# Optimal Super Twisting Sliding Mode Control of 3-DOF Manipulator

Eyob Mosisa Gurmu

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 Masters in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

June, 2023

Adama, Ethiopia

# Optimal Super Twisting Sliding Mode Control of 3-DOF Manipulator

Eyob Mosisa Gurmu

Advisor: prof. Gang Gyoo Jin

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 Masters in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

June, 2023

Adama, Ethiopia

# DECLARATION

I hereby declare that this Master Thesis entitled "Optimal Super Twisting Sliding Mode Control of 3-DOF 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 OF ADVISOR

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "Optimal Super Twisting Sliding Mode Control of 3-DOF Manipulator" prepared under my guidance by Eyob Mosisa Gurmu submitted in partial of the requirements for the degree of Masters of Science in Electrical Power and Control Engineering. Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

| | | |
|---|---|---|
| ______________________ | ______________________ | ______________________ |
| Major Advisor/Supervisor | Signature | Date |

# APPROVAL PAGE

I, the advisor of the thesis entitled "Optimal Super Twisting Sliding Mode Control of 3-DOF Manipulator" and developed by Eyob Mosisa Gurmu hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

| ___________________ | ___________________ | ___________________ |
|---|---|---|
| Major Advisor/Supervisor | Signature | Date |

We, the undersigned, members of the Board of Examiners of the thesis by Eyob Mosisa Gurmu have read and evaluated the thesis entitled "Optimal Super Twisting Sliding Mode Control of 3-DOF Manipulator" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in control Engineering.

| ___________________ | ___________________ | ___________________ |
|---|---|---|
| Chairperson | Signature | Date |
| Internal Examiner | Signature | Date |
| External Examiner | Signature | Date |

Final 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

I would like to begin by expressing my heartfelt gratitude to the Almighty God for providing me with unwavering strength and guidance throughout this challenging thesis journey. I would also like to extend my deepest appreciation to my Major advisor, Professor Gang Gyoo Jin, whose exceptional expertise, inspiration, and continuous encouragement have been instrumental in shaping the outcome of my thesis. His support and motivation have been invaluable in guiding me towards success. Finally, I want to express my sincere thanks and blessings to my family and all those who have provided assistance and support during the course of my thesis.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AFBSMC | Adaptive Fuzzy Back-stepping Sliding Mode Control |
| AFSMC | Adaptive Fuzzy Sliding Mode Control |
| AFSTSMC | Adaptive Fuzzy Super-Twisting Sliding Mode Control |
| BSC | Back-stepping Sliding Mode Control |
| DH | Denavit-Hartenberg |
| DOF | Degree of Freedom |
| EKF | Extended Kalman Filter |
| FLS | fuzzy logic system |
| HOSMC | High Order Sliding Mode Controller |
| LQG | Linear Quadratic Gaussian |
| LQG | Linear Quadratic Gaussian |
| LQR | Linear Quadratic Regulator |
| MATLAB | Matrix Laboratory |
| MIMO | Multi Input Multi Output |
| MPC | Model Predictive Control |
| PID | Proportional Integral Derivative |
| PMDC | Permanent Magnet Direct Current Motor |
| PSO | Particle Swarm Optimization |
| RM | Robotic Manipulator |
| SMC | Sliding mode control |
| ST-SMC | Super-Twisting Sliding Mode Control |

# LIST OF SYMBOLS

| | |
|---|---|
| $C(\theta,\dot{\theta})$ | Coriolis and centripetal matrix |
| $G(\theta)$ | Gravity term |
| $M(\theta)$ | Inertia matrix |
| $\tau$ | Torque |
| $F_v$ | Viscous damping coefficient |
| $F_s$ | Coulomb friction coefficient |
| $\tau_d$ | Torque disturbance |
| $k_1, k_2$ | Gain of super twisting sliding mode control |
| $\eta, k$ | Gain of sliding mode control |
| $F(\dot{\theta})$ | Frictional force |
| c | Sliding coefficient |
| *e(t)* | Tracking error |
| *KE* | Kinetic energy |
| *PE* | Potential energy |
| s | Sliding surface |
| *Sgn(s)* | Signum of sliding surface |
| *u(t)* | Input torque |
| *v(t)* | Angular velocity |
| *w* | Weighing factor |
| *x(t)* | Angular position |

