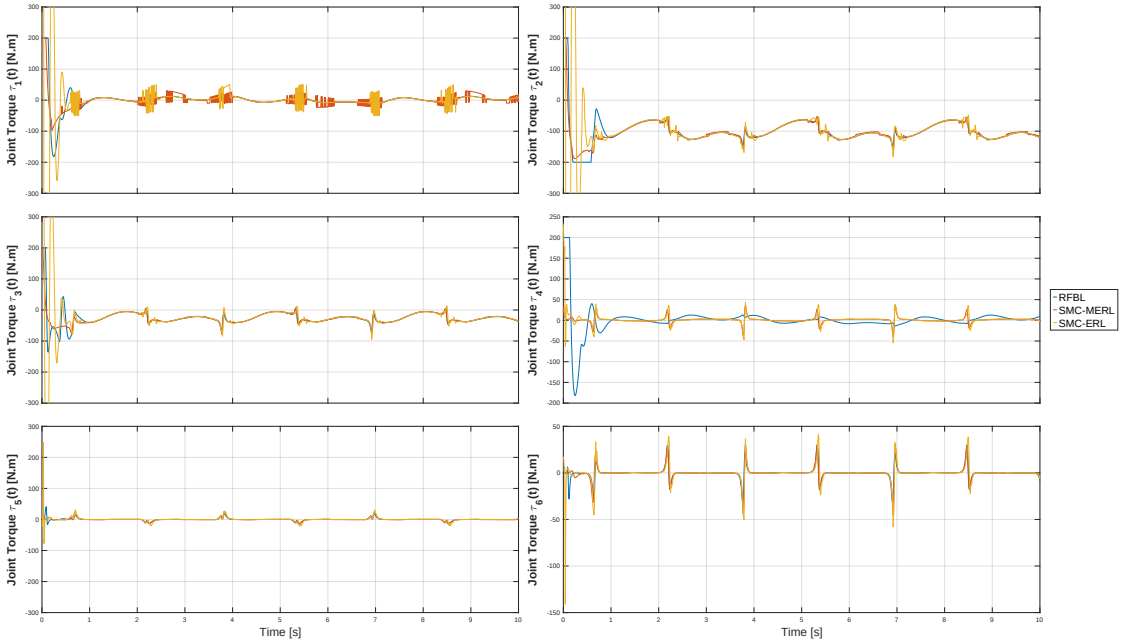


Figure 5.33: Joint torques at $\epsilon = 0.5$

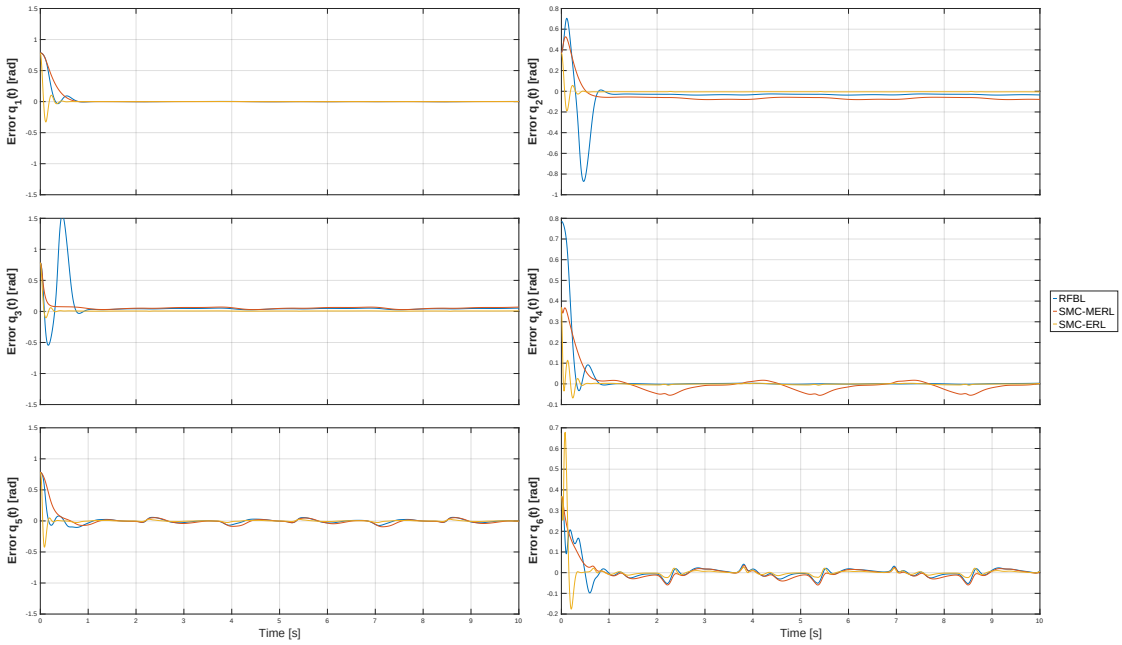


Figure 5.34: Joint error at $\epsilon = 1$

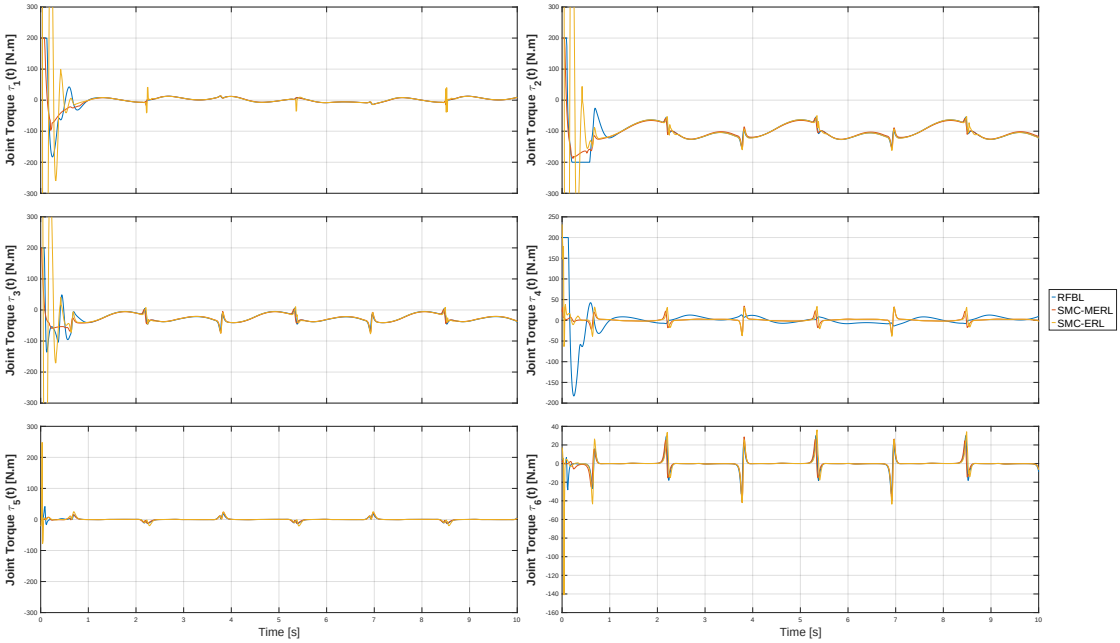


Figure 5.35: Joint torques at $\epsilon = 1$

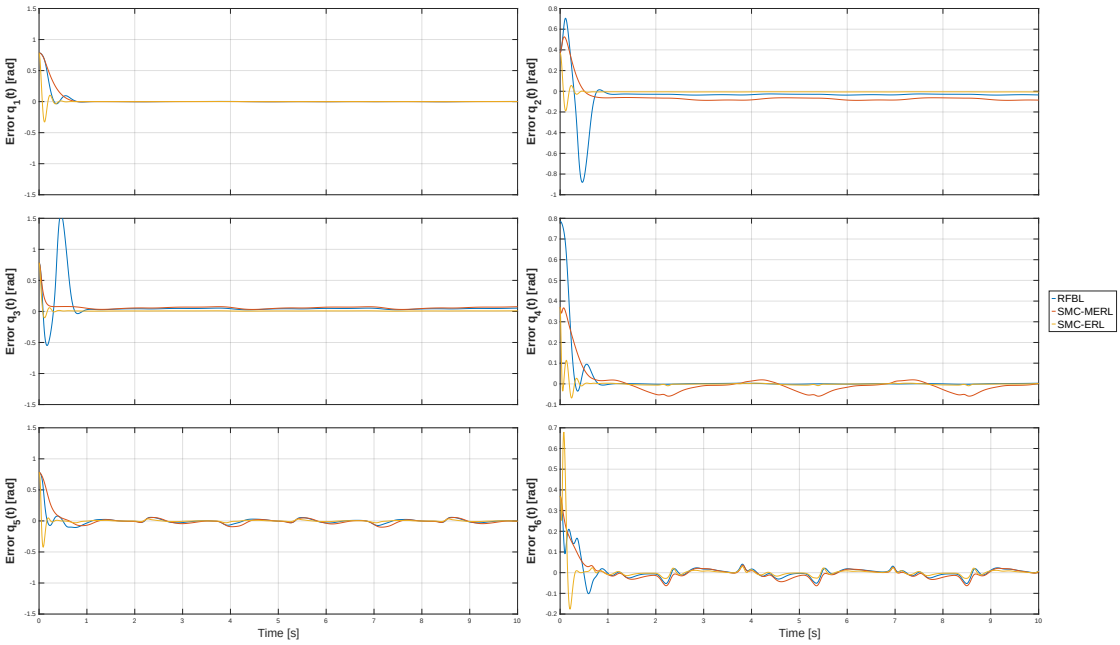


Figure 5.36: Joint error at $\epsilon = 3$

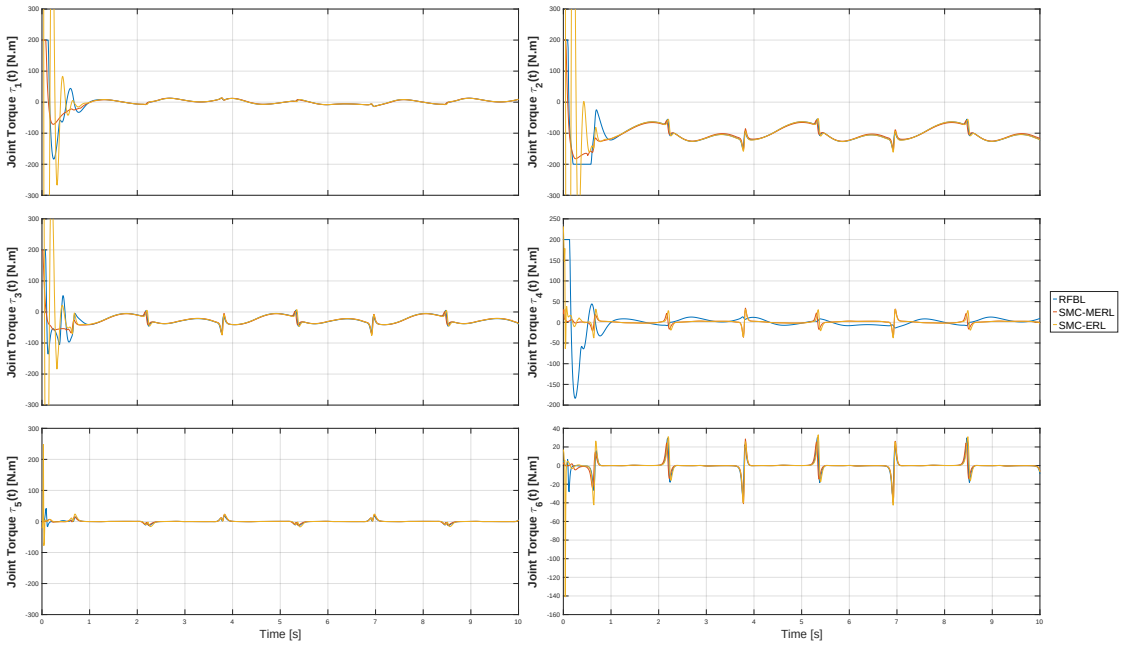


Figure 5.37: Joint torques at $\epsilon = 3$

	ASMC	**RFBL**	**SMC-ERL**	**SMC-MERL**
$\epsilon = 0.5$	0.3005	0.1326	0.0264	0.1133
$\epsilon = 1$	0.3002	0.1332	0.0288	0.1297
$\epsilon = 3$	0.3077	0.1337	0.0309	0.1379
$\epsilon = 5$	0.3147	0.1338	0.0315	0.1397

Table 5.9: RMSE of different controllers under different ϵ values

Comparing the behaviors depicted in Figures 5.30 and 5.31, specifically the Sliding Mode Control (SMC) employing a constant reaching law with ρ as a constant matrix against the case where ρ is adaptive, it is evident that the Adaptive Sliding Mode Control (ASMC) exhibits smoother trajectories characterized by smaller bounded tracking errors. However, both methods display slower convergence tendencies.

