Design of an Adaptive Fuzzy Super-Twisting Sliding Mode Controller for an Uncertain 2-DOF Robotic Manipulator

Hayleyesus Girma Dirara

A Thesis Submitted to

The Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

Design of an Adaptive Fuzzy Super-Twisting Sliding Mode Controller for an Uncertain 2-DOF Robotic Manipulator

Hayleyesus Girma Dirara

Advisors Dr. Tafesse Asrat (assistant professor)

A Thesis Submitted to
The Department of Electrical Power and Control Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies
Adama Science and Technology University

September, 2022
Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "Design of an Adaptive Fuzzy Super-Twisting Sliding Mode Controller for an Uncertain 2-DOF Robotic Manipulator" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

________________	________________	________________
Name of student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "Design of the Adaptive Fuzzy Super-Twisting Sliding Mode Controller for an Uncertain 2-DOF Robotic Manipulator" prepared under my guidance by Hayleyesus Girma Dirara submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Power and Control Engineering, postgraduate in control engineering.

Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Tafesse Asrat	taabera	Oct 29/2022
Name of Advisor's	Signature	Date

APPROVAL PAGE

I, the advisors of the thesis entitled "Design of the Adaptive Fuzzy Super-Twisting Sliding Mode Controller for an Uncertain 2-DOF Robotic Manipulator" and developed by Hayleyesus Girma Dirara, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Tafesse Asrat	taabera	Oct 29/2022
Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Hayleyesus Girma Dirara have read and evaluated the thesis entitled "Design of the Adaptive Fuzzy Super-Twisting Sliding Mode Controller for an Uncertain 2-DOF Robotic Manipulator" and examined the candidate during open defense. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Masters in Electrical Power and Control Engineering (Control)

______________	______________	______________
Chairperson	Signature	Date
Prof. Gang Gyoo Jin	Jin	Oct 29/2022
Internal Examiner	Signature	Date
Dr. Lebsework Negash		Oct 29/2022
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

______________________	______________	______________
Department Head	Signature	Date
______________________	______________	______________
School Dean	Signature	Date
______________________	______________	______________
Office of Postgraduate studies, Dean	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I would like to thank the supreme power the almighty God who is the one who has always guided me to work on the right path in life. Without his grace, this master's thesis could not become a reality. Next to him, I would like to give special thanks to my mother (**Zenebech Dengo**) and my family as a whole for their continuous support and understanding when undertaking my research and writing my thesis.

I would like to express my sincere gratitude to my advisor (**Dr. Tafesse Asrat**) for imparting his knowledge and expertise in this research. His guidance and advice carried me through all the stages of writing my master's thesis.

Finally, I would also like to acknowledge and give my warmest thanks to my supervisor (**Dr. Tefera Terefe**) for his support of my thesis, for his patience, motivation, interest, and most importantly his guidance helped me at all times of writing, modeling, designing, and simulating the thesis document.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AF	Adaptive Fuzzy
AF-SMC	Adaptive Fuzzy Sliding Mode Controller
AF-STSMC	Adaptive Fuzzy Super-Twisting Sliding Mode Controller
DC	Direct current
DH	Denavit-Hartenberg
DOF	Degree of Freedom
EKF	Extended Kalman Filter
EMF	Electromotive Force
FK	Forward Kinematics
FLC	Fuzzy Logic Controller
FK	Forward Kinematics
HOSMC	High Order Sliding Mode Controller
IAE	Integral Absolute Error
IK	Inverse Kinematics
ISE	Integral Square Error
ITAE	Integral Time Absolute Error
ITSE	Integral Time Square Error
MIMO	Multi Input Multi Output
NB	Negative Big
NS	Negative Small
PB	Positive Big
PID	Proportional Integral Derivative Control
PMDC	Permanent Magnet Direct Current Motor
PS	Positive
RHR	Right-Hand Rule
RM	Robotic Manipulator
SMC	Sliding mode controller
ST-SMC	Super-Twisting Sliding Mode Controller
TSM	Terminal Sliding mode control
ZO	Zero

LIST OF SYMBOLS

$\hat{\theta}^T$	Adaptive Rule
$\dot{q}$	Angular Velocity
$\hat{F}\left(q,\dot{q},\ddot{q}/\Theta\right)$	Approximating Fuzzy System
I_a	Armature Current
L_a	Armature Inductance
R_a	Armature Resistance
V_a	Armature Voltage
ω_a	Back emf
$C(q,\dot{q})$	Coriolis Matrix
$C\theta_i$	Cosine of Joint Angle
B	Damping Coefficient
T_e	Electromagnetic Torque
e	Error
$F(q,\dot{q})$	Frictional Uncertainties
k_1, k_2	Gain of Super-Twisting Control
g	Gravitational Constant
$G(q)$	Gravitational Matrix
T_n^0	Homogeneous Transformation Matrix
V_c	Induced Voltage
$M(q)$	Inertia Matrix
J	Inertia of Rotor
J	Jacobian Matrix
θ_i	Joint Angle
d_i	Joint Offset
τ_d	Joint Torque Disturbance
K	Kinetic Energy
$L(q,\dot{q})$	LaGrange Formula
L_1	Length of Link 1
L_2	Length of Link 2
a_i	Link Length
α_i	Link Twist
V	Lyapunov Candidate Function
m_1	Mass of Link 1
m_2	Mass of Link 2
$\mu_{A_i}^{l_i}(x_i)$	Membership Function of x_i
I	Moment of Inertia
R_n^o	Orientation

q_1	Output Position of Link 1
q_2	Output Position of Link 2
P_n^o	Position
P	Potential Energy
q_{r1}	Reference Position of Link 1
q_{r2}	Reference Position of Link 2
Rot_{x,α_i}	Rotation Around x Axis
Rot_{z,θ_i}	Rotation Around z Axis
$\mathrm{sgn}(s)$	Signum of Sliding Surface
$s\theta_i$	Sine of Joint Angle
c	Sliding Coefficient
s	Sliding Surface
U_{sw}	Switching Controller Input
τ	Torque
k_t	Torque Constant
$T_{\omega'}$	Torque Due to Acceleration of Rotor
T_L	Torque Due to Mechanical Load
T_ω	Torque Due to Velocity of Rotor
$Trans_{z,d_i}$	Translation of z-axis
$Trans_{z,a_i}$	Translation of x-axis
$\dot{X}$	The Velocity of End Effector

ABSTRACT

Robotic manipulators are highly coupled, multi-input multi-output (MIMO) nonlinear systems with uncertainties and extremely time-varying dynamic capabilities. These features make it very difficult to control the trajectory of a robotic manipulator. To overcome this problem, the Adaptive Fuzzy Super-Twisting Sliding Mode Controller (AF-STSMC) is presented in this paper. It allows 2-DOF robot manipulators to accurately track trajectories while controlling uncertainty parameters that have an impact on the robot manipulator's ability to track the target joint angular position. The Lagrange Equation is solved to produce the dynamic model of a two-link robot manipulator. The Lyapunov direct method was used to investigate the stability of the system. The controller was implemented using MATLAB/Simulink software and its performance was evaluated. The simulation results demonstrated that the proposed controller eliminated chattering phenomena from the control effort and reduced the integral absolute tracking error (IAE) to 0.01465 rad and 0.003862 rad for link-1 and link-2 respectively. However, the integral absolute tracking error (IAE) of link-1 increased to 14.24 rad for the Sliding Mode Controller (SMC), 0.0977 rad for Super-Twisting Sliding Mode Controller (ST-SMC), and 0.06819 rad for Adaptive Fuzzy Sliding Mode Controller (AF-SMC). while the IAE of link-2 increased to 5.723 rad for the Sliding Mode Controller (SMC), 0.306 rad for Super-Twisting Sliding Mode Controller (ST-SMC), and 0.04391 rad for Adaptive Fuzzy Sliding Mode Controller (AF-SMC). The conventional controllers (SMC, ST-SMC, and AF-SMC) were designed for comparison with the proposed AF-STSMC. The simulation results verified that the designed controller (AF-STSMC) has a superior tracking performance, is robust, and is not sensitive to applied frictional uncertainties as compared to other three controllers.

Keywords: *– 2-DOF Robotic manipulator, Lagrange equation, Lyapunov direct method, uncertainties, AF-STSMC, SMC, ST-SMC, and MATLAB/Simulink.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background

Today, the use of robotic manipulators has a huge impact on production and industry all over the world. Robots can perform the simple repetitive tasks that human workers hate to do very efficiently and economically. Robots is particularly advantageous for jobs like painting, welding, and the removal or handling of hazardous materials. Recent advancements in vision technology and the integration of force-torque sensors have enabled manipulators to accomplish relatively difficult assembly and inspection tasks. These improvements make robotics systems more flexible and enable them to carry out a wide variety of tasks.

However, to benefit from this versatility, the robotic controller needs to be able to work precisely at high speeds without being affected by frictional uncertainty and disturbance. it is challenging to achieve all these qualities without introducing burdensome computations to the control algorithm. The classes of controllers that employ sliding mode and fuzzy logic theories provide an excellent compromise. The addition of some adaption rules is also highly beneficial in the situation of uncertainty. The introduction of the proposed controller used throughout this thesis is provided in the section that follows.

A robot manipulator is a programmable, multipurpose manipulator that is used to move about different types of objects, parts, items, or special devices through changeable programmed movements to perform different types of tasks (Utkin. V. I., 1977).

Sliding mode control (SMC) is a special type of control technology that can make a control system extremely robust to the changes in system parameters and external disturbances. In addition, the technique provides simple way to design the control law for a plant, linear or nonlinear. It was pioneered in the Soviet Union in the early 1950s by S. V. Emelyanov and his cohorts (Utkin. V. I., 1977) the technique did not receive wide attention in the western world until recently. In the 1970s, researchers discovered additional attractive properties of sliding mode control and have developed methods for control law design. The feasibility of the technique has not only been predicted by theory but has also been demonstrated by numerous computer simulations and hardware experiments. Therefore, the sliding mode control technique has become mature and ready to be applied.

However, some research indicates that it has significant drawbacks. The main problem with applying the sliding mode is the high-frequency oscillation around the sliding surface, the so-called chattering, which strongly reduces the control performance. As a result, the high-order sliding mode (HOSMC) control technique has been developed to enhance the performance of the conventional SMC (Angalaeswari, 2016). So, in this research, my main focus also became on the way to get rid of chattering accompanied with sliding mode control using one of HOSMC control technique.

In HOSMC, the designed sliding variable (s) have a relative degree greater than 1 concerning the control. The chattering issue is then solved by using the discontinuous control action on the higher derivative of s to start the sliding motion on s = 0. The HOSMC offers a higher-order precision and finite-time convergence of the states and exhibits improved robustness against various uncertainty. So, in this research, I used one of those HOSMCs which are super-twisting control.

Fuzzy logic system has been around 1965 when (Zadeh, 1965) laid the foundation of the so-called linguistic model. The fuzzy sets theory was proposed by L.A. Zadeh to provide a tool to help solve ill-defined problems. Fuzzy sets theory provides a systematic framework for dealing with different types of uncertainty within a single conceptual framework. L.A. Zadeh established the fundamental principles of fuzzy control in a paper that was published in 1973 and 1974. The idea of linguistic variables, fuzzy IF-THEN rules, fuzzy algorithms, the compositional rule of inference, and the execution of fuzzy instructions are a few of the concepts that are described in his paper.

One of the most research areas of fuzzy sets theory is the field of the fuzzy logic controller (FLC). Fuzzy logic control has many advantages which makes it a very attractive area. First, it is suitable for both linear and non-linear systems. Second, it allows for imprecise mathematical models and measuring sensors. Third, it is more robust than classical controllers. Fourth, it can combine both linguistic and crisp information in the same framework.

In recent years, Fuzzy logic has been effectively used to increase the effectiveness of SMC in controlling uncertain non-linear systems. Fuzzy SMC algorithms (FLSMCs) for robot control systems have already been developed in substantial numbers (Hollerbach, 1980; Medewar, 2015). So, in this research universal fuzzy approximation is used for uncertainty estimation.

1.2. Statement of the Problem

Robotic manipulators are designed to perform various sophisticated tasks, such as drilling, welding, assembly, painting, etc., with high accuracy as a prerequisite (Craig, 2005). However, due to the presence of inherent uncertainties and disturbances, developing a high-performance controller for a robotic manipulator is extremely difficult (Corless, 1993). If these uncertainties are not completely eliminated, they may have a negative impact on the desired control performance and positioning accuracy, which may further contribute to system instability (D. Zhao, 2010). Therefore, an efficient and intelligent control strategy is essentially needed to achieve high trajectory tracking accuracy and faster convergence.

To handle the above-aforementioned problem, the adaptive fuzzy super-twisting sliding mode controller is designed. To show the efficacy of the proposed method, a 2-DOF robotic manipulator is used to simulate the controller under unknown model uncertainty and external disturbances. Also, the key characteristics and contributions of the proposed controller are compared with the existing conventional controller such as Sliding Mode Controller (SMC), Super-Twisting Sliding Mode Controller (ST-SMC), and Adaptive Fuzzy Sliding Mode Controller (AF-SMC).

1.3. Objectives of Study

1.3.1. General Objective

The main aim of this thesis is to design an adaptive fuzzy super-twisting sliding mode controller for a 2-DOF robotic manipulator under uncertainties.

1.3.2. Specific Objectives

- Developing the dynamic model of a 2-link planar robotic manipulator using Euler-Lagrange formulation.
- To design an adaptive fuzzy super-twisting sliding mode controller (AF-STSMC).
- To design three other controllers for comparative analysis.
- To investigate the stability of the proposed controller using the Lyapunov stability theorem.
- Simulate and analyze the proposed controller through MATLAB/Simulink.
- Making relevant conclusions and recommendations for the system.

1.4. Scope of the thesis

These theses involve modeling and controlling a two-link planar robotic manipulator with uncertainties using an adaptive fuzzy super-twisting sliding mode controller in MATLAB/Simulink. To demonstrate the effectiveness of the proposed controller, these theses also involve a comparative analysis with a conventional controller such as Sliding Mode Controller (SMC), Super-Twisting Sliding Mode Controller (ST-SMC), and Adaptive Fuzzy Sliding Mode Controller (AF-SMC).

1.5. Limitation of the thesis

The limitation of this thesis is that it solely addresses computer simulation, not implementation. This is because the implementation of the prototype is difficult. Because the materials of the system components are expensive and some components are not found in the country. Along with-it additional time is also required.

1.6. Significance of the thesis

A robotic system is essential to the development of the industry, which is currently expanding throughout Ethiopia. Today, a wide range of robots are used in a variety of industries and for a variety of purposes. However, all robots are required to be incredibly accurate in every industry. Unfortunately, their accuracy suffers as a result of uncertainty and disturbance. Both of these factors practically always exist in different systems due to various reasons. It is essential to use a more intelligent control strategy since the conventional linear controlling method does not provide a satisfactory response to these conditions. That is why in this thesis, I intended to develop an adaptive fuzzy super-twisting sliding mode controller for a 2-DOF robotic manipulator.

In industrial applications, typical PID controllers are used to control such complex systems. PID controllers are simple to use, only require three parameters to be tuned, and provide acceptable tracking performance for a robot. However, the linear PID controller is inadequate for achieving the desired tracking performance of nonlinear dynamic systems. A variety of studies have been conducted to improve PID control using a nonlinear controller, and many methods for developing nonlinear controllers including FLC, SMC, and FSMC have also been proposed. Although each of these three controllers has been successfully implemented in several applications, however, they do still have significant drawbacks. So, the proposed controller overcomes that limitation.

The robot I intend to use for this thesis work mostly serves for pick-and-place functions. The pick-and-place robots, as we all know, must be precise, or else they risk damaging the items they are carrying. So, the obvious solution to this issue is the use of an adaptive (self-tuning) controlling method.

This paper presents the modeling and design of a controller for a two-link planar robotic manipulator which requires minimum effort with minimum error. Due to the system's highly coupled MIMO and highly time-varying dynamic nature, controlling the trajectory of robotic manipulators is a challenging job. Furthermore, there are always a lot of uncertainties in the system model, such as parameter uncertainty and external disturbances, which results in the unstable performance of the control system. Consequently, it is crucial to model and evaluate robot manipulators using appropriate control techniques.

1.7. Thesis Organization

This thesis is organized into six chapters.

Chapter 1: - Presents the background of the research work, statement of the problems, objectives, scope, limitations and significance of the research.

Chapter 2: - This chapter presents a literature review on two-wheel self-balancing electric scooters and related work.

Chapter 3: - Describes the methods followed in this thesis and the mathematical model of the 2-degree of freedom robotic manipulator and Actuator dynamics (PMDC motors) are modeled.

Chapter 4: - This chapter includes the design of the proposed controller (AF-STSMC) and the design of other three controllers (SMC, ST-SMC and AF-SMC). It also includes the stability analysis of the proposed control system using Lyapunov direct method.

Chapter 5: - Presents the proposed results and discussions.

Chapter 6: - This chapter includes conclusions and recommendations.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Introduction

In today's environment, humans and robots collaborate to fulfill their given duties. An assistance robot is a robot that is self-contained and capable of performing duties on its own. Industries, military operations, and the medical field are just a few of the domains where robots are now being deployed. Working in high-temperature environments or doing jobs such as defusing bombs or handling molten metal can be dangerous. As a result, robots can conduct these kinds of dangerous tasks in place of humans. (Yagna Jadeja, 2019)

A robotic arm is a machine that is very similar to a human hand, it consists of a combination of links attached in series or parallel. It can be controlled by programming it to accomplish a particular task. (Rahul gautam, 2017) The joints of the manipulator connect the links that lead to the displacement which is either translational or rotational. A kinematic chain is a group of links joined together by joints that permit relative motion. The end effector is the terminating part of this kinematic chain and it can be considered as the hand of a human. The below points will give you a brief introduction to robotic manipulators.

2.2. Parts of manipulators

It's a rigid-arm mechanism made up of a sequence of segments, mostly sliding or jointing, known as cross-slides, that grab and move objects with varying degrees of freedom. It's also an electronically controlled mechanism, with each segment interacting with its workspace to execute distinct functions. They're also referred to as robotic arms. The majority of robot manipulator research and study focuses on the orientations and positions of the many sections that make up the robot manipulators (Vidyasagar, 2008). They are primarily made up of a collection of links, joints, and end-effectors. The following is a summary of these components.

Links: These are the rigid parts of the manipulator.

Joints: Part of the mechanism that connects the two links. Joints permit controlled or limited relative motion between the links. The table mentioned below provides brief information related to different types of joints.

Table 2.1 Types of joints

No	Type	Description
1	Revolute	These type of joints permits rotation motion about an axis
2	Cylindrical	The types of joints which permit the rotation and translation of an axis
3	Prismatic	Permits relative translation motion around an axis
4	Spherical	Gives three degrees of rotational motion freedom around the center of the joint. They are also referred to as ball socket joint
5	Planar	Provides translational motion on a plane and rotation motion around an axis perpendicular to the plane

End effector: It is the part of the manipulator, which interacts with its workspace to perform various tasks.

2.2.1. Classification of Manipulators

The power source or mechanism by which the joints are actuated, the geometry or kinematic structure, the intended application area, or the method of control can all be used to classify robot manipulators. I will categorize them here based on their kinematic structure.