# ABSTRACT

*The term "robot" is derived from the Czech word "robota," which means "forced labor" or "drudgery." Robotic manipulators are built from a series of links and joints. They are extremely intricate, unpredictable, and multi-input multi-output nonlinear systems with dynamic characteristics that fluctuate greatly over time. Robot manipulator trajectory control is challenging due to these factors. To track the joint trajectory of the manipulator accurately and efficiently, an optimal STSMC is proposed in this paper. It allows 3-DOF articulated robot manipulator to accurately track input trajectories while controlling the impact of uncertainty parameters on the manipulator. DH parameter is employed for forward kinematics, and algebraic method is employed for inverse kinematics of three degree of freedom robotic manipulator. By using Euler-Lagrange equation of motion nonlinear dynamic model of 3-DOF robotic manipulator is produced. For the investigation of the stability of the system, Lyapunov direct method was used. The settings of the controllers were fine-tuned using Particle swarm optimization (PSO) technique. The controller was implemented by using MATLAB/Simulink software and its performance was evaluated. Performance of Simulation result depicted that the proposed controller is chattering free. Step, ramp and sine wave tracking are discussed for STSMC, and position tracking to set point change is also discussed for STSMC controller. STSMC controller is compared with PID and SMC controllers. For 1st, 2nd and 3rd articulation the rise time of PID, SMC and STSMC are 0.2260s, 0.1101s, 0.1001s; 0.3891s, 0.2675s, 0.2967s; and 0.3602s, 0.2112s, 0.1228s respectively. In second link SMC has smaller rise time than STSMC, however in first link and in third link STSMC has smallest rise time. Settling time for 1st, 2nd and 3rd articulation using PID controller are 0.3543s, 3.2329s and 1.2746s respectively. For 1st, 2nd and 3rd articulation settling time using SMC controller are 0.2071s, 0.5316s, 0.4122s respectively. Using STSMC controller settling time for 1st, 2nd and 3rd articulation are 0.2051s, 0.5095s, 0.2683s respectively. According to the result STSMC controller achieves fastest convergence.*

**Keywords:** 3-DOF robotic arm, Dynamics, Lagrange equation, STSMC, PSO

# CHAPTER ONE
# INTRODUCTION

## 1.1 Background

The term "robot" is derived from the Czech word "robota," which means "forced labor" or "drudgery."(Corke, 2017) Robots are automated machines designed to replace human labor and increase efficiency. The ultimate goal of robotics is to develop intelligent machines that can assist people in their daily lives and improve safety. Robots are divided into two primary sectors: industrial and service. Industrial robots are frequently employed within the manufacturing and assembly lines, whereas service robots have widespread usage in industries such as healthcare, security, education, and others. Robotic manipulators come in various configurations, including rectangular, cylindrical, spherical, revolute, and horizontally articulated. One particular type of articulated robotic arm is known as SCARA, which has a horizontal revolute configuration and four degrees of freedom. The SCARA arm consists of three or two horizontal servo-controlled connections, including the elbow, shoulder, and wrist. This particular sort of robot arm was initially created in Japan and is frequently utilized for putting small items into assembly lines, such as electrical components. (Raza et al., 2018).

Robotic technology has become a ubiquitous tool for achieving greater efficiency, precision, and reliability in various industries. Robotic manipulators, which consist of electrically actuated 3-DOF arm-like serial manipulators with sliding or jointed segments, are used to perform tasks that are often challenging for humans to complete, such as space exploration, robotic surgery, or welding. These robotic manipulators are extensively employed in factories across the globe, where they help increase productivity and improve product quality, with articulated geometric types being the most prevalent (Ashagrie et al., 2021; Gambhire et al., 2021).