Further investigation, presented in Figures 5.32-5.37, focuses on the most effective controllers Sliding Mode Control with linear sliding surface and exponential reaching law (SMC-ERL) as defined in Equation (4.38), Sliding Mode Control with nonlinear sliding surface and exponential reaching law (SMC-MERL), and RFBL. These figures highlight how the sliding mode controllers exhibit chattering behaviors when ϵ is small, while RFBL and ARFBL strategies

effectively mitigate chattering, even in cases of small ϵ.

The dataset provided in Table 5.9 examines the impact of changing the parameter ϵ on the root mean square error (RMSE) values associated with different controllers: Adaptive Sliding Mode Control (ASMC), Robust Feedback Linearization (RFBL), Sliding Mode Control with Exponential Reaching Law (SMC-ERL), and Sliding Mode Control with Nonlinear Sliding Surface and Exponential Reaching Law (SMC-MERL).

Upon analyzing the table, discernible trends emerge that highlight the interplay between ϵ variations and controller performance. As ϵ increases, there is a general upward trend in the RMSE values across all controllers. This suggests that heightened values of ϵ are associated with greater RMSE values but on the other hand smoother joint torques.

Examining the controllers individually, the Adaptive Sliding Mode Control (ASMC) consistently exhibits the highest RMSE values across all ϵ values, indicating its comparatively weaker performance in minimizing tracking errors. This is due to the choice of proper reaching mode is crucial for the performance and convergence speed to the desired trajectory.

Conversely, the Robust Feedback Linearization (RFBL) method maintains relatively stable RMSE values irrespective of variations in ϵ, underlining its robustness against changes in this parameter.

Furthermore, the Sliding Mode Control with Exponential Reaching Law (SMC-ERL) has the best performance in terms of smoothness of joint torques and values of RMSE even if we increase ϵ . This again highlights the importance of choosing the proper sliding surface and the reaching law.

On the other hand, the Sliding Mode Control with Nonlinear Sliding Surface and Exponential Reaching Law (SMC-MERL) portrays an ascending trend in RMSE values with increasing ϵ, indicating its relatively deteriorating performance as ϵ grows.

In summary, the data presented in the table unravels the nuanced connection

between ϵ variations and the performance of different controllers. The trends underscore the importance of selecting an appropriate ϵ value to optimize the controller's tracking accuracy while maintaining smooth input torque . Furthermore, the comparison discerns the controllers with the most favorable and unfavorable RMSE outcomes. RFBL emerges as a robust contender, whereas ASMC tends to yield higher errors. The observed trends in this analysis offer valuable insights into controller dynamics and serve as a guide for selecting optimal control strategies across different scenarios.