Serial manipulators: links and joints are connected to form an open kinematic chain. All joints are actuated. The controller designed in this thesis is also applied to these types of manipulators.

Figure 2.1 Serial manipulator

Parallel manipulators: links and joints are connected to form a closed kinematic chain. Contain unactuated joints.

Figure 2.2 Parallel robot

Hybrid manipulators: combining serial and parallel manipulators.

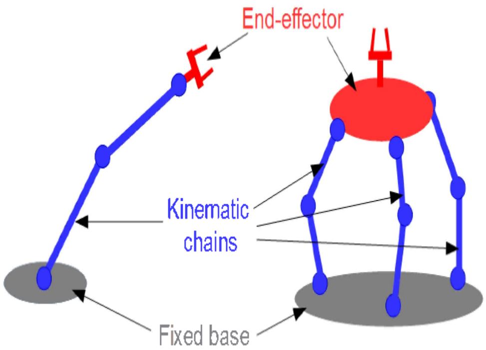

Figure 2.3 Hybrid robot

2.3. Related works

The most essential area of research right now is the trajectory tracking control of robotic manipulators. Many researchers have contributed significantly to the solution of the trajectory tracking problem thus far. However, because of the inherent presence of uncertainties and disturbances, a trajectory tracking control problem has recently become a study unit. Many scholars have presented a vast variety of solutions for trajectory tracking control problems in the literature, some of which are listed here.

Modeling and control of a 2-link robot using a typical PID controller were presented by Nasr M. Ghaleb and Ayman A. Aly in 2018. (Ghaleb, 2018) To move the robot's arm, the author applied a permanent magnet DC (PMDC) motor. They've also modeled the actuator's dynamics. However, instead of proposing a method to compensate for all of the dynamics, they just simply employed a traditional PID controller. Because of this, the system's response initially indicated a significant overshoot. Also, they haven't taken into account unwanted disturbances and uncertainties.

In 2018, Getachew (Getachew, 2018) proposed sliding mode control of a two-degree-of-freedom (2-DOF) robot utilizing a permanent magnet synchronous motor. in this literature, the author was able to achieve a decent system response at a disturbance of approximately 25%, but at a disturbance of more than 40%, the system response became more chattering. the author was also able to demonstrate that a sliding mode controller had a better response than a traditional PID controller. To minimize the chattering associated with the system, the author employed the saturation function instead of the signum function for sliding mode control, but it is still not sufficient. instead, he could have utilized someone else's high order sliding mode control (HOSMC) technique to reduce chattering at any percent of input amplitude, but the authors didn't. As a result, the author's suggested that in the future, another chattering removal method to be adopted, which could overcome any degree of disturbance introduced to a robot.

In 2011 Ohri et al. (Ohri, 2011) the authors present a comparison of the conventional PID controller and sliding mode control (SMC) for robotic manipulators. Both techniques have good performance, but the SMC is insensitive to parametric variations and has good disturbance rejection. The simulation results of the study show that the performance of the SMC is better than PID under both sine and cosine trajectory. As the payload is changed or during uncertainty, the tracking error in the PID controller is increased, but in the case of SMC, the performance remains the same, which clearly shows that the SMC is more robust than the PID controller. In this paper, the authors assume that the actuators do not have dynamics of their own, and arbitrary forces can be commanded at the joint of the robotic manipulator for simplicity. The results obtained were promising, but the authors didn't consider the actuator dynamics and ignored the effects of joint friction which brought about low controller performance.

In the year 2015, S.a.B.Jolly Shah (Shah, 2015) presented the design of a computed torque control mechanism for a two-link robot manipulator in his paper. and the author demonstrated the method's validity for the first 10 seconds and the second 120 seconds. Position errors are less than 1% for practically the first 10 seconds, but they are growing as time goes on. While velocity inaccuracies are higher in the first 10 seconds, they gradually decrease and eventually disappear as time passes. For the first 10 seconds, the error between the derived and expected position of link 1 and link 2 is virtually negligible, but as time passes, the error increases, although it remains under control. Finally, the authors suggested that a computed torque controller is adequate for controlling a 2 DOF robot for a few seconds, but that if it is used for a long time, it will generate a performance error.

In 2020, Anisa Ulya Darajat (Darajat, 2020) presented a design and control of a two-link robot manipulator with uncertainty parameters using self-tuning sliding mode control. Sliding Mode Control (SMC) is a robust nonlinear control method, but its performance in tracking the desired joint angular position against uncertainty parameters such as mass parameter changes caused by robot activity is still insufficient. In this instance, the author suggested a least-squares-based Self-Tuning method to enhance SMC tracking performance of the target joints angular position against uncertainty parameters such as mass parameter changes caused by robot activity. To demonstrate the effectiveness and robustness of the suggested method, the author used computer simulation to conduct a comparative analysis with the traditional sliding mode control signum function and saturation function.

The authors compared the SMC saturation function to the SMC signum function in a computer simulation. And according to findings, the conventional SMC method with the signum function can control the angular position, but with a steady-state error of 0.48 radian when the mass parameter changes, and the signum function of the conventional Sliding Mode Control has a chattering torque input, which consumes more energy or power. Meanwhile, when the mass parameter changes, the Conventional Sliding Mode Control Saturation function is unable to maintain the joints' angular position, as evidenced by the angular position of both joints going to infinity.

Finally, the author can conclude from the simulation results of the proposed methods that the proposed Self-tuning Sliding Mode Control is more efficient and robust than the conventional Sliding Mode Control when uncertainty parameters such as mass parameter changes caused by robot manipulator lifting load are present.

Different gaps are mentioned in the literature reviewed above. As previously stated, the dynamics of a practical manipulator change over time due to a variety of factors. Unwanted uncertainty and disturbance are one of the most difficult challenges. Chattering is also associated with traditional SMC controllers. This paper developed adaptive fuzzy super-twisted sliding mode control (AFSTSMC) to overcome all of these challenges. Adaptive fuzzy was employed in this study to evaluate the undesirable uncertainty and disturbance. To eliminate chattering and maintain the system’s robustness, super twisting sliding mode control is also implemented.

CHAPTER THREE

3. MATERIALS AND METHODS

3.1. Chapter Overview

This chapter focuses primarily on the mathematical modeling of a 2-DOF planar robotic manipulator, as well as the general block diagram and brief description. This chapter also focuses on robot kinematics and dynamics and actuator dynamics.

3.2. Materials

MATLAB/2017a, Microsoft Office 2013, and the Math Type 6.0 equation editor are all the materials used in this study. Simulink and technical toolboxes are part of the MATLAB program. To modify the thesis documentation, Microsoft Office is used. In Microsoft Office Word and PowerPoint presentation tools, Math Type 6.0 is used to write mathematical formulas and equations.

3.3. Methods

To accomplish the paper's general and specific objectives, the author focused on studying literature bout the theoretical and structural background of the robotic manipulator and its controller. Following that, I analyzed the performance of other controllers to see if the proposed controller (AF-STSMC) could overcome the disadvantages of existing linear and nonlinear controllers.

Since identifying the appropriate dynamic model equations is crucial when constructing a robotic controller. First, dynamic equations for the robot manipulator were established. This begins with the Lagrange equation, followed by forward and inverse kinematics, dynamic analysis and forces, and kinematic and potential energy derivations. Then Following the development of dynamic models, the suggested controller (Self-Tuning FSMC) for a 2-DOF robot manipulator was theoretically designed. The proposed controller's stability was then tested using the Lyapunov stability theorem.

Finally, the suggested method's performance is evaluated and compared to that of a conventional controller using MATLAB software, and the results of the simulation are analyzed and interpreted.

In general, the method followed throughout the research study can be summarized in the following flow chart in figure 3.1.

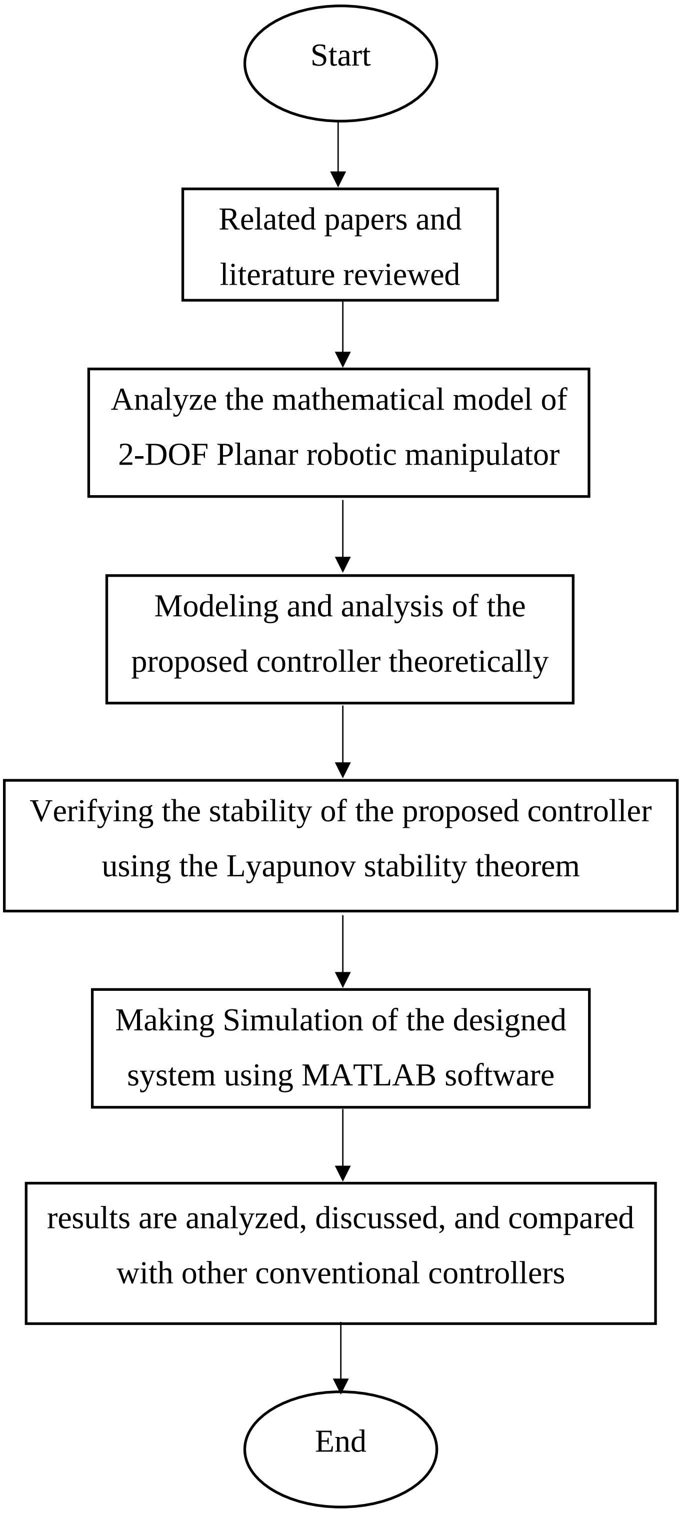

Figure 3.1 Workflow

3.4. System Block Diagram and Its Description

Figure 3.2 illustrates the overall block diagram of the proposed control system.

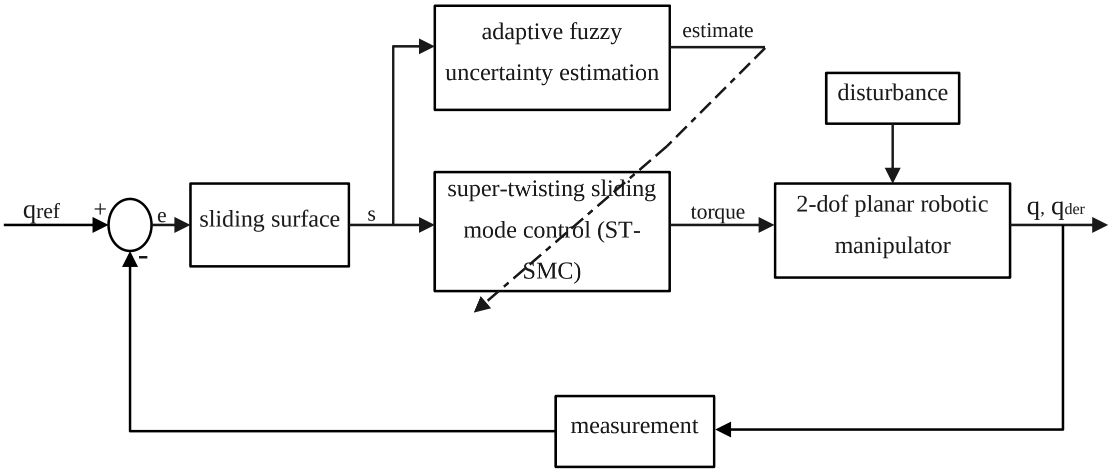

Figure 3.2 Overall block diagram of the proposed control system

The main aim of this work, as stated above, is to create an appropriate control law that allows the system's control output to track the reference trajectory rapidly and efficiently while also estimating system uncertainty parameters online. To achieve this, the overall system is first supplied with a reference trajectory position as a set point, as shown in the block diagram above. The difference between the reference input position trajectory and the 2-DOF uncertain robot manipulator output position trajectory, referred to as trajectory tracking error is then used to build the sliding surface. Sliding surface design is also another feature of our controller system design that helps to increase controller performance by reducing or eliminating the time it takes to reach the sliding phase. After that, the sliding surface is sent into a robust non-linear controller, which is known as sliding mode control as well as to adaptive fuzzy estimator.

Sliding mode control is an effective tool for designing reliable controllers for sophisticated high-order nonlinear dynamic plants that operate in uncertain conditions. Unfortunately, chattering becomes one of the most significant barriers to the success of the sliding mode controller. However, the author proposed the super-twisting sliding mode controller to overcome the chattering problem in this work. At the same time, the adaptive fuzzy approximation technique was employed to correct for unknown errors and disturbances entering the system.

Finally, the output of both controllers is fed to the robot manipulator resulting in an output that is chatter reduced and has improved tracking accuracy.

3.5. Kinematics modeling

The kinematics of a robot manipulator describes the relationship between the motion of the joints of the manipulator and the resulting motion of the rigid bodies which form the robot. (Murray, 1994) Robot kinematics is the study of the motion of kinematic chains with several degrees of freedom that makes up the structure of robotic systems. The emphasis on geometry leads to the modeling of the robot's linkages as rigid bodies and the assumption that its joints can only do rotation or translation. (Richard, 1981)

Hence, manipulator kinematics is the field of science that investigates the motion of manipulator links without regard to the force that causes it. (Craig, 2005) From the viewpoint of spatial geometry, kinematics investigates the position, orientation, velocity, and acceleration of the robot manipulator. For the purpose of describing the geometry, each link of the manipulator is given a link frame based on the Denavit-Hartenberg (DH) description. The kinematics are divided into forward (FK) and Inverse (IK) kinematics. The following section focuses on the forward and inverse kinematics of a two-link robotic manipulator.

3.6. Forward kinematics (FK)

A manipulator is made up of serial links from the base frame through the end effector that are attached to one another by revolute or prismatic joints. Forward kinematics is the process of expressing the position and orientation of an end-effector in terms of joint variables. One should employ a proper kinematics model in order to have forward kinematics for a robot mechanism in a systematic manner. The most popular approach for describing the robot kinematics is the Denavit-Hartenberg (DH) method, which has four parameters.

When we simply define FK, the forward kinematics of a robot determines the configuration of the end effector given the configuration of each pair of adjacent links of a robot. Forward kinematics can be expressed simply using the equation below.

$$x = f(\theta) \tag{3.1}$$

So, in the following section, the Denavit-Hartenberg (DH) approach is explained, and the forward kinematics of a 2-DOF planar robot manipulator are developed using the DH parameter.

3.6.1. Denavit-Hartenberg (DH) approach

Denavit and Hartenberg showed that a general transformation between two joints requires four parameters. (Denavit, 1955) These parameters known as the Denavit-Hartenberg (DH) parameters have become the standard for describing robot kinematics. These parameters are mentioned below in the table with its definition.

Table 3.1 Definition of DH parameters *(Craig, 2005)*

DH Parameters	**Its Definition**
Joint angle (θ_i)	The rotation about z_{i-1} axis needed to make axis x_{i-1} parallel with the axis x_i .
Joint offset (d_i)	The translation along z_{i-1} axis needed to make axis x_{i-1} intersect with axis x_i .
Link Length (a_i)	The translation along x_i axis needed to make axis z_{i-1} intersect with axis z_i .
Link twist (α_i)	The rotation about x_i axis needed to make axis z_{i-1} parallel with axis z_i

In order to establish DH parameters, each joint must have a coordinate frame attached to it before anything else. It is crucial to follow the frame assignment rule for assigning coordinate frames. When assigning coordinate frames, manipulators must have at least one extra frame than the number of joints. That means the end effector must have one frame.

In addition, the frame rule is provided below to allocate all three axes on each frame.

- The z-axis must be the axis of rotation or the direction of motion
- The x-axis must be perpendicular to the z-axis of the frame that comes before it.
- The x-axis must intersect the z-axis of the frame that comes before it.
- The y-axis must be drawn so that the whole frames follow the right-hand rule (RHR).

As a result, this paper used the above frame assignment rule to allocate coordinate frames to 2-DOF planar robot manipulators.

The DH frame assignment of a 2-dof planar robotic manipulator is then shown below in detail.

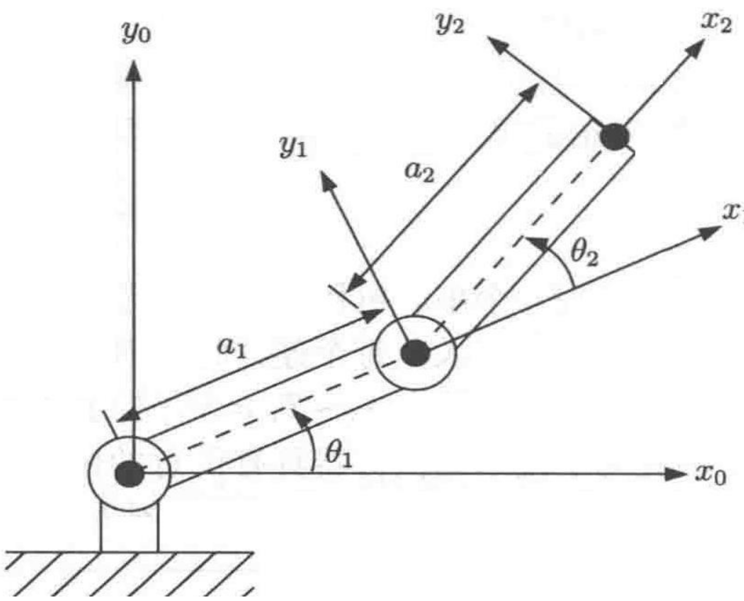

Figure 3.3 Frame assignments for the 2-DOF robot arm

The Forward kinematics of a robotic arm is determined by a group of parameters called Denavit-Hartenberg (DH) parameters which are used for deriving the homogenous transformation matrices between the different frames assigned on the robot arm structure. (Darajat, 2020) The DH parameters for a two degree of freedom robotic arm are defined as follows:

Table 3.2 DH parameters for the 2-DOF robotic arm

link	Link Length (a_i)	Link twist (α_i)	Joint offset (d_i)	Joint angle (θ_i)
1	L1	0	0	θ_1
2	L2	0	0	θ_2

In the Denavit-Hartenberg (DH) convention, each homogeneous transformation is represented as a product of four basic transformations.

$$T_n^0 = Rot_{z,\theta_i} * Trans_{z,d_i} * Trans_{x,a_i} * Rot_{x,\alpha_i} \tag{3.2}$$

$$T_n^0 = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$T_n^0 = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$T_n^0 = \begin{bmatrix} R_n^0 & P_n^0 \\ 0_{1*3} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

So,

$$R_n^0 = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix} \text{ and } P_n^0 = \begin{bmatrix} a_i c\theta_i \\ a_i s\theta_i \\ d_i \end{bmatrix} \tag{3.6}$$

Therefore, R_n^0 and P_n^0 means the orientation and position of the end effector respectively.