The use of robotic arms has become increasingly prevalent in industries and other fields due to their ability to perform dangerous, repetitive, and unpleasant tasks with higher precision and accuracy than humans. Robotic arms are used in a variety of applications such as material handling, assembly, welding, painting, and more. However, controlling these machines is a challenging and demanding task, requiring sophisticated controllers to ensure their safe and efficient operation. The control of robotic arms is an active area of research, with ongoing efforts to develop advanced control algorithms and technologies that can enhance their performance and expand their applications in various fields, including military and healthcare (Ghaleb & Aly, 2018; War Naing et al., 2018).

Controlling a robotic manipulator poses a challenging endeavor due to the intricate and sophisticated dynamics inherent in the system. The controller is a critical component for controlling the robot, but it is challenging to design an effective controller due to uncertainties and highly time-varying dynamics in the system. These factors make it difficult to regulate the trajectory tracking of the robot accurately (Piltan et al., 2012).

SMC is a crucial non-linear controller in the context of partially uncertain system dynamics and is frequently employed in controlling highly nonlinear systems, such as robotic manipulators. To ensure stable and reliable control of robotic systems, SMC is a form of control approach utilized in robotics (Mohammed & Eltayeb, 2018). Creating a sliding surface, which is a function of the system states and moves the system toward a desired equilibrium point, is the core tenet of sliding mode control. A specific kind of SMC called STSMC is intended to enhance the system's transient response. This is accomplished by adding a nonlinear function to the sliding surface, which reduces chattering that is frequently observed in conventional sliding mode control.

## 1.2 Statement of the problem

Robotic manipulators are built from a series of links and joints to enable flexibility and precise control over the robot's movements, allowing it to adapt to different tasks and environments. They are extremely intricate, unpredictable, and multi-input multi-output nonlinear systems with dynamic characteristics that fluctuate greatly over time. Robot manipulator trajectory control is challenging due to these factors. The system model also contains a lot of unknowns, such as parameter fluctuations and disturbances, which lead to unstable control system performance. The overall dynamics of a model can be more complex due to joint friction, as well as by a number of uncertainties in the system model.

The majority of research has focused on designing and controlling 2-link and 3-link robotic manipulators utilizing PID, SMC, model MPC and other methods, without taking into account disturbance and joint friction. As a result, the robotic manipulator's performance control had decreased. This thesis makes a contribution by addressing disturbance, joint friction, and meeting input restrictions on the dynamic model of the system to assure and solve the problem of PID trajectory tracking performance of 3-DOF robotic manipulator. With the aim of improving system stability, STSMC controller is developed to address the problem of handling uncertainties and time-varying dynamics because it is a nonlinear robust controller.

## 1.3 Objectives of thesis

### 1.3.1 General objective

The general objective of this thesis is to design optimal STSMC control law to assure that the manipulator tracks the desired trajectory input using MATLAB/Simulink.

### 1.3.2 Specific objectives

- ✓ To develop dynamic model of 3-DOF robotic manipulator using Euler-Lagrange formulation.
- ✓ To design super-twisting sliding mode controller.
- ✓ To design two other controllers for comparison.
- ✓ To investigate stability of proposed controller using Lyapunov stability theorem.
- ✓ To compare performance of the developed STSMC with PID and SMC in MATLAB/Simulink.
- ✓ To make relevant conclusions and recommendations based on results.

## 1.4 Scope of the study

The thesis involved employing a super-twisting sliding mode controller in MATLAB/Simulink to drive a 3-DOF articulated robotic manipulator. Efficiency performance criteria are used to assess the simulation's output. To enhance the performance of the intended trajectory tracking of the approximate actual system, disturbance, joint friction, and parameter uncertainty are taken into consideration. The design of the dynamic modeling and simulation for the 3-DOF robotic arm is the main emphasis of this thesis. The STSMC control technique is built on the basis of dynamic modeling. It has been compared with an optimized PID and SMC at the end.

