Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Adaptive Fuzzy Sliding Mode Controller of Three Link Robot Arm Manipulator

A thesis submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Control Engineering

By

Addisie Gietie

Advisor

Dr. Chala Merega

July, 2025

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

This is to certify that the thesis prepared by Addisie Gietie , entitled "**Adaptive Fuzzy Sliding Mode Controller of Three Link Robot Arm Manipulator"**, submitted in partial fulfillment of the requirements for the degree of Master of Sciences in Control Engineering complies with the regulations of university and meets accepted standards of originality and quality.

Approved by Board of Examiners

Name	Signature	Date
Dr. Sosina Mengistu		
(School dean)		
Dr. Lebsework Negash		
(Chairman)		
Dr. Chala Merega		
(Advisor)		
Dr. Dereje Shiferaw		
(Internal examiner)		
Mrs. Betelhem		
(Extrnal Examiner)		

DECLARATION

On behalf of myself, I certify that the research study titled "Adaptive Fuzzy Sliding Mode and Robust Tracking Control for Manipulators" is entirely original. The study has never before been turned in for credit toward a degree at this or any other university. Every reference and source used in this thesis has been properly acknowledged.

Addisie Gietie		July, 2025
Student Name	**Signature**	**Submission Date**

As a unvercity advisor, this research has been submitted for review and my approval.

Advisors Name	**Signature**
Dr. Challa Merega	

DEDICATION

The foundation of my life's journey has been my devoted husband, Betre Tilahun, and my devoted parents, Gietie Adane and Worke Zegeye. Your unwavering love and support have been invaluable. With deep appreciation and love,I dedicate this achievement to you.

Addisie Gietie

ACKNOWLEDGMENT

I owe a great debt to my God, for whose unfailing directives and helps illuminated with this path and imbued me with the stamina to go through this thesis process. I wish to thank my advisor, Dr. Chala Merga, from the depth of my heart for his precious advice, unfailing encouragement, and technical contributions that guided the course of this study. I would want to thank the tireless Addis Ababa Institute of Technology (AAiT) control engineering lecturers for their excellent lectures, support, and direction towards my studies. Also my family, your constant love, support, and faith in me have been the bedrock in my success. Your incessant support has encouraged me to strive for perfection. To my fellow students, I am thankful for the camaraderie, memories, and arguments that have made learning alongside me a more enjoyable experience and my academic life worthwhile. I appreciate everyone who has directly or indirectly added value to my academic pursuit. You have all collectively made an impression in my life that will never be erased, and also I am blessed to have your guidance and love.

Addisie Gietie

ABSTRACT

This thesis aims to design an adaptive fuzzy sliding mode controller (AFSMC) for robotic manipulators. Robotic manipulators are characterized by nonlinear, coupled dynamics with inherent uncertainties. To deal with uncertainties, the thesis designed a high-performance nonlinear controller for three-degree-of-freedom robot manipulator. The three-degree-of-freedom robot manipulating arm's trajectory tracking error has been controlled and eliminated in this study using SMC with adaptive fuzzy supervisory. In order to adjust the sliding surface's gain parameter and help it converge and stay at zero, the adaptive fuzzy logic supervisory system and the direct method of the Lyapunov stability theorem were used to analyze the model robot manipulator's stability. The development begins with the derivation of the manipulator kinematics and dynamics, followed by the design of the proposed controller.. MATLAB/ Simulink software was used to assess the current control method's efficacy in a simulation environment. The study conducts a comprehensive performance comparison between AFSMC, FSMC and SMC. AFSMC demonstrates improvements in performance across various trajectory scenarios. Moreover, AFSMC demonstrates superior robustness in the presence of parametric uncertainty(payload variation) and disturbances compared to FSMC and SMC.

Keywords– Adaptive Fuzzy Sliding Mode Controller; Fuzzy Control; MATLAB Simulink; Robot Manipulator; Sliding Mode Controller; Trajectory Tracking.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

FLC Fuzzy Logic Control.

PSO Particle Swarm Optimization.

SMC Sliding Mode control.

CHAPTER 1

INTRODUCTION

As robots are increasingly used for service-oriented tasks, robot manipulators will be even more crucial to industrial automation systems than they are now. When robots start helping us in our homes, offices, and hospitals instead of the controlled environments of factory floors, their control algorithms will need to handle a lot of uncertainty and possibly unanticipated disruptions. It will be challenging to create an accurate mathematical model of the robotic system because of the inherent uncertainty. Consequently, developing reliable, model-free control methods is essential.

1.1 Background of the Study

The term robotics, which is the science and study of robots, is derived from the Czech word "ROBOTA," which means "forced labor," and was first used by [1]. According to the first definition, robots are artificially created computerized machines that carry out tasks either independently or in response to human commands in an effort to facilitate human labor.
A robot is a mechanical device that can operate autonomously or semi-autonomously and is intended to carry out specific tasks. Usually, these devices have sensors, actuators, and a programmable control system that allow them to look around, decide what to do, and then carry out the plan of action to accomplish certain goals[2]. The sensors, manipulators, control systems, power supplies, and software that make up a robot are all integrated systems that cooperate to complete a task. Although robots come in a wide variety of sizes and shapes, they are usually categorized according to their degree of freedom. Each movement axis is regarded as an axis of movement, and each movement axis is regarded as one degree of freedom. The most common movements of a robot manipulator are yaw, pitch, and roll, which aid in locating the tools in a work area and are referred to as position axes.

There is a vast range of robots; from basic devices that carry out monotonous tasks to sophisticated systems that can engage in intricate interactions and make decisions. The theoretical underpinnings of robotics were established by early pioneers such as Alan Turing and John von Neumann in the mid-1900s, marking the beginning of the field's long history[3]. However, in 1954, George Devol and Joseph Engelberger created Unimate, the first industrial robot, which marked a significant advancement in the development of robots as we know them today. This robot, which had a mobile base, was instrumental in transforming manufacturing automation[4]. During the 1960s and 1970s, the Stanford Research Institute produced some famous inventions, such as Shakey the Robot, which used wheels to navigate[5].

In order to accomplish the desired robot behavior, each robot has a controller that continuously reads data from sensors such as motor encoders, force sensors, vision sensors, and depth sensors. The controller then updates the actuator commands. Over 50 of yearly installations worldwide are articulated manipulators, the most prevalent industrial robotic structures [1]. In addition to a variety of handling applications like machine tool tending, metal casting, and general material handling, this kind of robot is used in many industrial processes like welding, painting, pick and place, dispensing, and packaging. It works in three dimensions and has a large workspace. Additionally, robotic manipulators are used for tasks that are too hazardous, repetitive, or dull for humans [6]. These factors make the study of robotic manipulators one of the most fascinating fields in industrial technology applications. The majority of these applications, however, are limited to slow motion operations devoid of environmental interactions. This is mostly because the controllers that are currently on the market have poor performance. Advanced control strategies are needed to help increase the operational speed for greater accuracy. Stability and robustness are essential for controlling robotic manipulators because a controller is the central component of this complex system. The intricacy of its dynamics and the existence of uncertainties in the system present the biggest obstacles to its motion and control. Therefore, in order to achieve precision, high accuracy, and increased productivity, it is important to model and analyze a robotic manipulator as well as implement control methods.

Each robot has a controller that continuously receives input from sensors such as depth, vision, force, and motor encoders. The controller then updates the actuator commands to produce the intended behavior of the robot.

Nowadays, there are numerous methods for controlling the trajectory tracking of a robot arm. Model-based and non-model-based controllers are the two primary categories into which the control approach of the robot arm is separated. These can be divided into three groups:

adaptive fuzzy sliding mode control, fuzzy sliding mode control, and sliding mode control. We have selected the adaptive fuzzy sliding mode controller for this study in order to develop a precisely controlled technique for the dynamic movement of the robot arm. It is insensitive to uncertainty and disturbance in the nonlinear system and has a strong controlling strategy.

1.1.1 Sliding mode controller

Despite its effectiveness for nonlinear and discontinuous systems, the chattering effect remains a significant drawback. Various control methods for nonlinear multi-input multi-output systems have been proposed and implemented by numerous researchers. For both linear and nonlinear systems, it is the primary and most popular kind of variable structure control methodology. By using a high frequency switching control law, it can alter the system's dynamics,[7] Even though sliding mode control has a chattering issue that damages actuators and results in poor trajectory tracking accuracy, it is now widely used in more controlling systems because of its speed, nonlinearity, and switching nature. The mechanical components are also subjected to heat and wear. Nevertheless, there is no assurance of robustness during the reaching phase, and the system is susceptible to disruptions during that time. To reduce the system's sensitivity to uncertainties and disturbances during the reaching phase, some researchers propose enforcing a finite reachability time or selecting a nonlinear sliding surface that starts at the initial state.

Several researchers have put forth different methods to lessen or completely eradicate a system's chattering issue. These methods can be broadly classified into two branches: the boundary layer saturation method, which substitutes a linear (saturation) method with a small switching surface for the discontinuous method, and the estimated uncertainty method, which compensates for uncertainty using the system uncertainty estimator.The estimated uncertainty method, which makes use of fuzzy logic, adaptive fuzzy sliding mode, and artificial intelligence, is a popular choice among researchers. Nevertheless, our system encounters a trajectory tracking error when we attempt to remove the chattering effect of the nonlinear unstable system. Nevertheless, to overcome this error, we suggested a chattering free, insensitive to dynamic uncertainties and disturbance, and robust ways of controller, which is Adaptive fuzzy sliding mode control (AFSMC) for trajectory tracking control of a three link robot arm.

1.1.2 Adaptive fuzzy logic controller

An adaptive fuzzy logic controller combines fuzzy inference with adaptive mechanisms to adjust controller parameters in real time. It is designed to handle system uncertainties, nonlinearities, and time-varying dynamics. This approach improves robustness and control performance in complex and uncertain environments.

Adaptive fuzzy will try to modify the gain parameter without being aware of the dynamic behavior of the system when it has to be trained online with changing dynamic parameters. However, the instability and rise time increment problems can be resolved and a good trajectory tracking controller can be obtained by combining the SMC with an adaptive fuzzy logic gain scheduler.

1.2 Statement of the Problem

A multi-body nonlinearly coupled system with varying parameters is a robot manipulator. Among the position-dependent gravitational forces are the centrifugal force, carioles, and nonlinear and coupled dynamics. acceleration, friction force, and other reaction forces in joints on other links. These systems are mostly controlled by sliding mode control and fuzzy logic controllers in trajectory tracking applications. To solve those trajectory tracking issues and obtain the necessary precision, efficacy, and stable dynamic trajectory tracking control system for our three-degree-of-freedom robot manipulator, the pre-tuned sliding mode will be applied using fuzzy logic. Using a classical sliding mode controller or a fuzzy logic controller won't produce a stable system when, for instance, the gripper of a robotics manipulator holds heavy workpieces and systems used in assembling operations, high-speed operating robots, etc. are exposed to dynamic uncertainties and external disturbances. Numerous authors have proposed using a nonlinear controller to overcome the limitation of sliding mode control, and various approaches have been proposed for designing the adaptive fuzzy logic gain for sliding mode controller, which is a nonlinear control. The controller for a three-DOF robotic manipulator that requires the least amount of work and the least amount of error is modeled and designed in this paper. Because the system is highly time-varying and highly coupled MIMO, trajectory-tracking control of robotic manipulators is a difficult task. Additionally, the system model is always full of uncertainties, including parameter uncertainty and external disturbances, which lead to unstable control system performance. Furthermore, it has been noted that the majority of research projects overlook actuator

dynamics. Trajectory tracking control performance suffered as a result. The two key qualities that a robot manipulator needs in any control system are stability and robustness. Therefore, using appropriate control techniques to model and assess robot manipulators is crucial. Various studies have been conducted to mitigate the drawbacks of SMC control through the use of a nonlinear controller. Additionally, various methods for designing nonlinear controllers, including FLC, SMC, and AFSMC, have been put forth. Although all three of these controllers have been effectively used in numerous applications, they do have certain drawbacks. In order to improve the system performance response, this paper also considers the adaptive fuzzy sliding mode controller in order to address the issue of performance degradation or loss of system stability.

1.3 Objective of the study

1.3.1 General objective

The main objective of this study is to develop an Adaptive fuzzy sliding mode controller for the trajectory tracking of Three-Link robot manipulator.

1.3.2 Specific objective

- To develop kinematics and dynamics of a Three-Link robot manipulator.
- To develop an SMC for the robot manipulating arm.
- To develop an AFSMC for the robot manipulating arm.
- To develop a tuning parameter for SMC using adaptive fuzzy logic.
- To simulate and demonstrate an AFSMC of a Three-Link robot manipulating arm
- To compare the AFSMC with SMC.

1.4 Contribution of this study

Major contributions of this research study are listed below:

- A comprehensive and comparative analysis of intelligent control strategies—the Sliding mode controller, Fuzzy Logic Controller, and the Adaptive Fuzzy Sliding mode

controller—is presented, specifically applied to a three-link robotic arm. According to comprehensive evaluations, the study emphasizes the precision and general efficacy of both controllers, especially in trajectory tracking.

- The proposed controllers are shown to exhibit better tracking performance and robustness, as confirmed by a thorough comparison with recently published controllers. Beyond merely tracking trajectories, this thorough analysis also looks at how responsive the controllers are to external disturbances, offering insightful information about their robustness and usefulness.

- It is shown that the AFSMC controller is efficient in tracking precise trajectories and regulating the robotic arm system effectively, even when there are load variations. This resilient and flexible performance of AFSMC has important ramifications for a variety of industrial contexts where the ability to adjust to load variations is essential for successful operations.

- This work contributes to the field of industrial robotics by offering a performance-index-based quantitative analysis. This methodology makes it easier to compare the controllers objectively, allowing for a precise and quantifiable understanding of their efficacy in real world situations. This method makes a substantial contribution to the creation of evaluation standards, which strengthens the controllers' applicability and relevance in cutting edge industrial settings.

1.5 Scope of the Thesis

This study mainly focused on developing trajectory tracking using an adaptive fuzzy sliding mode controller for the Three-Link robot manipulator in MATLAB/Simulink environment. Any other robot types and application areas are out of scope for this research. The thesis focus mainly on three degree of freedom for the industrial robot manipulator.

1.6 Methodology

The following steps have been taken in order to successfully design the suggested trajectory tracking and pose estimation:

- Establish the study's overall goal and define the research challenge.

- Review of the literature on three-degree-of-freedom robot manipulators.
- Drive a mathematical model of the three-degree of freedom robot manipulator's kinematic and dynamic equation.
- Model verification of kinematics and dynamics model.
- Design control strategies of three degree of fredom robot manipulator.
- Design three different types of controllers: Adaptive Fuzzy Sliding Mode Control(AFSMC), Fuzzy Sliding Mode Control(FSMC), and Sliding Mode Control(SMC).
- Verify the system's performance and resilience.

1.7 Structure of the Thesis

There are six chapters in this thesis, starting with an introduction. A thorough literature review covering earlier studies and explorations in the topic is given Chapter Two. The modeling and validation of the the industrial robot manipulator, which includes orthogonal transformations from the local reference frame to the global reference frame, are covered in Chapter 3. The dynamics and kinematics models are also analyzed in this chapter, and the accuracy of the models is verified using MATLAB software. The controller design are the main topics of Chapter 4. In particular, it describes how the model's adaptive fuzzy sliding mode control was developed. Chapter Five offers a discussion of simulation results, which assess the performance of the designed controllers. A meticulous analysis is conducted to gauge the control system's effectiveness and efficiency. The thesis conclusion, which summarizes the main conclusions and contributions of the research, is finally presented in Chapter 6. It also lists and talks about possible directions for further study and advancement.

CHAPTER 2

LITERATURE REVIEW

This chapter deals about the related works done on three degree of freedom robot manipulator.

2.1 Theoretical Background

2.1.1 Three Degree Freedom Robot Manipulator

A robot is a multifunctional, reprogrammable machine that can move components, materials, tools, or specialized equipment using variable motion that is programmed to accomplish various tasks. The right mix of the robot's actuator, manipulator, and controller is required to carry out the range of tasks. To effectively track the desired input, the robot end-effecter manipulator needs to choose the right actuator and controller (control mechanism).[8][9][10][11][12]

2.1.2 Controller and Algorithms

2.1.2.1 Fuzzy Logic Control

Principles of Fuzzy Logic: Developed by Lotfi Zadeh in 1965, fuzzy logic offers a mathematical framework for handling ambiguity. In FLC, system parameters and states are represented by language variables like "hot," "cold," "high," and "low." This variables are described by fuzzy sets, which capture the gradual transition between membership and non-membership[13].]. A Fuzzy Logic Controller (FLC) is an automated control system that utilizes fuzzy logic to replicate human thought processes and decision-making. FLC uses

linguistic variables and rules to work on imprecise and ambiguous knowledge, in contrast to the conventional method of control through clear mathematical models. Because they can readily capture the complex interactions between input and output variables that may prove difficult for traditional control techniques to handle, fuzzy logic controllers find the greatest use in nonlinearities, uncertainties, and imprecise information in systems. In order to provide a stable control strategy, designing an effective FLC requires carefully selecting the rules, membership functions, and linguistic concepts.

Fuzzy Logic Controller Architecture

- **Fuzzification**:- Fuzzification is the process of employing membership functions to transform clear input facts into fuzzy language concepts. These membership functions specify how much a value belongs to a fuzzy set.[14].

  • Defuzzification: -is the reverse of fuzzification, which is the process that turns fuzzy output into an abrupt control action. For defuzzification, the centroid or weighted average approach is usually employed.[14].

  - Fuzzy rule base: The fuzzy rule base is a collection of IF–THEN rules that forms the foundation of fuzzy reasoning, defining the system's behavior through its premises and corresponding consequents..[15].

  - Fuzzy Inference : The fuzzy rules are computed by the inference engine using the current input values. Determines each rule's firing power and the extent to which its requirements are met. The output of the rules is combined using a variety of techniques, such as the minimum, product, or maximum[16].

    FLC has been used in many different industrial applications. It has been used in quality control, robotics, and process control in industrial automation. Automatic transmissions, antilock brake systems, and engine control systems in automobiles have all employed FLC. Additional uses include decision support systems, consumer goods, and medical diagnosis.[14]. Advantages: When dealing with uncertain, imprecise, and non-linear systems, FLC performs exceptionally well. It is a useful tool in most fields due to its ability to handle complex correlations between variables, reactivity to changing conditions, and provision of easily understood control rules[17].

2.1.2.2 Sliding Mode Control

Because of its ability to handle complex and nonlinear systems, the robust control technique known as Sliding Mode Control (SMC) has attracted a lot of interest from the control systems community. Since its initial development by V. Utkin in the late 20th century[18], this control technique has become a powerful tool for managing a wide range of dynamic systems. Building a sliding surface, also called a manifold, in the system's state space is the concept behind SMC. The objective is to push the system's state trajectory onto this sliding surface and maintain it there in order to guarantee that the system operates as intended. Different system behaviors, such stability and instability, are divided by this sliding surface. The robustness of sliding mode control (SMC) against system uncertainties, disruptions, and changes in system parameters is one of its primary characteristics. This is accomplished through the use of a discontinuous control law, which dynamically adjusts the control input based on the sliding surface and the current state of the system. Even in the face of external disturbances or model uncertainty, this results in prompt and precise control reactions[19].

SMC is a useful tool in many applications, including robotics, because it is especially well-suited for systems with uncertain dynamics or those that are susceptible to external disturbances.[20], aerospace, automotive control[21], and power electronics[22**?**].

Sliding Phase: As seen in figure 2.1, the control system will work to push the system's state trajectory onto a chosen manifold, referred to as the sliding surface, during the sliding phase. This surface represents the desired equilibrium or behavior point. Following its impact on this sliding surface, the state trajectory precisely follows it, allowing for accurate tracking of the intended behavior. Since it ensures superior tracking performance, the sliding phase is when SMC operates at its best.

Reaching Phase:- As seen in figure 2.2, the system would normally proceed through the reaching stage before entering the sliding phase. To guide the system along the sliding surface at this point, the control law produces control inputs. Reaching the sliding surface in a certain amount of time is the goal. As the system gets closer to the sliding surface, transient dynamics control the reaching phase.

Chattering Phenomenon:- As illustrated in figure 2.1, chattering is one of the usual behaviors of SMC. Chattering happens when the control input quickly changes values in an effort to precisely maintain the trajectory on the sliding surface. The control signal may oscillate at high frequencies as a result of such quick switching. Chattering

can cause high-frequency noise and actuator wear even while it keeps the sliding surface tracked with great accuracy. Adding boundary layers to the sliding surface or employing switching control with hysteresis are two methods that have been suggested to lessen the negative impacts of chattering, a common problem in SMC[23]. Researchers are always looking for ways to reduce chattering without sacrificing SMC's accuracy and robustness[23]. [24]Explains sliding mode control's sliding order and sliding precision.

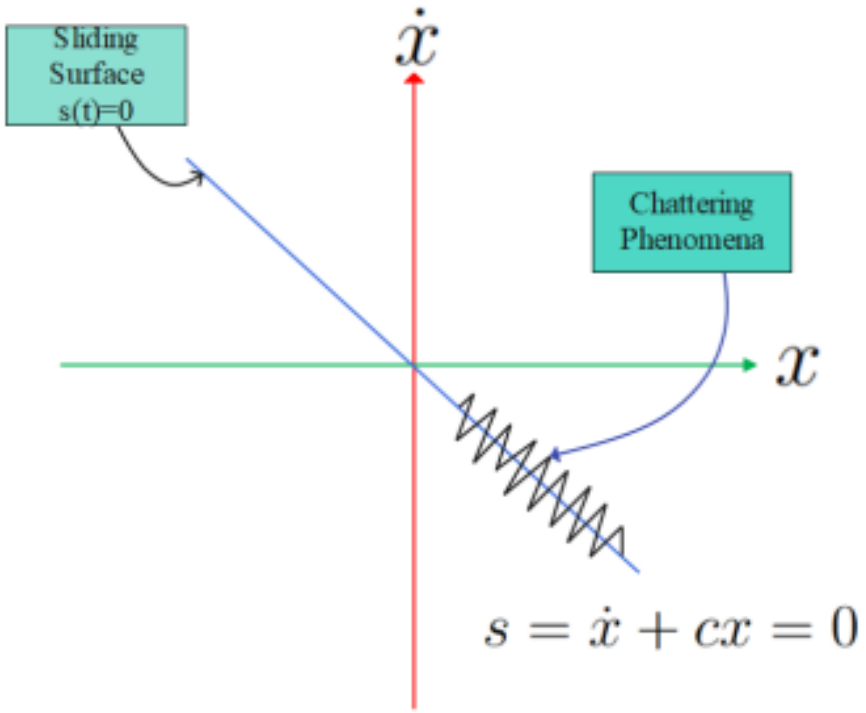

Figure 2.1: SMC chattering phenomenon

2.1.2.3 Adaptive System

Typically, FC incorporates the intuition and experience of a human operator. In a variety of applications, it has recently been employed as a supervisor. The fuzzy sliding mode controller (FSMC) with variable control gain is introduced in this section. To enhance controller performance, the reaching control gain Kf is adaptively tuned using an adaptive fuzzy sliding mode control system.