The homogenous transformation matrices for the 2-DOF robotic arm are derived as follows:

Firstly, based on the DH parameter table 3 mentioned above, Let's start by calculating the transformation matrices for each link.

$$T_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 c\alpha_1 & s\theta_1 s\alpha_1 & a_1 c\theta_1 \\ s\theta_1 & c\theta_1 c\alpha_1 & -c\theta_1 s\alpha_1 & a_1 s\theta_1 \\ 0 & s\alpha_1 & c\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{and}$$
$$T_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 c\alpha_2 & s\theta_2 s\alpha_2 & a_2 c\theta_2 \\ s\theta_2 & c\theta_2 c\alpha_2 & -c\theta_2 s\alpha_2 & a_2 s\theta_2 \\ 0 & s\alpha_2 & c\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

So now by substituting the parameters from the DH table.

$$T_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & L1c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & L1s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{and } T_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & L2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & L2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Finally, the overall transformation matrices of the 2-DOF robotic arm are derived as follows:

$$T_2^0 = T_1^0 * T_2^1 \tag{3.9}$$

$$T_2^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & L1c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & L1s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & L2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & L2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

$$T_2^0 = \begin{bmatrix} c\theta_{12} & -s\theta_{12} & 0 & L1c\theta_1 + L2c\theta_{12} \\ s\theta_{12} & c\theta_{12} & 0 & L1s\theta_1 + L2s\theta_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

The homogenous transformation matrix defined in Equation (3.11) is the one that defines the forward kinematics of the 2-DOF robotic manipulator. From this matrix, it is possible to obtain the position and orientation of the end-effector from the individual joint angle. The first three rows and columns of this overall transformation matrix in Equation (3.11) represent the end-effector orientation matrix, whereas the first three rows and the last columns of this overall transformation matrix in Equation (3.11) represent the end-effector position matrix.

The following Equation (3.12) -(3.14) are the forward kinematics (position of end-effector) of the 2-DOF robotic manipulator. also, for simplicity let's use q instead of θ.

$$x_2 = L1\cos(q_1) + L2\cos(q_1 + q_2) \tag{3.12}$$

$$y_2 = L1\sin(q_1) + L2\sin(q_1 + q_2) \tag{3.13}$$

$$z_2 = 0 \tag{3.14}$$

The following Equation (3.15) is the forward kinematics (orientation of end-effector) of the 2-DOF robotic manipulator.

$$R_o = \begin{bmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

By taking the first derivative of the above two expressions Equation (3.12) and (3.13)

$$\dot{x}_2 = -L1\dot{q}_1 \sin(q_1) - L2(\dot{q}_1 + \dot{q}_2)\sin(q_1 + q_2) \tag{3.16}$$

$$\dot{y}_2 = L1\dot{q}_1 \cos(q_1) + L2(\dot{q}_1 + \dot{q}_2)\cos(q_1 + q_2) \tag{3.17}$$

When both Equation (3.16) and (3.17) expressed in matrix form

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -L1\sin(q_1) - L2\sin(q_1+q_2) & -L2\sin(q_1+q_2) \\ L1\cos(q_1) + L2\cos(q_1+q_2) & L2\cos(q_1+q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \tag{3.18}$$

$$\dot{X} = J \cdot \dot{q} \text{ and so, } J = \begin{bmatrix} -L1\sin(q_1) - L2\sin(q_1+q_2) & -L2\sin(q_1+q_2) \\ L1\cos(q_1) + L2\cos(q_1+q_2) & L2\cos(q_1+q_2) \end{bmatrix} \tag{3.19}$$

Where $\dot{X}$ is the velocity of end effectors, J is the Jacobian matrix and $\dot{q}$ is the angular velocity of end effectors.

3.7. Inverse kinematics

Inverse kinematics is a mathematical method for calculating the joint positions required to set the end effector of a robot in a certain position and orientation. Programming a robot to perform tasks requires a reliable inverse kinematic solution. The inverse kinematic can be calculated using a variety of different methods. but, in this section, this thesis used the geometrical approach to solve the inverse kinematics equations for the 2-DOF robotic manipulator as given below.

By squaring and summing the forward kinematics of Equation (3.12) and (3.13) we have

$$x_2^2 = L_1^2\cos^2(q_1) + L_2^2\cos^2(q_1+q_2) + 2L_1L_2\cos(q_1)\cos(q_1+q_2) \tag{3.20}$$

$$y_2^2 = L_1^2\sin^2(q_1) + L_2^2\sin^2(q_1+q_2) + 2L_1L_2\sin(q_1)\sin(q_1+q_2) \tag{3.21}$$

$$x_2^2 + y_2^2 = L_1^2 + L_2^2 + 2L_1L_2\cos(\theta_2) \tag{3.22}$$

Now, by rearranging and taking the inverse of Equation (3.22)

$$\cos(\theta_2) = \frac{x_2^2 + y_2^2 - (L_1^2 + L_2^2)}{2L_1L_2} \tag{3.23}$$

$$\theta_2 = \cos^{-1}\left(\frac{x_2^2 + y_2^2 - (L_1^2 + L_2^2)}{2L_1L_2}\right) \tag{3.24}$$

As seen above, getting θ_2 via a geometry technique is simple, but getting θ_1 requires a special manipulator arrangement, as illustrated in the image below

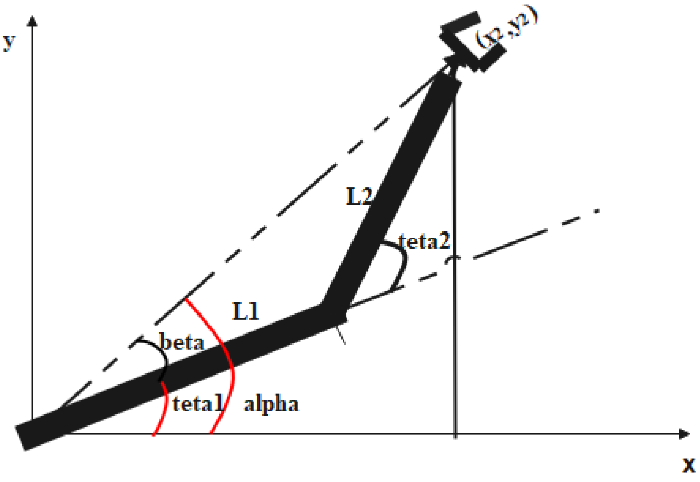

Figure 3.4 Geometric arrangement of the 2-DOF robot manipulator

Now, the following section demonstrates how to determine theta1 by utilizing trigonometric and geometric techniques in Figure 3.4.

$$\theta_1 = \alpha - \beta \tag{3.25}$$

$$\tan(\beta) = \frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)} \text{ and so, } \beta = \tan^{-1}\left(\frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)}\right) \tag{3.26}$$

$$\alpha = \tan^{-1}\left(\frac{y_2}{x_2}\right) \tag{3.27}$$

$$\theta_1 = \tan^{-1}\left(\frac{y_2}{x_2}\right) - \tan^{-1}\left(\frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)}\right) \tag{3.28}$$

By substituting Equation (3.24) into (3.26) it is possible to express β in terms of position (x2, y2).

Remember that inverse kinematics means expressing joint variables in terms of joint position, therefore the inverse kinematics of a 2-DOF robotic manipulator are the above two equations such as Equation (3.24) and (3.28).

3.8. Dynamic model of 2-DOF Robot Manipulator

The dynamic model of the robot manipulator is its response in terms of the movements of its arms to the forces applied by electrical, pneumatic, or any other type actuator at its joints. (R. Kelly, 2006) A robot's dynamic model is an ordinary differential equation in which the variable corresponds to the vector of positions and velocities, which might be in joint coordinates or operational coordinates. In this thesis, we focus on the dynamic model in joint space.

The dynamics of the robot arm concerns with the mathematical formulations of the robot arm equations of motion. A manipulator's dynamic equations of motion are a set of mathematical equations that describe the manipulator's dynamic behavior. These types of equations motion are important for computer simulations of robot arm motion, the development of appropriate control equations for a robot arm, and the structure of a robot arm. This section describes the dynamics of manipulators for the control purpose.

In general, a manipulator's dynamic performance is directly relying on the efficiency of its control algorithms and its dynamic model. Obtaining dynamic models of the physical robot arm system and then specifying appropriate control laws or methods to accomplish the required system response and performance is the main control problem.

By using known physical laws such as the laws of Newtonian mechanics and Lagrangian mechanics, the actual dynamic model of a robot arm can be obtained. But, the disadvantage of Newtonian mechanics is that the complexity of the analysis increases with the number of joints in the robot. Due to this case, Lagrangian approach is mostly used in deriving the complicated mathematical model of a robot manipulator (Hollerbach, 1980). Therefore, we have used the Euler-Lagrange method in this paper to develop a 2-DOF robotic manipulator dynamics.

3.8.1. Langragian formulation

A manipulator's equations of motion are essentially a description of the relationship between the input joint torques and the motion of the arm linkage. This relationship is needed in a robotic simulation because the control system only specifies the torques that should be applied to the joints of the manipulator. The exact motion that results from the application of the torques needs to be found to insure an accurate simulation. Unfortunately, the precise model of a manipulator can never be found. However, as long as the major dynamic effects are included, a reasonable estimation is sufficient.

The Lagrangian formulation is a popular method for determining the equations of motion of a dynamic system. (B., 1965) (Symon, 1971) The Lagrangian $L(q,\dot{q})$ of a robot manipulator of n DOF is the difference between its kinetic energy K and its potential energy P, that is

$$L(q,\dot{q}) = K(q,\dot{q}) - P(q) \tag{3.29}$$

Then, the Lagrange equations of motion for a manipulator of n DOF, are given by (Ed., 1989)

$$\frac{d}{dt}\left[\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right]-\frac{\partial L(q,\dot{q})}{\partial q}=\tau \tag{3.30}$$

Or in the equivalent form by

$$\frac{d}{dt}\left[\frac{\partial L(q,\dot{q})}{\partial \dot{q}_i}\right]-\frac{\partial L(q,\dot{q})}{\partial q_i}=\tau_i, \qquad i=1,\cdots,n \tag{3.31}$$

where τ_i corresponds to the external forces and torques (delivered by the actuators) at each joint, q_i corresponds to the angular positions of each joint, and $\dot{q}_i$ corresponds to the angular velocity of each joint.

Equation (3.30) is considered to be the equation of motion of the dynamic system. Using a set of partial derivatives of the Lagrangian with respect to the angle and its derivative, the torque τ is obtained. This torque represents the external force applied to a joint so that it moves to a certain position. It is the force in charge of executing the rotatory movement of the revolution joints. There are two different torques in this robotic arm. Each torque is calculated, but it needs to be done step by step.

The derivation of robot dynamics using Lagrange's equations can be reduced to the following stages:

1. Computation of the kinetic energy function $K(q,\dot{q})$
2. Computation of the potential energy function $P(q)$
3. Computation of the Lagrangian (3.28) $L(q,\dot{q})$
4. Development of Lagrange's equations (3.30)

In the rest of the section, we present methods that illustrate the process of obtaining the 2-DOF robot dynamics by the use of Lagrange's equations of motion. The above four stages will be applied to a simple model of a 2-link robot manipulator shown in Figure 8. The mass of each link, m, and m2, is lumped at the end of a massless rod of lengths l1 and l2. The links are connected to each other and to the base via a revolute joint. the moments of inertia are I1 and I2 for link 1 and link 2 respectively. The generalized coordinates, q, describe the position of each of the revolute joints, and the generalized torques τ are applied at the joints by actuators.

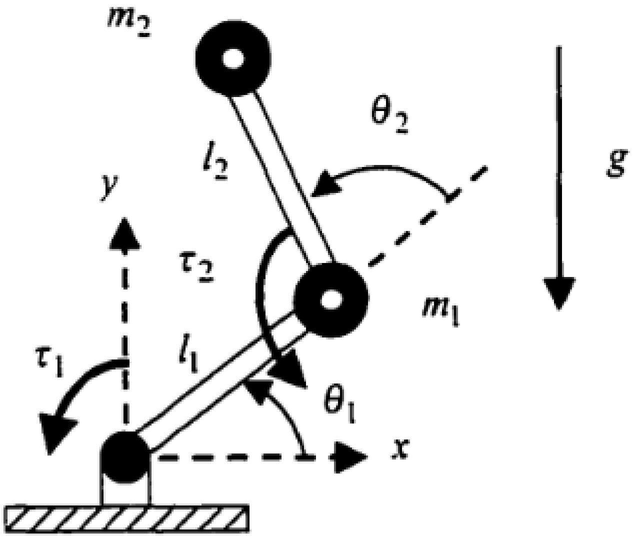

Figure 3.5 Dynamic configuration of the 2-DOF robotic manipulator

The joint variables for the two-link manipulator are

$$q = [q_1\, q_2] \tag{3.32}$$

and the generalized force vector is

$$\tau = [\tau_1\, \tau_2] \tag{3.33}$$

with torques τ_1 and τ_2 supplied by the actuators.

3.8.2. Kinetic and Potential Energy

For link1 the velocity square is

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 \tag{3.34}$$

From Equation (3.8) we have

$$x_1 = l_1 \cos(q_1) \text{ and } y_1 = l_1 \sin(q_1) \tag{3.35}$$

By taking the first derivative

$$\dot{x}_1 = -l_1 \dot{q}_1 \sin(q_1) \text{ and } \dot{y}_1 = l_1 \dot{q}_1 \cos(q_1) \tag{3.36}$$

By substituting Equation (3.36) to (3.34)

$$v_1^2 = l_1^2 \dot{q}_1^2 \tag{3.37}$$

Therefore, the kinetic energy and potential energy of link1 are

$$K_1 = \frac{1}{2}m_1 l_1^2 \dot{q}_1^2 + \frac{1}{2}I_1 \dot{q}_1^2 \tag{3.38}$$

$$P_1 = m_1 g y_1 = m_1 g l_1 \sin(q_1) \tag{3.39}$$

For link2 the velocity square is

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \tag{3.40}$$

By substituting Equation (3.16) and (3.17) to Equation (3.40)

$$v_2^2 = (-l_1\dot{q}_1 \sin(q_1) - l_2(\dot{q}_1 + \dot{q}_2)\sin(q_1 + q_2))^2 + (l_1\dot{q}_1 \cos(q_1) + l_2(\dot{q}_1 + \dot{q}_2)\cos(q_1 + q_2))^2$$

$$v_2^2 = (l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2))\dot{q}_1^2 + (2l_1 l_2 \cos(q_2) + 2l_2^2)\dot{q}_1\dot{q}_2 + l_2^2\dot{q}_2^2 \tag{3.41}$$

Now by using Equation (3.41) let us compute Kinetic energy

$$K_2 = \frac{1}{2}m_2 v_2^2 + \frac{1}{2}I_2\dot{q}_2^2$$

$$K_2 = \frac{1}{2}m_2((l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2))\dot{q}_1^2 + (2l_1 l_2 \cos(q_2) + 2l_2^2)\dot{q}_1\dot{q}_2 + l_2^2\dot{q}_2^2) + \frac{1}{2}I_2\dot{q}_2^2$$

$$K_2 = \frac{1}{2}m_2(l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2))\dot{q}_1^2 + m_2(l_1 l_2 \cos(q_2) + l_2^2)\dot{q}_1\dot{q}_2 + (\frac{1}{2}m_2 l_2^2 + \frac{1}{2}I_2)\dot{q}_2^2 \tag{3.42}$$

The potential energy of link2

$$P_2 = m_2 g y_2 \tag{3.43}$$

By substituting Equation (3.13) into (3.43)

$$P_2 = m_2 g(l_1 \sin(q_1) + l_2 \sin(q_1 + q_2)) \tag{3.44}$$

$$P_2 = m_2 g l_1 \sin(q_1) + m_2 g l_2 \sin(q_1 + q_2)) \tag{3.45}$$

3.8.3. Lagrange's Equation

The Lagrangian for the entire robot arm is

$$L = K - P = K_1 + K_2 - P_1 - P_2 \tag{3.46}$$

By substituting Equation (3.38), (3.39), (3.42), and (3.45) into Equation (3.46)

$$\begin{aligned} L &= \frac{1}{2}m_1 l_1^2 \dot{q}_1^2 + \frac{1}{2}I_1 \dot{q}_1^2 + \frac{1}{2}m_2(l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2))\dot{q}_1^2 + m_2(l_1 l_2 \cos(q_2) + l_2^2)\dot{q}_1 \dot{q}_2 \\ &+ (\frac{1}{2}m_2 l_2^2 + \frac{1}{2}I_2)\dot{q}_2^2 - m_1 g l_1 \sin(q_1) - m_2 g l_1 \sin(q_1) - m_2 g l_2 \sin(q_1 + q_2)) \end{aligned} \tag{3.47}$$

The Lagrangian equation of motion for link 1 is

$$\tau_1 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1} \tag{3.48}$$

To compute the above equation Equation (3.48) let us find the derivatives of Lagrangian

$$\frac{\partial L}{\partial \dot{q}_1} = ((m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) + I_1)\dot{q}_1 + (m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2))\dot{q}_2 \tag{3.49}$$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) &= ((m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) + I_1)\ddot{q}_1 + (m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2))\ddot{q}_2 \\ &- m_2 l_1 l_2 \sin(q_2)(\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) \end{aligned} \tag{3.50}$$

$$\frac{\partial L}{\partial q_1} = -(m_2 l_2 g \cos(q_1 + q_2) + (m_1 + m_2) l_1 g \cos(q_1)) \tag{3.51}$$

By inserting Equation (3.49), Equation (3.50), and Equation (3.51) in to Equation (3.48) the Lagrangian equation of motion for link 1 becomes

$$\begin{aligned} \tau_1 &= ((m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) + I_1)\ddot{q}_1 + (m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2))\ddot{q}_2 \\ &- m_2 l_1 l_2 \sin(q_2)(\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + m_2 l_2 g \cos(q_1 + q_2) + (m_1 + m_2) l_1 g \cos(q_1)) \end{aligned} \tag{3.52}$$

Again, the Lagrangian equation of motion for link 2 are

$$\tau_2 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2} \tag{3.53}$$

To compute the above equation Equation (3.53) let us to find the derivatives of Lagrangian

$$\frac{\partial L}{\partial \dot{q}_2} = (m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2))\dot{q}_1 + (m_2 l_2^2 + I_2)\dot{q}_2 \tag{3.54}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) = (m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2))\ddot{q}_1 + (m_2 l_2^2 + I_2)\ddot{q}_2 - m_2 l_1 l_2 \sin(q_2)(\dot{q}_1 \dot{q}_2) \tag{3.55}$$

$$\frac{\partial L}{\partial q_2} = -(m_2 l_1 l_2 \sin(q_2)\dot{q}_1^2 + m_2 l_1 l_2 \sin(q_2)\dot{q}_1 \dot{q}_2 + m_2 l_2 g \cos(q_1 + q_2)) \tag{3.56}$$

By inserting Equation (3.54), (3.55) and Equation (3.56) in to Equation (3.53) the Lagrangian equation of motion for link 2 becomes

$$\tau_2 = (m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2))\ddot{q}_1 + (m_2 l_2^2 + I_2)\ddot{q}_2 + m_2 l_1 l_2 \sin(q_2)\dot{q}_1^2 + m_2 l_2 g \cos(q_1 + q_2) \tag{3.57}$$

Therefore, according to the above derivation, the 2-DOF robot dynamics are two coupled non-linear differential Equation (3.52) and (3.57).