## 1.5 Limitations of the study

The thesis is restricted to taking the system's dynamic model into account. So, it could have an impact on how well the system performs in real-time applications. Analytically constructing the 3-DOF dynamic model is challenging, so the computation will be time-consuming. Hence, the model of the system may be unclear. As a result, system performance may be decreased. Hence, a digital system should be developed to identify the dynamics equation and overcome the model mistake for the sake of improving performance. The study's limitation is the lack of implementation.

## 1.6 Motivation of the study

The primary justification for selecting this thesis above others is because robots are interesting. Three-DOF robotic manipulators have received very little research due to the dynamics component's complexity. Several of the studies used a simple mathematical model with 2-DOF. The study is motivated in part by the fact that it has a high degree of freedom that means the end effector may select from a wider range of options. Robots may operate in hazardous conditions, liberating people from risky occupations. They are able to handle dangerous materials, lift and carry big things, and do repetitive activities. So, the robust STSMC controller is developed for this electromechanical system so that it may move in any direction to do a certain task.

## 1.7 Significance of the Study

Robots possess adaptability, allowing them to function effectively in various environments while maintaining a constant level of precision and reliability. Since they can operate in unsafe areas, robots limit the number of harmful occupations that people must perform. Robots possess the ability to undertake repetitive tasks, handle hazardous materials, and perform heavy lifting. This has resulted in businesses avoiding numerous accidents while also saving both time and money. Robots outperform humans in terms of precision since they can access and operate in confined spaces that are inaccessible to human hands. Consequently, industrial robots have significantly enhanced productivity, safety, and time efficiency. They can generate work that is exceptionally accurate, dependable, and of superior quality without requiring breaks or vacation time. In order to give the robot the capabilities it needed, the suggested controller regulated the robotic manipulator's trajectory. Using the proposed controller, a robotic manipulator can be employed for various tasks such as welding, painting, and pick-and-place operations.

## 1.8 Thesis organization

This thesis is organized into six chapters.

**Chapter 2**:- Within this chapter, an extensive examination of the literature regarding 3-DOF articulated robotic manipulators and related research is conducted.

**Chapter 3**:- The kinematic and dynamic models of a 3-DOF articulated robotic arm are elucidated in this section, along with a description of the materials and methods employed in the study.

**Chapter 4**:- Within this chapter, the design of three controllers is covered.

**Chapter 5**:- The simulation results of STSMC controller are analyzed and compared with the PID and SMC controllers.

**Chapter 6**:- Within this chapter, the study concludes by presenting the main findings and insights derived from the research. Additionally, based on these findings, recommendations are provided for further improvements or actions to be taken in the field.

# CHAPTER TWO
# LITERATURE REVIEW

## 2.1 Introduction

A robot is a multipurpose machine that can move objects, materials, tools, or specialized equipment using a variety of programmed motions. To finish a variety of jobs, the right actuator, manipulator, and robot controller must be used. The robot's end-effector manipulator needs a suitable selection of controllers in order to accurately track the required input (Kim et al., 2020).

## 2.2 Associated robotic manipulator research

Several modeling and control approaches are presented in many research publications on the trajectory control of robotic manipulators. An overview of studies on robotic manipulator trajectory control and other related controls is given in this section. In general, the research in the literature review concentrate on designing and controlling industrial robotic manipulators using either classical or modern control methods.

In (Saraf & N, 2019), the authors developed an optimal stability controller for an articulated manipulator with three degrees of freedom (3-DOF) that tracks point-to-point trajectory. The forward and inverse kinematics of the manipulator are determined using the Denavit Hartenberg (DH) convention. The Euler-Lagrange approach is used to construct the manipulator dynamics in order to create a nonlinear system. The effectiveness of a 3-DOF spherical articulated manipulator is assessed by carrying out a pick-and-place operation utilizing an ideal LQR and a standard PID controller. The results are contrasted and reported. Although the response to the study was positive, the robustness of the controller was not tested using external disturbance, and tracking was only step set point tracking.