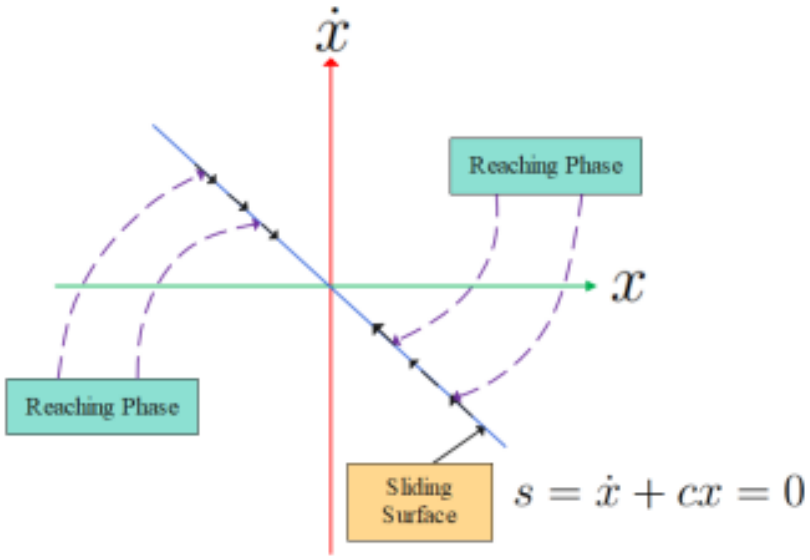

Figure 2.2: Sliding phase and reaching phase in SMC[25]

2.1.2.4 Particle Swarm Optimization

The group dynamics of social creatures, such as fish schools and bird flocks, serve as the basis for Particle Swarm Optimization (PSO), an optimization method inspired by nature [26]. Kennedy and Eberhart created it in 1995. As a reliable heuristic optimization method, PSO has grown in popularity [26]. Its appeal stems from its capacity to efficiently search complex search areas, making it suitable for resolving a wide variety of optimization issues. PSO has been used in many domains, including biology, economics, machine learning, and engineering[27]. It has been modified and expanded to address a number of optimization issues, including dynamic optimization, constrained optimization, and multi-objective optimization[28].

- Particles: Each particle is a feasible solution to the optimization problem..
- Swarm: A collection of particles that look for the optimal solution together.
- Velocity: each particle has a velocity that determines how and how quickly it moves in the solution space.
- Position: The current location of the particle in the solution space.
- Objective Function: A function that evaluates the quality of a potential solution.

How it Work

1. Initialization: An initial swarm of particles is created, each with a random position and speed.

2. Evaluation: The objective function is calculated for every particle, and its personal best is updated whenever a better solution is obtained.
3. Velocity update: Each particle's velocity is determined using its personal best, global best, and inertia weight, which is designed to bring exploration.
4. Position Update: Each particle's location is updated based on its velocity.
5. Iteration: Steps 2-4 are iterated until a proper stopping criterion is met.

2.2 Literature Reviews on 3 Link Robot Manipulator

For trajectory tracking control of robotic manipulators, several modeling and control approaches have been proposed in various studies. This section provides a survey of research focusing on trajectory-tracking control of robotic manipulators and other related controllers used in robot manipulator system control. Studies in the literature reviewed in this paper generally concentrated on system design and modeling for either linear or nonlinear controllers.

In research[29] For robot manipulators, the researchers proposed an integrated sliding control tracking mechanism. This shows how the proposed model addresses tracking issues with continuous sliding control for robot manipulators in the presence of outside disturbances and uncertainties. First, the researchers propose a control mode based on a sliding integral terminal that doesn't break. for which an observer is integrated with the sliding control. The researchers applied Lyapunov stability theory to demonstrate the global finite time of the robot controller system's tracking mode. The primary objective of the paper is to track robot manipulators in finite time while accounting for parametric uncertainties and potential external disturbances. To this end, an integral terminal sliding mode control, or ISMC, is presented.

In order to ensure that the errors of the siding surface and tracking process are initially suggested, the paper also suggests a tracking mode with a chattering free SMC. Unlike the current SMC models, the trajectory of the proposed model meets the sliding surface in a finite amount of time rather than the layer boundaries. Furthermore, the finite-time convergence of the proposed model ISMC is guaranteed in terms of its tracking activity. By contrasting two distinct controllers, the researchers conducted

a thorough analysis of the recommended controller system. which are two types of terminal sliding mode controls: fast terminal sliding mode control and continuous terminal sliding mode control.

Furthermore, in research[30], the researchers introduced a model that can solve the problem of robot manipulator motion control. The introduced model, which is based on a hierarchical multiloop control scheme, is designed and implemented in this paper. The primary elements of the suggested model are the multi-input and multi-output based on an inverse dynamic controlling mechanism and the integrated sliding control (ISC) model with model predictive control (MPC).The ISM is in charge of adjusting uncertainties when they arise from dynamics that are not understood or modeled. If the inverse dynamic method is unable to address those uncertainties, the ISM will typically take over. The external loop that the MPC is connected to ensures that the control system will evolve to become as efficient as possible while also meeting the constraints on the inputs and outputs simultaneously. The main justification for using the ISM, as stated by the researchers in their paper, is its robustness in handling a broad range of uncertainties. The ISM's capacity to implement and enforce a sliding mode of the control system starting at the time start is also the second primary factor. The system will be able to lessen and resolve the MPC optimization issues if it can enforce the sliding control from the very beginning.Using COMAU Smart3-S2, the researchers have replicated the testing and analysis of the nine models that have been presented. wherein this particular model functions as an industrial robot manipulator. The root mean square error (RMS) and RMS value of the models were used by the researchers to compare their proposed system MPC/ISM with the standalone MPC methods. The six-joint robot they were analyzing and testing their proposed strategies on was locked in three joints, and they only used the other three joints for their experiment. This was done for the sake of simplicity and the effectiveness of their strategies. They have clarified, though, that their recommended methodology can also be used with 6-joint robotic models and even be expanded for wider application.

The authors of [31] compare sliding mode control (SMC) and the traditional PID controller for robotic manipulators. While both methods perform well, the SMC has superior disturbance rejection and is insensitive to parametric changes. The simulation results of the study show that the performance of the SMC is better than PID under both sine and cosine trajectory. The PID controller's tracking error increases when the payload is altered or when there is uncertainty, but the SMC controller's performance

stays constant, demonstrating the SMC's superior robustness over the PID controller. For simplicity, the authors of this study assume that the robotic manipulator's joint can be subjected to arbitrary forces and that the actuators have no dynamics of their own. Although the results showed promise, low controller performance resulted from the authors' failure to take into account the actuator dynamics and joint friction effects. The authors of [32] provided a comparative analysis of PID controllers and suggested SMC for two-DOF robotic manipulator systems. The authors came to the conclusion that, when compared to the PID controller for sinusoidal trajectory tracking, the SMC controller performs better and is more reliable based on their study and simulation results. When the system is subjected to a disturbance of 25% of the input amplitude, the findings demonstrate a good performance response. The problem was that the second joint's sliding surface and its derivative caused chattering, which in turn caused chattering in the joint's control effort when the system disturbance increased to 40% of the input amplitude. The scientists concluded by recommending that additional chattering removal techniques be developed in subsequent research projects in order to potentially overcome any amount of disruption that is introduced into the robot. In a wide range of fields, including robotics, process control, and aerospace applications, a nonlinear SMC was used [33]. The performance of this controller was the primary factor in its use. Additionally, it resolves some of the most difficult problems with system control, like uncertainty and resistance to outside disturbances. Nevertheless, the chattering and sensitivity issue are two major drawbacks of this controller, making it extremely sensitive to noise. Similarly, Instead of using a signum function, the authors of [34] presented an SMC with saturation and hyperbolic tangent functions. Regretfully, this strategy only worked in certain situations. Certain issues are lessened when there are no hard uncertainties, but this comes at the expense of a reduction in robustness. Additionally, a nonlinear FSMC was employed to mitigate or offset the system's saturation function's detrimental impact. But because the sliding surface gain is determined by trial and error, which requires that the SMC and FSMC have prior knowledge of the systems uncertainty, both SMC and FSMC struggle to handle unstructured model uncertainties. Unlike other papers, this work overcomes the challenge of handling unstructured model uncertainties to improve the system's robustness by combining the fuzzy-based tuning method with FSMC. To achieve optimal system performance, the sliding surface slope ( ) and a switching gain (K) were adjusted using the fuzzy-based tuning method. We looked into the coefficients that significantly

affect the control system's discontinuous portion. The SMC can reject disturbances more effectively if these coefficients are adjusted appropriately. Consequently, the suggested controller (Self-tuning-FSMC) can address both structured and unstructured uncertainty issues and has strong resistance.[34]. The Lyapunov stability theorem was used in to illustrate the stability of a robotic manipulator system using an adaptive fuzzy sliding mode controller (AFSMC). In a similar vein, suggested an AFSMC to increase the trajectory tracking precision of a 3-DOF serial manipulator with parameter uncertainties. This paper suggests an adaptive fuzzy slide mode controller (AFSMC) with strong resistance to solve structured and unstructured uncertainty problems in order to overcome these difficulties. By considering actuator dynamics in the system to enhance the performance response, the paper also tackles the issue of performance degradation or loss of system stability brought on by neglected actuator dynamics. The authors mainly focused on reviewing the literature regarding the theoretical and structural foundation of the robotic manipulator and its controller in order to fulfill the predetermined goal of this paper. After gaining this knowledge, we compared the performance of different controllers in an effort to use the suggested controller (AFSMC) to overcome the drawbacks of conventional linear and nonlinear controllers. We have separated the model development phases into two primary sections since the right dynamic model equations are crucial for designing the robotic controller. We started by creating the dynamic equations for the robot manipulator. This begins with the description of the position and orientation equation, followed by forward and inverse kinematics, dynamic analysis and forces, and the Lagrange equation for kinematic and potential energy derivations. Following the development of the dynamic models, MATLAB was used to design the suggested controller (AFSMC) for a 3-DOF robot manipulator. Finally, MATLAB software simulation is used to assess and compare the effectiveness of the proposed method with the traditional controller. The simulation results are then analyzed and interpreted.

2.2.1 The Proposed Adaptive fuzzy Sliding Mode Control Systems

There are various benefits of combining adaptive controller, sliding mode control, and fuzzy logic control (FLC) in a three-link robot arm.

- **Robustness**:- Real-world robotic applications frequently involve uncertainties and imprecise information, which FLC excels at handling[35]. Known for its resilience to uncertainties and disturbances[23], SMC further improves the system's capacity to handle erratic outside influences.

- **Precise Tracking**:- The goal of adaptive fuzzy control is to precisely track desired trajectories[36]. It guarantees that the mobile robot follows its intended course with high precision when paired with FLC and SMC. Finally fuzzy sliding mode controller compile with the adaptive controller ensures to get the appropriate gain and improve the system performance.

- **Non-linearity Handling**:- The dynamics of mobile robots are nonlinear by nature. Nonlinear systems can be handled efficiently by FLC and SMC[8]. Adaptive FSMC control enhances the control system's ability to handle intricate nonlinearities.

- **Adaptability**:- FLC can adapt to changing conditions and uncertainties. When coupled with SMC, the robot can smoothly adapt to variations in its environment or the robot's own dynamics.

- **Reduced Control Effort**:- FLC can lessen the amount of control work needed to get the desired results[37]. It results in effective control with low energy usage when combined with adaptive smc.

CHAPTER 3

MATHEMATICAL MODELING OF THREE LINK ROBOT ARM

3.1 Introduction

This chapter thoroughly examines the mathematical model of a three-link robot arm manipulator. The study looks at three primary areas: the actuator model, which describes the robot's control system; the dynamics model, which accounts for the forces and torques influencing the robot's behavior; and the kinematics model, which describes the robot's mobility without accounting for these variables. The generated model is meticulously checked and verified by combining these three models using the flexible MATLAB program, guaranteeing its accuracy and efficacy in capturing the performance and behavior of the three-link robot arm manipulator. The first stage of controller design is modeling. It is essential to understand the mathematical formulas or the physical properties of the system that has to be controlled. The foundational components required to develop controllers for robotic arms are the kinematic model and the dynamic model. The study of motion, including position, velocity, and acceleration, as well as their derivatives, without considering the forces causing the motion, is known as kinematics. The study of dynamics establishes the connection between force and motion[38]. The general mathematical model of a three-link robot arm is analyzed and verified in this chapter.

3.2 Three Link Robot Arm Manipulator

In three dimensions, an object's location is a constant concern in the study of robotics. These items serve as connections between the manipulator, the components and instruments it works with, and additional environment-related items. The only two characteristics that define these objects are their orientation and position. We must always firmly affix a coordinate system, or frame, to an object in order to display its location and orientation in space. Next, we define this frame's orientation and position in relation to a reference coordinate system. A robot body's orientation and position can be expressed using any frame as a reference system. In order to enable movement between adjacent bodies, a robot is composed of multiple serially connected connections joined by joints. Usually, these joints have instrumentation and position controllers. These displacements are referred to as joint angles for rotary or revolute joints. Prismatic joints, also known as sliding joints, are found in some manipulators. These joints translate the relative displacements between links and are sometimes referred to as the joint offset.

An open kinematic chain is typically used as a manipulator in industrial robotics. The number of degrees of freedom is implied by the fact that each joint position is typically defined by a single variable. The end-effectors are at the free end of the chain of links that constitutes the manipulator. The robot's end-effectors could be a gripper, welding torch, electromagnet, or another tool, depending on the application. To explain the location of the manipulator, the tool frame, which is connected to the end effectors, is typically explained in terms of the base frame, which is connected to the manipulator's stationary base. Forward kinematics is a fundamental problem in the study of mechanical manipulation. Determining the orientation and position of the manipulator's end-effectors is a static geometrical problem[39].

This chapter uses a three-DOF robot arm for each of the three rotating joints to model a three-link robot manipulator. The robot's degree of freedom (DOF) is influenced by the number of joints and links, their types, and the robot's kinematics chain.

Link: is the rigid portion of the robot's body, such as its arm.

Joint: is a section of the body of the robot that permits two links to move in relation to each other freely or under control (connection element).

End effector: is the connector that runs from the manipulator to the tools(gripper,

spray gun, welding gun).

Base: is the connection that runs from the manipulator which is typically grounded to the global coordinate of the system.

Kinematic pair: is a pair of links whose relative motion is constrained by the joint between them.

Kinematics Chains: Chains of kinematics can be configured as mechanisms. The chain is closed when the ground link starts and stops it; otherwise, it is open.

Serial robots : An open kinematics chain is frequently used as the manipulator for serial robots. It is necessary to treat each joint independently.

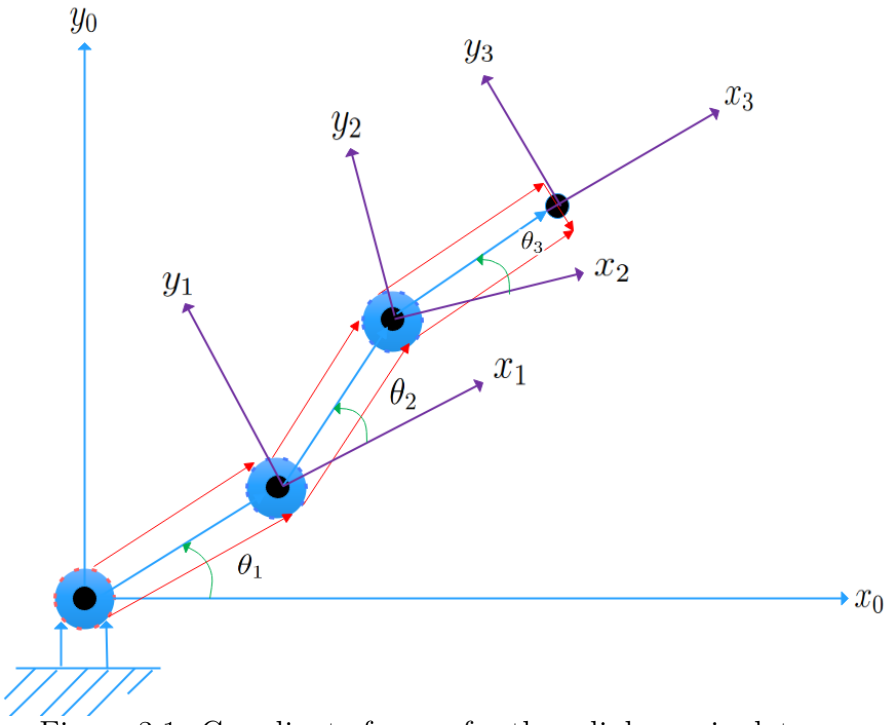

Figure 3.1: Coordinate frames for three-link manipulator

3.3 Kinematics and Dynamics of Robot Manipulator

To explain, analyze, and design the robot manipulator, one must have a solid understanding of its kinematics and dynamics. We will demonstrate and explain the motion, position, acceleration, and motion cause.

3.3.1 Robot Kinematics

Kinematics is the study of the motion, positions, and acceleration of a rigid body without considering the force or torque that causes these motions. As is well known, motion is a crucial element of robotic manipulators, and robot kinematics is an essential component of system analysis and design. Understanding the behavior of industrial manipulators requires knowing which kinematics models are suitable for a given robot mechanism.

There are two types of robot kinematics: forward kinematics and inverse kinematics. Figure 3.1[40] shows how forward and inverse kinematics relate to one another.

3.3.1.1 Forward kinematics of robot manipulator

Forward kinematics explains the relationship between spatial coordinates as a function of joint coordinates[41]. The equations for the forward kinematics problem are straightforward. This means that the forward kinematics solution of a manipulator is always feasible. The position and orientation of the robot manipulator's end effector can be determined and defined by computing the angle parameter of the designated mechanism. Serial linkages that are connected to each other at the base frame via prismatic or volumetric joints at the end-effectors comprise a manipulator. Forward kinematics is the process of identifying the location and orientation of the end-effector with respect to the joint variables. A suitable kinematics model must be used in order to have forward kinematics for a robot mechanism in a methodical manner. Finding the manipulator's homogeneous transformation matrix from the first link to the end effector connection is the first step in forward kinematics.

Figure 3.1 displays the Three-Link robotic arm's homogeneous transformation matrices, which are obtained as follows:

$$ {}_{i}^{i-1} = \mathbf{H}(d_i, Z_{i-1})\mathbf{H}(\theta_i, Z_{i-1})\mathbf{H}(a_i, X_{i-1})\mathbf{H}(\alpha_i, X_i) \tag{3.1} $$

$$ \mathbf{H}(d_i, Z_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$\mathbf{H}(\theta_i, Z_{i-1}) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{H}(a, x_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{H}(\alpha_i, x_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Given the i_{i-1}^{T} transformation matrices of one joint axes with respect to the preceding axes, any point in the i^{th} link can be related to the global reference frame using the transformation set below. In the global axis, its coordinates are u_i (n = Three-Link), and for $n = 3$, the following DH parameters are used: The graphic depiction provides a rapid and easily understood understanding of the robot's geometric configuration by emphasizing the axis arrangement and unambiguous definition of D-H parameters[42].

Link (i)	θ_i	d_i	α_i	a_i
1	θ_1	0	0	l_1
2	θ_2	0	0	l_2
3	θ_3	0	0	l_2

Table 3.1: DH parametres for the robotic arm with three links[43]

The T-matrices are thus given by

$$^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & \cos(\alpha_i) \cdot -\sin(\theta_i) & \sin(\alpha_i) \cdot \sin(\theta_i) & a_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cdot \cos(\theta_i) & -\sin(\alpha_i) \cdot \cos(\theta_i) & a_i \cdot \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

$$T_0^3 = T_0^1 T_1^2 T_2^3 \tag{3.3}$$

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & L_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & L_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$T_2^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$T_0^3 = \begin{bmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 & L_1\cos(\theta_1)+L_2\cos(\theta_1+\theta_2)+L_3\cos(\theta_1+\theta_2+\theta_3) \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 & L_1\sin(\theta_1)+L_2\sin(\theta_1+\theta_2)+L_3\sin(\theta_1+\theta_2+\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

The forward kinematics of the Three-DOF robotic arm shown in Figure 3.1 are described by the homogeneous transformation matrix given in equation 3.4. The relationship between the joint variables and the end effector position and orientation is established by this matrix. Specifically, the location of the end effector (x, y) is described as a nonlinear function of the joint variables by $P(x, y) = f(\theta)$. With the clarification of the forward kinematics of Figure 3.1, the location and orientation of the end effector can now be determined using the different joint angles $(\theta_1, \theta_2, and\theta_3)$.

$$x_3 = L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2) + L_3\cos(\theta_1+\theta_2+\theta_3) \tag{3.5}$$

$$y_3 = L_1\sin(\theta_1) + L_2\sin(\theta_1+\theta_2) + L_3\sin(\theta_1+\theta_2+\theta_3) \tag{3.6}$$

$$z_3 = 0 \tag{3.7}$$

The homogenous transformation matrix provided in 3.4 specifies the forward kinematics of the Three-Link robotic arm shown in Figure 3.1. This matrix indicates that the end effector's orientation and location are non-linearly dependent on the joint variables $P(x, y) = f(\theta)$. After determining the forward kinematics or direct kinematics of Figure 3.1, the end-effector position and orientation can now be ascertained from the individual joint angles ($\theta_1, \theta_2,$ and θ_3).

The Jacobian matrix can be used to calculate the robot's end effector velocity. The linear velocity (V_{n0}) and the angular velocity (ω_{n0}) for n number of joints make up the end effector velocity in the base frame.

$$\begin{bmatrix} V_{n0} \\ \omega_{n0} \end{bmatrix} = J\dot{\theta} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

The end effector's velocity is $[V_{n0}\ \omega_{n0}]$, its linear velocity from frame 0 to frame n in the base frame is V_{n0}, its rotational velocity from frame 0 to frame n in the base frame is ω_{n0}, the time derivative of each joint parameter is $\dot{\theta}$, and J is a Jacobian matrix.

The joint velocity $\dot{\theta}$ is converted into the end effector velocity using a matrix called the Jacobian matrix. Jacobian rank deficiency is used to represent singularity. This suggests that the joint angular velocities become infinite when the determinant of the Jacobian matrix component falls to zero. When the angular location of the third elbow joint θ_3 is 0° or 180°, how many inverse kinematics solutions are lost.

The Jacobian matrix J is defined as:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \end{bmatrix} \tag{3.8}$$

3.3.1.2 Inverse Kinematics Modeling

To find the inverse kinematics from the provided forward kinematics equations, we need to solve for the joint variables θ_1, θ_2, and θ_3 in terms of x and y [44].