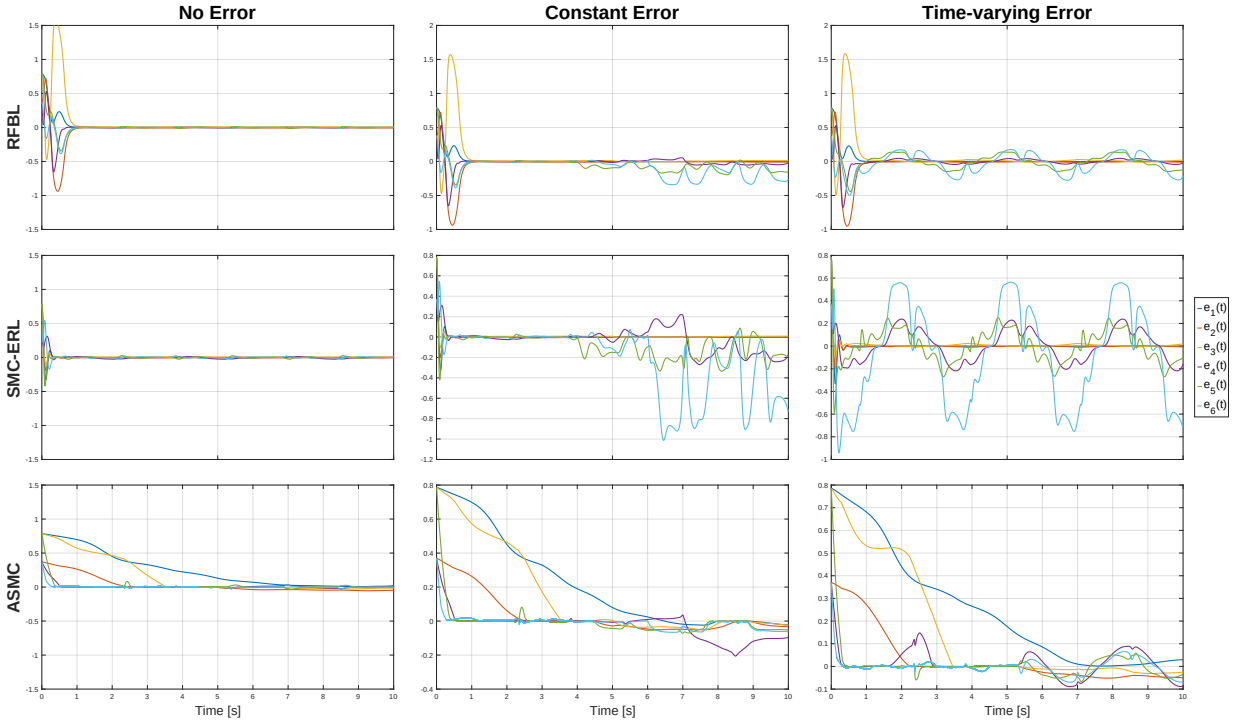


Figure 5.38: Comparison with best performance controllers under different levels of disturbances

	RFBL	SMC-ERL	ASMC
No Error	0.1057	0.0336	0.3444
Constant Error	0.2087	0.2867	0.3217
Time-varying Error	0.2342	0.4249	0.3475

Table 5.10: RMSE of different controllers under different external disturbances

The investigation entails a comparative analysis utilizing Figure 5.38 and Table 5.10 to contrast two distinct sliding mode controllers (SMCs) with a robust

feedback linearization (RFBL) controller across three distinct external disturbance scenarios. The observations reveal distinctive controller behaviors under these conditions.

When there is no external disturbance, the SMC-ERL demonstrates notable superiority in performance, showcasing both rapid convergence and the minimal root mean square error (RMSE) among all controllers. However, upon the introduction of noise to the system, the RFBL controller emerges as the most resilient in maintaining minimal tracking error. This resilience is particularly evident when contrasted with the other controllers. Conversely, the adaptive sliding mode controller (ASMC) exhibits sluggish convergence, contributing to higher RMSE values in all scenarios. This outcome can be attributed to the ASMC's utilization of a constant reaching law, leading to a plateauing of tracking error upon reaching the sliding surface.