The development and presentation of a mathematical model for a robot arm with two degrees of freedom (2-DOF) is presented in (Okubanjo et al., 2017). Due to the model's foundation in a set of precise second-order nonlinear ordinary differential equations, its behavior can be described. Successfully mathematical model is derived and established using the Euler-Lagrange equations. To achieve independent control over each arm, the coupling effect of the robot arm was eliminated, enhancing its flexibility. The control methodology was expanded by utilizing mathematical equations specifically developed to govern the robot arm's joint angle positioning. The simulation model was created with the help of MATLAB and Simulink and integrated proportional-integral-derivate controllers (PIDs). The paper has no comparison between controllers, and simulation is done with only one controller, which is conventional PID controller. Parameters of PID controller are not optimized. Hence, the in the simulation there is large settling time.

In (Ashagrie et al., 2021) a self-tuning fuzzy controller for sliding mode has been presented. The study discussed how to eliminate chattering in the system control law and gave modeling of 3-DOF and its trajectory tracking. In comparison of the proposed controller to other controllers like PID, SMC and FSMC, the tracking error has decreased according to the simulation findings. Using Lyapunov stability, the system stability has been tested. But, there is no specified simulation regarding to position tracking of each link other than position tracking error.

A 3-DOF hydraulic manipulator with significant payload variation is suggested in (Truong et al., 2019) with an adaptive fuzzy position control. Electrohydraulic actuators are the main torque generators used by the hydraulic manipulator to increase payload carrying capacity. The suggested control integrates fuzzy logic system (FLS), a nonlinear disturbance observer, and back-stepping sliding mode control. The back-stepping sliding mode control consists of a Proportional and Integral control for actuator dynamics as well as a SMC for manipulator dynamics. The payload is compensated using the fuzzy logic system, which modifies the control gain and robust gain of the SMC based on the output of the nonlinear disturbance observer. The resilience and stability of the entire system are demonstrated using the Lyapunov method and back-stepping technique. Although the paper has interesting concept with hydraulic manipulator, chattering phenomena on the control signal has not eliminated.

(Zhang et al., 2020) Presented that a motion simulator platform, a parallel kinematic machine, a medical rehabilitation device, and other application domains may all use parallel manipulators due to their advantages of being small structures with great stiffness, stability, and precision. Owing to the closed-loop structural system's complexity, it is highly challenging to develop an accurate dynamic model in the absence of specific uncertainty factors and outside perturbations. The authors discussed the design and implementation of AFSMC for a 3-DOF parallel manipulator to improve the accuracy of tracking trajectories in the presence of varying and nonlinear parameters. Lyapunov theory is used to derive the proposed controller, which increases trajectory tracking performance while ensuring stability. Lastly, the analysis shows how the suggested control mechanism is insensitive to uncertainties and disruptions. Despite the fact that the result shows the controller's insensitivity to uncertainties and disturbances, it has chattering in system control input.

A two-degree-of-freedom robot has been controlled by two reliable methods (Massaoudi et al., 2019). For this uncertain non-linear system, a back-stepping control has been investigated as well as a modified sliding mode control. Results from simulations demonstrate the usefulness and performance of both controllers. When uncertainties are taken into account, sliding mode control is more reliable while back-stepping is quicker. The performance and durability of modified sliding mode control have been equaled by adaptive sliding mode control. The paper requires more robustness; in order to do so, high-order sliding mode control might be formed to increase robustness.

(Darajat & Istiqphara, 2020) Discussed the control of a two-link robot manipulator in the presence of uncertainty factors that have an impact on the robot manipulator's ability to control the angular position of the designated joint. The dynamic characteristics of a two-link robot manipulator can be determined by resolving the Lagrange Equation. Although SMC is a reliable non-linear control approach, its performance in particular uncertainty factors is still insufficient to monitor the intended joints' angular position. To improve the tracking of the target joints' angular position in the presence of uncertain parameters, such as variations in mass parameters caused by the manipulator's lifting activity, this study proposed the adoption of a self-tuning mechanism based on least squares. To showcase the effectiveness and dependability of the proposed method, a comparison is made between the Self-Tuning Sliding Mode Control (SMC) and the conventional SMC utilizing the Signum function and Saturation function. Chattering is eliminated in presented simulation, but settling time is more than 4 second.