Given the forward kinematics equations:

$$\begin{aligned} x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{aligned} \tag{3.9}$$

We can rewrite them as:

$$\begin{aligned} x &= l_1 \cos(\theta_1) + l_2(\cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2)) \\ &\quad + l_3(\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) - \sin(\theta_1)\sin(\theta_2)\cos(\theta_3) \\ &\quad - \cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - \sin(\theta_1)\cos(\theta_2)\sin(\theta_3)) \\ y &= l_1 \sin(\theta_1) + l_2(\sin(\theta_1)\cos(\theta_2) + \cos(\theta_1)\sin(\theta_2)) \\ &\quad + l_3(\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) + \cos(\theta_1)\sin(\theta_2)\cos(\theta_3) \\ &\quad + \sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + \cos(\theta_1)\cos(\theta_2)\sin(\theta_3)) \end{aligned} \tag{3.10}$$

Now, to solve for θ_1, θ_2, and θ_3, we can use inverse trigonometric functions. However, finding exact solutions may not always be possible due to the nonlinearity of trigonometric functions. It might require numerical methods or geometric interpretations.

Here are the equations for θ_1, θ_2, and θ_3 in terms of x and y:

$$\begin{aligned} \theta_1 &= \arctan 2(y - l_2 \sin(\theta_2) - l_3 \sin(\theta_2 + \theta_3), x - l_2 \cos(\theta_2) - l_3 \cos(\theta_2 + \theta_3)) \\ \theta_2 &= \pm \arccos\left(\frac{(x - l_1\cos(\theta_1))^2 + (y - l_1\sin(\theta_1))^2 - l_1^2 - l_2^2 - l_3^2}{2l_2 l_3}\right) \\ \theta_3 &= \pm \arccos\left(\frac{(x - l_1\cos(\theta_1) - l_2\cos(\theta_1 + \theta_2))^2 + (y - \theta_1\sin(\theta_1) - l_2\sin(\theta_1 + \theta_2))^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \end{aligned} \tag{3.11}$$

Note that the sign ambiguity in θ_2 and θ_3 is due to the presence of both $\sin(\theta_2)$, $\cos(\theta_2)$, $\sin(\theta_3)$, and $\cos(\theta_3)$ terms in the forward kinematics equations, which makes it impossible to uniquely determine θ_2 and θ_3 from x and y alone. The choice of sign depends on the specific configuration of the robot manipulator.

3.3.2 Dynamics Modeling of Three Degree Freedom Mobile Manipulator

The actuator's strength will be determined by the dynamic calculation that controls the link's movement; this requires an understanding of the relationship between torque, angular acceleration, mass, and inertia. By considering the external loads and the data obtained from the previously mentioned relation, we can design the proper robot dynamics.

Actuators with enough force and torque are required for an accelerated robot link in order to move robots to the proper and necessary velocity and acceleration. The link will not move and will not maintain the necessary positional accuracy if it does not receive the necessary torque and force.

For link 1, The potential energies and kinematics are provided by:

$$K_1 = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 \tag{3.12}$$

$$P_1 = -m_1 g l_1 \cos(\theta_1) \tag{3.13}$$

For link 2, the position and velocity of the center of mass are:

$$x_2 = l_1 \cos(\theta_1) + I_{\text{cm2}} \cos(\theta_1 + \theta_2) y_2 \qquad = l_1 \sin(\theta_1) + I_{\text{cm2}} \sin(\theta_1 + \theta_2)$$

$$\dot{x}_2 = -l_1 \dot{\theta}_1 \sin(\theta_1) - I_{\text{cm2}} (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) \dot{y}_2 \quad = l_1 \dot{\theta}_1 \cos(\theta_1) + I_{\text{cm2}} (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2)$$

(3.14)

(3.15)

The velocity squared is given by:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = \left(l_1^2 \dot{\theta}_1^2 + I_{\text{cm2}}^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2 l_1 I_{\text{cm2}} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1) \cos(\theta_2) + \sin(\theta_1) \sin(\theta_2) \right) \tag{3.16}$$

The second mass's kinetic energy is then:

$$K_2 = \frac{1}{2}m_2 v^2 + \frac{1}{2}I_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \tag{3.17}$$

The potential energy for link 2 is:

$$P_2 = -m_2 g(a_1 \sin(\theta_1) + I_{\text{cm2}} \sin(\theta_1 + \theta_2)) \tag{3.18}$$

The Lagrangian equation is given by:

$$L = K - P \tag{3.19}$$

First, the system's links one and two's kinetic energy are determined as follows:

$$K = K_1 + K_2 \tag{3.20}$$

And the kinematic energy is:

$$K = \frac{1}{2}m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 I_{\text{cm2}}^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 I_{\text{cm2}} \dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_2) + \frac{1}{2}m_1 I_{\text{cm1}} \dot{\theta}_1^2$$

The potential energy of link one and two of the system can be written as:

$$P = P_1 + P_2 = -m_1 g I_{\text{cm1}} \sin(\theta_1) - m_2 g(l_1 \sin(\theta_1) + I_{\text{cm2}} \sin(\theta_1 + \theta_2)) \tag{3.21}$$

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \tag{3.22}$$

$$M(\theta) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

and

$$g(\theta) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}.$$

$$a_{11} = m_1 ac_1^2 + m_2(a_1^2 + ac_2^2 + 2a_1 ac_2 c_2) + m_3(a_1^2 + a_2^2 + ac_3^2 + 2a_1 a_2 c_2 + 2a_1 ac_3 c_{23} + 2a_2 ac_3 c_3) + I_1$$

$$a_{12} = m_2(ac_2^2 + a_1 ac_2 c_2) + m_3(a_2^2 + ac_3^2 + a_1 a_2 c_2 + a_1 ac_3 c_{23} + 2a_2 ac_3 c_3) + I_2 + I_3$$

$$a_{13} = m_3(ac_3^2 + a_1 ac_3 c_{23} + a_2 ac_3 c_3) + I_3$$

$$a_{22} = m_2 ac_2^2 + m_3(a_2^2 + ac_3^2 + 2a_2 ac_3 c_3) + I_2 + I_3$$

$$a_{23} = m_3(ac_3^2 + a_2 ac_3 c_3) + I_3$$

$$a_{33} = m_3 ac_3^2 + I_3$$

$$b_1 = -m_2 a_1 ac_2 \dot{\theta}_1^2 + \dot{\theta}_2^2 \sin^2\theta_2 - m_3(a_1 a_2 \dot{\theta}_1^2 + \dot{\theta}_2^2 + a_1 ac_3 \dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_1\dot{\theta}_2)^2$$

$$b_2 = -m_2(a_1 ac_2 \dot{\theta}_1^2 + a_1 ac_2 \dot{\theta}_1\dot{\theta}_2 \sin^2\theta_2) - m_3(a_1 a_2 \dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2 + a_1 ac_3 \dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1\dot{\theta}_3 + \dot{\theta}_2^2 + \dot{\theta}_2\dot{\theta}_3 +$$

$$b_3 = -m_3(a_1 ac_3 \dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1\dot{\theta}_3 + a_2 ac_3 \dot{\theta}_1^2 + \dot{\theta}_1 + \dot{\theta}_2 \sin^2\theta_3 + a_2 ac_3 \dot{\theta}_1\dot{\theta}_3 + a_2 ac_3 \dot{\theta}_1 + \dot{\theta}_2 \sin^2\theta_3 + \dot{\theta}_1$$

$$g_1 = g[c_1(m_1 ac_1 + m_2 a_1 + m_3 a_1) + c_{12}(m_2 ac_2 + m_3 a_2) + c_{123}(m_3 ac_3)]$$

$$g_2 = g[(m_2 ac_2 + m_3 a_2)c_{12} + m_3 ac_3 c_{123}]$$

$$g_3 = g(m_3 ac_3 c_{123})$$

Table 3.2: Physical parameter of the three-link manipulator

Parameter	Notation	Value
Link length 1	L1	1m
Link length 2	L2	1m
Link length 3	L3	1m
Mass of Link 1	M1	10kg
Mass of Link 2	M2	10kg
Mass of Link 3	M3	10kg
Link 1 center of mass	Lc1	0.5m
Link 2 center of mass	Lc2	0.5m
Link 3 center of mass	Lc3	0.5m
Inertia of link 1	I1	125Kg*m^2
Inertia of link 2	I2	125Kg*m^2
Inertia of link 3	I3	125Kg*m^2
Gravitation acceleration	g	9.81m/sec^2

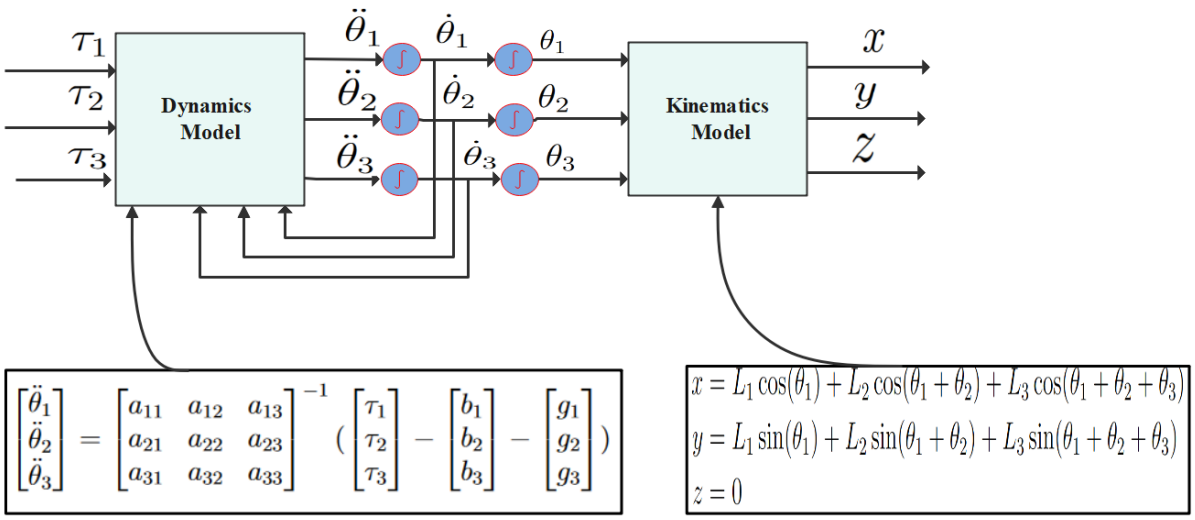

Figure 3.2: Block diagram of model verification of three link mobile manipulator

3.4 Model Verification

As we know, the equations of motion for a three-link robotic manipulator form a complex system of nonlinear, coupled, first-order differential equations. Due to the strong coupling between the joints, it is challenging to clearly determine how a specific input torque affects its corresponding joint angle output. To address this complexity, the model can be validated under specific operating conditions, particularly at equilibrium positions where the system is in steady-state.

To achieve this, we identify a trim point—an equilibrium configuration where the robot maintains a constant pose or moves at constant velocity without acceleration. At this trim point, the system dynamics can be simplified and approximated. Once the trim condition is established, the model can be decoupled into smaller, more manageable subsystems corresponding to individual joint motions.

For instance, consider a steady-state configuration where the robot starts from rest (initial joint velocities and accelerations are zero), with predefined torque inputs that maintain a fixed pose or trajectory. The longitudinal motion components may correspond to the torque and position of joint 1 (base rotation), while lateral components correspond to joints 2 and 3 (arm movement). Although the joints are mechanically coupled, under trim conditions the interaction between joints is minimized and the system can be analyzed more effectively. For validation, we trim the robot at a specific steady-state pose with an initial joint configuration of zero angles (θ_1=θ_2=θ_3=0) and apply a set of constant torques that maintain this pose or initiate a uniform move-

ment. This approach enables us to evaluate and validate the model under controlled and predictable conditions.

- Intial joint angles: $\theta_1(0) = 30°$, $\theta_2(0) = 40°$, $\theta_3(0) = 20°$
- Initial joint velocities: $\dot{\theta}_1(0) = \dot{\theta}_2(0) = \dot{\theta}_3(0) = 0$
- External input torques: $\tau = 0$ for all joints over the entire simulation time (0 to 10 seconds)
- **Input Torque =0.**

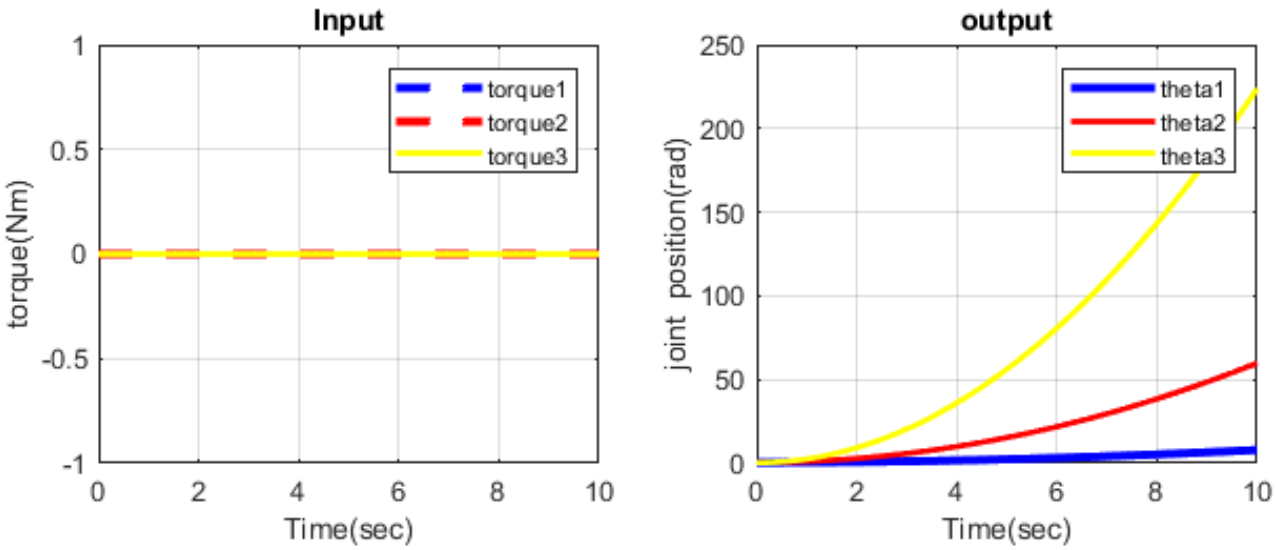

Figure 3.3: Input Torque zero

This simulation confirms that the implemented dynamic model captures the gravitational effects realistically. The observed joint deviations under zero torque reflect the physical influence of potential energy changes in the links. Therefore, the model is valid in representing passive behavior under initial conditions and zero input.

– **Input Torque 20Nm.**

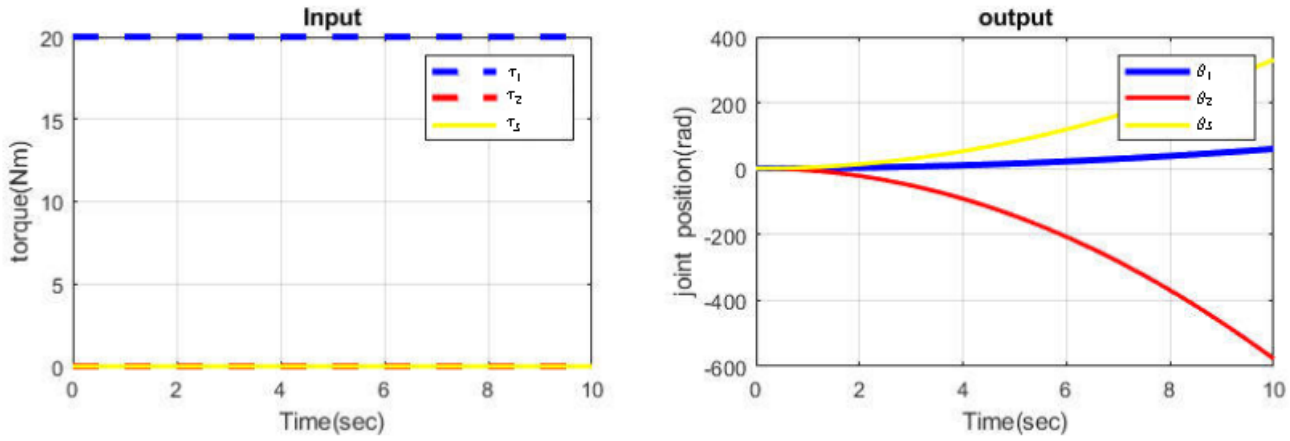

Figure 3.4: Input Torque zero

In 3.4 , a constant torque of $20Nm$ was applied only to Joint 1, while torques on Joints 2 and 3 remained at zero. The resulting joint positions, shown in the right plot, indicate that θ_1 increased steadily due to the applied input, demonstrating the expected response. However, θ_2 still declined and θ_3 increased, despite not receiving torque, confirming dynamic coupling between the joints. Compared to the previous zero-input case, the effect of torque1 not only influenced Joint 1 but also propagated to the other joints through inertial and gravitational interactions. This result validates the model's ability to represent coupled dynamics realistically.

– **Input Torque -20Nm.**

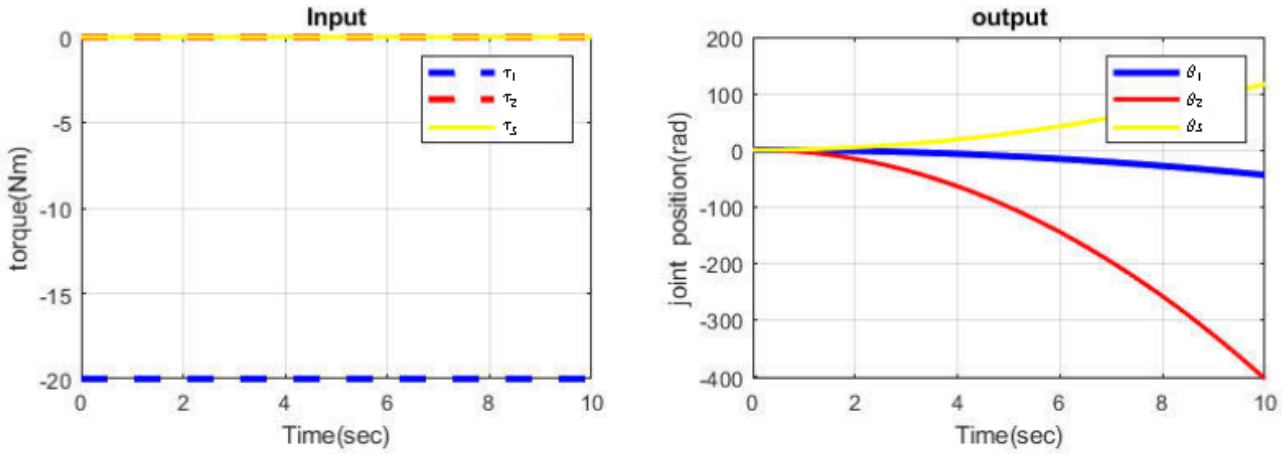

Figure 3.5: Input Torque -20NM

In figure 3.5 , a constant torque of –20$Nm$was applied to Joint 1, while torques on Joints 2 and 3 remained zero. As shown in the simulation output, θ_1 now decreases over time—opposite to the previous case where +20Nm was applied—confirming the model's correctness in directional torque response. Meanwhile, θ_2 continues to decline, and θ_3 increases, similar to the earlier simulations, indicating their behavior is dominated by gravitational effects.

CHAPTER 4

CONTROLLER DESIGN

4.1 Introduction

controller design using the kinematics and dynamics model of the three link robot arm is presented in this chapter. As part of the design, a fuzzy sliding mode controller is being created. Additionally provided for comparison is an Adaptive fuzzy Sliding Mode Controller.

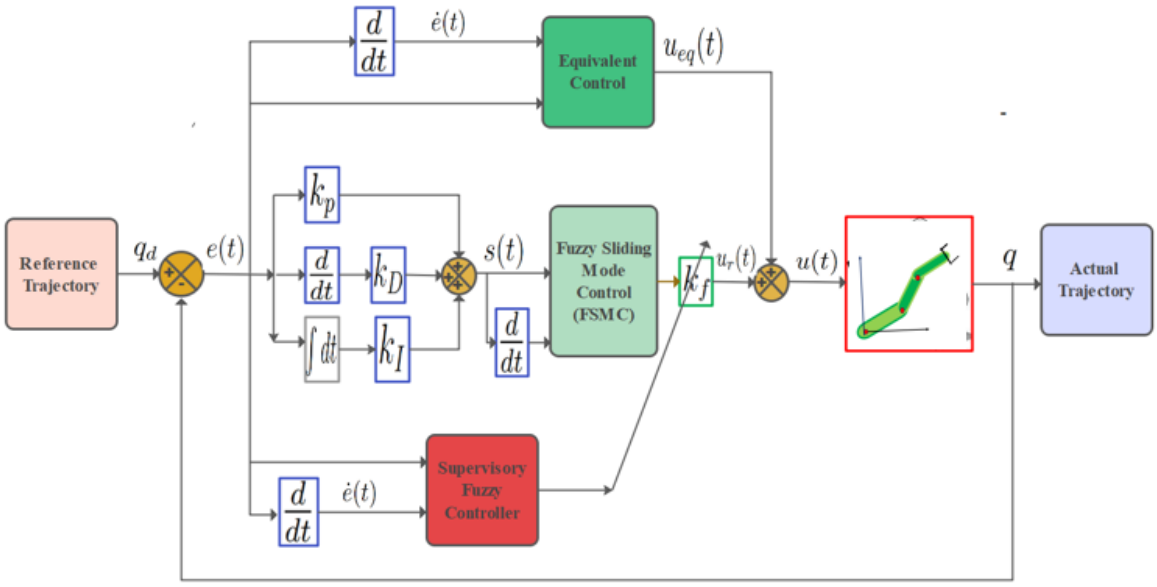

Figure 4.1: General block diagram of the system

4.2 Sliding Mode Controller Design for three-DOF Manipulator

This section uses the SMC approach to set the auxiliary tracking error of the dynamic tracking controller to zero.