3.8.4. Manipulator Dynamics

The equations of motion for the manipulator are most conveniently written in matrix form. So, let's put the above two equation of motion Equation (3.52) and (3.57) into matrix form.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -m_2 l_1 l_2 \sin(q_2)(\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) \\ m_2 l_1 l_2 \sin(q_2)\dot{q}_1^2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2) l_1 g \cos(q_1) + m_2 l_2 g \cos(q_1 + q_2) \\ m_2 l_2 g \cos(q_1 + q_2) \end{bmatrix} \tag{3.58}$$

Once both torques are expressed in matrix form, it is recommended to re-arrange them into the following form of a non-linear equation: (Vidyasagar, 2008)

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \tag{3.59}$$

Where $q \in R^n$ represents the link positions and n represents the joint numbers, which in this case is 2. $M(q)$ is the inertia matrix, $C(q, \dot{q})$ represents the centripetal Coriolis matrix and $G(q)$ represents the gravitational matrix. They are defined as:

$$M(q) = \begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 + I_2 \end{bmatrix} \tag{3.60}$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 l_1 l_2 \sin(q_2)(\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) \\ m_2 l_1 l_2 \sin(q_2)\dot{q}_1^2 \end{bmatrix} \tag{3.61}$$

$$G(q)=\begin{bmatrix}(m_1+m_2)l_1 g\cos(q_1)+m_2 l_2 g\cos(q_1+q_2)\\ m_2 l_2 g\cos(q_1+q_2)\end{bmatrix} \quad (3.62)$$

By considering external disturbance and uncertainties, the above dynamic manipulator equation (Eq. (3.59)) can be rewritten as

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+F(q,\dot{q})=\tau+\tau_d \quad (3.63)$$

Where $F(q,\dot{q})$ is the unstructured uncertainties including friction and other disturbance and τ_d denote joint torque disturbance.

For convenience, the following assumptions are considered about robot dynamics of equation. (M. Zhihong, 1994)

Assumption 1: Inertia matrix $M(q)$ is bounded for all q even in presence of uncertainty and a symmetric positive definite matrix.

$$m_1\|x\|^2 \le x^T m(q)x \le m_2\|x\|^2 ;\forall x\in R^n \quad (3.64)$$

where m_1, m_2 are known positive scalar constants, $x\in R^n$ is a vector, $\| \;\|$ denotes the Euclidean vector norm.

Assumption 2: $(\dot{M}(q)-2C(q,\dot{q}))$ is a skew symmetric matrix. That is

$$x^T[\dot{M}(q)-2C(q,\dot{q})]x=0 \quad (3.65)$$

For any vector $x\in R^n$.

Assumption 3: The external torque disturbance τ_d is bounded and unknown but Lipschitz continuous. Also, system modeling uncertainty $F(q,\dot{q})$ is assumed to be bounded. (Swarup, march 16, 2016)

3.9. Actuator's Dynamics

In a number of literatures, the design of a controller for the robot manipulator control system without introducing the actuator dynamics has been presented, (Wai, 2003) while actuator dynamics plays an important role in the overall robot dynamics. The effectiveness of a robot's performance is determined by the type of actuators it uses. Due to their improved performance, reliability, simplicity, low cost, and variable speed control, dc drives play a significant role in modern industrial drives. Therefore, in this paper, a PMDC motor is used to drive all joints of the robotic manipulator.

Permanent magnet direct current (PMDC) motors are widely used in industrial application and proving ground due to simplicity, low cost and efficiency. (Angalaeswari, 2016) (Medewar, 2015) These types of motors need no excitation current; hence no energy consumption due to field fluxes producing. Additionally, PMDC motors do not need a winding field, therefore their size is smaller than that of ordinary DC motors and their price is substantially lower. (Moussavi, 2012)

The mathematical model of a permanent magnet direct current (PMDC) motor is described in the next section. the goal in the development of the mathematical model is to relate the voltage applied to the armature to the velocity of the motor. In most of previous research, the control input has been a torque or force, but in this case, author used the voltage of the actuator from the derived mathematical model instead of torque or force. Two balance equations can be developed in the following section by considering the electrical and mechanical characteristics of the actuators.

3.9.1. Electrical characteristics

The electrical equivalent of the armature coil is an inductance (La) in series with a resistance (Ra) in series with an induced voltage (Vc) that opposes the voltage source. The induced voltage is generated by the rotation of the electrical coil through the fixed flux lines of the permanent magnets. This voltage is often referred to as the back emf (electromotive force). (Ghazanfar Shahgholian, 2010)

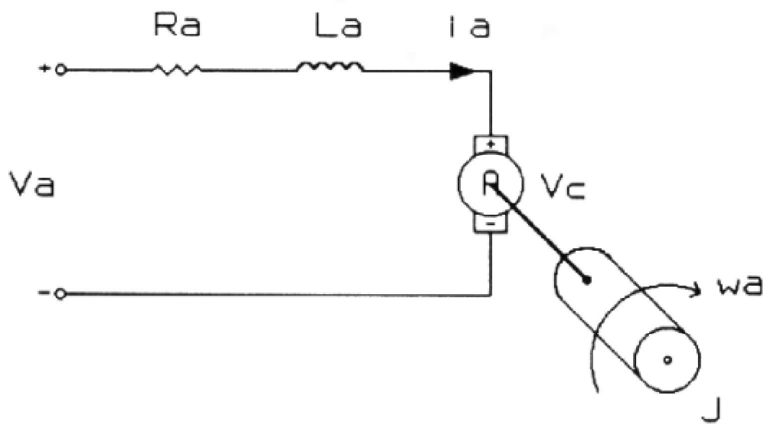

Figure 3.6 Equivalent Electrical Circuit of PMDC motor

A differential equation for the equivalent circuit can be derived by using Kirchoff's voltage law around the electrical loop. Kirchoff's voltage law states that the sum of all voltages around a loop must equal zero, or

$$V_a - V_{Ra} - V_{La} - V_c = 0 \tag{3.65}$$

According to Ohm's law, the voltage across the resistor can be represented as

$$V_{Ra} = i_a R_a \tag{3.66}$$

where i_a is the armature current. The voltage across the inductor is proportional to the change of current through the coil with respect to time and can be written as

$$V_{La} = L_a \frac{d}{dt} i_a \tag{3.67}$$

where L_a is the inductance of the armature coil. Finally, the back emf can be written as

$$V_c = k_v \omega_a \tag{3.68}$$

where k_v is the velocity constant determined by the flux density of the permanent magnets, the reluctance of the iron core of the armature, and the number of turns of the armature winding. ω_a is the rotational velocity of the armature.

Substituting Equation (3.66), (3.67), and (3.68) into Equation (3.65) gives the following differential equation:

$$V_a - i_a R_a - L_a \frac{d}{dt} i_a - k_v \omega_a = 0 \tag{3.69}$$

3.9.2. Mechanical Characteristics

Performing an energy balance on the system, the sum of the torques of the motor must equal zero. Therefore,

$$T_e - T_{\omega'} - T_\omega - T_L = 0 \tag{3.70}$$

where T_e is the electromagnetic torque, $T_{\omega'}$ is the torque due to rotational acceleration of the rotor, T_ω is the torque produced from the velocity of the rotor, and T_L is the torque of the mechanical load.

The electromagnetic torque is proportional to the current through the armature winding and can be written as

$$T_e = k_t i_a \tag{3.71}$$

where k_t is the torque constant and like the velocity constant is dependent on the flux density of the fixed magnets, the reluctance of the iron core, and the number of turns in the armature winding. $T_{\omega'}$ can be written as

$$T_{\omega'} = J \frac{d}{dt} \omega_a \tag{3.72}$$

where J is the inertia of the rotor and the equivalent mechanical load. The torque associated with the velocity is written as

$$T_{\omega} = B\omega_a \tag{3.73}$$

where B is the damping coefficient associated with the mechanical rotational system of the machine.

Substituting Equation (3.71), (3.72) and (3.73) into Equation (3.70) gives the following differential equation:

$$k_t i_a - J \frac{d}{dt} \omega_a - B\omega_a - T_L = 0 \tag{3.74}$$

3.9.3. Transfer Function Block Diagram

A block diagram for the system can be developed from the differential equations given Equation (3.69) and (3.74). Taking the Laplace transform and re-arranging of each equation gives:

$$sI_a(s) - I_a(0) = -\frac{R_a}{L_a} I_a(s) - \frac{k_v}{L_a} \omega_a(s) + \frac{1}{L_a} V_a(s) \tag{3.75}$$

$$s\omega_a(s) - \omega_a(0) = \frac{k_t}{J} I_a(s) - \frac{B}{J} \omega_a(s) - \frac{1}{J} T_L(s) \tag{3.76}$$

If perturbations around some-steady state values are considered, the initial conditions go to zero and all the variables become some change around a reference state, and the equations can be expressed as follows:

$$I_a(s) = \frac{-k_v \omega_a(s) + V_a(s)}{L_a s + R_a} \tag{3.77}$$

$$\omega_a(s) = \frac{-k_t I_a(s) - T_L(s)}{Js + B} \tag{3.78}$$

The block diagram obtained from these equations for a permanent magnet dc motor (PMDC) is shown in Figure 3.7.

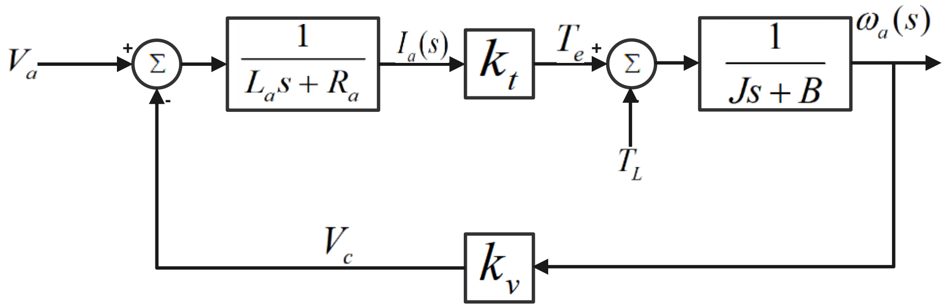

Figure 3.7 Block diagram representation of Equation (3.77) and (3.78)

Therefore, the overall transfer function between the output angular velocity and input applied voltage given in Equation (3.79).

$$T(s) = \frac{\omega(s)}{V_a(s)} = \frac{k}{(R + Ls)(Js + b) + k^2} \tag{3.79}$$

For a negligibly small L, Equation (3.79) can be simplified as

$$T(s) = \frac{k}{(JL)s^2 + (bL + RJ)s + (Rb + k^2)} \tag{3.80}$$

In this thesis, two permanent magnet DC motors have been used to drive the joint of 2-DOF robot manipulators. The electrical and mechanical parameters of a PMDC motor are depicted in Table 3.3.

Table 3.3 Permanent magnet dc (PMDC) motor parameters *(Ghazanfar Shahgholian, 2010)*

Parameters	Joint-1	Joint-2
Moment of inertia (J)	$0.0236\, kg.m^2$	$0.0236\, kg.m^2$
Armature resistance (R_a)	$7.72\,\Omega$	$7.72\,\Omega$
Armature inductance (L_a)	0.16273H	0.16273H
Damping coefficient (B)	$0.003\, N.m.s/r$	$0.003\, N.m.s/r$
EMF or torque constant (K)	$1.25\, Nm$	$1.25\, Nm$

Finally, by inserting each parameter of Table 3.3 in to Equation (3.80). we have

$$T(s) = \frac{1.25}{0.0038s^2 + 0.1827s + 1.5856} \tag{3.81}$$

CHAPTER FOUR

4. CONTROLLER DESIGN

4.1. Introduction

Since robot manipulators are complex systems with nonlinear dynamics, they require a nonlinear controller technique for optimal functionality and performance (Vidyasagar, 2008). As a result, in this chapter, we'll focus on a class of nonlinear control method known as Sliding Mode controller (SMC) for trajectory tracking control of the robotic arm manipulator as discussed in the previous chapter. This control technique has been employed in a variety of applications for optimal and reliable performance over the years, including flight control, robotic manipulators, tracking systems, the electrical and mechanical fields, adaptive techniques, and many others (Li, 1991; Sastry, 1983; V. Utkin, 1999).

Apart from the benefits, it has one disadvantage, known as chattering. The oscillations of the input control signal are caused by the discontinuous behavior of the control mechanism. These undesired oscillations are extremely dangerous since they have the potential to entirely damage the system and controller performance. To overcome the above-mentioned problem, we propose a super-twisting sliding mode controller, which is a higher-order sliding mode controller. The super-twisting sliding mode control (ST-SMC) has the ability to reduce chattering and also is robust against varying parameters (C. Mu, 2013).

In this thesis, a 2-degrees-of-freedom (DOF) robotic manipulator is employed to test the suggested controller. However, due to the inherent presence of undesirable uncertainties and disturbances, it is difficult to control these plants in practice. The author proposed an adaptive method to deal with this problem. The adaptive control philosophy is commonly used to control uncertain systems (Frank L.Lewis). In the adaptive approach, the author will design a controller that tries to "learn" the system's uncertain parameters.

However, since the uncertainty parameters are unknown in practice, the author introduced the fuzzy system approximation technique, which is a universal approximation technique. A fuzzy system can approximate any nonlinear function over a compact set with arbitrary precision, according to previous research on the universal approximation theorem (Wang, 1997). Therefore, this fuzzy system approximation is applied to estimate the unknown uncertainty parameters based on fuzzy logic rule. The adaptive controller will then compensate based on the estimated uncertainty.

In general, the author proposes the adaptive fuzzy super-twisting sliding mode controller (AFST-SMC) by integrating the super-twisting sliding mode controller (ST-SMC) with an adaptive fuzzy (AF) system. The rest of this chapter explains how the proposed controller is developed and how it is implemented to a 2-DOF robotic manipulator.

4.2. Design of Super-Twisting Sliding Mode Controller (ST-SMC)

We first establish the expression of the tracking error dynamics in order to solve the stated control problem using the super-twisting sliding mode approach. The tracking errors defined below are the outputs we would like to force to zero.

$$\begin{aligned} e_1 &= q_1 - q_{r1} \\ e_2 &= q_2 - q_{r2} \\ e &= [e_1, e_2] \end{aligned} \tag{4.1}$$

The key stage in SMC design is designing a sliding surface so that the plant restricted to the sliding surface has a desired system response. This means that state variables of the plant dynamics are constrained to satisfy another set of equations which define the so-called switching surface. Therefore, the sliding mode surface of the proposed controller can be designed as follows:

$$s = ce + \dot{e} \tag{4.2}$$

where c is the constant of the sliding mode surface; e and $\dot{e}$ are the tracking error and the tracking error rate, respectively. The expression of e and $\dot{e}$ is as follows.

$$e = q - q_r = \left[q_1 - q_{r1}, q_2 - q_{r2}\right]^T \tag{4.3}$$

$$\dot{e} = \dot{q} - \dot{q}_r = \left[\dot{q}_1 - \dot{q}_{r1}, \dot{q}_2 - \dot{q}_{r2}\right]^T \tag{4.4}$$

Where q_1, q_2, q_{r1} and q_{r2} are the output trajectory of link-1, the output trajectory of link-2, the reference trajectory of link-1 and the reference trajectory of link-2 respectively.

The super-twisting sliding mode control algorithm is one of the outstanding robust control algorithms which handle a system with a relative degree equal to one (Peng, 2021). According to the super-twisting control algorithm, the switching control law is designed as follows:

$$u_{sw} = -\begin{bmatrix} k_{11}\sqrt{|s_1|}\,\mathrm{sgn}(s_1) \\ k_{12}\sqrt{|s_2|}\,\mathrm{sgn}(s_2) \end{bmatrix} - \begin{bmatrix} k_{21}\int \mathrm{sgn}(s_1)d\tau \\ k_{22}\int \mathrm{sgn}(s_2)d\tau \end{bmatrix} \tag{4.5}$$

Set k_1, k_2 as $k_1 = \begin{bmatrix} k_{11} & 0 \\ 0 & k_{12} \end{bmatrix}$, $k_2 = \begin{bmatrix} k_{21} & 0 \\ 0 & k_{22} \end{bmatrix}$.

Note k_1, k_2 are the user-defined gains of the super-twisting sliding mode controller satisfy $k_{1i} > 0$ and $k_{2i} > 0$. where $i = 1, 2$.

$\mathrm{sgn}(s)$ is a symbolic function, which is defined as:

$$\mathrm{sgn}(s) = \begin{cases} -1 & if\ s < 0 \\ 0 & if\ \ s = 0 \\ +1 & if\ s > 0 \end{cases} \tag{4.6}$$

Then the final control law can be obtained as follows:

$$u_{sw} = -k_1\sqrt{|s|}\,\mathrm{sgn}(s) - k_2\int \mathrm{sgn}(s)d\tau \tag{4.7}$$

4.3. Design of Fuzzy Logic-Based Adaptive Laws

4.3.1. Uncertainty Approximation Using of Fuzzy System

A brief introduction for the approximation principle of fuzzy system is given in this part, if $f(x)$ is unknown, we can replace $f(x)$ with the fuzzy system $\hat{f}(x)$ to realize feedback control. $\hat{f}(x)$ is the unknown part of the system model (uncertainty) used to approximate $f(x)$ according to the universal approximation property of fuzzy system. Designing 5 fuzzy sets for the input x_1, x_2 of the fuzzy system respectively and setting $n = 2$, $i = 1, 2,\ p_1 = p_2 = 5,$ there will be 25 $(p_1 \times p_2 = 25)$ fuzzy rules. The following two steps are the universal approximation theorem which are used to construct a fuzzy system $\hat{f}(x/\theta)$.

Step-1. Defining fuzzy sets for the variable $x_i (i = 1, 2)$ as $A_i^{l_i} (l_i = 1, 2, 3, 4, 5),$ the number of fuzzy sets is p_i.