In summary, this comparative assessment highlights the varying performance profiles of the analyzed controllers in response to different external disturbance scenarios. The SMC-ERL excels in disturbance-free environments, the RFBL controller demonstrates robustness against noise, and the ASMC presents limitations in convergence and tracking accuracy due to its control strategy. These findings deepen our comprehension of controller behavior and inform controller selection based on specific operational contexts.

6

Conclusions and Future Works

6.1 Discussion

In this thesis, an investigation was undertaken to address the challenges associated with controlling robot manipulators in the presence of unknown uncertainties. In Chapter 4, the focus centered on two primary categories of robot controllers: robust feedback approaches, which introduced a novel update law to mitigate uncertainty, and sliding mode approaches. These controllers were rigorously examined across diverse scenarios, encompassing varying levels of structural and non-structural uncertainties.

Chapter 5 presented simulation outcomes for a range of scenarios, including a comparative analysis of existing control methods. The simulation encompassed three distinct manipulator types two-link RR planar, SCARA, and UR10 with increasing complexity to validate the results. The findings underscored the efficacy of the proposed adaptive robust feedback linearization controller in effectively addressing diverse forms of uncertainty across all manipulators.

6.2 Conclusion

The research outcomes yielded the following conclusions:

- The study of robot manipulator control in the context of uncertain time-

varying parameters is of paramount importance, given its potential to enhance operational accuracy and tracking precision.

- The introduced adaptive robust controller successfully achieved control objectives, demonstrating the ability to minimize tracking errors, ensure fast convergence, and provide smooth input torque.
- Notably, for sliding mode control, the selection of a suitable sliding surface emerged as a critical factor with a direct impact on performance. This was evident in the simulation results of the UR10 manipulator, where the choice of sliding law significantly influenced outcomes. Employment of the exponential reaching law yielded the best performance in terms of convergence and root mean square error (RMSE), contrasting with the poorer performance associated with the constant reaching law.

6.3 Future avenues of research

- Development of a specialized adaptive robust controller tailored for tracking control in robot manipulators. This controller would amalgamate the strengths of adaptive and robust control techniques, reducing online computations while ensuring robustness against bounded disturbances and eliminating the need for prior knowledge of system uncertainty. Furthermore, this controller would strive to achieve asymptotic tracking error performance.
- Enhanced system modeling through the utilization of Gaussian Processes based on measured data, thereby facilitating improved cancellation of nonlinearities.
- Advancements in the Adaptive sliding mode control methodology, involving the refinement of sliding surface choices and the exploration of novel reaching laws.
- Extension of the adaptive robust control concepts to address force control scenarios, broadening the scope of their applicability.

References

[1] S. Barnett, *Matrix Methods for Engineers and Scientists*. McGraw-Hill, London, 1979.

[2] J.J. Craig, *Introduction to Robotics - Mechanics and Control*, second edition, Addison-Wesley, 1986.

[3] C. Chen, *Linear System Theory and Design*. Oxford University Press Inc., 1995.

[4] A.M. Lyapunov, *The general problem of the stability of motion*, Inf. J. Control, Vol. 55, No. 3, pp. 531-773, 1992.

[5] J. Denavit and R. S. Hartenberg, *A kinematic notation for lower pair mechanisms*. Applied Mechanics, 22:215221, 1955.

[6] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control (1st. ed.)*. Cambridge University Press, USA, 2017.

[7] E. T. Whirraker, *Dynamics of Particles and Rigid Bodies*. Cambridge University Press, London, 1904.

[8] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, John Wiley & Sons, 2006.

[9] P. Rocco, *Stability of PID control for industrial robot arms*, IEEE Trans. Robot. Autom. 1996, 12, 606614. [CrossRef]

[10] C. Aguilar-Ibanez, J. Moreno-Valenzuela, O. García-Alarcón, M. Martinez-Lopez, J.Á. Acosta, M.S. Suarez-Castanon, *PI-Type Controllers and SD Modulation for Saturated DC-DC Buck Power Converters*, IEEE Access 2021, 9, 2034620357. [CrossRef]

[11] L.A. Soriano, E. Zamora, J. Vazquez-Nicolas, G. Hernández, J.A. Barraza Madrigal, D. Balderas, *PD control compensation based on a cascade neural network applied to a robot manipulator*, Front. Neurorobotics 2020, 14, 78. [CrossRef] [PubMed]

[12] H. A. Abbas, M. Belkheiri, and B. Zegnin, *Feedback linearization control for highly uncertain nonlinear systems augmented by single-hidden-layer neural networks*, Journal of Engineering Science and Technology Review, vol. 8, no. 2, pp. 215224, 2015.