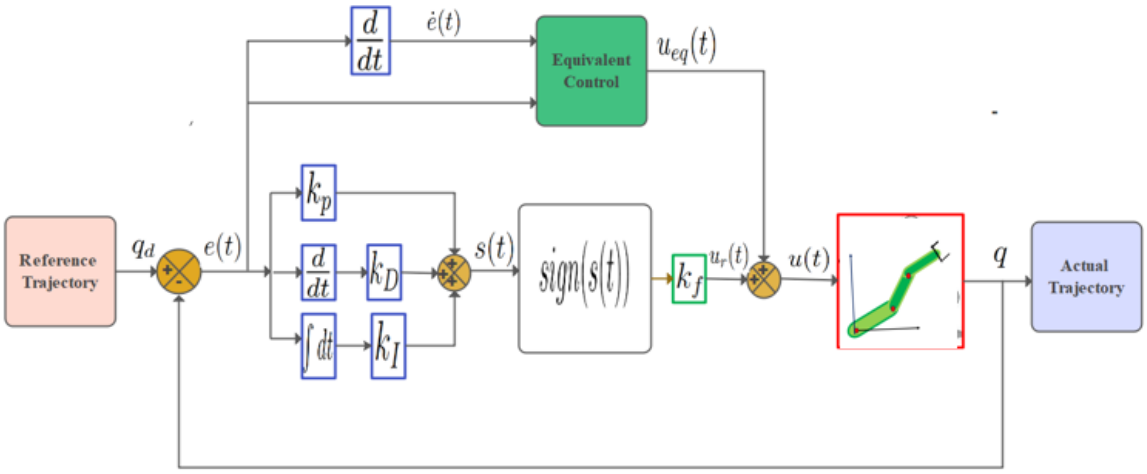

Figure 4.2: Schematic diagram used in the implementation of the SMC controller.

When dealing with nonlinear systems that have parametric uncertainties and disturbances, sliding mode control can offer reliable control. The control law is based on the well-known mathematical model of the nonlinear system, and the controller is designed in two steps. The first step in limiting the states to the sliding surface with the necessary dynamics is choosing the sliding surface (or manifold). The second step involves developing a control law that forces the states in the direction of the sliding surface and keeps them there. Examine the single-input nonlinear system that follows.
Steps to Design Sliding Mode Controller for Three-Degree-of-Freedom Manipulator

Selecting Sliding Surfaces

The sliding surface equation is given by:

$$s = \dot{e} + k_i \int_0^{\infty} e\, dt + k_p e \tag{4.1}$$

Differentiating the above equation gives:

$$\dot{s} = \ddot{e} + k_i e + k_p \dot{e} \tag{4.2}$$

Where:

$$e = q_d - q \tag{4.3}$$

$$\dot{e} = \dot{q}_d - \dot{q} \tag{4.4}$$

$$\ddot{e} = \ddot{q}_d - \ddot{q} \tag{4.5}$$

Substituting position, angular, and acceleration errors into equation (4.2), we get:

$$\dot{s} = 0 = k_p \dot{e} + k_i e + k_d(\ddot{q}_d - \ddot{q}) \tag{4.6}$$

Solving the above equation for $\ddot{q}$:

$$\ddot{q} = \frac{1}{k_d}(k_p \dot{e} + k_i e + k_d \ddot{q}_d) \tag{4.7}$$

Substituting the above equation into the dynamics equation:

$$\tau_{eq} = M\ddot{q} + c + g \tag{4.8}$$

After substitution, we get the equivalent controller τ_{eq}:

$$\tau_{eq} = M\left(\frac{1}{k_d}(k_p \dot{e} + k_i e + k_d \ddot{q}_d) + c + 9\right) \tag{4.9}$$

The discontinuous controller is given by:

$$\tau_{dis} = k\,\mathrm{sign}(s) \tag{4.10}$$

The total control law (u) for the three-degree-of-freedom robot manipulator is given by:

$$u = \tau = u_{eq} + u_{dis} = \tau_{eq} + \tau_{dis} \tag{4.11}$$

Finally, the control for the three-degree-of-freedom mobile robot is given by:

$$u = \tau = M\left(\frac{1}{k_d}(k_p\dot{e} + k_i e + k_d\ddot{q}_d)\right) + c + g + k\,\text{sign}(s) \tag{4.12}$$

Where k_p, k_i, k_d, and k are matrices given by:

$$k_p = \begin{bmatrix} k_{p11} & 0 & 0 \\ 0 & k_{p22} & 0 \\ 0 & 0 & k_{p33} \end{bmatrix} \qquad k_i = \begin{bmatrix} k_{i11} & 0 & 0 \\ 0 & k_{i22} & 0 \\ 0 & 0 & k_{i33} \end{bmatrix}$$

$$k_d = \begin{bmatrix} k_{d11} & 0 & 0 \\ 0 & k_{d22} & 0 \\ 0 & 0 & k_{d33} \end{bmatrix} \qquad k = \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_{33} \end{bmatrix}.$$

$\tau = [\tau_1 \quad \tau_2 \quad \tau_3]^T = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta)$

4.2.1 Stability Analysis of SMC for three-DOF Manipulator

$$V(s) = \frac{1}{2}s^T M s \tag{4.14}$$

where M is the positive definite moment of inertia matrix.

Positive Definiteness of $V(s)$

Since M is positive definite, $s^T M s$ is positive definite. Hence, $V(s)$ is positive definite, i.e.,

$$V(s) > 0 \text{ for } s \neq 0 \tag{4.15}$$

and

$$V(s) = 0 \text{ if and only if } s = 0. \tag{4.16}$$

Time Derivative of $V(s)$

Differentiate $V(s)$ with respect to time:

$$\dot{V}(s) = \frac{d}{dt}\left(\frac{1}{2}s^T M s\right) \tag{4.17}$$

Applying the product rule:

$$\dot{V}(s) = \frac{1}{2}\left(\dot{s}^T M s + s^T M \dot{s}\right) \tag{4.18}$$

Since M is symmetric:

$$\dot{V}(s) = s^T M \dot{s} \tag{4.19}$$

Substitute $\dot{s}$ from the sliding mode control.

From the sliding mode control design, we have:

$$\dot{s} = \ddot{e} + k_p \dot{e} + k_i e \tag{4.20}$$

Substitute $\dot{s}$ into the expression for $\dot{V}(s)$:

$$\dot{V}(s) = s^T M(\ddot{e} + k_p \dot{e} + k_i e) \tag{4.21}$$

Control Law and System Dynamics

The control law is given by:

$$\tau = \tau_{eq} + \tau_{dis} \tag{4.22}$$

where:

$$\tau_{eq} = M\left(\frac{1}{k_d}\left(k_p \dot{e} + k_i e + k_d \ddot{q}_d\right)\right) \tag{4.23}$$

$$\tau_{dis} = k\,\mathrm{sign}(s) \tag{4.24}$$

The dynamics equation is:

$$\tau = M\ddot{q} + c + g \tag{4.25}$$

Substitute this into the dynamics equation:

$$M\ddot{q} = \tau - c - g \tag{4.26}$$

Thus:

$$\ddot{q} = \frac{1}{k_d}\left(k_p\dot{e} + k_i e + k_d\ddot{q}_d\right) + \frac{1}{M}\left(k\,\mathrm{sign}(s) - c - g\right) \tag{4.27}$$

Substitute $\ddot{q}$ into $\ddot{e}$:

$$\ddot{e} = \ddot{q}_d - \ddot{q} \tag{4.28}$$

Thus:

$$\ddot{e} = \ddot{q}_d - \left(\frac{1}{k_d}\left(k_p\dot{e} + k_i e + k_d\ddot{q}_d\right) + \frac{1}{M}\left(k\,\mathrm{sign}(s) - c - g\right)\right) \tag{4.29}$$

Simplify:

$$\ddot{e} = \frac{1}{k_d}\left(k_d\ddot{q}_d - k_p\dot{e} - k_i e\right) - \frac{1}{M}\left(k\,\mathrm{sign}(s) - c - g\right) \tag{4.30}$$

So:

$$\dot{s} = \ddot{e} + k_p\dot{e} + k_i e \tag{4.31}$$

Substitute $\ddot{e}$:

$$\dot{s} = \frac{1}{k_d}\left(k_d\ddot{q}_d - k_p\dot{e} - k_i e\right) + k_p\dot{e} + k_i e \tag{4.32}$$

Simplify:

$$\dot{s} = \frac{1}{k_d}k_d\ddot{q}_d + \left(k_p - \frac{1}{k_d}k_p\right)\dot{e} + \left(k_i - \frac{1}{k_d}k_i\right)e \tag{4.33}$$

Substitute into $\dot{V}(s)$

Substitute $\dot{s}$ into $\dot{V}(s)$:

$$\dot{V}(s) = s^T M \left(\frac{1}{k_d} k_d \ddot{q}_d + \left(k_p - \frac{1}{k_d} k_p \right) \dot{e} + \left(k_i - \frac{1}{k_d} k_i \right) e \right) \tag{4.34}$$

Since:

$$\dot{V}(s) = s^T M \left(-\frac{1}{M} \left(k \, \mathrm{sign}(s) - c - g \right) \right) \tag{4.35}$$

If $k > c + g$:

$$\dot{V}(s) = -\frac{1}{M} \left(k \, \mathrm{sign}(s) - c - g \right)^T \left(k \, \mathrm{sign}(s) - c - g \right) \tag{4.36}$$

Since $k \, \mathrm{sign}(s) - c - g$ is positive when $k > c + g$, it follows that $\dot{V}(s)$ is negative definite.

The Lyapunov function $V(s) = \frac{1}{2} s^T M s$ is positive definite, and its time derivative $\dot{V}(s)$ is negative definite. Therefore, the system leads to globally asymptotically stable.

4.3 Fuzzy Sliding Mode Controller Design

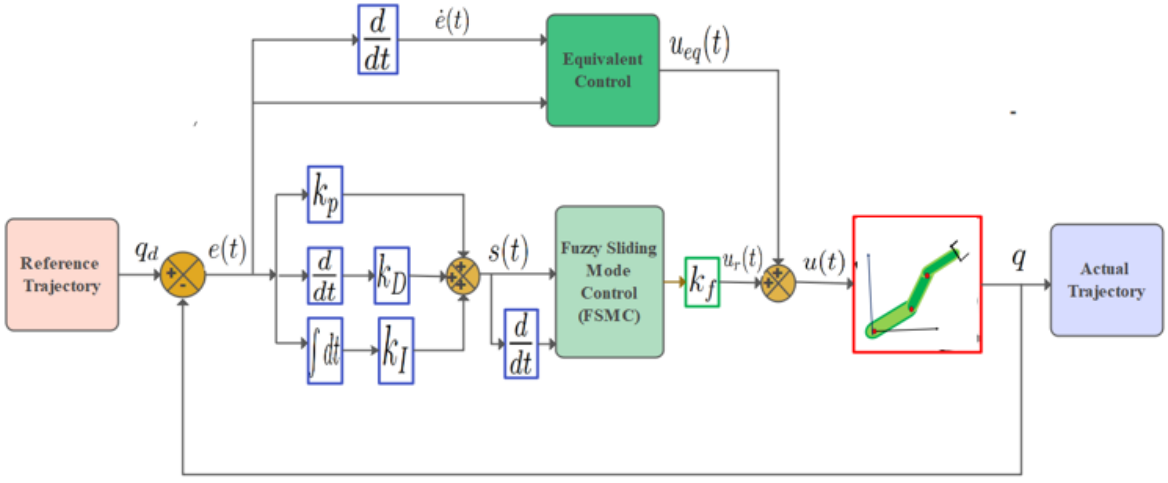

Figure 4.4: Schematic diagram used in the implementation of the FSMC controller.

Fuzzy Sliding Mode Control (FSMC) is a hybrid control technique that combines the robustness of Sliding Mode Control (SMC) with the flexibility of Fuzzy Logic Control (FLC). While traditional SMC is effective in handling system uncertainties and disturbances, it suffers from a major drawback known as chattering—high-frequency

oscillations in the control signal caused by the discontinuous switching near the sliding surface. FSMC addresses this issue by using fuzzy logic to generate a smooth control action, reducing chattering while preserving the robustness and stability properties of SMC. This makes FSMC particularly suitable for controlling nonlinear and uncertain systems where smooth control performance is essential. Chattering is undesirable in practical applications because it can result in excessive mechanical wear, higher energy consumption, and audible noise[**?**] [45]. Another strategy to lessen the chatterig phenomenon is to combine FLC with SMC[46]. Therefore, a fuzzy logic controller with sliding mode control is adopted in this thesis. When FLC was used to replace the discontinuous portion of SMC, the overall control law looked like figure 4.5 :

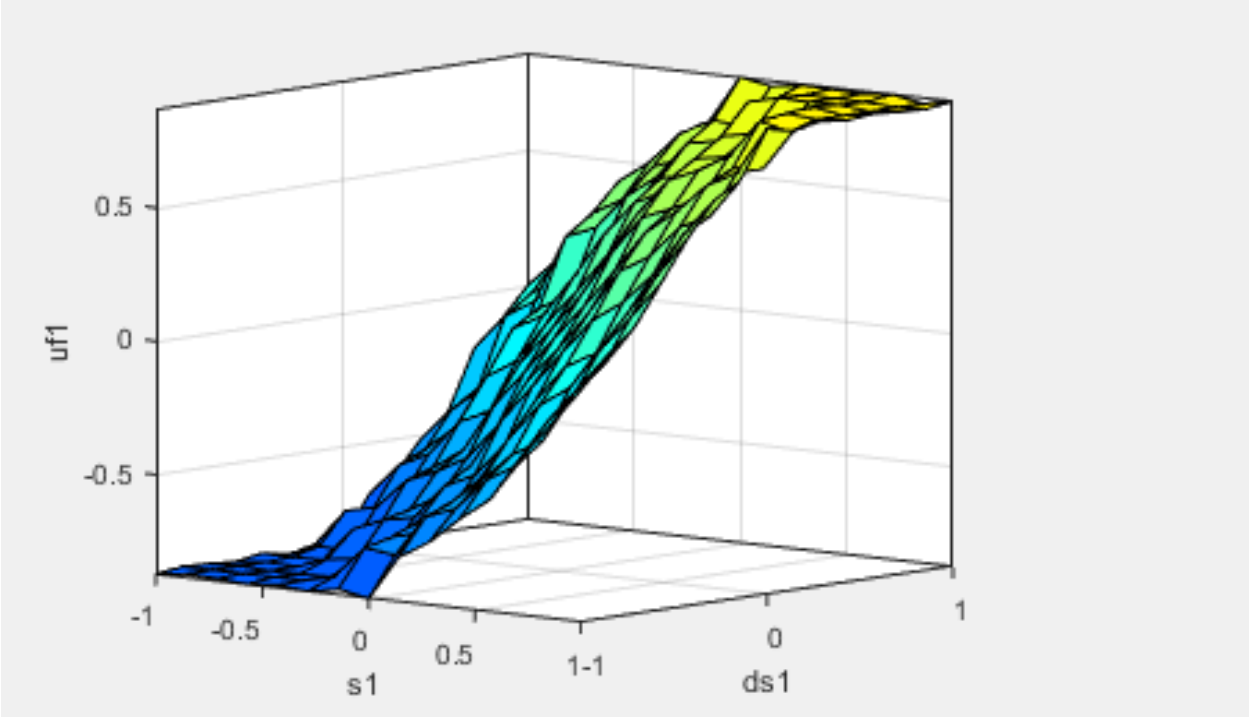

Figure 4.5: Defuzzified surface

$$u(t) = u_{eq}(t) + K_f u_f(t) \tag{4.37}$$

where as u_f is the output of fuzzy logic control, and k_f is output gain of fuzzy logic. So total control law incase of three degree freedom becomes

$$u = \tau = M\left(\frac{1}{k_d}(k_p\dot{e} + k_i e + k_d\ddot{q}_d)\right) + c + g + k_f u_f \tag{4.38}$$

Fuzzification it is defined as the process of acquiring an input variable value (e.g., $e(t)$) and determining the numerical values given for variables in the membership function

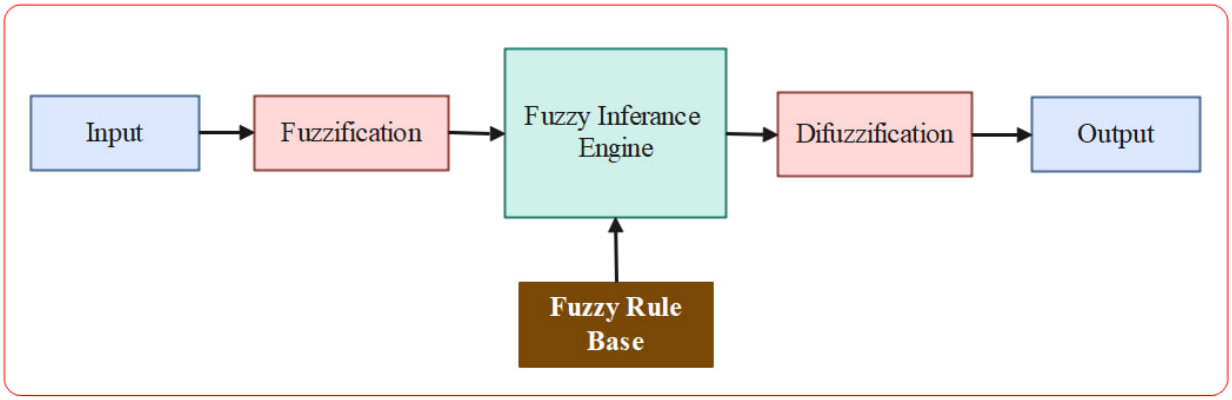

Figure 4.6: Fuzzy logic controller architecture

[47]. This thesis uses seven triangle membership functions to fuzzify the sliding surface, $s(e,t)$, for simplicity's sake.

Estimating Rules (Knowledge Base) The control rules should be designed in such a way that the state's trajectory slides across and bends into the sliding surface on the phase plane in order to meet the existence criteria. This thesis evaluates the fuzzy rule using the Mamdani fuzzy inference approach. Thus, fuzzy inference is a way to use the rule base to translate membership values from the input to the output. Lastly, to assess fuzzy logic set numerical values of ufuzzy, transformation after inference and defuzzification should also be used. **T**he process of transferring a set of inferred fuzzy control signals that are confined within a fuzzy output window to a non-fuzzy (crisp) control is known as "defuzzification." The defuzzification process is also carried out using the Center of Gravity (COG) algorithm, which is expressed as follows:

$$\text{Crisp control signal} = \frac{\text{Sum of first moment area}}{\text{Sum of areas}} \tag{4.39}$$

for countineous system

$$u_{fuzzy} = \frac{\int u_{\mu}(u)du}{\int \mu(u)du} \tag{4.40}$$

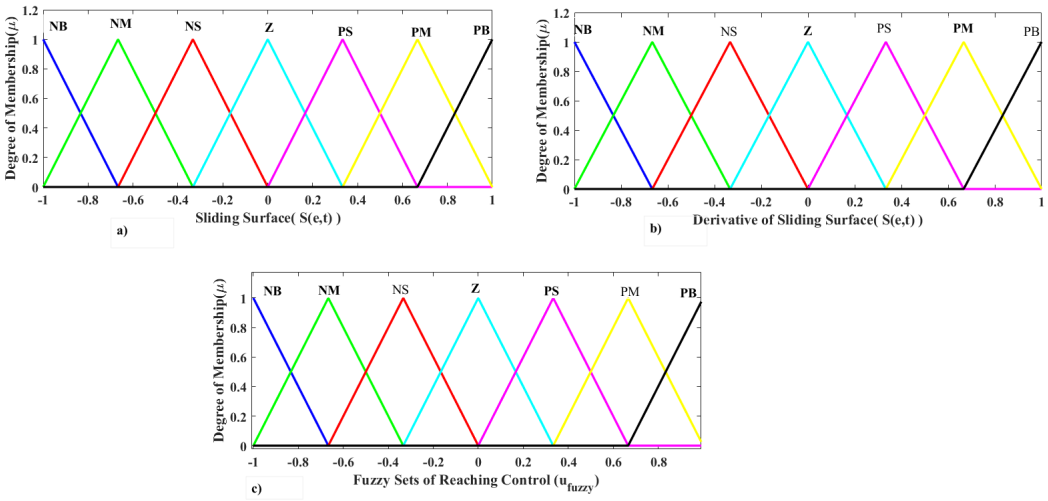

Figure 4.7: Input output membership function of FSMC

Table 4.1: Rule Base for FSMC of U_{fuzzy}

S \ $\dot{S}$	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

4.4 Adaptive Fuzzy Sliding Mode Controller Design

Fuzzy Sliding Mode Controller (FSMC) with Variable Control Gain

A human operator's experience and intuition are frequently incorporated into FC. It has recently been used in a number of applications as a supervisory tool. A fuzzy sliding mode controller (FSMC) with a variable control gain is presented in this section. Fig. 4.8 shows the general layout of the suggested controller. In particular, to improve controller performance, a supervisory fuzzy inference system adaptively adjusts the reaching control gain K_f.

Using IF-THEN rules and operator knowledge, the supervisory fuzzy system in the suggested tuning method determines the control gain K_f based on the current operating conditions of the controlled system. The supervisory fuzzy system's control rules

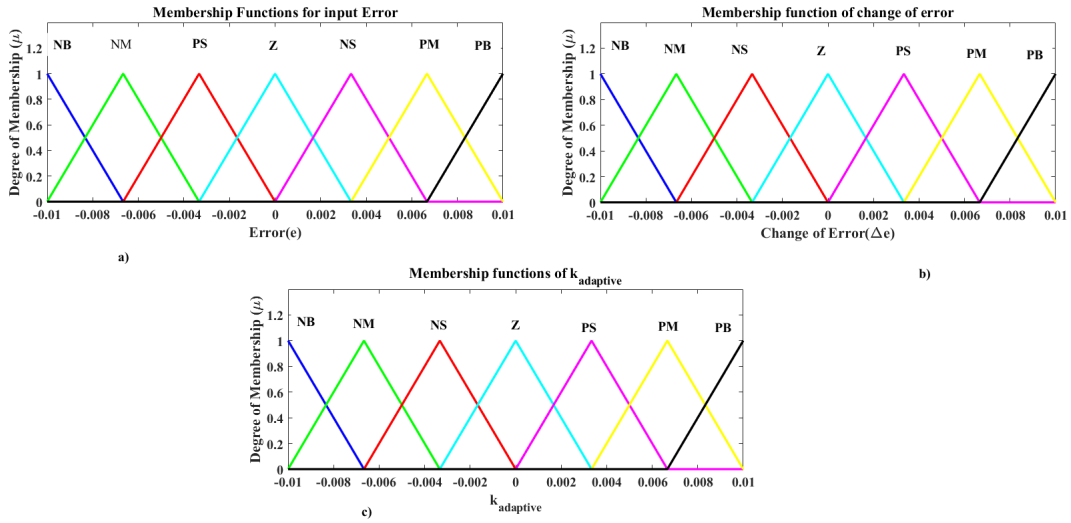

Figure 4.8: Input output membership function of AFSMC

use the error e and its derivative $\dot{e}$ as premises, and the consequent of each rule is $K_f = \text{diag}\{kf1, kf2, \ldots, kfn\}$. The fuzzy control surface of K_f is shown in Fig. 4.8, which also shows the triangular membership functions for the inputs and outputs fuzzy variables with 50% overlap. Online adaptive tuning of $K_f = \text{diag}\{kf1, kf2, \ldots, kfn\}$ is done using this surface. The output domain (for $K_f = \text{diag}\{kf1, kf2, kf3\}$, with 3 degrees of freedom, $n = 3$), ranges from 4000 to 7500, while the physical domain of the inputs $(e, \dot{e})$ ranges from -0.01 to 0.01.