Step-2. Constructing fuzzy systems $\hat{f}(x/\theta)$ with 25 ($\prod_{i=1}^{n} p_i = p_1 \times p_2 = 25$) fuzzy rules, then the j_{th} fuzzy rule is:

$$R^{(j)}: \text{IF } x_1 \text{ is } A_1^{l_1} \text{ and } x_2 \text{ is } A_1^{l_1} \text{ THEN } \hat{f} \text{ is } B^{l_1 l_2} \tag{4.8}$$

Where $l_i = 1,2,3,4,5,\ \ i = 1,2,\ j = 1,2,\cdots,25,$ and $B^{l_1 l_2}$ is the fuzzy set defined in the consequent part. Whole fuzzy rules are expressed as:

$$\begin{aligned} &R^{(1)}: \text{IF } x_1 \text{ is } A_1^1 \text{ and } x_2 \text{ is } A_2^1 \text{ THEN } \hat{f} \text{ is } B^1 \\ &\qquad\qquad \vdots \\ &R^{(25)}: \text{IF } x_1 \text{ is } A_1^5 \text{ and } x_2 \text{ is } A_2^5 \text{ THEN } \hat{f} \text{ is } B^{25} \end{aligned} \tag{4.9}$$

The following four steps are adopted in the process of fuzzy inference (Kosko, NOVEMBER,1994).

Step-1. Using product inference engine to realize the pre-requisite inference of rules, the result of inference is $\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i)$.

Step-2. Adopt singleton fuzzifier to solve $\bar{y}_f^{l_1 l_2}$.

Step-3. Using product inference engine to realize the inference of the precondition and the conclusion of the rule, the result of the inference is $\bar{y}_f^{l_1 l_2}(\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i))$. Performing the union operations on all fuzzy rules. Then the output of the fuzzy system is $\sum_{l_1=1}^{5}\sum_{l_2=1}^{5} \bar{y}_f^{l_1 l_2}(\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i))$.

Step-4. The output of the fuzzy system is obtained by using the average de-fuzzifier:

$$\hat{f}(x/\theta) = \frac{\sum_{l_1=1}^{5}\sum_{l_2=1}^{5} \bar{y}_f^{l_1 l_2}\left(\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i)\right)}{\sum_{l_1=1}^{5}\sum_{l_2=1}^{5}\left(\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i)\right)} \tag{4.10}$$

where $\mu_{A_i}^{l_i}(x_i)$ is the membership grade of x_i.

Let $\bar{y}_f^{l_1 l_2}$ be a free parameter and put it in the set of $\theta \in R^{(25)}$. Introducing the fuzzy basis vector $\xi(x)$, then Equation (4.10) can be modified as

$$\hat{f}(x/\theta) = \hat{\theta}^T \xi(x) \tag{4.11}$$

Where $\hat{\theta}^T$ is the adaptive law based on Lyapunov stability theory. $\xi(x)$ is a 25-dimensional ($\prod_{i=1}^{n} p_i = p_1 \times p_2 = 25$) fuzzy basis vector and the $l_1 l_2^{th}$ element is

$$\xi_{l_1 l_2}(x) = \frac{\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i)}{\sum_{l_1=1}^{5}\sum_{l_2=1}^{5}\left(\prod_{i=1}^{2} \mu_{A_i}^{l_i}(x_i)\right)} \tag{4.12}$$

4.3.2. Design of Adaptive Fuzzy Super-Twisting Sliding Mode Controller (AFST-SMC)

There have been significant research efforts on the adaptive fuzzy control for non-linear systems (LX, 1993). In this section an adaptive fuzzy super-twisting sliding mode control algorithm for a class of continuous time uncertain robotic manipulator is derived. The unknown uncertainties that are associated with robot manipulator are approximated by the fuzzy system with a set of fuzzy if-then rules whose parameters are adjusted on-line according to some adaptive laws. This aids in controlling the output of a robotic manipulator to track a given trajectory effectively by compensating the uncertainty in the system. The Lyapunov synthesis approach is used to develop an adaptive control algorithm which is based on the adaptive fuzzy model. As a result, the methods for designing the adaptive fuzzy super-twisting sliding mode controller for a 2-DOF robotic manipulator will be detailed in the next part.

As previously stated, the dynamic model of the uncertain 2-dof robotic manipulator takes the form of the equation below.

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q}) \tag{4.13}$$

Suppose $M(q)$, $C(q,\dot{q})$, and $G(q)$ are known, and all the states are measured.

Let the desired output be q_d, and denote the tracking error as

$$e = q_d - q \tag{4.14}$$

Select a sliding surface variable as

$$s = ce + \dot{e} \tag{4.15}$$

Where c is a positive constant, e is tracking error.

Denote

$$\dot{q}_r = \dot{q}_d - ce \tag{4.16}$$

Based on the above Equation (4.15) and (4.16), it is possible to select Lyapunov candidate function as

$$V(t) = \frac{1}{2}\left(s^T M s + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \tilde{\Theta}_i \right) \tag{4.17}$$

Where Θ^*_i is the desired adaptive parameter, $\tilde{\Theta}_i = \Theta^*_i - \Theta_i$ is error of adaptive parameter and $\Gamma_i > 0$.

Now by taking the first derivative of the Lyapunov function (Eq. (4.17)) we have

$$\dot{V}(t) = s^T M \dot{s} + \frac{1}{2} s^T \dot{M} s + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i \tag{4.18}$$

By substituting Equation (4.14) and (4.16) in to Equation (4.15) and by rearranging

$$\begin{aligned} s &= \dot{e} + ce \\ &= \dot{q} - \dot{q}_d + ce \\ &= \dot{q} + (ce - \dot{q}_d) \\ s &= \dot{q} - \dot{q}_r \\ \dot{s} &= \ddot{q} - \ddot{q}_r \end{aligned} \tag{4.19}$$

multiplying both side of Equation (4.19) by inertia matrix (M)

$$M\dot{s} = M(\ddot{q} - \ddot{q}_r) \tag{4.20}$$

From the general torque equation of robot (Eq. (4.13))

$$\ddot{q} = M^{-1}(\tau - C\dot{q} - G - F) \tag{4.21}$$

By applying Equation (4.21) in to (4.20) and rearranging

$$\begin{aligned} M\dot{s} &= M(M^{-1}(\tau - C\dot{q} - G - F) - \ddot{q}_r) \\ M\dot{s} &= (\tau - C\dot{q} - G - F) - M\ddot{q}_r \end{aligned} \tag{4.22}$$

Then the Equation (4.22) is applied to the derivative of Lyapunov function Equation (4.18)

$$\begin{aligned} \dot{V}(t) &= s^T (M\dot{s}) + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i \\ &= s^T (\tau - C\dot{q} - G - F - M\ddot{q}_r) + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i \\ \dot{V}(t) &= -s^T (-\tau + C\dot{q} + G + F + M\ddot{q}_r) + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i \end{aligned} \tag{4.23}$$

where F or $F(q,\dot{q},\ddot{q})$ is unknown non-linear function or unwanted uncertainty entering to the robot manipulator. Fuzzy system $\hat{F}(q,\dot{q},\ddot{q}/\Theta)$ is adopted to approximate $F(q,\dot{q},\ddot{q})$ using a technique of universal fuzzy approximation rule as defined above in section 4.3.1.

So now, the overall adaptive fuzzy super-twisting sliding mode controller is designed as:

$$\tau = M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) + \hat{F}(q,\dot{q},\ddot{q}/\Theta) - k_1|s|^{\frac{1}{2}}\operatorname{sgn}(s) - \int_0^t k_2 \operatorname{sgn}(s)dt \tag{4.24}$$

where $k_1 > 0$ and $k_2 > 0$.

Also, the approximated uncertainty is obtained as:

$$\hat{F}(q,\dot{q},\ddot{q}/\Theta) = \begin{bmatrix} \hat{F}_1(q,\dot{q},\ddot{q}/\Theta_1) \\ \hat{F}_2(q,\dot{q},\ddot{q}/\Theta_2) \end{bmatrix} = \begin{bmatrix} \Theta_1^T\xi(q,\dot{q},\ddot{q}) \\ \Theta_2^T\xi(q,\dot{q},\ddot{q}) \end{bmatrix} \tag{4.25}$$

where Θ_1 and Θ_2 are the adaptive parameter which are obtained from adaptive rule in section below.

4.4. Stability Analysis

At this stage the stability of the overall system is analyzed using the Lyapunov direct method. According to the Lyapunov stability criterion, If $\dot{V}$ is semi-negative definite, it guarantees the global asymptotic stability of the system and ensures that the controlled system can reach stable state in limited time (La Salle J., 1961). The semi-negative definite matrices of $\dot{V}$ guarantee that V and s are bounded. According to Lasalle theorem and its corollaries, $s(t)$ will tend to zero, and then e and $\dot{e}$ will also tend to zero in limited time, which guarantees the robustness and stability of the system.

The stability analysis and proof will be given in this part:

Now the fuzzy approximating error is

$$\omega = F(q,\dot{q},\ddot{q}) - \hat{F}(q,\dot{q},\ddot{q}/\Theta^*) \tag{4.26}$$

From Eq. (4.23) and (4.24), we have

$$\dot{V}(t) = -s^T(F(q,\dot{q},\ddot{q}) - \hat{F}(q,\dot{q},\ddot{q}/\Theta) + k_1|s|^{\frac{1}{2}}\operatorname{sgn}(s) + \int_0^t k_2 \operatorname{sgn}(s)dt) + \sum_{i=1}^{2}\tilde{\Theta}_i^T\Gamma_i\dot{\tilde{\Theta}}_i \tag{4.27}$$

By inserting $F(q,\dot{q},\ddot{q}/\Theta^*) - \hat{F}(q,\dot{q},\ddot{q}/\Theta^*)$ in to Eq. (4.27) and rearranging

$$\dot{V}(t) = -s^T(F(q,\dot{q},\ddot{q}) - \hat{F}(q,\dot{q},\ddot{q}/\Theta) + F(q,\dot{q},\ddot{q}/\Theta^*) - \hat{F}(q,\dot{q},\ddot{q}/\Theta^*) + k_1|s|^{\frac{1}{2}}\operatorname{sgn}(s)$$

$$+\int_0^t k_2 \operatorname{sgn}(s)dt) + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i .$$

$$= -s^T ((F(q,\dot{q},\ddot{q}) - \hat{F}(q,\dot{q},\ddot{q}/\Theta^*)) + (F(q,\dot{q},\ddot{q}/\Theta^*) - \hat{F}(q,\dot{q},\ddot{q}/\Theta)) - k_1 |s|^{\frac{1}{2}} \operatorname{sgn}(s)$$

$$-\int_0^t k_2 \operatorname{sgn}(s)dt) + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i$$

$$= -s^T (\omega + \tilde{\Theta}_i^T \xi(q,\dot{q},\ddot{q}) + k_1 |s|^{\frac{1}{2}} \operatorname{sgn}(s) + \int_0^t k_2 \operatorname{sgn}(s)dt) + \sum_{i=1}^{2} \tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i$$

$$= -s^T \omega - k_1 |s|^{\frac{1}{2}} (s^T \operatorname{sgn}(s)) - \int_0^t k_2 (s^T \operatorname{sgn}(s))dt) + \sum_{i=1}^{2} (\tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i - s_i \tilde{\Theta}_i^T \xi(q,\dot{q},\ddot{q}))$$

$$\dot{V}(t) = -s^T \omega - k_1 |s|^{\frac{1}{2}} \|s\| - \int_0^t k_2 \|s\| dt + \sum_{i=1}^{2} (\tilde{\Theta}_i^T \Gamma_i \dot{\tilde{\Theta}}_i - s_i \tilde{\Theta}_i^T \xi(q,\dot{q},\ddot{q})) \tag{4.28}$$

where $\tilde{\Theta} = \Theta^* - \Theta$, $\xi(q,\dot{q},\ddot{q})$ is the fuzzy system.

In order to ensure $\dot{V} \le 0$, the adaptive rules of the super-twisting sliding mode controller are designed as:

$$\dot{\Theta}_i = -\Gamma_i^{-1} s_i \xi(q,\dot{q},\ddot{q}),\ i = 1,2,\cdots,n \tag{4.29}$$

To check $\dot{V} \le 0$, let us to apply Eq. (4.29) in to Eq. (4.28) as

$$\dot{V}(t) = -s^T \omega - k_1 |s|^{\frac{1}{2}} \|s\| - \int_0^t k_2 \|s\| dt + \sum_{i=1}^{2} (\tilde{\Theta}_i^T \Gamma_i (-\Gamma_i^{-1} s_i \xi(q,\dot{q},\ddot{q})) - s_i \tilde{\Theta}_i^T \xi(q,\dot{q},\ddot{q}))$$

$$\dot{V}(t) = -s^T \omega - k_1 |s|^{\frac{1}{2}} \|s\| - \int_0^t k_2 \|s\| dt + \sum_{i=1}^{2} (s_i \tilde{\Theta}_i^T \xi(q,\dot{q},\ddot{q}) - s_i \tilde{\Theta}_i^T \xi(q,\dot{q},\ddot{q}))$$

$$\dot{V}(t) = -s^T \omega - k_1 |s|^{\frac{1}{2}} \|s\| - \int_0^t k_2 \|s\| dt \le 0$$

$$\dot{V}(t) \le 0 \tag{4.30}$$

Thus, it is proved that the proposed Adaptive fuzzy Super-twisting Sliding Mode Controller (AFST-SMC) guarantees that the trajectory tracking error in robotic system will reach zero in finite time under the presence of uncertainties and disturbance.

Suppose the joint number of manipulators is n, and if MIMO fuzzy system $\hat{F}(q,\dot{q},\ddot{q}/\Theta)$ is adopted to approximate to $F(q,\dot{q},\ddot{q})$, then for each joint, the number of input variables is 3. If k membership functions are designed for each input variable, then the whole number of rules is k^{3n} (Ham, APRIL, 2000). For instance, the joint number of the manipulator used in this thesis is 2, the number of input variable is 3, there are 5 membership functions, then the whole rule number is $5^{3\times2} = 5^6 = 15625$. The two many rules will bring out excessive computation. In order to decrease the number of fuzzy rules, independent design should be adopted with respect to $F(q,\dot{q},\ddot{q},t)$.

Therefore, for this thesis case author assume that the uncertainties on the robot manipulator results from the friction. Because the friction is relative to velocity, the fuzzy system which approximates friction can be written as $\hat{F}(\dot{q}/\theta)$. So, the adaptive fuzzy super-twisting sliding mode controller is designed as:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \hat{F}(\dot{q}/\theta) - k_1|s|^{\frac{1}{2}}\operatorname{sgn}(s) - k_2\int_0^t \operatorname{sgn}(s)dt \tag{4.31}$$

Adaptive Rule is

$$\dot{\theta}_i = -\Gamma_i^{-1} s_i \xi(q,\dot{q},\ddot{q}), \quad i = 1,2,\cdots,n \tag{4.32}$$

And the fuzzy system is

$$\hat{F}(\dot{q}/\theta) = \begin{bmatrix} \hat{F}_1(\dot{q}_1) \\ \hat{F}_2(\dot{q}_2) \end{bmatrix} = \begin{bmatrix} \theta_1^T \xi^1(\dot{q}_1) \\ \theta_2^T \xi^2(\dot{q}_2) \end{bmatrix} \tag{4.33}$$

Previous research has shown that sliding mode controllers are more robust than the traditional PID controller due to this case author did not design the PID controller for making comparison (Getachew, 2018; Darajat, 2020; Abbas, 2018). In order to illustrate the superiority of the proposed controller, comparative analysis have been made with some conventional sliding mode controllers such as Sliding Mode Controller (SMC), Super-Twisting Sliding Mode Controller (ST-SMC) and Adaptive Fuzzy Sliding Mode Controller (AF-SMC)

Sliding Mode Controller (SMC) is

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) - k\operatorname{sgn}(s) \tag{4.34}$$

Super-Twisting Sliding Mode Controller (ST-SMC) is

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) - k_1|s|^{\frac{1}{2}}\operatorname{sgn}(s) - k_2\int_0^t \operatorname{sgn}(s)dt \tag{4.34}$$

Adaptive Fuzzy Sliding Mode Controller (AF-SMC) is

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \hat{F}(\dot{q}/\theta) - k\operatorname{sgn}(s) \tag{4.35}$$

The adaptive fuzzy compensation rule is also the same in this conventional controller

CHAPTER FIVE

5. SIMULATION RESULTS AND DISCUSSION

5.1. Introduction

In this chapter, the efficacy of the proposed AF-STSMC is analyzed through simulation on a 2-DOF RM (MIMO) system. The simulation works were implemented in MATLAB/Simulink environment using ODE45 (Dormand-prince) solver with variable step. Simulink is a graphical programming environment for modeling, simulating and analyzing multidomain dynamical systems. A Simulink file consists of interconnected graphical components that emulate mathematics of the system they represent.

In this thesis the custom components in Simulink are developed using level 1 S-function that takes mathematical description of a system and simulates it as a standard Simulink component. Level 1 S-function is implemented in the native MATLAB language with .m file extension. The main advantage of M-file S-function is speed of development. In the rest of this section M-file S-function structure and how S-function work is given.

5.1.1. M-file S-function structure

M-file S-function consists of a MATLAB function of the following form:

$$[sys, x0, str, ts] = f(t, x, u, flag, p1, p2, ...) \tag{5.1}$$

Where f is the S-function's name, t is the current time, x is the state vector of the corresponding S-function block, u is the block's inputs, flag indicates a task to be performed, and $p1, p2, ...$ are the block's parameters.

The top-level function f consists of subfunctions called callback methods that perform specific tasks during life cycle of S-function. During the simulation of a model, Simulink repeatedly invokes top level function f, using value of flag to indicate the callback method to be used task to be performed for a particular invocation. The following table lists the contents of an M-file S-function that follows this standard format.

Table 5.1 Content of M-file S-function

Simulation stage	Callback Method	Flag
Initialization	$mdlinitializeSizes;$	$flag = 0$
Calculation of next sample hit (variable sample time blocks only)	$mdlGetTimeOfNextVarHit(t,x,u);$	$flag = 4$
Calculation of outputs	$mdlOutputs(t,x,u);$	$flag = 3$
Update of discrete states	$mdlUpdate(t,x,u);$	$flag = 2$
Calculation of derivatives	$mdlDerivatives(t,x,u);$	$flag = 1$
End of simulation tasks	$mdlTer\min ate(t,x,u);$	$flag = 9$

5.1.2. How S-functions work

The flow chart in the figure below illustrates the stages how S-functions work.