[13] Y. Kali, M. Saad, K. Benjelloun, *Optimal super-twisting algorithm with time delay estimation for robot manipulators based on feedback linearization*, Robot. Auton. Syst. 2018, 108, 8799. [CrossRef]

[14] Q. Hu, L. Xu, A. Zhang, *Adaptive backstepping trajectory tracking control of robot manipulator*, J. Frankl. Inst. 2012, 349, 10871105. [CrossRef]

[15] M. Liu, *Decentralized control of robot manipulators: Nonlinear and adaptive approaches*, IEEE Trans. Autom. Control 1999, 44, 357363.

[16] A. Safaei, Y. C. Koo, and M. N. Mahyuddin, *Adaptive model-free control for robotic manipulators*, in Proc. IEEE Int. Symp. Robot. Intell. Sensors (IRIS), Oct. 2017, pp. 712.

[17] J. J. E. Slotine and W. Li, *Adaptive manipulator control: a case study*, IEEE Trans. on Automatic Control, vol. 33, no. 11, 1988.

[18] M. Bueno Lopez and D. Marino Lizarazo, "A comparative analysis of adaptive visual servo control for Robots Manipulators in 2D," *2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, Jun 12-13, 2015.

[19] S. Islam, X.P. Liu, *Robust sliding mode control for robot manipulators*, IEEE Trans. Ind. Electron. 2010, 58, 24442453. [CrossRef]

[20] Y. Stepanenko and J. Yuan, *Robust adaptive control of a class of nonlinear mechanical systems with unbounded and fast-varying uncertainties*, Automatica, vol. 28, no. 2, pp. 265–276, 1992.

[21] F. Wang, Z.q. Chao, L.b. Huang, H.y. Li, C.q. Zhang, *Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode*, Clust. Comput. 2019, 22, 57995809. [CrossRef]

[22] P. R. Naik, J. Samantaray, S. K. Pattanayak, and B. K. Roy, "2-DOF Robot Manipulator Control Using Fuzzy PD Control With SimMechanics and Sliding Mode Control: A Comparative Study," *2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, Jun 12-13, 2015.

[23] F. Bouakrif, M. Zasadzinski, *Trajectory tracking control for perturbed robot manipulators using iterative learning method*, Int. J. Adv. Manuf. Technol. 2016, 87, 20132022. [CrossRef]

[24] P. Ji, F. Ma, F. Min, *Terminal Traction Control of Teleoperation Manipulator With Random Jitter Disturbance Based on Active Disturbance Rejection Sliding Mode Control*, IEEE Access 2020, 8, 220246220262. [CrossRef]

[25] H.H. Tsai, C.C. Fuh, J.R. Ho, C.K. Lin, P.C. Tung, *Controller Design for Unstable Time-Delay Systems with Unknown Transfer Functions*, Mathematics 2022, 10, 431. [CrossRef]

[26] Y. Danik, M. Dmitriev, *Symbolic Regulator Sets for a Weakly Nonlinear Discrete Control System with a Small Step*, Mathematics 2022, 10, 487. [CrossRef]

[27] X. Tian, Z. Yang, *Adaptive stabilization of a fractional-order system with unknown disturbance and nonlinear input via a backstepping control technique*, Symmetry 2019, 12, 55.

[28] H. A. Abbas, M. Belkheiri, and B. Zegnin, *Feedback linearization control for highly uncertain nonlinear systems augmented by single-hidden-layer neural networks*, Journal of Engineering Science and Technology Review, vol. 8, no. 2, pp. 215224, 2015.

[29] W. Peng, Z. Lin, and J. Su, *Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques*, IET Control Theory Appl., vol. 3, no. 6, pp. 701711, Jun. 2009.