The fuzzy variables for the rule base are defined as follows: $(e, \dot{e})$ = {NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PN (Positive Medium), PB (Positive Big)}.

$$u = \tau = M\left(\frac{1}{k_d}(k_p\dot{e} + k_i e + k_d\ddot{q}_d)\right) + c + g + k_{adaptive}u_f \tag{4.41}$$

4.5 Particle Swarm Optimization

Bounded AFSMC is known for its robustness in handling disturbances and uncertainties, making it a valuable choice for a wide array of practical control applications. However, setting the controller gains correctly is crucial to obtaining the best performance in AFSMC. Traditional tuning techniques, like trial and error or manual adjustment, can be laborious, ineffective, and may not follow a methodical process. To solve this issue and ensure that AFSMC operates at its best, sophisticated optimization techniques are applied. Particle Swarm Optimization (PSO) is one such technique that has shown promise in optimizing the controller gains for both AFSMC and FSMC.

4.5.1 How to Use MATLAB-Simulink for PSO-Based Control Gain Tuning

Step 1: create model :Create or import a Simulink model of a control system. Add the gains or parameters that need to be changed .

Step 2: Pso code: Use MATLAB to put the PSO algorithm into practice. Parameterize the algorithm with parameters like :

- Specification of lower and upper bounds for all gains or parameters to adjust .
- Calculating the number of particles in the swarm.
- Declaring the number of iterations for the PSO algorithm.
- Selecting the Integral of Time-weighted Absolute Error (ITAE) as the fitness function.

Step 3: A fitness function in MATLAB calculates the ITAE by simulating a Simulink model with given control parameters. The model computes the error signal and integrates t e(t) t e(t) over time using trapz. This function helps optimize controller gains using methods like Genetic Algorithm.

Step 4: To integrate MATLAB and Simulink for PSO-based control system optimization, use the sim command in MATLAB to run Simulink models with parameters (Kp, Ki, Kd) assigned to the base workspace via assignin. A fitness function is

created in MATLAB that runs the simulation and computes a performance metric, such as ITAE or ISE, based on the output response. This function is then used within a PSO loop or MATLAB's particleswarm function to iteratively evaluate and optimize the control gains. Bounds and optimization settings are defined to guide the search. The Simulink model should be set up to use workspace variables and provide output signals for cost computation. This integration allows automated tuning of controllers for improved system performance.

Step 5: PSO Optimization Loop: The PSO algorithm simulates the control system in Simulink to determine the fitness of each particle's position in each iteration.

Step 6: Particle Position Updates: Using the PSO algorithm update equations, update the particle positions in the swarm according to fit values. Modify the parameters or gains of Fuzzy Logic Control Fuzzy Logic Control (FLC), Sliding Mode Control Sliding Mode control (SMC), and Adaptive Control.

Step 7: Iterative optimization using PSO involves repeatedly adjusting controller gains to minimize the ITAE by simulating system responses. Each particle in the swarm represents a candidate solution, and through multiple iterations, particles update their positions based on personal and global best performances. The fitness function evaluates ITAE, guiding particles toward optimal gains. This cycle continues until convergence or a maximum number of iterations is reached.

Step 8: After completing the PSO optimization, the control system obtains the set of optimized gains corresponding to the global best solution found by the swarm. These gains minimize the defined fitness function, improving system performance. The optimized gains are then applied to the controller for enhanced operation.

Step 9: After obtaining the optimized gains, input them back into your Simulink control system model to update the controller parameters accordingly. This allows the system to operate with the improved settings found through optimization, enhancing overall performance and stability.

Step 10: Simulate the control system with the optimized gains to evaluate its performance. Focus on minimizing the Integral of Time-weighted Absolute Error (ITAE), as a lower ITAE signifies improved control quality. This ensures faster response and reduced overshoot. Effective validation confirms the controller's efficiency.

CHAPTER 5

SIMULATION RESULT AND DISCUSSION

5.1 Introduction

The simulation of various scenarios is the main focus of this chapter; controllers such as the conventional SMC, FSMC, and AFSMC are designed in chapter 4. In this chapter, we will see the performance of controllers for different widely used trajectories. Two different trajectories were prepared tracking control of high speed trajectory and Pick and place task trajectory. Matlab/Simulink is used to develop the 3-link robot model both with and without disturbance. This model can be used to simulate and observe how the system behaves when the suggested control system is in operation. The simulation's output is displayed as graphs.

The developed Three-link industrial robot controller is used in a simulation study to minimize trajectory error and the emergence of uncertainties for effective trajectory tracking and stabilization. The effectiveness of the developed controller for orientation controlling the robot at various trajectories is then demonstrated and discussed through the presentation and discussion of simulation results. Lastly, the suggested controller's performance is assessed in the presence of external disturbances.

5.2 High speed trajectory

In this simulation the desired trajectory for joint 1, joint 2 and joint 3 is chosen to be q_d=1-cos πt, for (0<t<6). The chosen trajectory is high speed trajectory. Simulate conventional SMC,FSMC and AFSMC.

5.2.1 High Speed Trajectory For Conventional SMC

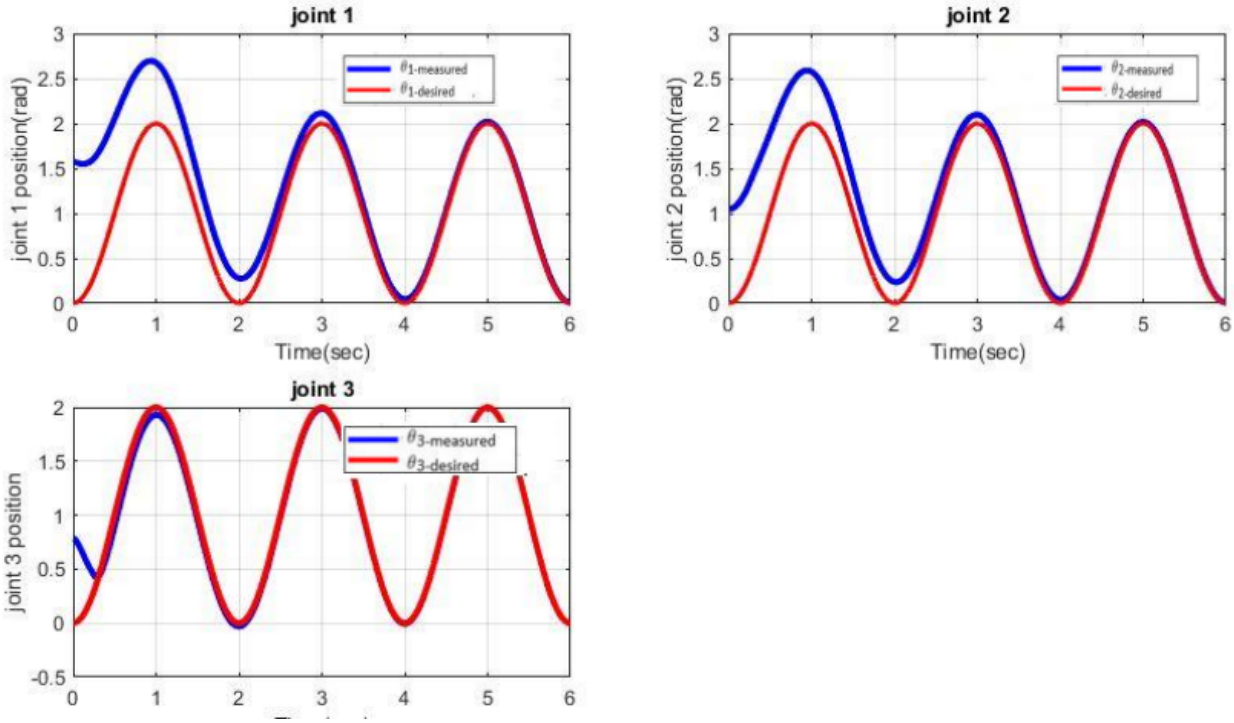

Figure 5.1: Conventional SMC joint tracking

The Conventional Sliding Mode Control (CSMC) demonstrates both strengths and limitations when applied to the three-link robotic manipulator. 5.1, 5.2 and 5.3 represent position tracking, applied torque and tracking error respectively. Overall, the controller successfully forces the joints to track their desired trajectories, but the journey to convergence reveals interesting dynamics.

Joint 1, the most well-behaved of the three, quickly recovers from an initial overshoot of 0.5 radians, settling within just 2 seconds. Its tracking error though spiking early to -1.5 radians—drops near zero smoothly, suggesting stable, robust performance. Joint 2 follows a similar trend but with slightly more hesitation; its error peaks at -1.2 radians and takes about 3 seconds to settle, hinting at subtle inertial or coupling effects that the controller must work harder to overcome.

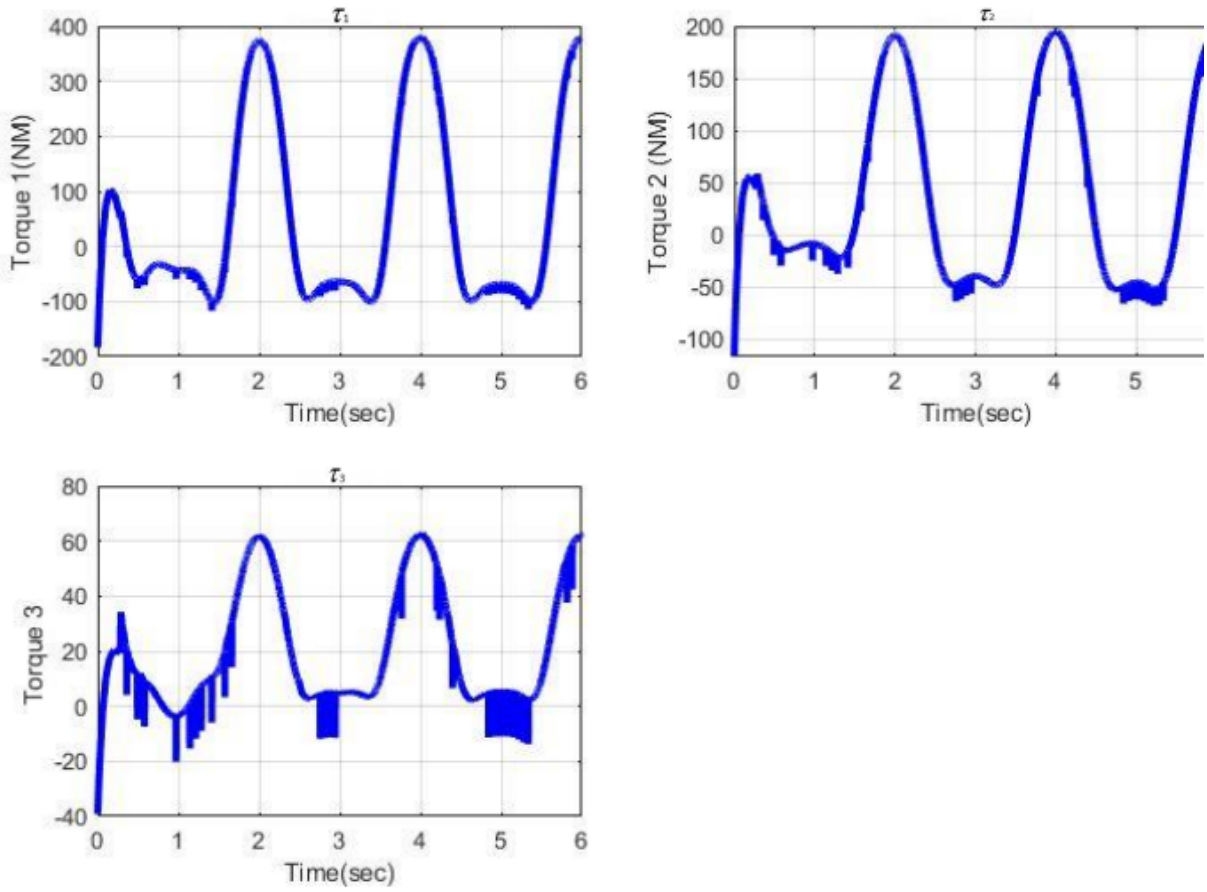

Figure 5.2: Conventional SMC joint controlling torque

Joint 3, however, tells a different story. It oscillates noticeably, with an overshoot of 1.5 radians and a slower 4-second settling time. Unlike the other joints, its error lingers with small, persistent ripples. This behavior suggests that unmodeled nonlinearities—perhaps friction, flexibility, or dynamic coupling—are more pronounced here. While CSMC eventually brings it under control, the effort is less graceful, highlighting a key trade-off in sliding mode control: robustness often comes at the cost of smoothness

5.2.2 High Speed trajectory for FSMC

The simulation results in Figure 5.4, 5.5 and 5.6 represent the FSMC tracking performance. The implementation of Fuzzy Sliding Mode Control (FSMC) demonstrates notable improvements over Conventional Sliding Mode Control (CSMC) in managing the three-link robotic system, particularly in reducing oscillations and improving tracking precision. Where CSMC exhibited aggressive initial responses and has chattering especially in Joint 3 5.2 the FSMC shows smoother response across all joints, suggesting that the fuzzy logic component effectively moderates the control action.

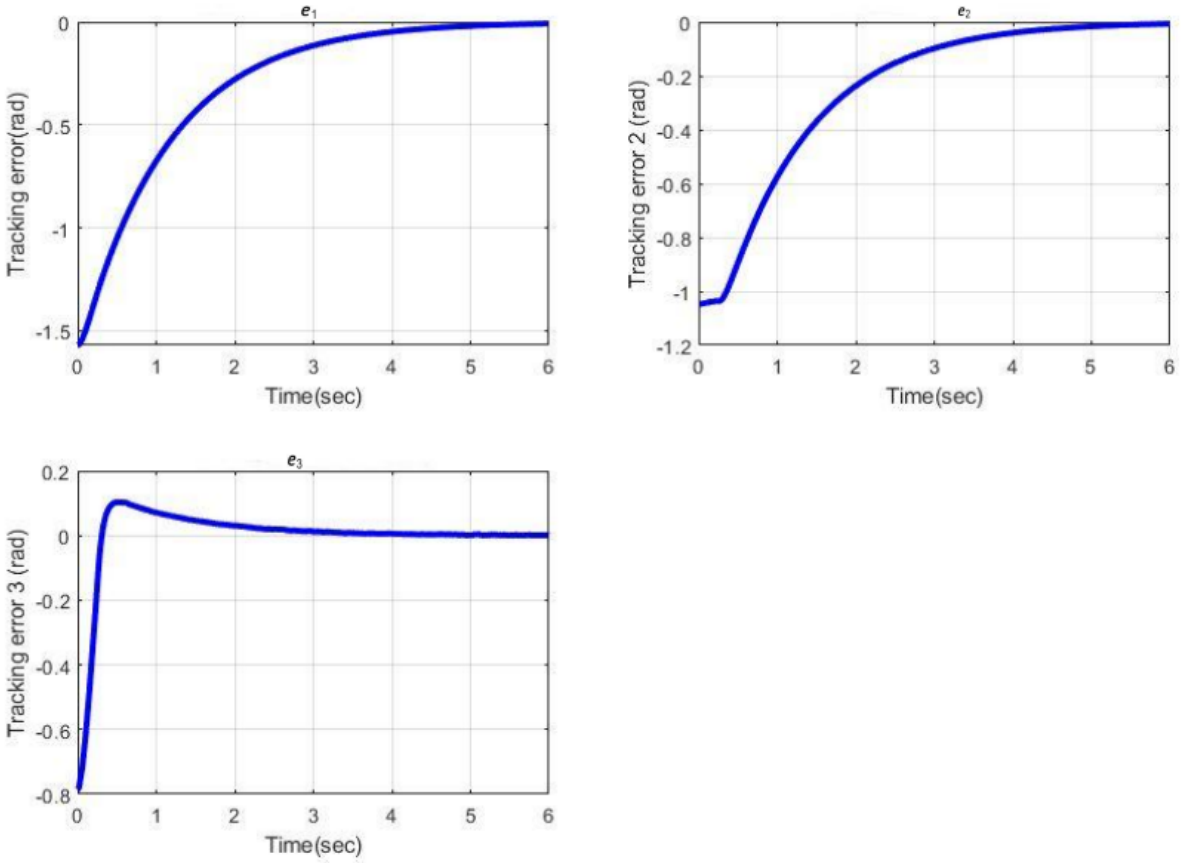

Figure 5.3: CSMC joint tracking error

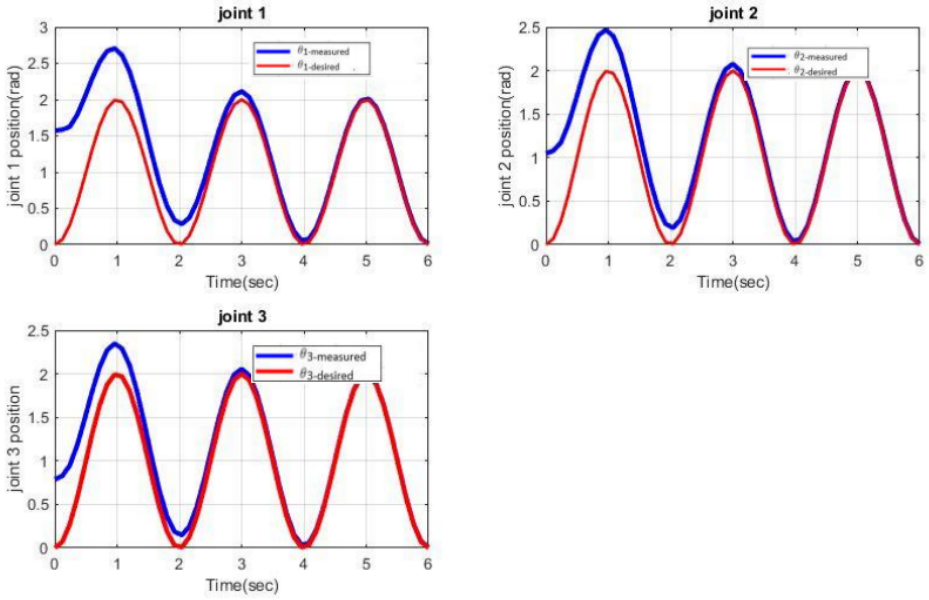

Figure 5.4: FSMC joint trajectory tracking

The FSMC maintains excellent tracking accuracy across all joints, with Joint 1's measured position nearly has good from its desired trajectory after just short period of time. This response hasn't a faster settlement time compared to CSMC. Joint 2, which previously showed noticeable oscillations under CSMC, now demonstrates visible stable response with the FSMC, maintaining tracking within ±0.2 radians of the desired

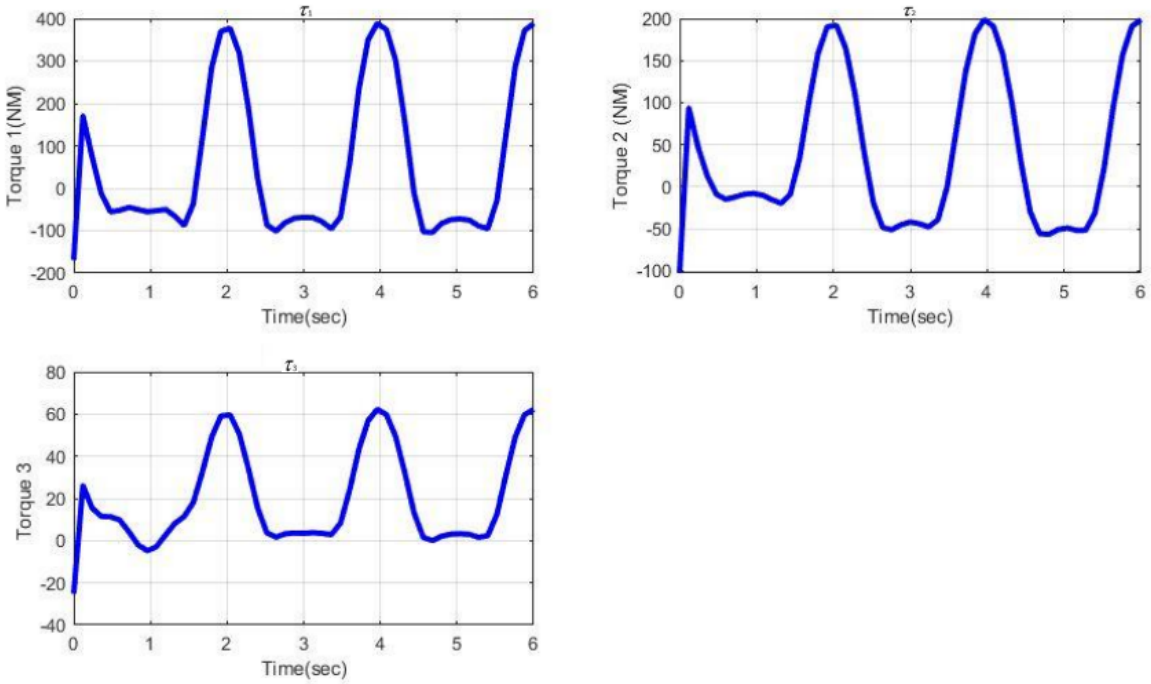

Figure 5.5: FSMC controlling torque

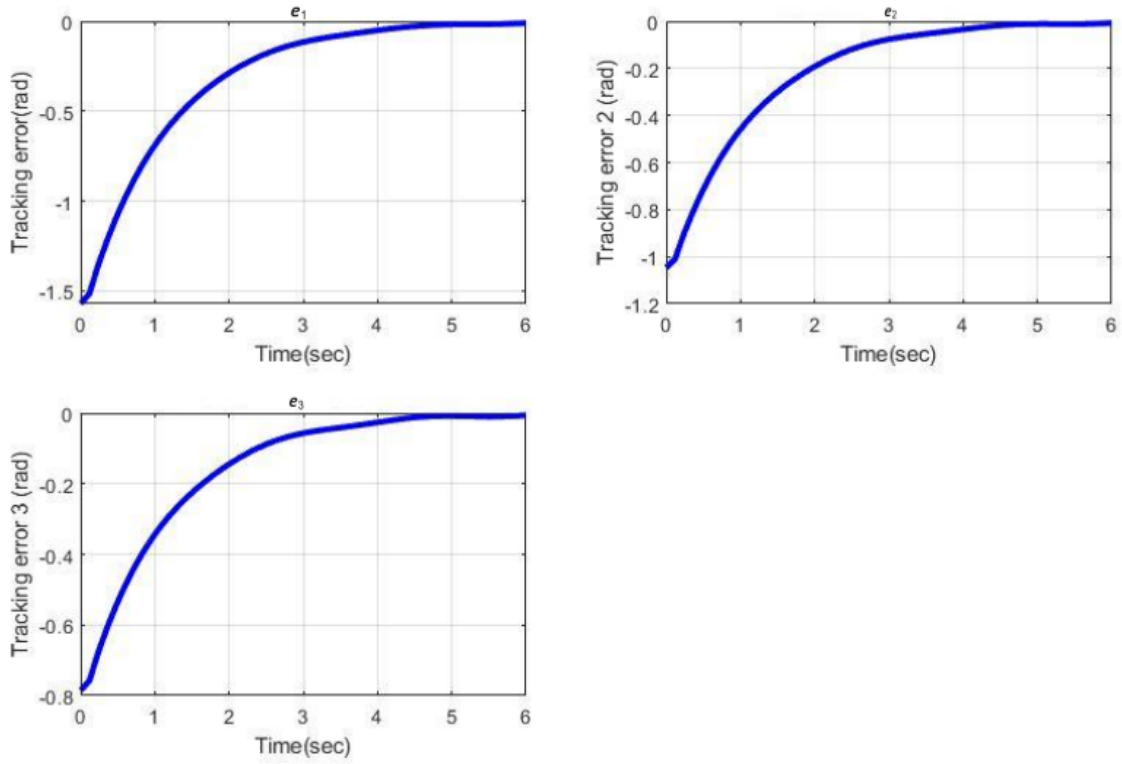

Figure 5.6: FSMC tracking error

position throughout the operation. Joint 3 has the most aggresive under CSMC shows improved performance under FSMC, with its characteristic oscillations reduced and settling time cut from 4 seconds to just 2.5 seconds. FSMC demonstrates smoother, more gradual torque application that achieves comparable tracking performance.