The standard robot identification approach is described in (Brunot et al., 2017). It is based on the well-known Least-Squares approach, but in order to handle closed-loop difficulties and estimate the non-measured signals, a thorough, custom pre-filtering is needed. This special pre-filtering procedure is outlined. A fresh pre-filtering technique is also created. The one is based on a Kalman filter and fixed interval smoother combo. The findings obtained imply that the novel approach is a good substitute when the system bandwidth is unknown prior to identification. While the authors applied the least-square approach, it is preferable to apply the extended Kalman filter since the system is a MIMO nonlinear system.

In (Abbas, 2018), the position control of a robotic arm manipulator is addressed using two control approaches: PID control and Sliding Mode control. Both PID control and sliding mode control are used for the position control of a robotic arm manipulator. However, sliding mode control demonstrates superior performance in terms of robustness against parametric fluctuations, disturbance rejection, and handling uncertainty compared to PID control. The SMC settings in this study have not been tuned, and as a result, the settling time is after 2s. Moreover, higher order sliding mode controls like STSMC are preferable for eradicating the chattering phenomenon.

In (Endris, 2022) PIDGA was used to evaluate the performance of the designed MPC control, and MATLAB/Simulink was used to interface and simulate the model predictive control rule with plants. The authors offered a model predictive control law for an industrial articulated manipulator robot with three degrees of freedom. This solution starts with the feedback linearization method and converts the robot's nonlinear dynamic model into a linear one. The weight factor and forecast horizon time are modified in accordance with the torque energy constraint calculation technique. Positive findings were found when the potential for disturbance caused by MPC controllers was investigated. MPC parameters has not optimized as of PID using one of optimization technique.

(Mohammad & Ehsan, 2008) presents a design for a sliding mode control (SMC) system that is free from chattering for a robot manipulator. The design incorporates a proportional-integral-derivative (PID) component with a fuzzy tunable gain to combine the robustness property of SMC and the favorable response characteristics of PID, resulting in improved performance. The PID sliding surface is initially considered, and the robot dynamic equations are rewritten in terms of the sliding surface and its derivative. The SMC control law includes a PID component. The stability of the sliding mode PID-controller is demonstrated using a lemma with a direct Lyapunov method. In this paper authors didn’t consider disturbance uncertainty and friction.

In summary, by thoroughly reading and understanding the research objectives, while also actively seeking unanswered research questions or unresolved issues, several gaps are identified. The research gap that observed and solved in this thesis are: some researchers used PID and their result shows high overshoot and oscillation in the systems response. Certain researchers did not account for disturbances and friction, which significantly impact the system's behavior in real-time applications. Chattering was observed in the control input of some researchers' results, and some researchers solely focused on tracking step set points. These gaps have been recognized and addressed within this thesis

.

# CHAPTER THREE

# MODELLING OF 3-DOF ROBOTIC MANIPULATOR

## 3.1 Introduction

The material and system approach of the 3-DOF articulated robotic manipulator are given in this chapter as well as general block diagram, and work flow. Robot kinematics, dynamics, and actuator dynamics are also discussed in this chapter

## 3.2 Methods

To create the thesis, a flowchart will be employed to outline and identify all the essential steps that need to be completed. The initial step in completing this thesis was reviewing the literature reviews from various journal papers on robotic manipulators. Chapter 2 discusses the literature reviews, which were earlier related work. The mathematical model is then developed using the Euler-Lagrange technique by computing the kinetic energy and potential energy for each link of the robot arm. As the resultant dynamic model for a 3-DOF articulated robot arm is non-linear, a non-linear controller must be used to properly regulate the system behavior. Super-twisting sliding mode control law is created to achieve this. The assessment of controller stability was conducted using the Lyapunov stability theorem, and MATLAB/Simulink was employed for simulation purposes.