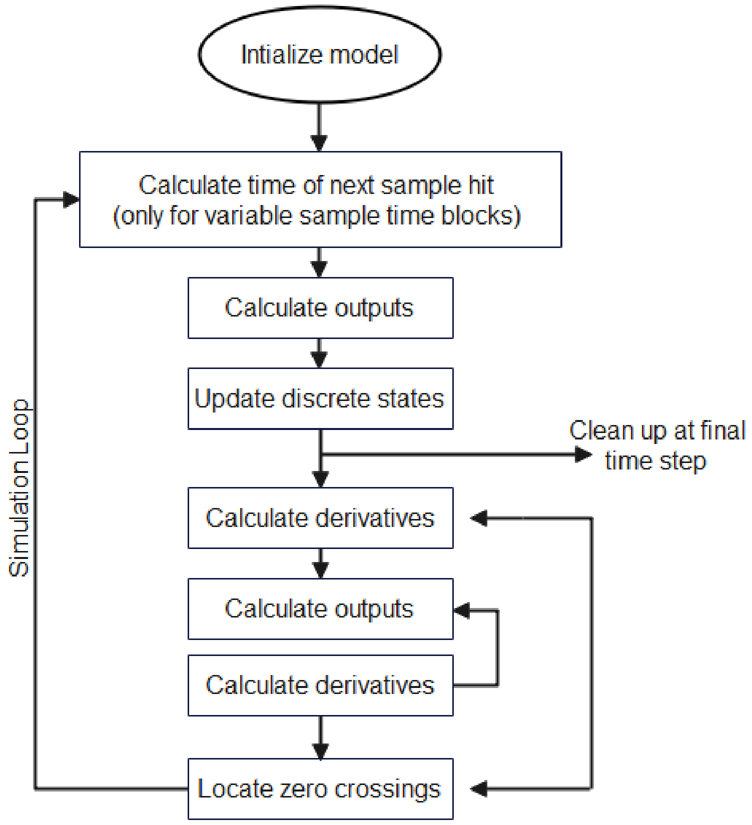

Figure 5.1 Flow chart of how S-function work

As shown in the above diagram prior to the first simulation loop, the engine initializes the s-function, including initializing the Sim-Struct, that means a simulation structure that contains information about the S-function, setting the number and dimensions of input and output ports, setting the block sample times and allocating storage areas. Then calculation of the next sample hit takes place, if you created a variable sample time block, this stage calculates the time of the next sample hit; that is, it calculates the next step size.

Then after the calculation of outputs in the major time step and update of discrete state in the major time step will be performed. Finally, the integration is being done. In this process continuous state derivatives are calculated and the process cycles.

5.2. Simulink Block Diagram of the proposed control system

Table 5.2 lists the physical parameters of the 2-DOF robotic manipulator used for the simulation analysis.

Table 5.2 Physical parameters of RM *(Kang, 2015; Swarup, march 16, 2016)*

RM Parameters	Its value
Mass of link-1 (m_1)	1kg
Mass of link-2 (m_2)	1.5kg
Length of link-1 (l_1)	1m
Length of link-2 (l_2)	0.8m
Moment inertia of link-1 (I_1)	$12\, kg.m^2$
Moment inertia of link-2 (I_2)	$26\, kg.m^2$
Gravitational constant (g)	$9.8\, m/s^2$

Here, the manipulator has four states $x = [q_1 \quad \dot{q}_1 \quad q_2 \quad \dot{q}_2]^T$, two output states $y = [q_1 \quad q_2]^T$ and the two inputs $u = [v_1 \quad v_2]^T$. The frictional effects in joints, including the viscous friction and the coulomb friction torques, are selected as

$u = [15\dot{q}_1 + 6\sin(\dot{q}_1) \quad 15\dot{q}_2 + 6\sin(\dot{q}_2)]^T$.

The main control objective is to make the output q_1, q_2 to track the desired trajectories $q_{d1} = 0.5\sin t$ and $q_{d2} = 0.3\sin t$ respectively. For simulation, the initial states of RM are selected as $q_1(0) = q_2(0) = \dot{q}_1(0) = \dot{q}_2(0) = 0$. The membership function for fuzzy systems is chosen as:

$$\mu_{A_j^l}(x_j) = \exp\left(-\left(\frac{x_i - \bar{x}_i^l}{\pi/4}\right)^2\right), \tag{5.2}$$

where $\bar{x}_i^l$ are $-\pi/6, -\pi/12, 0, \pi/12 \; and \; \pi/6$, respectively, $i = 1, 2, \cdots, 5$, A_i^l is the fuzzy set including NB, NS, ZO, PS, PB, which belong to l^{th} fuzzy rule.

Now by substituting $\bar{x}_i^l$ values to equation (Eq. (5.2)), all the selected five membership function can be expressed as below:

$$\begin{aligned}
&\mu_{NB}(x_j) = \exp\left(-\left(\frac{x_i + \pi/6}{\pi/4}\right)^2\right), \mu_{NS}(x_j) = \exp\left(-\left(\frac{x_i + \pi/12}{\pi/4}\right)^2\right),\\
&\mu_{ZO}(x_j) = \exp\left(-\left(\frac{x_i}{\pi/4}\right)^2\right), \mu_{PS}(x_j) = \exp\left(-\left(\frac{x_i - \pi/12}{\pi/4}\right)^2\right),\\
&and \; \mu_{PB}(x_j) = \exp\left(-\left(\frac{x_i - \pi/6}{\pi/4}\right)^2\right),
\end{aligned} \tag{5.3}$$

These membership functions are given in Figure 5.2 and its program is also given in appendix.

By the trial-and-error method the best design parameters of the proposed AF-STSMC controller are tabulated in Table 5.3.

Table 5.3 Proposed controller (AF-STSMC) controller

Controller	Tuning parameter
Sliding surface constant	$c = [10 \quad 0; \quad 0 \quad 10]$
Super-twisting constant	$k_1 = [0.5 \quad 0; \quad 0 \quad 0.5], k_2 = [2 \quad 0; \quad 0 \quad 2]$
Adaptive constant	$Gamma1 = Gamma2 = 0.0001$

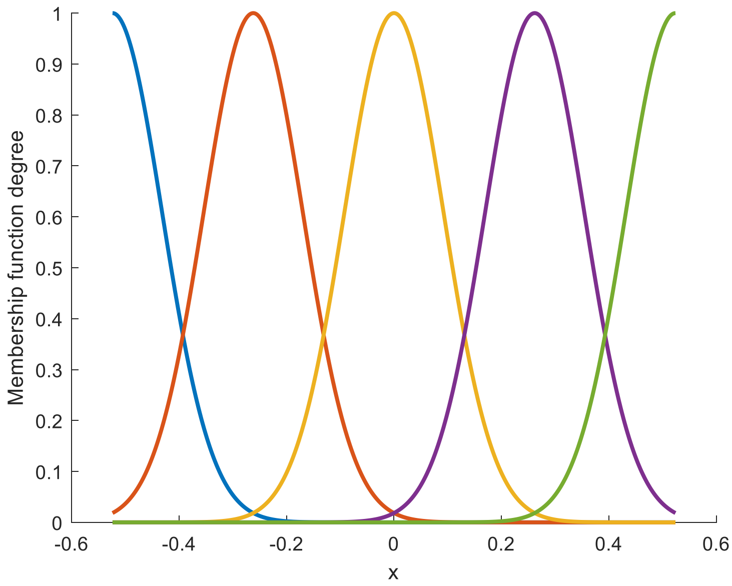

Figure 5.2 Membership function

Finally, the proposed controller (AF-STSMC) Eq. (4.24) implemented in MATLAB/Simulink R2019a environment as shown in the figure below.

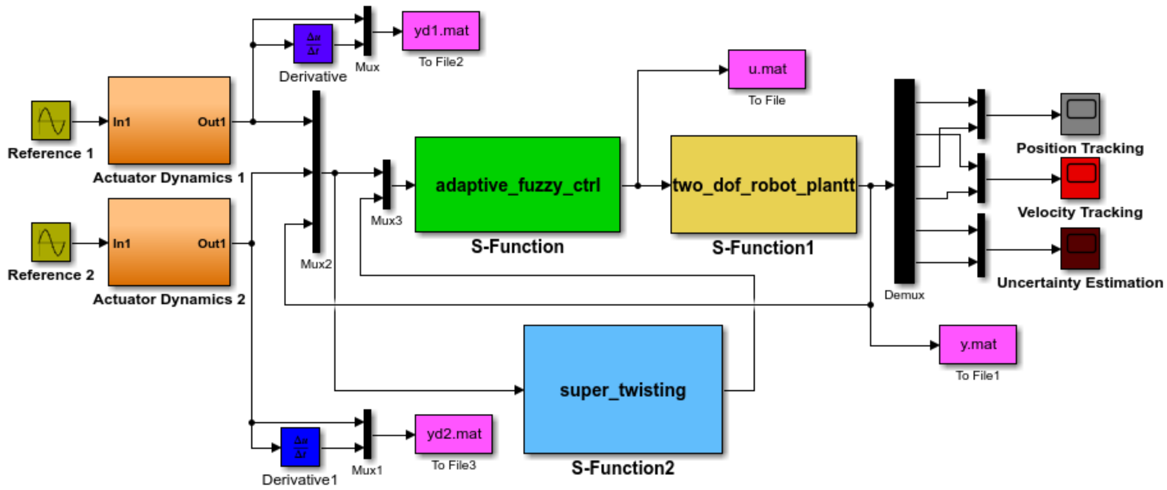

Figure 5.3 Simulink Model of the Proposed Controller (AF-STSMC)

5.3. Simulation Result of Tracking Control for Both Joints

In this section, we examine in greater detail how effectively the proposed controller can track the desired trajectories. For experimental purpose, the desired trajectory for the robotic manipulator is picked arbitrarily to check the tracking capability of the proposed controller (AFST-SMC). The simulation results obtained by applying the proposed controller [Eq. (4.31)] to uncertain robotic manipulator [Eq. (3.58)] are shown in figure 5.9-5.16. From Figure 5.9-5.12, it is observed that both joints 1 and 2 track the reference position trajectory faithfully. However, As seen in position error graph of figures 5.10 and 5.12, the output position initially departed from the reference position by an amplitude less than 0.01 of both joints, Nevertheless, it has been considered that position tracking response as excellent because the robot system is uncertain.

Again, figure 5.13-5.16 demonstrates that joints 1 and 2 properly follow the reference angular velocity trajectory. However, as can be observed in the angular velocity error graph of figures 5.14 and 5.16 the output angular velocity initially deviated from the reference angular velocity by an amplitude less than 0.6 of joint 1 and 0.4 of joint 2. Despite this, it is still rated that the angular velocity tracking response as good because the robot system is uncertain.

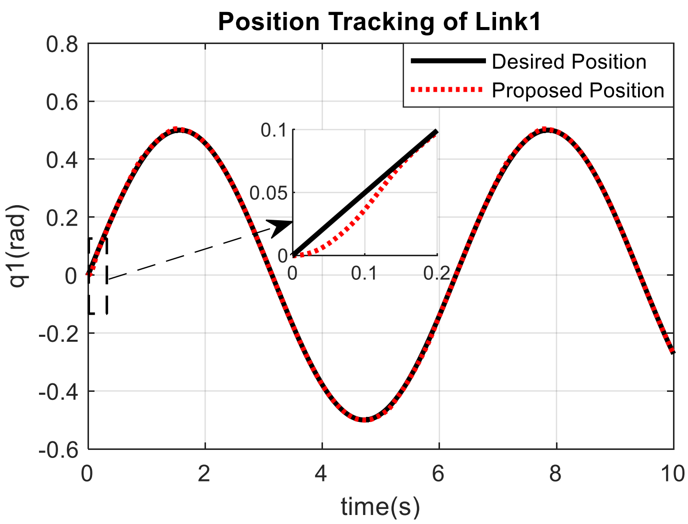

Figure 5.4 Position Trajectory Tracking of Link-1

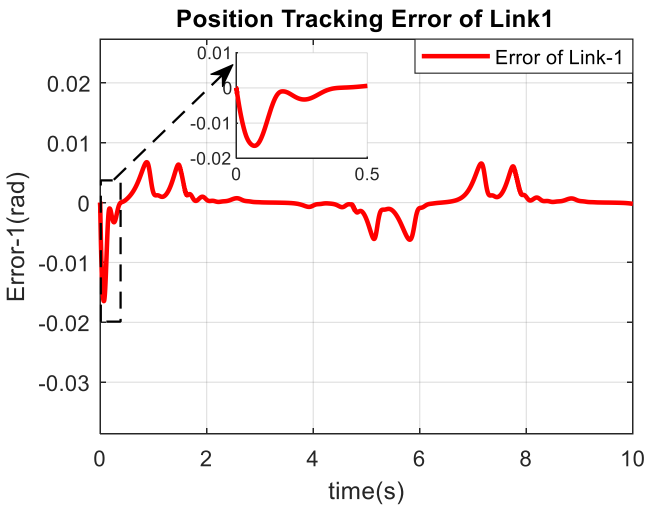

Figure 5.5 Position Trajectory Tracking Error of Link-1

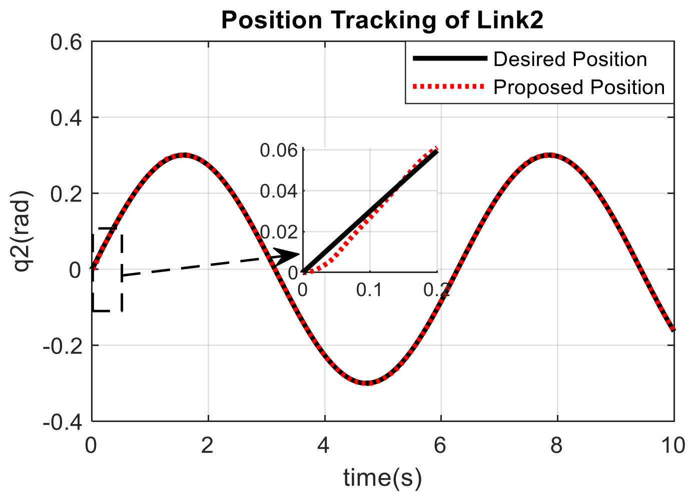

Figure 5.6 Position Trajectory Tracking of Link-2

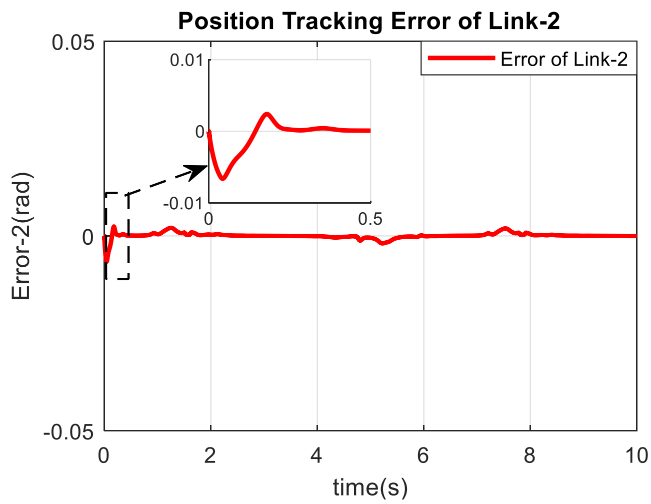

Figure 5.7 Position Trajectory Tracking Error of Link-2

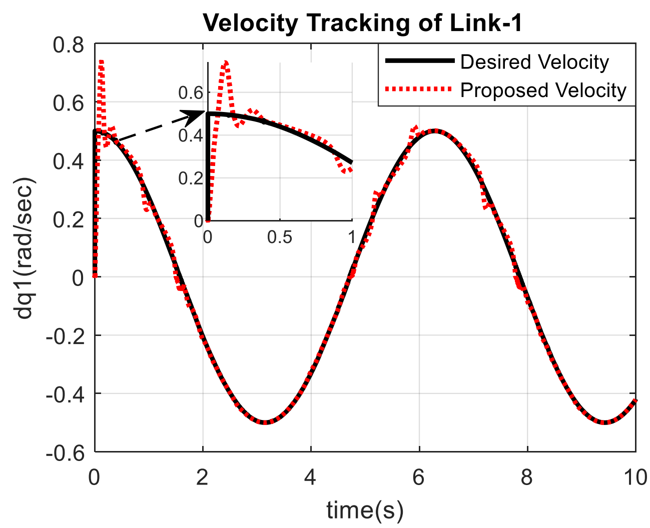

Figure 5.8 Angular Velocity Trajectory Tracking of Link-1

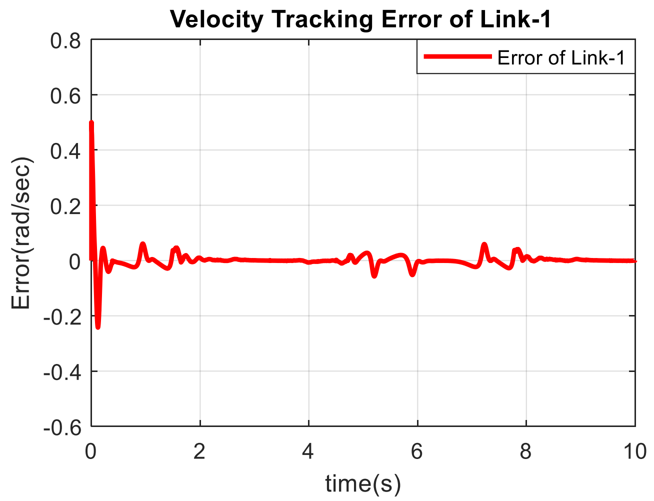

Figure 5.9 Angular Velocity Trajectory Tracking Error of Link-1

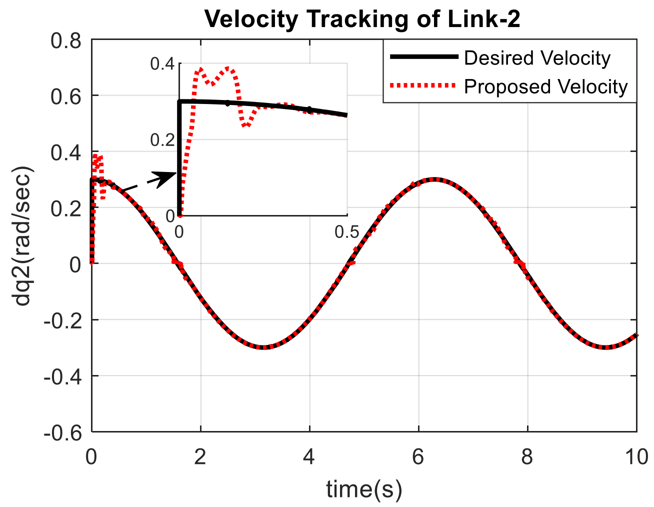

Figure 5.10 Angular Velocity Trajectory Tracking of Link-2

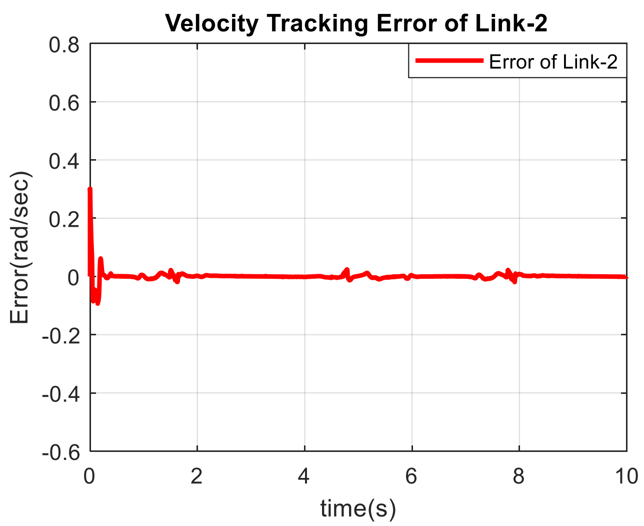

Figure 5.11 Angular Velocity Trajectory Tracking Error of Link-2

5.4. Simulation Result of Overall Controller Input and Uncertainty Estimation for Both Joints

The overall control inputs applied to both joints are continuous input and with initial short span of chattering, i.e, approximately 0.3sec for link-1 and 0.2sec for link-2 and after that the chattering phenomenon is substantially eliminated in both joints as evident in figure-5.17 and figure-5.18. As can be seen from diagram the chattering phenomenon of the overall control input can be avoided effectively.