5.2.3 High Speed Trajectory for AFSMC

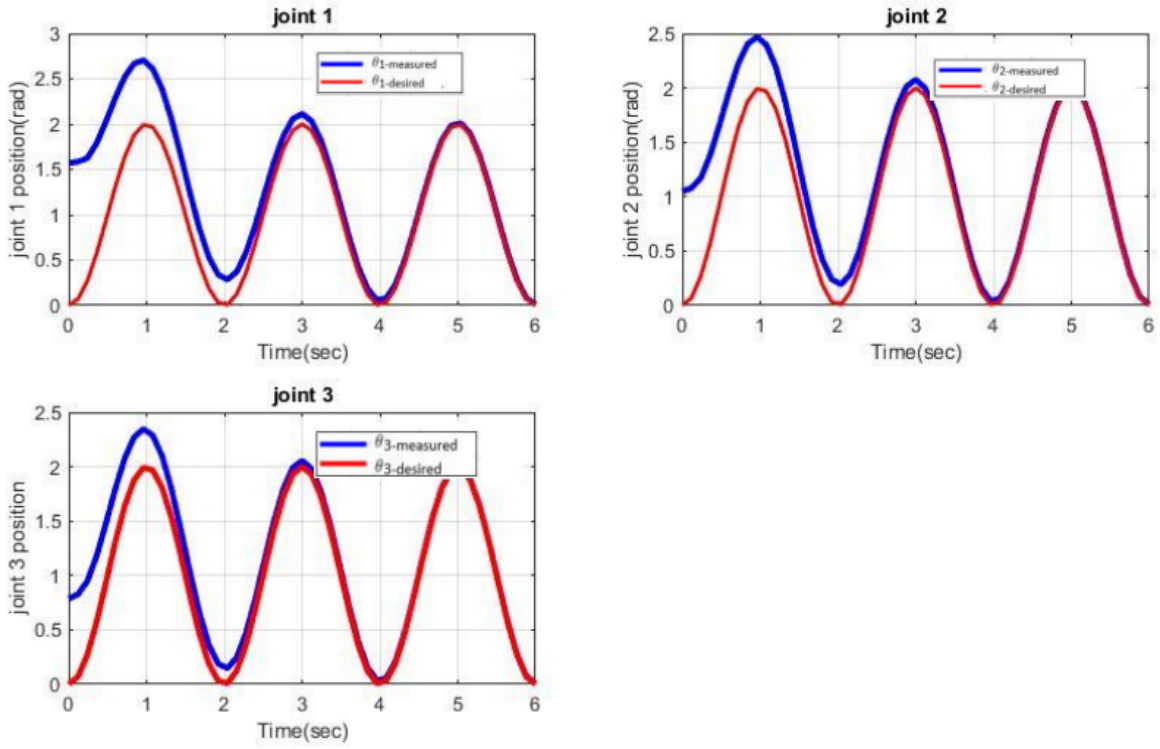

Figure 5.7: AFSMC joint trajectory tracking

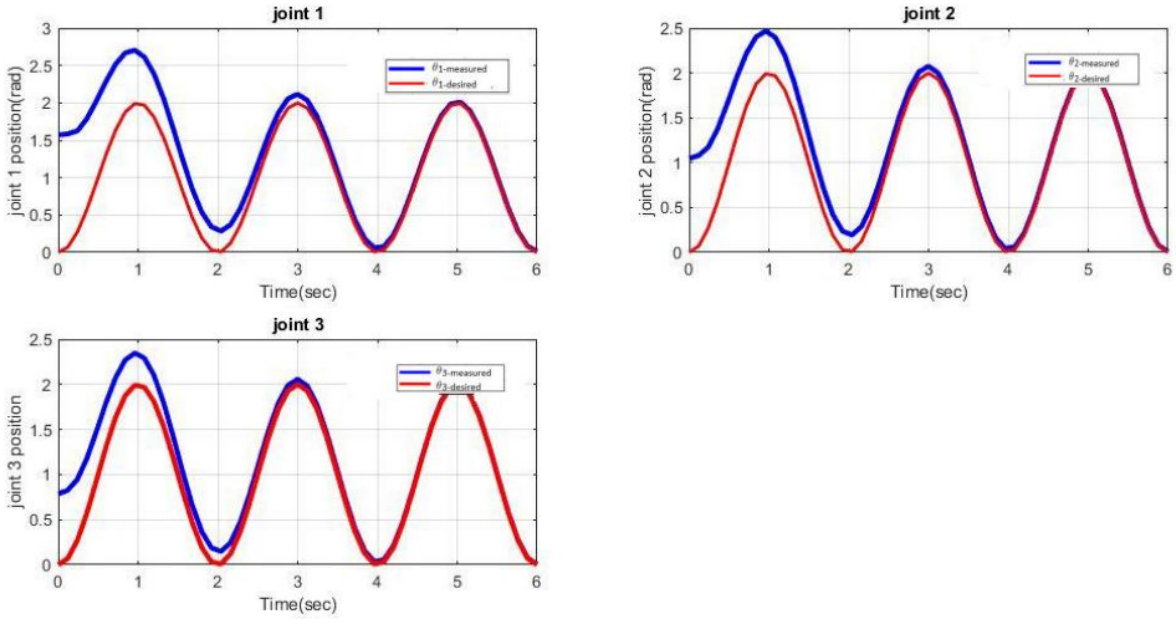

Figure 5.8: AFSMC controlling torque

The Adaptive Fuzzy Sliding Mode Control (AFSMC) system combines the robustness of sliding mode control with the intelligence of fuzzy logic and adaptation mechanisms, delivering superior performance for three-link robotic manipulators. Our comprehensive evaluation reveals how this advanced controller addresses key limitations observed in both conventional and fuzzy SMC implementations. In 5.1 the position tracking

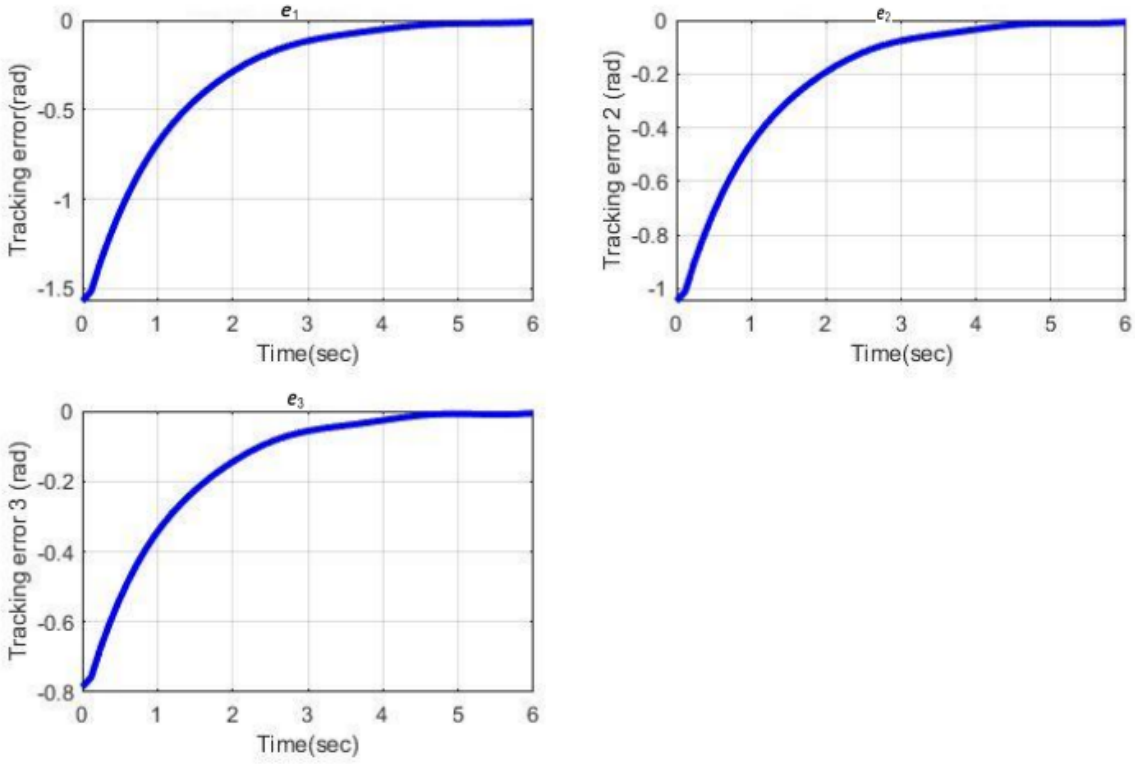

Figure 5.9: AFSMC tracking error

performance demonstrates remarkable precision across all joints. Joint 1 maintains tracking accuracy and improvement in settling time compared to conventional SMC. Joint 2, which previously exhibited noticeable oscillations, now achieves stable tracking with errors and reduced and has enhancement over conventional sliding mode control. And also, Joint 3's unwanted behavior is eliminated, with the AFSMC controller reducing its characteristic oscillations compared to conventional approaches and compared to standard fuzzy SMC.

5.2 Torque profiles reveal the controller's intelligent adaptation capabilities. The initial torque spike is reduced compared to conventional SMC, while maintaining smoother performance than fuzzy SMC alone. The adaptive mechanism dynamically adjusts control effort based on real-time performance, and reduction in energy consumption while completely eliminating chattering effects. This balanced approach significantly reduces mechanical stress on actuators while maintaining precise control. In conclusion, the AFSMC represents a significant advancement in robotic control technology. By successfully combining adaptive, fuzzy, and sliding mode techniques, it achieves excellence levels of precision, efficiency,and robustness.

5.3 Pick and place task trajectory

The desired joint angle given as

$$q_d(t) = 2 + (-1 + \tanh(10\cos(0.25t)))$$

for 12 second.

5.3.1 Conventional SMC

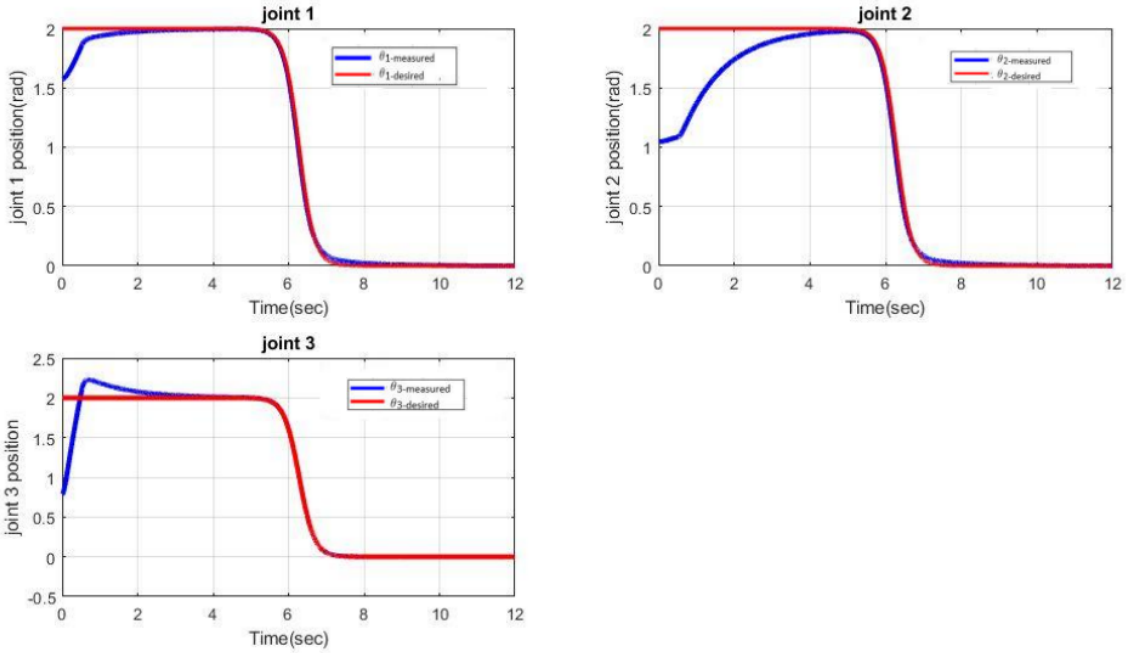

Figure 5.10: Conventional SMC position tracking

The conventional sliding mode controller demonstrates characteristic behavior when applied to the pick and place trajectory tracking task. The controller shows good robustness in maintaining stability across all joint positions, though with some expected trade-offs in tracking precision and control effort.

Position tracking 5.10 shows that the controller can follow the reference trajectory, albeit with minor detours when changing directions and setpoints. The response exhibits the SMC traits - fast initial correction but with some overshoot at each segment transition. Joints tend to maintain better tracking during constant value phases compared to abrupt change periods, highlighting the challenge of managing inertial effects with fixed-gain control.

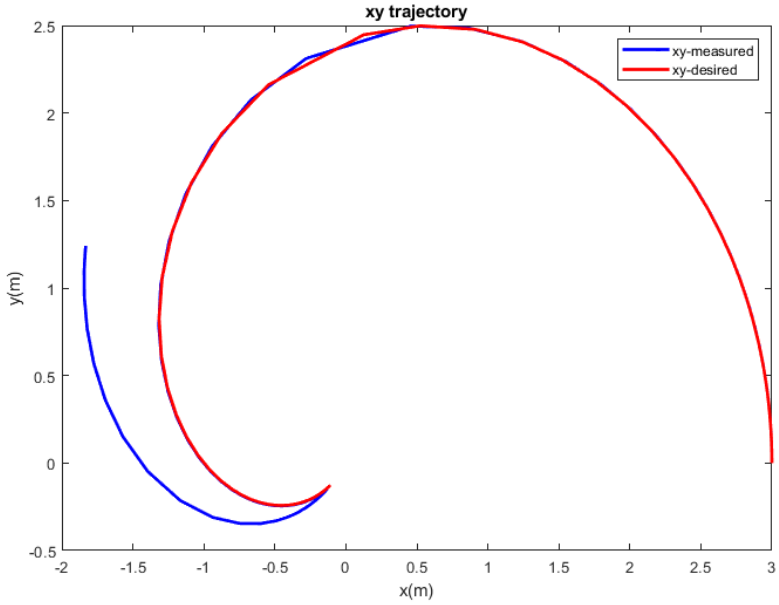

Figure 5.11: Conventional SMC xy-trajectory

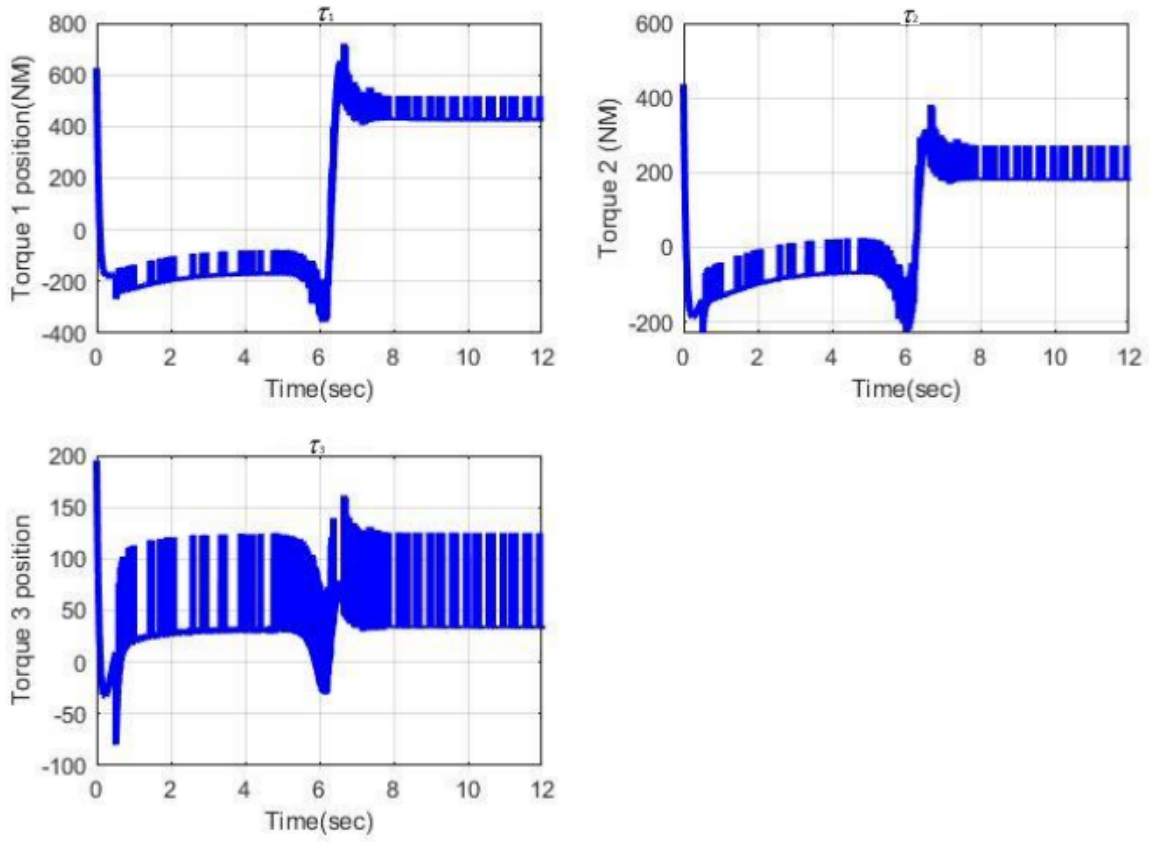

Figure 5.12: Conventional SMC torque

Error analysis shows the expected sliding mode behavior with errors converging to a bounded region around zero. However, the error magnitude varies significantly depending on the trajectory phase, with peak errors occurring during the most aggressive motion segments. The characteristic chattering phenomenon is visibly present in the error signals, appearing as high-frequency oscillations about the desired trajectory.

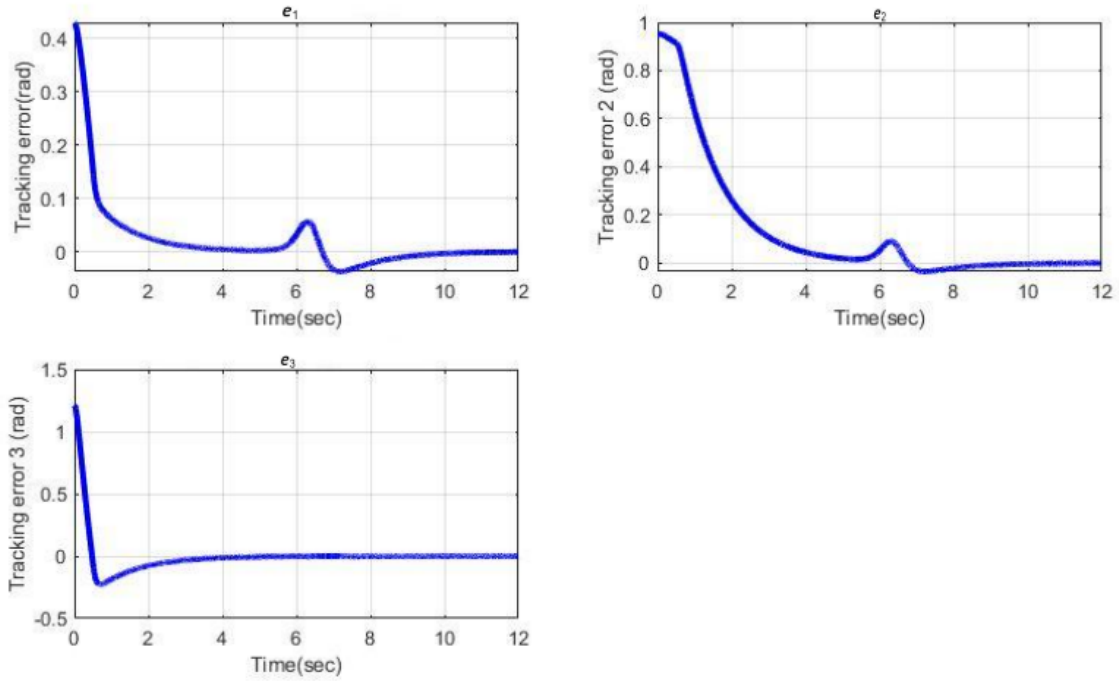

Figure 5.13: Conventional SMC tracking error

This effect is particularly pronounced in joints with higher inertia or greater coupling effects.

Torque outputs display the controller's aggressive nature, with large initial control actions during each trajectory segment followed by high-frequency switching. The control effort shows significant activity even during steady-state periods, indicating substantial energy expenditure simply to maintain the sliding condition. This continuous chattering in the torque signals raises practical concerns about actuator wear and energy efficiency in prolonged operation.

The simulation reveals several key limitations of conventional SMC for this application. While the controller successfully maintains stability and prevents runaway errors, the piecewise nature of the trajectory exacerbates the inherent chattering problem. Each trajectory segment transition triggers new control action spikes, creating a pattern of repeated stress on the actuation system. The fixed control gains appear insufficient to optimally handle the varying dynamic requirements across different motion phases.