By employing particle swarm optimization (PSO), the optimal parameter for the STSMC was determined. The performance of the proposed controller was then evaluated by comparing it to optimized PID and optimized SMC using PSO. This allowed for a comprehensive analysis of how well the proposed controller performed in relation to the other controllers. The work flow chart of this thesis is presented in Figure 3.1, providing a visual representation of the sequential steps to be followed.

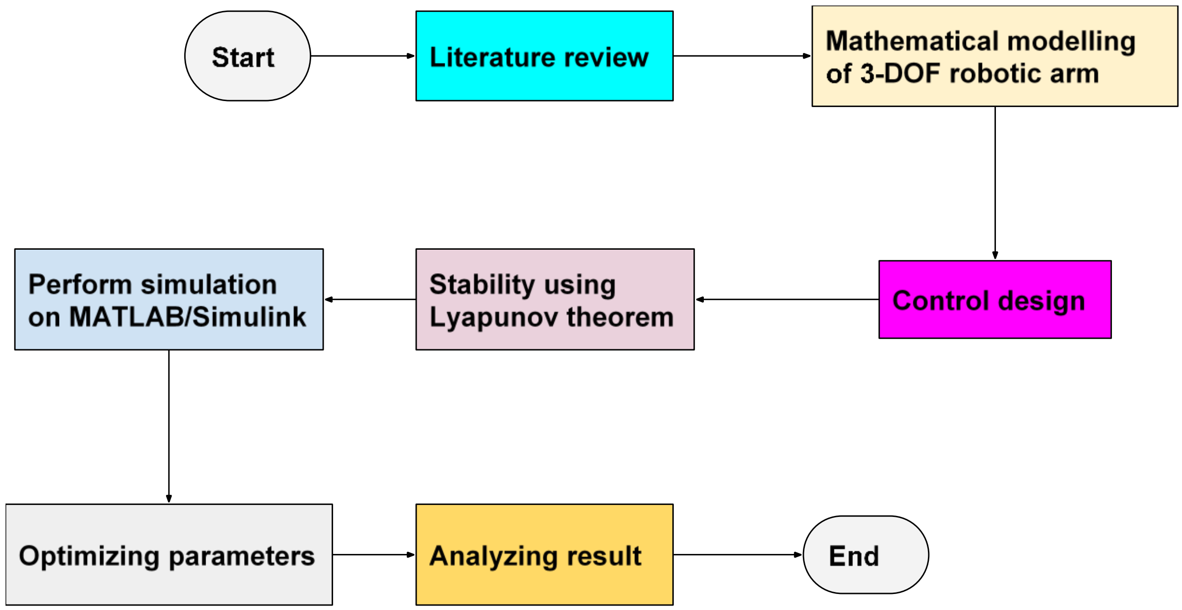


*Figure 3.1: flow chart of the methods.*

## 3.3 System block diagram and its description

Figure 3.2 describes the proposed control system's block diagram in detail.