The estimation values of actual frictional uncertainties under the adaptive fuzzy super-twisting sliding mode control are described in figure-5.19 and figure-5.21. It is observed that the estimated values of frictional uncertainties of Link-1 and Link-2 under the proposed control can quickly converge to their actual values. Figure-5.20 and figure-5.22 represents the error of actual frictional uncertainties present in the joints and the effectiveness of the proposed controller in cancelling this external frictional perturbation.

Clearly, the proposed controller is capable of rejecting the frictional uncertainties present in the 2-dof robotic manipulator, resulting in the very quick and finite position error convergence to zero as evident above in figure-5.20 and figure-5.22.

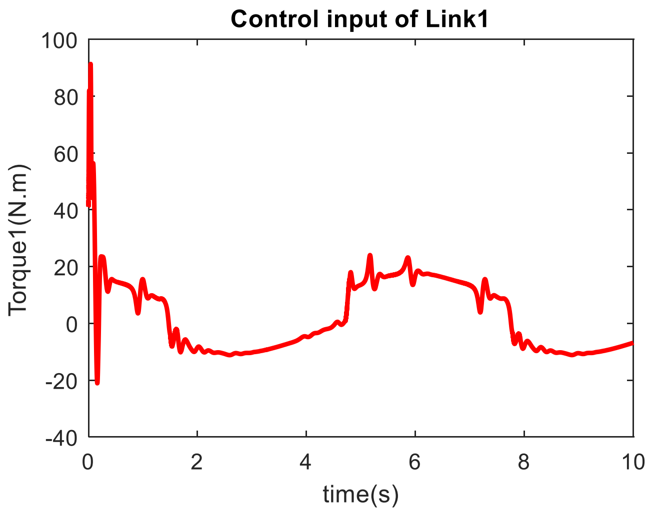

Figure 5.12 Overall Controller Input for Link-1

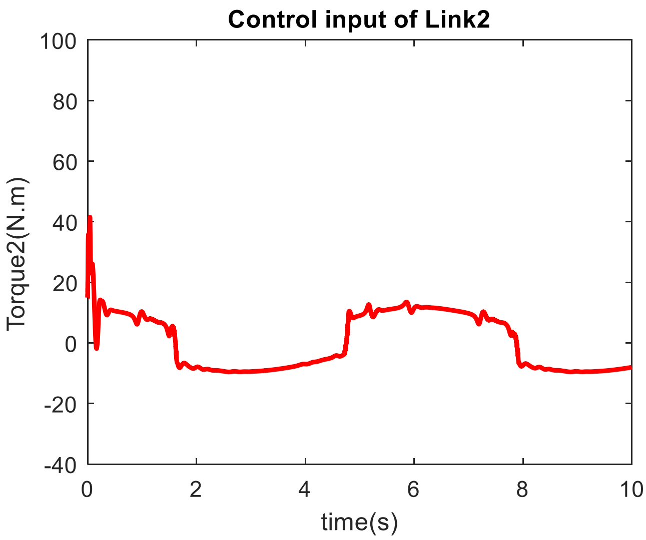

Figure 5.13 Overall Controller Input for Link-2

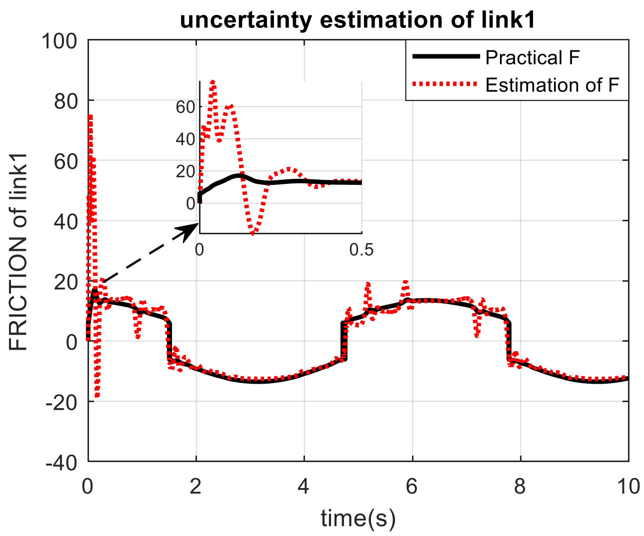

Figure 5.14 Frictional Uncertainty Estimation of Link-1

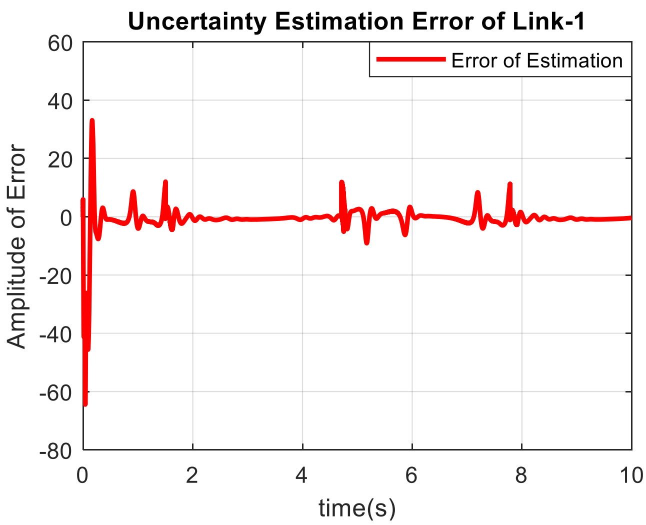

Figure 5.15 Frictional Uncertainty Estimation Error of Link-1

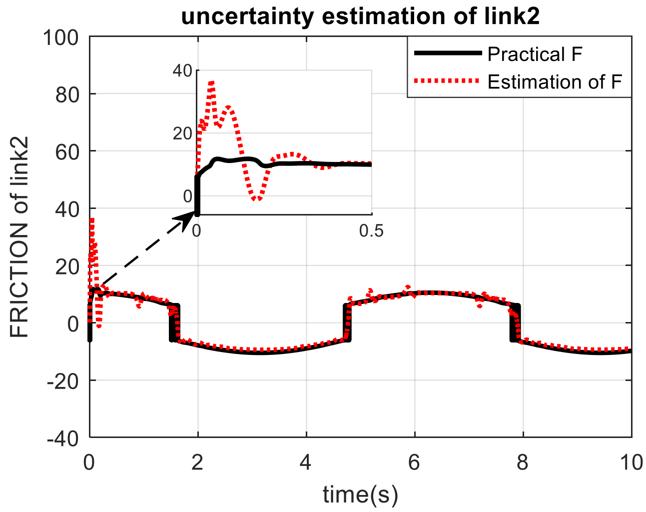

Figure 5.16 Frictional Uncertainty Estimation of Link-2

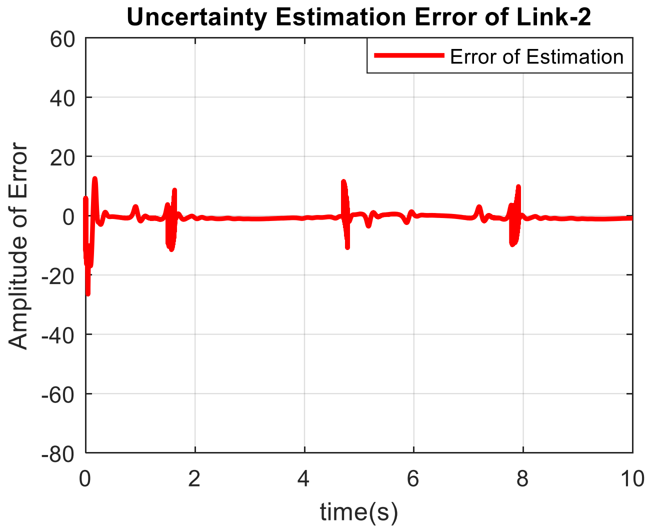

Figure 5.17 Frictional Uncertainty Estimation Error of Link-2

5.5. Comparison of the Proposed Controller with Existing Controller

In order to illustrate the efficacy of the proposed controller, a comparison has been performed with the existing sliding mode controllers applied to the same 2-dof robotic manipulator. for position tracking performance comparison author designed a sliding mode controller (SMC), super-twisting sliding mode controller (ST-SMC) and Adaptive fuzzy sliding mode controller (AF-SMC). Table 5.4 below lists the controller parameters of the existing sliding mode controller.

Table 5.4 Parameter of Conventional Controller Implemented in This Paper

Conventional Controller Parameters	
Controller Gain	
Sliding Mode Controller (SMC)	K=[20 0;0 20]
Adaptive Fuzzy Sliding Mode Controller (AF-SMC)	K=[20 0;0 20]
Super-Twisting Sliding Mode Controller (ST-SMC)	K1=[20 0;0 20] and K2=[50 0;0 50]

With the same initial condition, the position trajectory tracking response obtained by using the above-mentioned existing controllers, are illustrated in figure 5.23 and 5.25. Figure 5.24 and 5.26 show the comparison of their errors, obtained from the application of the proposed AF-STSMC and the rest of the controllers (SMC, ST-SMC and AF-SMC). As shown in the diagram below, the proposed controller achieves the faster trajectory tracking performance than the remaining controllers. The proposed controller error response also converges to zero much more quickly than the existing controllers while compensating the unwanted external disturbance, as shown in Figures 5.24 and 5.26. Moreover, the SMC and ST-SMC controller exhibited larger overshoot than the proposed AF-STSMC, which is undesirable for high-accuracy tracking performance.

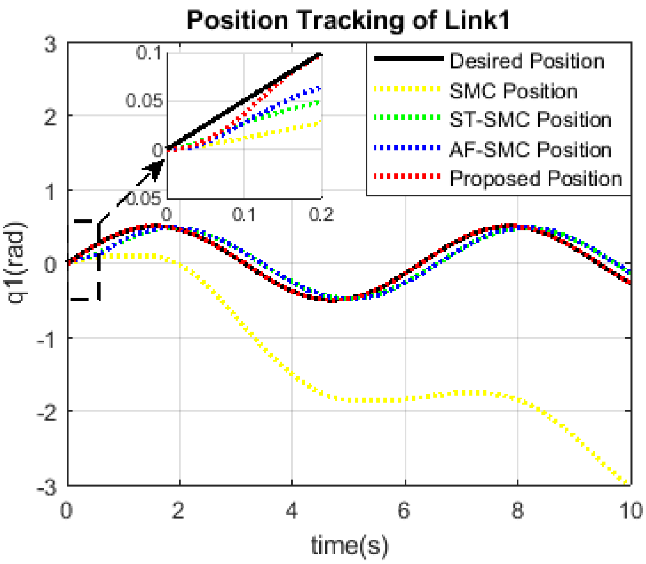

Figure 5.18 Comparison of Position Tracking of Link-1 By Various Controllers

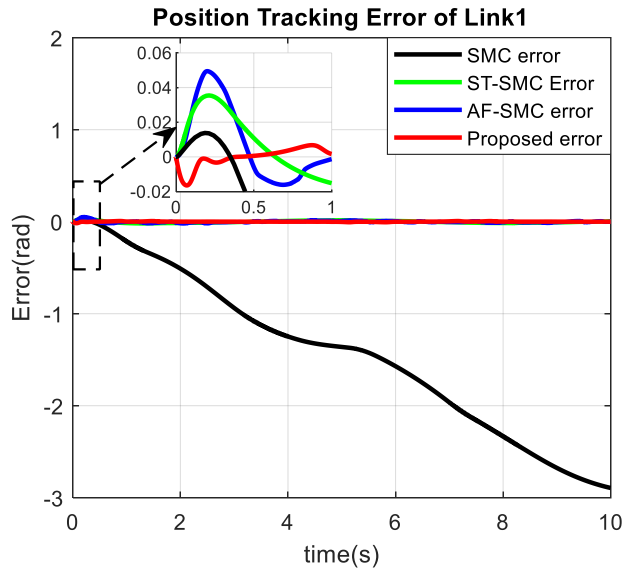

Figure 5.19 Tracking Error of Link-1 With Various Controllers

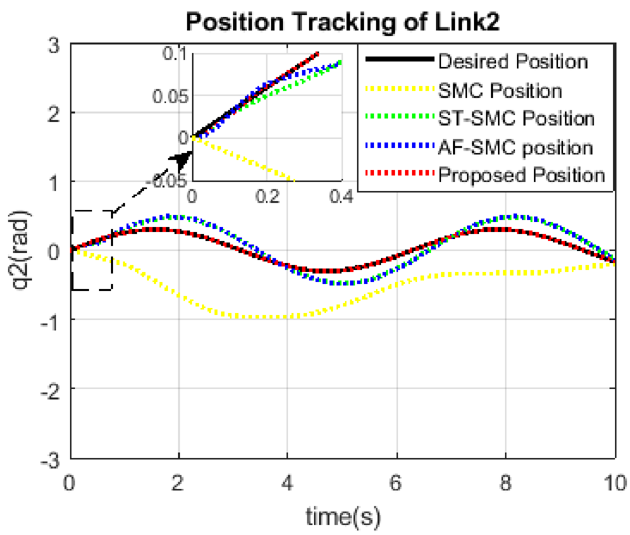

Figure 5.20 Comparison of Position Tracking of Link-2 By Various Controllers

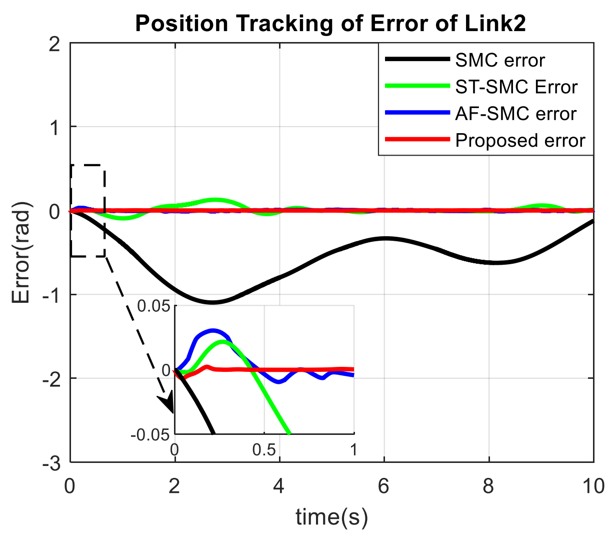

Figure 5.21 Tracking Error of Link-2 With Various Controllers

Generally, the above all figures indicates that the Adaptive Fuzzy Super-Twisting Sliding Mode Control (AF-STSMC) being insensitive to frictional uncertainties and has good performance with frictional uncertainties as compared to Sliding Mode Controller (SMC), Super-Twisting Sliding Mode Controller (ST-SMC) and Adaptive Fuzzy Sliding Mode Controller (AF-SMC).

5.6. Performance index Comparison of the Proposed Controller with Existing Controller

Furthermore, to have a more detailed comparison between the performance of the proposed controller and that of other controllers, an analysis has been performed for input and output characteristics in terms of standard performance index. Performance index is often used to consider how large the error from set point. There are four common ways to do performance assessment criteria. The first one is called the integral absolute error, or IAE. This method essentially adds up all the errors from set point over time. The equation of integral absolute error is given as below Eq. (5.4).

$$IAE = \int_0^\infty \left| Y_{sp}(t) - Y_s(t) \right| dt \tag{5.4}$$

The remaining three, Integral Time Absolute Error (ITAE), Integral Square Error (ISE), and Integral Time Square Error (ITSE), which are all listed below, similarly emphasize various aspects of the error.

$$ITAE = \int_0^\infty t \left| Y_{sp}(t) - Y_s(t) \right| dt \tag{5.5}$$

$$ISE = \int_0^\infty \left| Y_{sp}(t) - Y_s(t) \right|^2 dt \tag{5.6}$$

$$ITSE = \int_0^\infty t \left| Y_{sp}(t) - Y_s(t) \right|^2 dt \tag{5.7}$$

The Simulink Model is modified as shown in figure-5.27 below in order to compare the output performances of the proposed controller with the existing controllers. The link-1 and link-2 performances are calculated for the period from 0 to 10 sec with a sampling time of 0.05 sec. Tables 5.5 and 5.6 provide an overview of the results from the proposed (AFST-SMC) controller as well as other existing controllers for the 2-DOF robotic manipulator. From the table it is noted that the proposed controller offers comparable tracking performance by applying a smoother control input having minimal total variation as compared to the other sliding mode controllers.

Hence, by comparison, it is revealed that the proposed controller supports my important claim of having high-performance criterion and chattering-less control even in the presence of various frictional uncertainties.

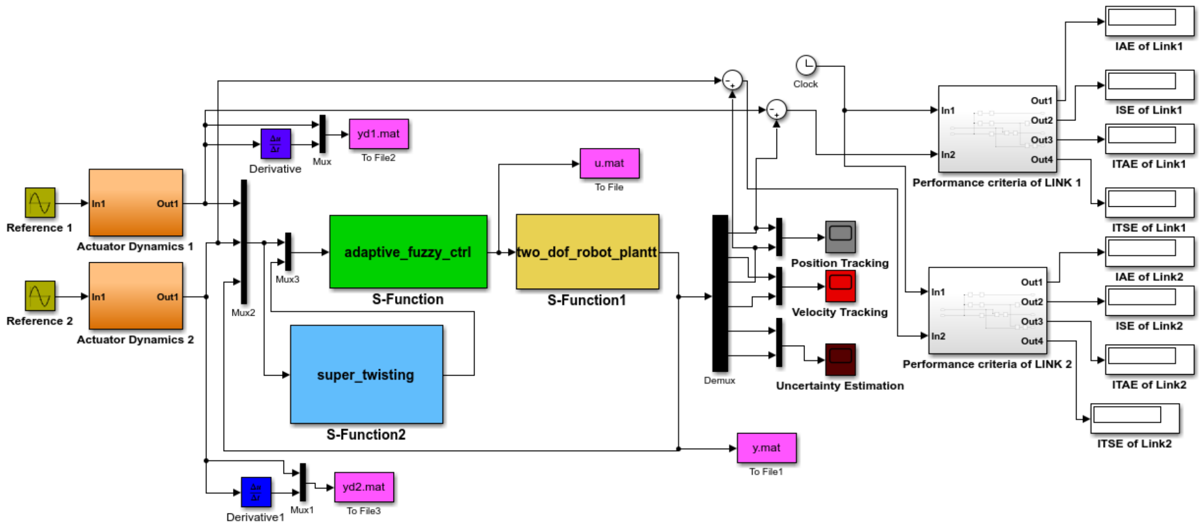

Figure 5.22 Simulink Model of Proposed controller for Performance criteria

Table 5.5 Performance Index of Joint-1

Performance Index of Link-1 by Different Controller				
Performance Index	SMC	ST-SMC	AF-SMC	Proposed
Integral Absolute Error (IAE)	14.24	0.09777	0.06819	0.01465
Integral Time Absolute Error (ITAE)	27.64	0.001344	0.0009952	0.00006811
Integral Square Error (ISE)	95.87	0.4132	0.2593	0.05919
Integral Time Square Error (ITSE)	209	0.004555	0.002123	0.0002109

Table 5.6 Performance Index of Joint-2

Performance Index of Link-2 by Different Controller				
Performance Index	SMC	ST-SMC	AF-SMC	Proposed
Integral Absolute Error (IAE)	5.723	0.306	0.04391	0.003862
Integral Time Absolute Error (ITAE)	4.042	0.02128	0.0003851	0.00005796
Integral Square Error (ISE)	26.69	1.076	0.179	0.01465
Integral Time Square Error (ITSE)	16.71	0.06188	0.0009493	0.0000141

CHAPTER SIX

6. CONCLUSIONS AND RECOMMENDATIONS

6.1. Conclusions

In this thesis paper, an adaptive fuzzy super-twisting sliding mode controller (AF-STSMC) was proposed for a 2-DOF robot arm manipulator system. A comparative analysis was conducted after the successful design of the sliding mode controller (SMC), super-twisting sliding mode controller (ST-SMC), and adaptive fuzzy sliding mode controller (AF-SMC). Simulations were performed to demonstrate that the AF-STSMC is more effective in terms of the response performance than other conventional controllers (SMC, ST-SMC, and AF-SMC) for a given desired trajectory tracking because the proposed controller can adjust itself to the system parameter's variations and external frictional uncertainties. The simulation results further demonstrate excellent AF-STSMC tracking performance and the controller's capacity to handle frictional uncertainty compensation issues and achieve stability, robustness, and reliability.