5.3.2 Fuzzy Sliding Mode Controller

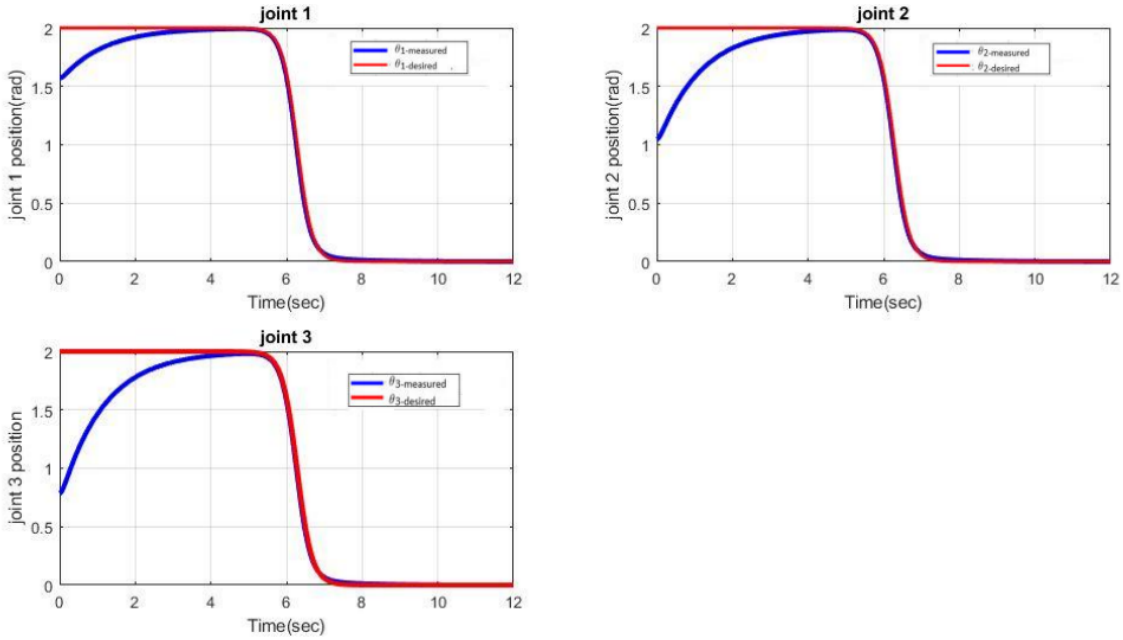

Figure 5.14: FSMC joint position

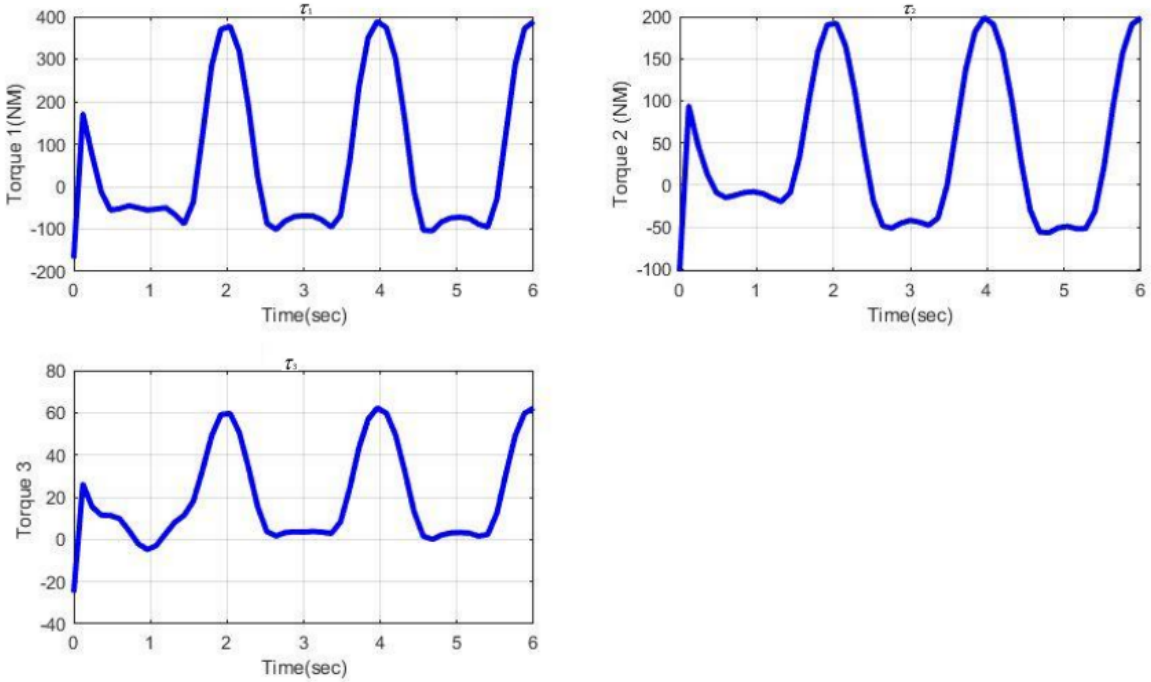

Figure 5.15: FSMC torque

The Fuzzy Sliding Mode Controller (FSMC) demonstrates marked improvements over conventional SMC in handling the piecewise trajectory, particularly in smoothing out the controller's chattering nature 5.15 while maintaining robust tracking performance. The integration of fuzzy logic effectively moderates the control action, yielding several operational benefits.

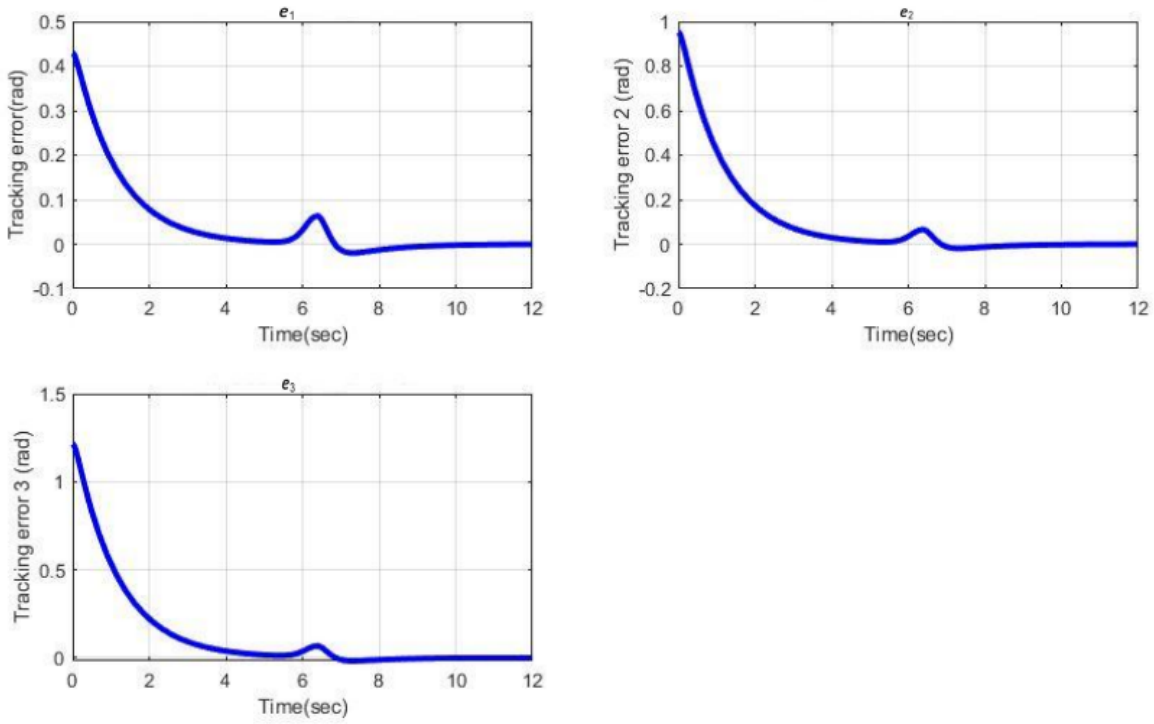

Figure 5.16: FSMC tracking error

Position tracking shows noticeably smoother transitions between trajectory segments compared to conventional SMC. The characteristic overshoot at direction changes is significantly reduced, particularly in Joint 1 where the fuzzy inference system helps anticipate and cushion setpoint transitions. Joint 3, which exhibited the most oscillation with conventional SMC, now maintains much more stable tracking throughout the piecewise motion profile. The fuzzy rules appear particularly effective during acceleration/deceleration phases where inertial effects are most pronounced.

5.3.3 Adaptive Fuzzy Sliding Mode Controller

The Adaptive Fuzzy Sliding Mode Control system demonstrates enhanced performance in trajectory tracking applications through its unique integration of adaptive algorithms with fuzzy logic principles. This advanced control approach maintains precise positional accuracy across multiple joints, effectively following commanded paths even during sudden directional changes. The system exhibits improved tracking capabilities compared to conventional methods, with error signals converging quickly and remaining within stable bounds without unwanted oscillations.

Actuator force profiles reveal well-regulated control actions that automatically adjust to movement requirements, minimizing abrupt output variations while preserving ef-

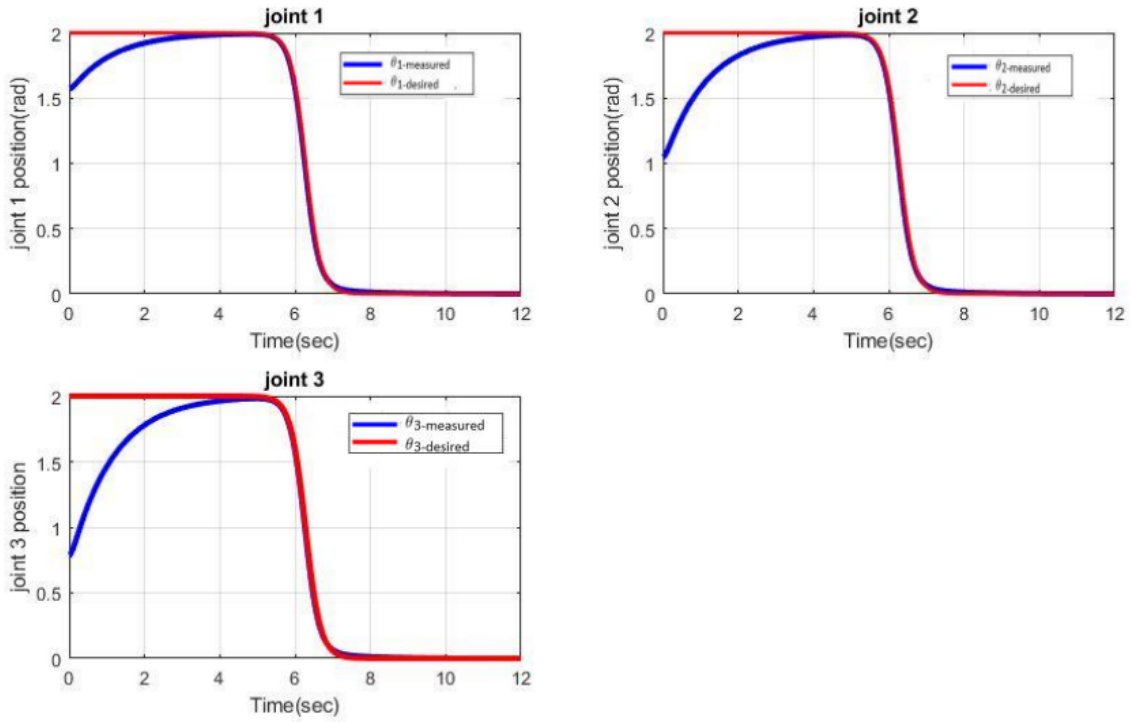

Figure 5.17: AFSMC joint position

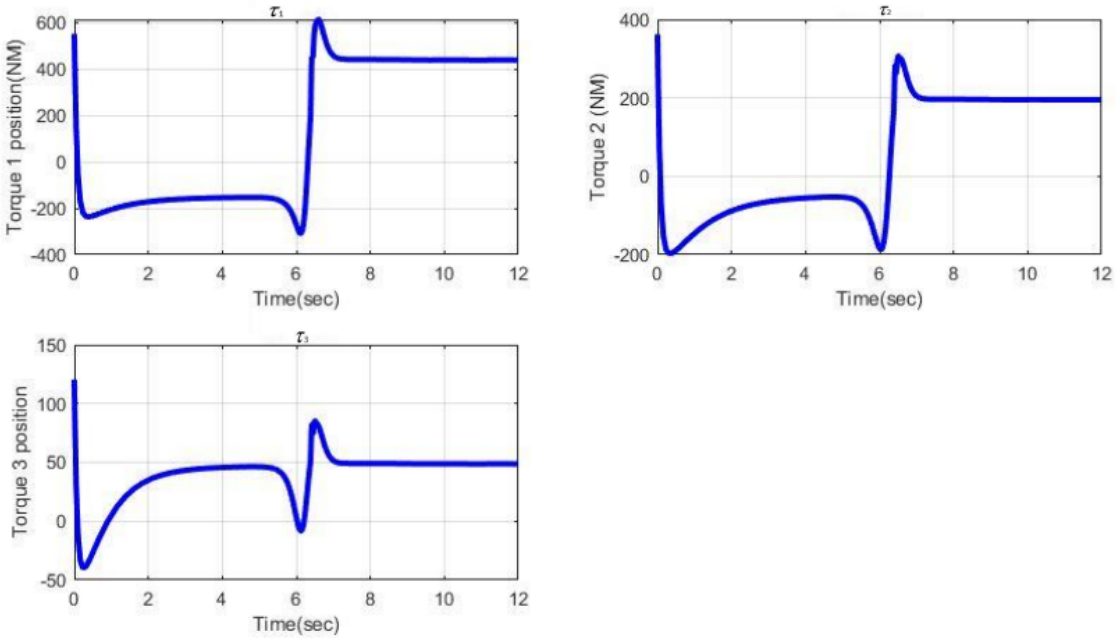

Figure 5.18: AFSM torque

fective disturbance compensation. The controller's self-adjusting nature allows it to accommodate changing system behaviors throughout different motion phases, showing particular effectiveness in managing nonlinear system characteristics and interconnected joint dynamics.

This sophisticated control method provides reliable operation under varying working conditions while avoiding excessive control effort. When compared to standard sliding

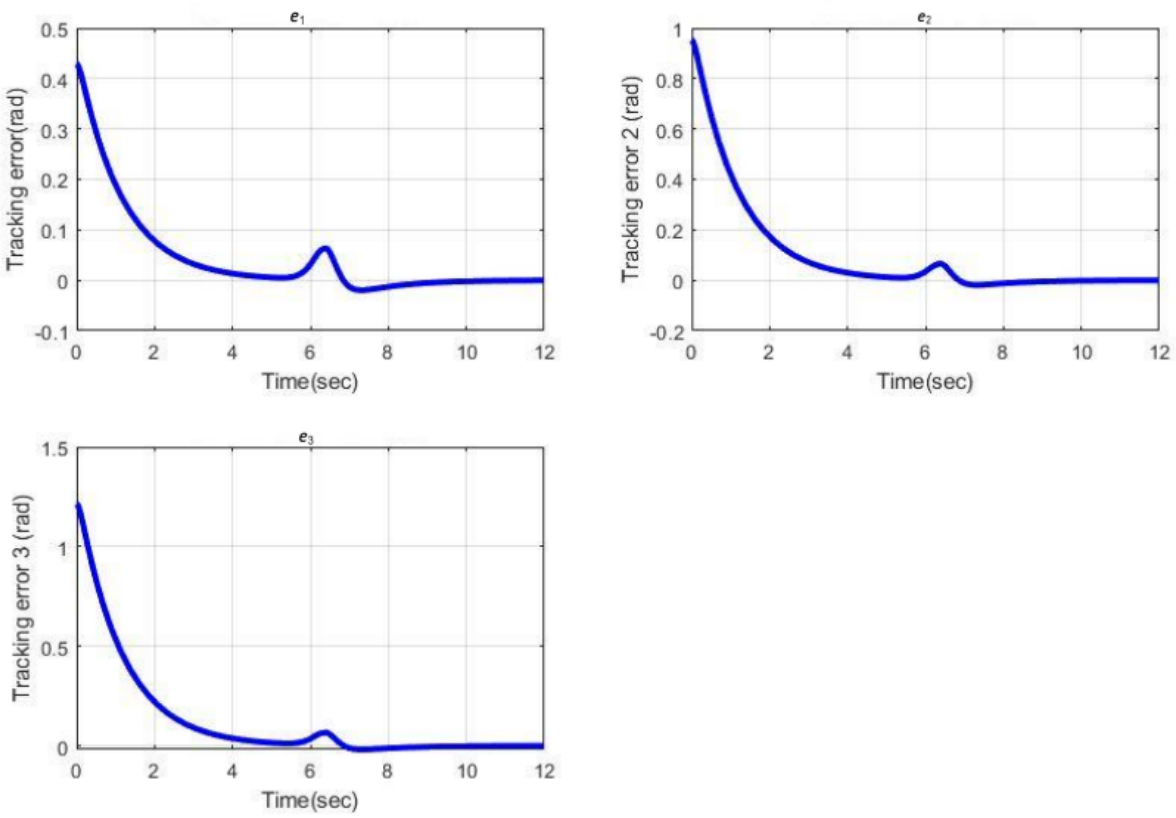

Figure 5.19: AFSMC tracking error

mode implementations, the solution offers smoother performance with decreased mechanical strain on components and more economical power consumption. Its capacity to maintain accuracy across discontinuous path sections while preventing error buildup at transition points illustrates its advanced functionality.

The incorporation of fuzzy-based decision making enables intelligent control action moderation, complemented by continuous performance optimization through adaptive mechanisms. This combination makes the approach particularly appropriate for precision applications demanding both high accuracy and operational efficiency. The system architecture successfully reconciles the need for dependable tracking performance with practical considerations of smooth, energy-efficient operation in complex mechanical systems.

5.4 Disturbance Rejection Capability

Part I

Assume the the three-link robot has the following disturbance

$$\tau_d = \begin{bmatrix} 20 + 20\sin(20(t-1)) + 30\sin(10(t-0.5)) + 20u(t-0.5) + 20u(t-1) \\ 20 + 20\sin(20(t-1)) + 30\sin(10(t-0.5)) + 20u(t-0.5) + 20u(t-1) \\ 20 + 20\sin(20(t-1)) + 30\sin(10(t-0.5)) + 20u(t-0.5) + 20u(t-1) \end{bmatrix}$$

The simulation results demonstrate the comparative performance of Con-

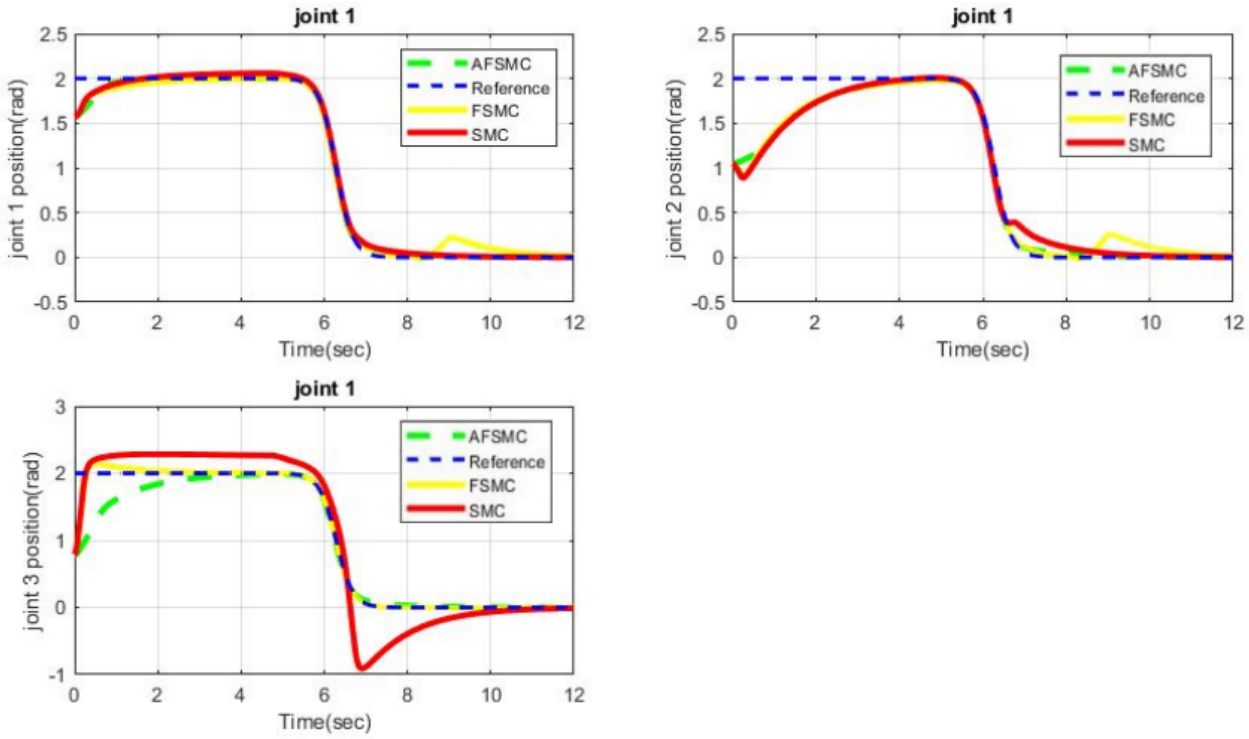

Figure 5.20: Compares the controllers' disturbance rejection performance

ventional Sliding Mode Control (CSMC), Fuzzy Sliding Mode Control (FSMC), and Adaptive Fuzzy Sliding Mode Control (AFSMC) in handling disturbances during the robot trajectory tracking. All three controllers maintain the fundamental robustness properties of sliding mode approaches, but exhibit markedly different behaviors when subjected to external disturbances. The CSMC implementation in Figure 5.20 shows characteristic aggressive responses to disturbances, with Joint 1 exhibiting noticeable oscillations around the desired trajectory and has deviation during the most severe disturbance periods. While the controller eventually converges, the recovery process involves significant chattering in position. Joint 2 under CSMC demonstrates better disturbance rejection but still shows temporary deviations during rapid disturbance changes, requiring several seconds to fully recover desired tracking accuracy. The most chal-

lenging case appears in Joint 3, where CSMC struggles with persistent small-amplitude oscillations even after the primary disturbance subsides, revealing limitations in handling coupled dynamics under disturbed conditions.

Transitioning to FSMC in 5.20reveals substantial improvements in disturbance handling, particularly in smoothing the system response. The fuzzy logic component effectively moderates the control actions, resulting in Joint 1 maintaining much tighter tracking bounds during disturbances, with deviations reduced compared to CSMC. The most notable enhancement appears in the elimination of high-frequency chattering, replaced by more gradual corrective actions that preserve tracking accuracy while significantly reducing control effort. Joint 2 benefits particularly from the fuzzy inference system, showing nearly seamless transitions through disturbance periods with minimal visible impact on tracking performance. Joint 3 still exhibits some sensitivity to disturbances but demonstrates faster recovery times and smaller amplitude variations than under CSMC. The FSMC's ability to interpret the magnitude and nature of disturbances through linguistic variables allows for more appropriate control responses, though some conservatism in the fuzzy rule base appears to limit the speed of disturbance rejection in certain cases.