[30] H. G. Sage, M. F. De Mathelin, and E. Ostertag, *Robust control of robot manipulators: A survey*, Int. J. Control, vol. 72, no. 16, pp. 14981522, 1999.

[31] K. Hashinloto, T. Kunoto, M. Kawabata, and H.Kimura, *Model-Based Robust Control of a Manipulator*, Proc. IEEE/XSIS Inf. Workshop on Intelligent Robots and Systems IROS '91, pp. 1597-1602, 1991.

[32] C. Kim and K. Lee, *Robust control of robot manipulators using dynamic compensators under parametric uncertainty*, Int. J. Innov. Comput., Inf. Control, vol. 7, no. 7, pp. 41294137, 2011.

[33] L.A. Soriano, J.d.J. Rubio, E. Orozco, D.A. Cordova, G. Ochoa, R. Balcazar, D.R. Cruz, J.A. Meda-Campaña, A. Zacarias, G.J. Gutierrez, *Optimization of Sliding Mode Control to Save Energy in a SCARA Robot*, Mathematics 2021, 9, 3160.

[34] S. Islam and X. P. Liu, *Robust sliding mode control for robot manipulators*, IEEE Transactions on Industrial Electronics, vol. 58, no. 6, pp. 2444-2453, Jun. 2011.

[35] J.-J. E. Slotine, *Sliding controller design for non-linear systems*, International Journal of Control, vol. 40, no. 2, pp. 421434, 1984.

[36] F. Piltan and N. B. Sulaiman, *Review of sliding mode of robotic manipulator*, World Applied Sciences Journal, vol. 18, no. 12, pp. 18551869, 2012.

[37] M. Zeinali and L. Notash, *Adaptive sliding mode control with uncertainty estimator for robot manipulators*, Mechanism and Machine Theory, vol. 45, no. 1, pp. 8090, 2010.

[38] Ji, P.; Li, C.; Ma, F. *Sliding Mode Control of Manipulator Based on Improved Reaching Law and Sliding Surface*. *Mathematics* 2022, 10, 1935.

[39] Gao, W.; Hung, J.C. *Variable structure control of nonlinear systems: A new approach*. *IEEE Trans. Ind. Electron.* 1993, 40, 4555.

[40] Fallaha, C.J.; Saad, M.; Kanaan, H.Y.; Al-Haddad, K. *Sliding-mode robot control with exponential reaching law*. *IEEE Trans. Ind. Electron.* 2010, 58, 600610. [CrossRef]

[41] Ma, H.; Wu, J.; Xiong, Z. *A novel exponential reaching law of discrete-time sliding-mode control*. *IEEE Trans. Ind. Electron.* 2017, 64, 38403850. [CrossRef]

[42] Zhang, J.; Liu, B.; Jiang, Z.; Hu, H. *The application of sliding mode control with improved approaching law in manipulator control*. In *Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, 31 May–2 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 807–812.

[43] B. Siciliano, L. Sciavicco, L. Villani, and G. C. H. Y. P. Bruno, *Robotics: Modelling, Planning and Control*, Springer, 2010.

[44] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, Springer, 2017.

[45] D. Seo, "Adaptive Control for Robot Manipulator with Guaranteed Transient Performance," *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 12-14, 2016.

[46] H. Wang, "Adaptive Control of Robot Manipulators With Uncertain Kinematics and Dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, Feb 2017.

[47] J. E. Slotine and W. Li, "Adaptive Manipulator Control: A Case Study," *IEEE Transactions on Industrial Electronics*, vol. 33, no. 11, Nov 1988.

Acknowledgments

I wish to extend my heartfelt appreciation to my research supervisor, Prof. Carli Ruggero, whose unwavering support, motivation, and guidance were integral to the successful completion of this work. His patience and consistent encouragement played a pivotal role in shaping my research journey.

Finally yet importantly, I am deeply indebted to my entire family, with a special mention of my wife, Hala, for her boundless love and invaluable assistance across various facets of my life throughout my academic pursuits.