The stability of the designed controller was verified using Lyapunov`s Direct Method. Also, performance index evaluation criteria were chosen to compare the performance of the controller designed (AF-STSMC) with other existing controllers such as SMC, ST-SMC, and AF-STSMC. Both SMC and ST-SMC controlling performance was observed to have problem in handling external disturbances (frictional uncertainties) because there is no estimation and adaptive technique, which means that the SMC and ST-SMC must have prior knowledge of the system uncertainty. In order to solve the problem of handling frictional uncertainties to increase the robustness of the system and give an improved output, adaptive fuzzy controller and super-twisting sliding mode controller methods were combined. This control method was implemented in the 2-DOF uncertain robotic manipulator to give a high-performance nonlinear control in the presence of uncertainties.

The simulation results finally revealed that the proposed controller successfully satisfies the performance criterion for accurate 2-DOF RM trajectory tracking. The proposed controller is compared against recently developed SMC-based controller and the study proved the suitability of the proposed controller in application to higher complex non-linear uncertain robotic systems.

6.2. Recommendation

Numerous research-related questions arise as a result of the experience obtained through the research activity presented in this paper. So, here are a few recommendations for the future work:

- Applying the same control strategies discussed in this research paper, or any other control methodology, to a system with a higher degree of freedom RM.
- Including other additional tasks like minimizing energy consumption, obstacle detection, or other cost-related goals. This is crucial for manufacturing tasks, which often require robots to perform a repetitive task while keeping energy and time requirements as low as possible.
- Using estimation technique such as extended Kalman filter (EKF) to measurement of feedback joint’s position and joint’s angular velocity.
- Hardware implementation.

***

RESEARCH FUND ACKNOWLEDGEMENTS

The research for this paper is financially supported by Adama Science and Technology University under the grant number of ASTU/SM-R/534/20, Adama, Ethiopia.

REFERENCE

Abbas, Z. (2018). Technique, Motion Control of Robotic Arm Manipulator Using PID and Sliding Mode. *A thesis submitted in partial fulfillment for the degree of Master of Science*.

Angalaeswari, S. K. (2016). Speed control of permanent magnet (PMDC) motors using Arduino and LabView. *international conference on computational intelligence and computing research.*

B., M. J. (1965). Classical Dynamics. *Academic Press*.

C. Mu, C. S. (2013). Super-twisting sliding mode control based on Lyapunov analysis for the cursing flight of hypersonic vehicles. *IEEE International Conference on Control and Automation(ICCA)*, 522-527. doi:10.1109/ICCA.2013.6565017

Corless, M. (1993). Control of uncertain nonlinear systems. *J. Dynam. Syst. Meas. Control, 115*, 362-372.

Craig, J. J. (2005). Introduction to Robotics. *Mechanics and Control, 3*.

D. Zhao, S. L. (2010). Robust finite-time control approach for robotic manipulators,. *IET Control Theory Appl, 4*, 1-15.

Darajat, A. U. (2020). Control of two-link robot manipulator with uncertainty parameter using self-tuning sliding mode control. *3rd, International Conference on Mechanical, Electronics,Computer, and Industrial Technology (MECnIT)*. doi: https://doi.org/10.1109/mecnit48290.2020.9166643

Denavit, J. a. (1955, June). A kinematic notation for lower-pair mechanisms based on matrices. *journal of applied mechanics*, 215-221.

Ed., M. B. (1989). robotic science.

Frank L.Lewis, D. M. (n.d.). Robot Manipulator Control Theory and Practice. *Control of Robot Manipulators*. Retrieved from www.dekker.com

Getachew, D. (2018). Sliding mode control of a two degree of freedom robot using permanent magnet synchronous motor. Retrieved from http://localhost:80/xmlui/handle/

Ghaleb, N. M. (2018). Modeling and control of 2-DOF robot arm. Retrieved from https://www.ijeert.org/papers/v6-i11/3.pdf

Ghazanfar Shahgholian, P. s. (2010). State space modeling and eigen value analysis of the permanent magnet DC motor drive system. *2nd international conference on electronic computer technology*.

Ham, B. K. (APRIL, 2000). Adaptive Control of Robot Manipulator Using Fuzzy Compensator. *IEEE TRANSACTIONS ON FUZZY SYSTEMS, 8*.

Hollerbach, J. M. (1980). A Recursive Lagrangian Formulation of Maniputator Dynamics and a Comparative Study of Dynamics Formulation Complexity. *IEEE Transactions on Systems Man and Cybernetics*, 730--736. doi:10.1109/TSMC.1980.4308393.

Kang, M.-D. T.-J. (2015). Adaptive Fuzzy Pid Sliding Mode Controller Of Uncertain Robotic Manipulator. 92–103. doi: 10.1007/978-3-319-22186-1_9

Kosko, B. (NOVEMBER,1994). Fuzzy Systems as Universal Approximators. *IEEE TRANSACTIONS ON COMPUTER, 43*.

La Salle J., L. S. (1961). Stability by Lyapunov’s direct method with applications. *Academic Press*.

Li, J.-J. S. (1991). Applied Non-linear Control. *Prentice Hall Englewood Cliffs, 199*.

LX, W. (1993). Stable adaptive fuzzy control of nonlinear systems. *IEEE Transactions on Fuzzy systems, 12*, 146-155.

M. Zhihong, A. P. (1994). A robust MIMO terminal sliding mode control scheme for rigid robotic manipulators. *IEEE Trans.Autom. Control., 39*, 2464–2469.

Medewar, P. G. (2015). PSO based PID Controller Tuning for PMDC Motor. *International Conference on Energy Systems and Applications*, 522-526.

Moussavi, S. Z. (2012). Speed Control of Permanent Magnet DC motor by using combination of Adaptive Controller and Fuzzy Controller. *International Journal of Computer Applications, 52*, 20.

Murray, R. M. (1994). A Mathematical Introduction to robotic manipulation.

Ohri, J. V. (2011). Comparison of robustness of PID control and sliding mode control of robotic manipulator.

Peng, Y.-R. L.-C. (2021). Super-Twisting Sliding Mode Control Law Design for Attitude Tracking Task of a Spacecraft via Reaction Wheels. *Mathematical Problems in Engineering*. doi:https://doi.org/10.1155/2021/6644033

R. Kelly, V. S. (2006). Control of Robot Manipulators in Joint Space. *Springer Science & Business Media*.

Rahul gautam, A. G. (2017, march). Review on development of industrial robotic arm. *international research journal of engineering and technology (IRJET), 04*(03).

Richard, P. (1981). the computer control of robot manipulators. *robot manipulators: mathematics, programming, and control*.

Sastry, J.-J. S. (1983). Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators. *international journal of control, 38*, 465-492.

Shah, S. a. (2015, january). 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*, 52-59.

Swarup, A. G. (march 16, 2016). Adaptive Fuzzy High-Order Super-Twisting Sliding Mode Controller for Uncertain Robotic Manipulator. *J. Intell. Syst. 2017*, 697-715.

Symon, K. R. (1971). Mechanics. *Addison-Wesley*.

Utkin. V. I. (1977, april). variable structure systems with sliding modes,. *AC-22*, 212-222.

V. Utkin, J. G. (1999). Sliding mode control in electromechanical systems. *Taylar & Francis Ltd*, 115-129.

Vidyasagar, M. W. (2008). Robot dynamics and control. *John Wiley*.

Wai, R.-J. (2003). Tracking control basedon neural network strategy for robot manipulator. *ELSEVIER, 51*(neuro computing), 425-444. doi:https://doi.org/10.1016/S0925-2312(02)

Wang, L. X. (1997). A Course in Fuzzy System and Control.

Yagna Jadeja, B. p. (2019, november). design and development of 5-dof robotic arm manipulators. *international journal of scientific and technology research*, *8*(11).

Zadeh, L. (1965). Fuzzy Sets. *8*, 338-353.

APPENDIX

APPENDIX A: MATLAB Code

Embedded code of Membership function

```
clear all; close all; L1=-pi/6; L2=pi/6; L=L2-L1; T=L*1/1000; x=L1:T:L2;
figure(1); for i=1:1:5 gs=-[(x+pi/6-(i-1)*pi/12)/(pi/24)].^2; u=exp(gs); hold on;
plot(x,u,'linewidth',2); end xlabel('x'); ylabel('Membership function degree');
```

Embedded code of S-function (Adaptive_fuzzy_ctrl)

```
function [sys,x0,str,ts] = adaptive_fuzzy_ctrl(t,x,u,flag)
switch flag, case 0, [sys,x0,str,ts]=mdlInitializeSizes; case 1, sys=mdlDerivatives(t,x,u);
case 3, sys=mdlOutputs(t,x,u); case {2,4,9} sys=[];
 otherwise error(['Unhandled flag = ',num2str(flag)]); end
% % initialization of S function
function [sys,x0,str,ts]=mdlInitializeSizes global nmn1 nmn2 c nmn1=10;nmn2=10;
c=[nmn1 0;0 nmn2]; sizes = simsizes; sizes.NumContStates = 10;
sizes.NumDiscStates = 0; sizes.NumOutputs = 4; sizes.NumInputs = 10;
sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 0; sys = simsizes(sizes);
x0 = [0.1*ones(10,1)]; str = []; ts = [];
% % since system are continuous model derivatives
function sys=mdlDerivatives(t,x,u)
global nmn1 nmn2 c qd1=u(1); qd2=u(2); dqd1=0.5*cos(t); dqd2=0.3*cos(t);
dqd=[dqd1 dqd2]'; ddqd1=-0.5*sin(t); ddqd2=-0.3*sin(t); ddqd=[ddqd1 ddqd2]';
q1=u(3); dq1=u(4); q2=u(5); dq2=u(6);
% % fuzzy membership for both link
fsd1=0; for l1=1:1:5 gs1=-[(dq1+pi/6-(l1-1)*pi/12)/(pi/24)]^2; u1(l1)=exp(gs1);
end
fsd2=0; for l2=1:1:5 gs2=-[(dq2+pi/6-(l2-1)*pi/12)/(pi/24)]^2; u2(l2)=exp(gs2);
end
for l1=1:1:5 fsu1(l1)=u1(l1); fsd1=fsd1+u1(l1);
end
for l2=1:1:5 fsu2(l2)=u2(l2); fsd2=fsd2+u2(l2);
end fs1=fsu1/(fsd1+0.001); fs2=fsu2/(fsd2+0.001);
e1=q1-qd1; e2=q2-qd2; e=[e1 e2]'; de1=dq1-dqd1; de2=dq2-dqd2; de=[de1 de2]';
s=de+c*e;
```

```
% % Adaptive rule
Gama1=0.0001;Gama2=0.0001; S1=-1/Gama1*s(1)*fs1; S2=-1/Gama2*s(2)*fs2;
for i=1:1:5 sys(i)=S1(i); end
for j=6:1:10 sys(j)=S2(j-5); end
% %S-function model output
function sys=mdlOutputs(t,x,u)
global nmn1 nmn2 c q1=u(3);dq1=u(4); q2=u(5);dq2=u(6); r1=1; r2=0.8; m1=1; m2=1.5;
% % inertia matrix
M11=(m1+m2)*r1^2+m2*r2^2+2*m2*r1*r2*cos(q2); M22=m2*r2^2;
M21=m2*r2^2+m2*r1*r2*cos(q2); M12=M21; M=[M11 M12;M21 M22];
% % Corolios matrix
C12=m2*r1*sin(q2); C=[-C12*dq2 -C12*(dq1+dq2);C12*q1 0];
% % Gravity matrix
g1=(m1+m2)*r1*cos(q2)+m2*r2*cos(q1+q2); g2=m2*r2*cos(q1+q2); G=[g1;g2];
% % % % %
qd1=u(1);qd2=u(2); dqd1=0.5*cos(t); dqd2=0.3*cos(t); dqd=[dqd1 dqd2]';
ddqd1=-0.5*sin(t); ddqd2=-0.3*sin(t); ddqd=[ddqd1 ddqd2]'; e1=q1-qd1; e2=q2-qd2;
e=[e1 e2]'; de1=dq1-dqd1; de2=dq2-dqd2; de=[de1 de2]'; s=de+c*e; dqr=dqd-c*e;
ddqr=ddqd-c*de; for i=1:1:5 thta1(i,1)=x(i); end for i=1:1:5 thta2(i,1)=x(i+5); end
fsd1=0; for l1=1:1:5 gs1=-[(dq1+pi/6-(l1-1)*pi/12)/(pi/24)]^2; u1(l1)=exp(gs1); end
fsd2=0; for l2=1:1:5 gs2=-[(dq2+pi/6-(l2-1)*pi/12)/(pi/24)]^2; u2(l2)=exp(gs2); end
for l1=1:1:5 fsu1(l1)=u1(l1); fsd1=fsd1+u1(l1); end for l2=1:1:5 fsu2(l2)=u2(l2);
fsd2=fsd2+u2(l2); end
% % Adaptive fuzzy compensation
fs1=fsu1/(fsd1+0.001); fs2=fsu2/(fsd2+0.001); Fp(1)=thta1'*fs1'; Fp(2)=thta2'*fs2';
usw1=u(9); usw2=u(10); usw=[usw1 usw2]';
% % controller input to the plant
tol=M*ddqr+C*dqr+G+1*Fp'+usw;
% % output of overall Controller
sys(1)=tol(1); sys(2)=tol(2); sys(3)=Fp(1); sys(4)=Fp(2);
```

Embedded code of S-function2 (Super_twisting)

```
function [sys,x0,str,ts] =super_twisting(t,x,u,flag) switch flag, case 0,
[sys,x0,str,ts]=mdlInitializeSizes;  case 1,sys=mdlDerivatives(t,x,u); case 3,
```

```
sys=mdlOutputs(t,x,u); case {2,4,9} sys=[]; otherwise error(['Unhandled flag = ',num2str(flag)]); end
% % S-function model initialization
function [sys,x0,str,ts]=mdlInitializeSizes
global nmn1 nmn2 c nmn1=10;nmn2=10; c=[nmn1 0;0 nmn2]; sizes = simsizes;
sizes.NumContStates = 2; sizes.NumDiscStates = 0; sizes.NumOutputs = 2;
sizes.NumInputs =8; sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 0; sys =
simsizes(sizes); x0 = [0 0]'; str = []; ts = [];
% % since system are continuous model derivatives
function sys=mdlDerivatives(t,x,u)
nmn1=10;nmn2=10; c=[nmn1 0;0 nmn2]; qd1=u(1); qd2=u(2); dqd1=0.5*cos(t);
dqd2=0.3*cos(t); dqd=[dqd1 dqd2]'; ddqd1=-0.5*sin(t); ddqd2=-0.3*sin(t); ddqd=[ddqd1
ddqd2]'; q1=u(3);dq1=u(4); q2=u(5);dq2=u(6); e1=q1-qd1; e2=q2-qd2; e=[e1 e2]';
de1=dq1-dqd1;de2=dq2-dqd2; de=[de1 de2]'; s=de+c*e; k1=[0.2 0;0 0.2]; v=-k1*sign(s);
sys(1)=v(1);sys(2)=v(2);
function sys=mdlOutputs(t,x,u)
nmn1=10;nmn2=10;  c=[nmn1 0;0 nmn2]; qd1=u(1); qd2=u(2);  dqd1=0.5*cos(t);
dqd2=0.3*cos(t); dqd=[dqd1 dqd2]';  ddqd1=-0.5*sin(t); ddqd2=-0.3*sin(t); ddqd=[ddqd1
ddqd2]';q1=u(3);dq1=u(4); q2=u(5);dq2=u(6); e1=q1-qd1;  e2=q2-qd2; e=[e1 e2]';
de1=dq1-dqd1; de2=dq2-dqd2; de=[de1 de2]'; s=de+c*e; k2=[5 0;0 5]; Usw1=-
k2.*(sqrt(abs(s))).*sign(s); Usww1=Usw1(1)+x(1); Usww2=Usw1(2)+x(2);
sys(1)=Usww1;  sys(2)=Usww2;
```

Embedded code of S-function1 (two_dof_robot_plantt)

```
function [sys,x0,str,ts]=two_dof_robot_plantt(t,x,u,flag)
switch flag, case 0, [sys,x0,str,ts]=mdlInitializeSizes; case 1, sys=mdlDerivatives(t,x,u);
case 3, sys=mdlOutputs(t,x,u); case {2, 4, 9 } sys = [];  otherwise  error(['Unhandled flag = ',num2str(flag)]); end
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes; sizes.NumContStates = 4; sizes.NumDiscStates = 0; sizes.NumOutputs =
6; sizes.NumInputs = 4; sizes.DirFeedthrough = 0; sizes.NumSampleTimes = 0;
sys=simsizes(sizes); x0=[0 0 0 0]; str=[]; ts=[];
function sys=mdlDerivatives(t,x,u)
r1=1;r2=0.8; m1=1;m2=1.5; M11=(m1+m2)*r1^2+m2*r2^2+2*m2*r1*r2*cos(x(3));
M22=m2*r2^2; M21=m2*r2^2+m2*r1*r2*cos(x(3));
```

```
 M12=M21;  M=[M11 M12;M21 M22]; C12=m2*r1*sin(x(3)); C=[-C12*x(4) -
C12*(x(2)+x(4));C12*x(2) 0]; g1=(m1+m2)*r1*cos(x(3))+m2*r2*cos(x(1)+x(3));
g2=m2*r2*cos(x(1)+x(3));G=[g1;g2];
% % input disturbance
Fr=[15*x(2)+6*sign(x(2));15*x(4)+6*sign(x(4))];
tol=[u(1) u(2)]';
% % output velocity
S=inv(M)*(tol-C*[x(2);x(4)]-G-Fr);
sys(1)=x(2); sys(2)=S(1); sys(3)=x(4); sys(4)=S(2);
function sys=mdlOutputs(t,x,u)
Fr=[15*x(2)+6*sign(x(2));15*x(4)+6*sign(x(4))];
% % output our system
sys(1)=x(1);  sys(2)=x(2);  sys(3)=x(3);
sys(4)=x(4);  sys(5)=Fr(1); sys(6)=Fr(2);
```

APPENDIX B: Simulink block