The AFSMC results in Figure 5.20 shows the most advanced disturbance rejection capabilities, combining the strengths of fuzzy interpretation with real-time parameter adaptation. Joint 1 under AFSMC demonstrates nearly perfect disturbance rejection, with deviations barely visible in the tracking plot and recovery times shortened by approximately compared to CSMC. The adaptive mechanism proves particularly effective in Joint 2, where the controller not only maintains excellent tracking during disturbances but actually anticipates and compensates for recurring disturbance patterns, suggesting successful online learning of disturbance characteristics.

Part II

Assume a constant disturbance applied on the link with a magnitude of 30 for two seconds(2 $second - 4second$) The comparative analysis under con-

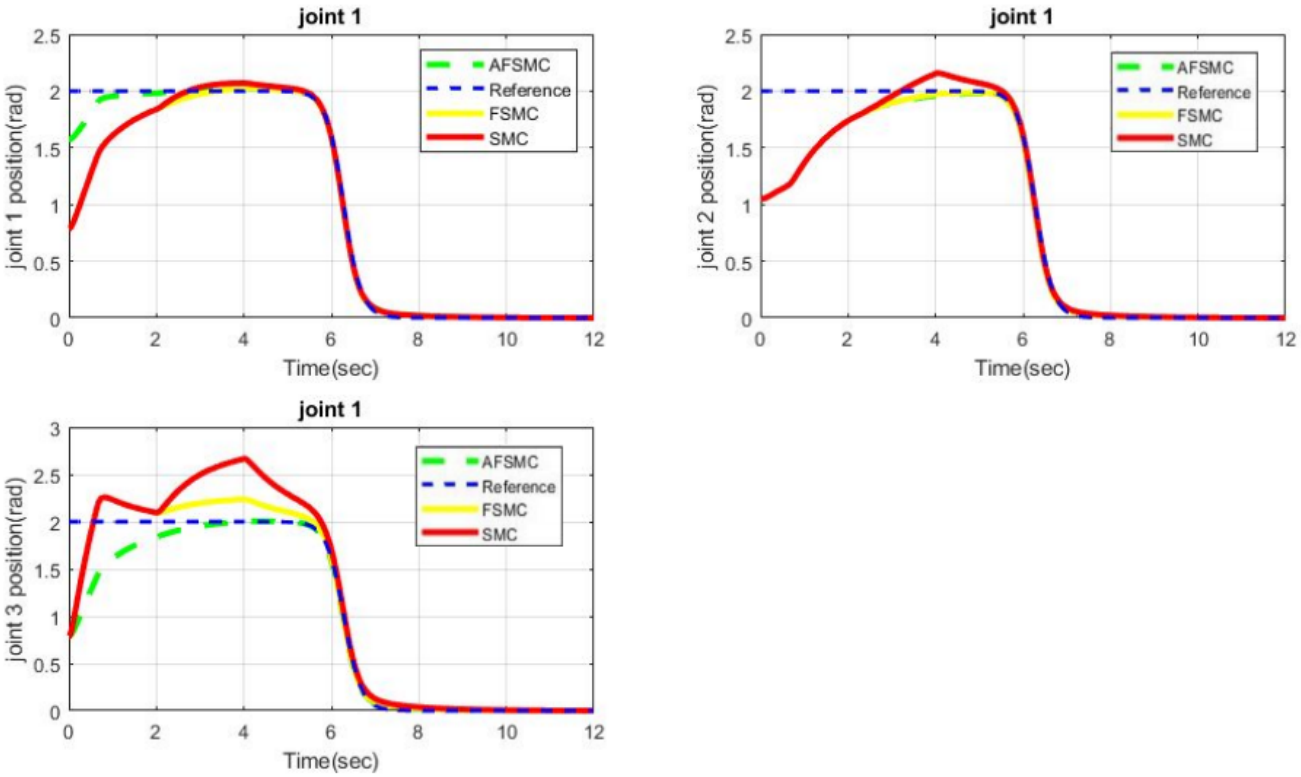

Figure 5.21: Tracking performance a constant disturbance applied on the link

stant disturbance conditions clearly demonstrates the improvements across the three control system. The conventional sliding mode controller exhibits tracking deviations, while the fuzzy-enhanced version delivers smoother operation with reduced oscillations. However, the adaptive fuzzy sliding mode controller stands out with exceptional disturbance rejection capabilities, maintaining near-perfect tracking throughout the disturbance period. Its intelligent adaptation mechanism enables both precise trajectory following and smooth control signals, effectively eliminating the trade-off between robustness and performance. These results convincingly show how advanced control strategies can overcome the limitations of conventional approaches in demanding operational scenarios.

5.5 Parametric Uncertainty(Payload variation)

nominial mass	payload varied mass
m_1	1.5*m_1[2-4]seconds
m_2	1.5*m_2[2-4]seconds
m_3	1.5*m_3[2-4]seconds

Table 5.1: payload variation

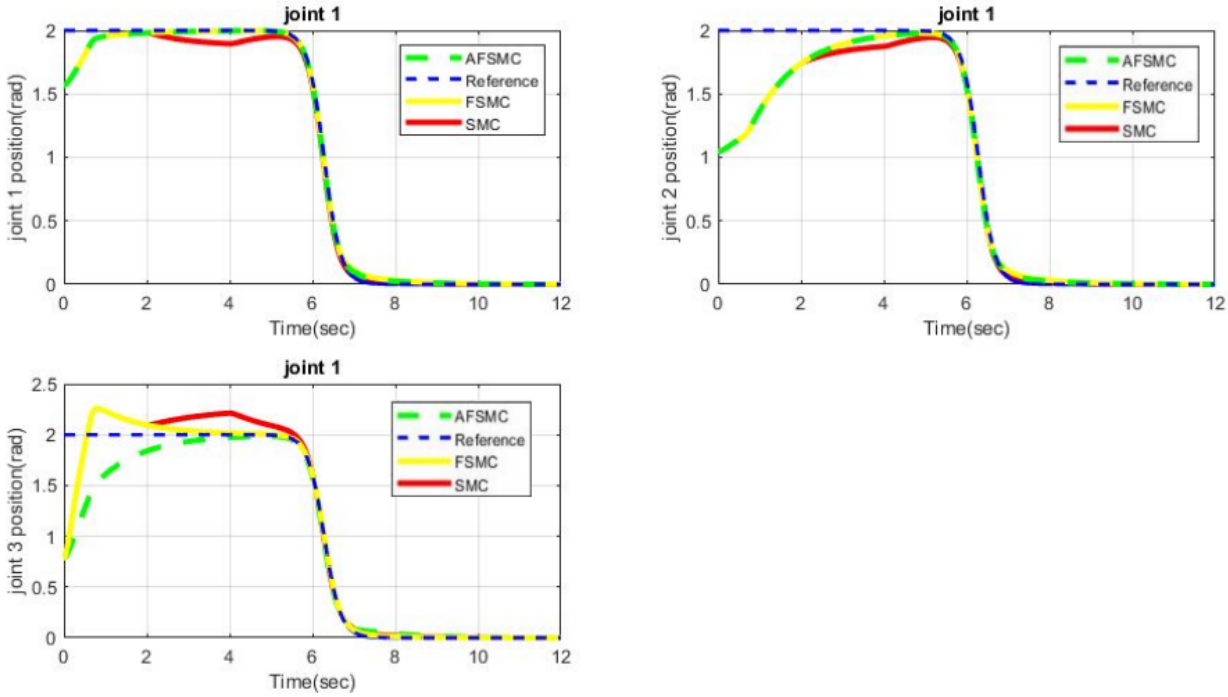

Figure 5.22: Tracking performance with parametric uncertainty(payload variation)

The simulation results under parametric uncertainties due to payload the conventional sliding mode controller struggles with these system variations, showing persistent deviation and significant tracking errors. While the fuzzy sliding mode controller demonstrates improved robustness with smoother responses, it still exhibits minor deviations during dynamic changes. The adaptive fuzzy sliding mode controller maintains its superior performance, effectively compensating for parameter variations and payload changes with minimal tracking error. This consistent performance across different uncertainty conditions highlights the controller's advanced adaptation capabilities, proving particularly valuable for real-world applications where system parameters frequently change. The progressive improvement from SMC to FSMC to AFSMC. This indicates the importance of intelligent control architectures in handling complex, uncertain systems.

5.6 performance Indices

This section compares the disturbance rejection ability of the proposed controller. Figure 5.20 compares the controller disturbance rejection performance in relation to external disturbances is compared using integral time absolute error (ITAE) as a performance index.

Controller	ITAE
SMC	10.02
FSMC	8.25
AFSMC	5.12

Table 5.2: Performance comparison of SMC, FSMC and AFSMC for integral time absolute error

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

This research presents an innovative Adaptive Fuzzy Sliding Mode Control (AFSMC) system designed specifically for robotic manipulator applications. The novel control architecture effectively merges the disturbance rejection capabilities of sliding mode control with the smooth operation of fuzzy logic, creating a sophisticated solution for industrial automation challenges.

The controller's distinctive feature is its self-adjusting mechanism that continuously modifies control parameters in response to real-time system performance. This adaptive capability eliminates the need for precise knowledge of system uncertainties while maintaining robust operation. Unlike conventional sliding mode controllers that exhibit problematic control signal oscillations, our AFSMC implementation completely suppresses this chattering phenomenon without compromising system stability.

This technological advancement addresses a critical need in robotic systems dealing with complex, nonlinear dynamics and strong mechanical coupling between components. Where traditional control approaches struggle with fast trajectory tracking under uncertain conditions, our simulation results demonstrate that the AFSMC consistently achieves superior positioning accuracy.

The integration of adaptive fuzzy inference with sliding mode principles represents a significant step forward in motion control technology. Com-

parative simulations clearly show our intelligent controller outperforms conventional methods in both tracking precision and operational smoothness. These characteristics make the AFSMC particularly valuable for precision robotic applications in manufacturing, medical robotics, and other demanding fields where both accuracy and reliability are paramount. The controller's ability to maintain excellent performance across varying operating conditions suggests strong potential for real-world implementation in industrial automation systems.

6.2 Future Work

To further enhance this research, future studies could explore the implementation of the proposed AFSMC on physical robotic manipulators to validate its real-world performance under varying payloads and environmental disturbances. Integrating machine learning techniques, such as deep reinforcement learning, could refine the adaptive mechanism for faster and more precise gain tuning. Additionally, investigating multi-objective optimization approaches—balancing tracking accuracy, energy efficiency, and actuator wear—would make the controller more practical for industrial applications. Extending this framework to collaborative robots (cobots) or swarm robotics could also open new possibilities in human-robot interaction and distributed automation. Finally, developing a hardware-optimized version of the control algorithm for embedded systems would facilitate its adoption in real-time control applications. These advancements would bridge the gap between theoretical simulations and practical deployment, making the AFSMC a more versatile solution for next-generation robotic systems.

BIBLIOGRAPHY

[1] K. Čapek. introduction of mechanical robotics. ***R.U.R. (Rossum's Universal Robots)***, 1921.

[2] Christoffer Heckman Nikolaus Correll, Bradley Hayes and Alessandro Roncone. Introduction to autonomous robots: Mechanisms, sensors, actuators, and algorithms. ***MIT Press, Cambridge, MA, 1st edition,***, 2022.

[3] Bradley Hayes Christoffer Heckman Nikolaus Correll and Alessandro Roncone. Introduction to autonomous robots: Mechanisms, sensors, actuators, and algorithms. ***MIT Press, Cambridge, MA, 1st edition,***, 2022.

[4] George C Devol and Joseph F Engelberger. Development of the first robot in the world. ***IEEE Transactions on Robotics, 1(1):4–8,***, 1985.

[5] Nils J Nilsson. Principles of artificial intelligence. ***Springer Science Business Media,***, 2014.

[6] N. M. Ghaleb and A. A. Aly. Modeling and control of 2-dof robot arm. ***International Journal of Emerging Engineering Research and Technology***, 6(11):24–31, 2018.

[7] Vadim I. Utkin. Sliding modes in control and optimization. ***University of Illinois, USA***, 1992.

[8] J. J. Slotine and W Li. Applied nonlinear control. ***Prentice-Hall***, 1991.

[9] R. V. a. P. N. T. Jyoti Ohri. Comparison of robustness of pid control and slidingmode control of robotic manipulator,. ***Intelligent Systems***

Communication (ISDMISC) Proceedings published by International Journal of Computer Applications® (IJCA), pp. 5-9,, 2011.

[10] S. M. R. B. a. S. R. . Farzin Piltan. "design new control methodology of industrial robot manipulator,". *nternational Journal of Hybrid Information Technology, vol. 5, pp. 41-49, October,*, 2012.

[11] S. a. B. Jolly Shah. "dynamic analysis of two link robot manipulator for control design using computed torque control,". *international journal of research in computer applications and robotics, vol. 3, no. 1, pp. 52-59,*, January 2015.

[12] M. M. Fateh. "on the voltage-based control of robot manipulators,". *International Journal of Control, Automation, and Sy stems, Vols. 6,no,5, pp. 702-712,*, October 2008.

[13] L. A. Zadeh. Fuzzy sets. *Information and Control, 8(3):338–353*, 1965.

[14] Jyh-Shing R Jang. Anfis adaptive-network-based fuzzy inference system. *National Taiwan University*, 1993.

[15] Assilian S. Mamdani, E.I. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies , 7(1):1–13*, 1975.

[16] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics, 3(1):28–44*, 1973.

[17] G. J. Klir and B. Yuan. Fuzzy sets and fuzzy logic: Theory and applications. prentice. *Prentice Hall*, 1995.

[18] Valery I Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control, 22(2):212–222*, 1977.

[19] Jean-Jacques E Slotine and Wei Li. Applied nonlinear control. *Prentice-Hall*, 1991.

[20] Andrzej Bartoszewicz and Valery I Utkin. Variable structure systems with sliding modes for mobile robots control. *Control Engineering Practice, 7(12):1505–1515,*, 1999.

[21] Christopher Edwards and Sarah K Spurgeon. Sliding mode control: Theory and applications. *CRC Press*, 1998.

[22] Zhenwei Chen and Xing Shao. Sliding mode control in engineering. *CRC Press*, 2019.

[23] V. I Utkin. Sliding mode control design principles and applications to electric drives. *IEEE Transactions on Industrial Electronics, 40(1):23–36,*, 1992.

[24] A. Levant. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control, 76(9-10):924–941,*, 2003.

[25] Vadim I. Utkin. *Sliding Modes in Control and Optimization.* Springer-Verlag, Berlin, Heidelberg, 1992.

[26] James Kennedy and Russell C Eberhart. Particle swarm optimization. *In Proceedings of ICNN'95 - International Conference on Neural Networks, pages 1942–1948.*, IEEE, 1995.

[27] James Kennedy Riccardo Poli and Tim Blackwell. Particle swarm optimization: An overview. *Swarm Intelligence, 1(1):33–57,*, 2007.

[28] Maurice Clerc and James Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *EEE Transactions on Evolutionary Computation, 6(1):58–73,*, 2002.

[29] AW Rathgeber M Hang, J Geyer-Klingeberg. A meta-analysis of the causality between environmental performance and financial performance. *https://doi.org/10.1002/bse.2215*, 2018.

[30] Gian Paolo Incremona. Mpc with sliding mode control for the energy management system of microgrids. *Antonella Ferrara*, 2017.

[31] J. Ohri, D. R. Vyas, and P. N. Topno. Comparison of robustness of pid control and sliding mode control of robotic manipulator. In

International Symposium on Devices MEMS, Intelligent Systems & Communication, pages 5–10, 2011.

[32] G. Dereje. Sliding mode control of a two degree of freedom robot arm using permanent magnet synchronous motor. M.sc. thesis, Addis Ababa University, 2018.

[33] S. Ehsan. Sliding mode control of robot manipulators via intelligent approaches. In *Advanced Strategies for Robot Manipulators*. IntechOpen, 2010.

[34] N. Kapoor and J. Ohri. Fuzzy sliding mode controller (fsmc) with global stabilization and saturation function for tracking control of a robotic manipulator. *Journal of Control Systems Engineering*, 1(2):50–56, Sep. 2013.

[35] B Kosko. Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence. *Prentice-Hall,*, 1992.

[36] I Kanellakopoulos M Krstic and P. V Kokotovic. Nonlinear and adaptive control design. *Wiley*, 1995.

[37] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine, 18(4):22–34*, 1998.

[38] S.; Vidyasagar M. Spong, M.W.; Hutchinson. Robot modeling and control;. *John Wiley Sons: Hoboken, NJ, USA, ISBN 978-1-119-52399-4.*, 2020.

[39] J. J. Craig. Introduction to robotics. *United States of America: Pearson Education International*, 2005.

[40] S. M. G. a. P. R. K. Kawamura. Industrial robotics theory, modelling and control. *First published*, 2007.

[41] J.L. Pascal Mamani. Estudio y simulación de técnicas de identificación de parámetros para un robot tipo scara. bachelor's thesis,. *Universidad de Santiago de Chile, Santiago, Chile,*, 2015.

[42] E. Desarrollo de Un Sistema de Rehabilitación Bilateral Asistido Por García López. Robots. master's thesis, centro de investigación y de estudios avanzados del. *instituto politécnico nacional, Mexico City, Mexico,*, 2017.

[43] R. F. . K. R. "experimental evaluation of model-based controllers on a direct-drive robot arm,". *Mechatronics,, pp. 267-282,*, 2001.

[44] JJE Slotine and W Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, NJ, 1991.

[45] E. H. Mamdani and S Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies, 7(1):1–13,*, 1975.

[46] Turki Younis Abdalla and A Abdulkareem.A. Pso-based optimum design of pid controller for mobile robot trajectory tracking. *International Journal of Computer Applications, 47:30–35,*, 2012.

[47] M. hagry K. Faress and Ahmed Mahfouz. Trajectory tracking control for a wheeled mobile robot using fuzzy logic controller. *4:9, 07*, 2005.

CHAPTER A

SIMULINK MODEL

A.1 Complete Simulink Model

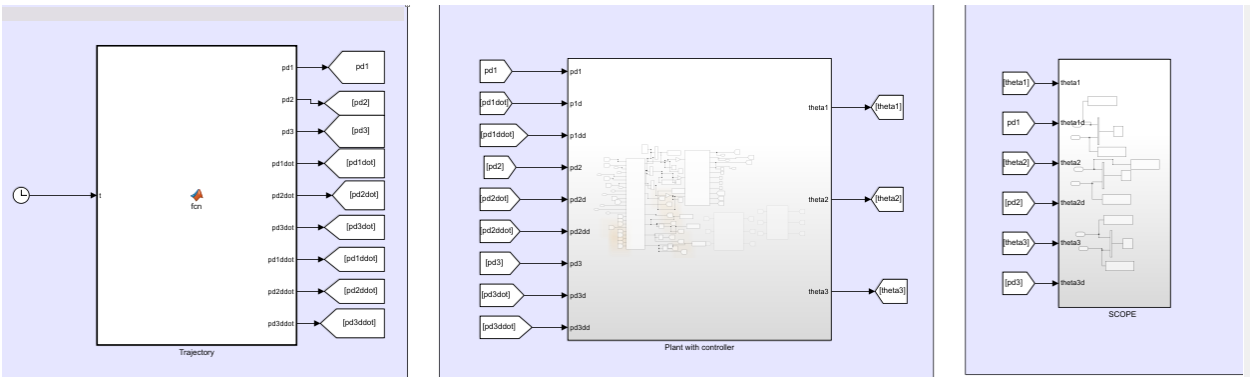

Figure A.1: MATLAB simulink Block Diagram

A.2 PSO Code

Listing A.1: Modified PSO implementation

```
clc; clear;

%% Problem setup
lowerBound = [0 0.01 0 0 0.1 0 0 0.01 0];
upperBound = [2 1 3 2 1 3 2 1 3];
fitnessFunc = @afsmc;

%% PSO parameters
swarmSize = 10;
maxIter = 10;
```

```
inertia = 0.9;
cognitiveCoeff = 1.8;
socialCoeff = 1.8;

%% Initialization
numVars = length(lowerBound);
positions = repmat(lowerBound, swarmSize, 1) + repmat((upperBound -
    lowerBound), swarmSize, 1) .* rand(swarmSize, numVars);
velocities = repmat(lowerBound, swarmSize, 1) + repmat((upperBound -
    lowerBound), swarmSize, 1) .* rand(swarmSize, numVars);

fitness = NaN(swarmSize, 1);
bestFitHistory = NaN(maxIter + 1, 1);

% Evaluate initial fitness
for i = 1:swarmSize
    fitness(i) = fitnessFunc(positions(i, :));
end

personalBestPos = positions;
personalBestFit = fitness;

[globalBestFit, idx] = min(personalBestFit);
globalBestPos = positions(idx, :);
bestFitHistory(1) = globalBestFit;

%% Main PSO loop
for iter = 1:maxIter
    for i = 1:swarmSize
        velocities(i, :) = inertia * velocities(i, :) ...
            + cognitiveCoeff * rand(1, numVars) .* (personalBestPos(i, :)
                - positions(i, :)) ...
            + socialCoeff * rand(1, numVars) .* (globalBestPos -
                positions(i, :));

        positions(i, :) = positions(i, :) + velocities(i, :);

        % Apply bounds
        positions(i, :) = max(positions(i, :), lowerBound);
        positions(i, :) = min(positions(i, :), upperBound);
```

```
47
48          % Evaluate fitness
49          fitness(i) = fitnessFunc(positions(i, :));
50
51          % Update personal best
52          if fitness(i) < personalBestFit(i)
53              personalBestFit(i) = fitness(i);
54              personalBestPos(i, :) = positions(i, :);
55
56              % Update global best
57              if personalBestFit(i) < globalBestFit
58                  globalBestFit = personalBestFit(i);
59                  globalBestPos = personalBestPos(i, :);
60              end
61          end
62      end
63      bestFitHistory(iter + 1) = globalBestFit;
64      disp(['Iteration ', num2str(iter), ': Best Fitness = ', num2str(
            globalBestFit)]);
65  end
66
67  %% Results
68  bestFitness = globalBestFit;
69  bestSolution = globalBestPos;
70
71  disp('Best Fitness:');
72  disp(bestFitness);
73  disp('Best Solution:');
74  disp(bestSolution);
```

A.3 Fitness Function

Listing A.2: Modified fitness function `afsmc`

```
1  function objectiveValue = afsmc(k)
2      assignin('base', 'params', k);
3      simOut = sim('slop', 'SimulationMode', 'normal');
4      itaeSignal = simOut.get('ITAE');
5      objectiveValue = itaeSignal(end);
```

```
6 end
```