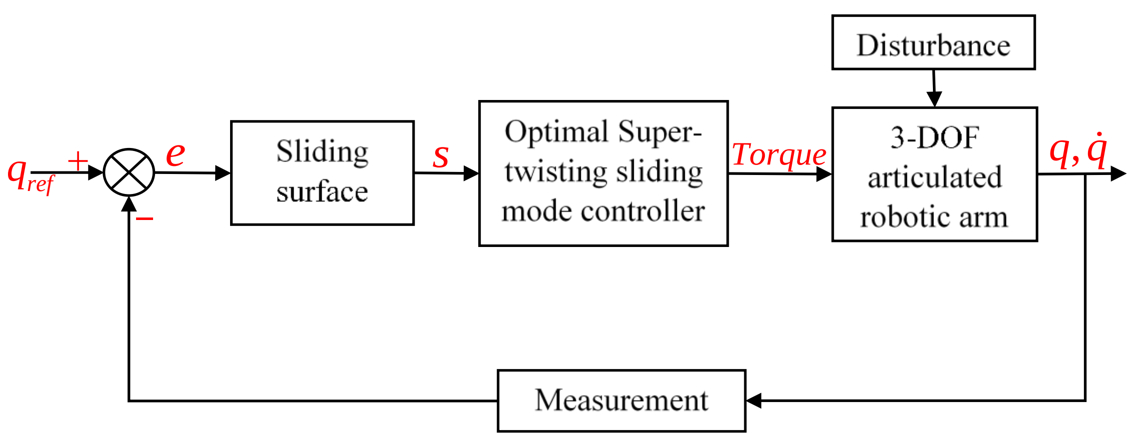


*Figure 3.2: Overall block diagram of the planned control system.*

As previously stated, the fundamental goal of this study is to design a suitable control law that enables the output of the system to efficiently and quickly match the reference trajectory while also online predicting the system uncertainty parameters. To do this, as indicated in the block diagram above, the total system is first given a reference trajectory location as a set point. The sliding surface is then constructed using the trajectory tracking error and its derivative, which is the difference between the reference input position trajectory, and the 3-DOF uncertain robot manipulator output position trajectory. The after, One aspect of our controller's design, which is the sliding surface, shortens the time needed to enter the sliding phase, improving controller performance. The sliding surface is transmitted to a powerful nonlinear controller, which is super twisting sliding mode control.

Sliding mode control is a powerful technique for creating dependable controllers for complex, high-order nonlinear dynamic plants that operate under ambiguous circumstances. Unfortunately, one of the biggest obstacles to the sliding mode controller's success is the chattering phenomenon. To solve the chattering issue in this work, the author therefore suggested the super-twisting sliding mode controller. And also to make sure that the controller is with possible minimum error, the parameters are optimized using particle swarm optimization (PSO).

### 3.4 Robot kinematics

Kinematics is a general term for the study of motions. It examines relationships between motion displacements or positions and their time-derivatives, including velocities, accelerations, and jerks, without taking into account the causes of force and/or torque for a particle, a body, or a multi-body system (Gu, 2022). Certainly, there are two main categories of robot kinematics; Forward kinematics and Inverse kinematics.

#### 3.4.1 Forward kinematics

Also referred to as Direct Kinematics: In direct kinematics, the location and orientation of a robot's end-effector are calculated based on the joint angles. In order to connect the location and orientation of each joint to the end-effector, transformation matrices are used. The position and alignment of a robot's end-effector can be ascertained in real time using direct kinematics (Lynch & Park, 2017). The Denavit-Hartenberg method, which comprises four parameters, is the most often used method for characterizing the robot's kinematics.

Here are parameters four parameters of DH: ‘a’ is the distance in x-direction between the previous and next joint centers. ‘d’ is the offset distance in z-direction between the previous and next joint centers. ‘α’ is the twist angle, this is the angle needed to transfer the previous joint's z-axis to the current joint's z-axis, and ‘θ' is the rotation angle, this is always along the z-axis .The frame assignments for the 3-DOF articulated robotic arm, as well as the joints and their connections, are shown in Figure 3.3, and The DH parameters for the 3-DOF robotic manipulator are displayed in Table 3.1.

*Table 3.1: Denavit-Hartenberg parameters for 3-DOF articulated robotic arm.*

| Link | $a_{i-1}$ | $\alpha_{i-1}$ | $d_i$ | $\Theta_i$ |
|---|---|---|---|---|
| 1 | 0 | $90^0$ | $l_1$ | $\Theta_1$ |
| 2 | $l_2$ | 0 | 0 | $\Theta_2$ |
| 3 | $l_3$ | 0 | 0 | $\Theta_3$ |

Here, a displacement column matrix vector and rotational matrix are both employed. Forward kinematics are explained via homogeneous matrix transformation. As a result, the homogenous transformation matrix is determined depending on the number of degrees of freedom. It is expressed in terms of a transformed displacement vector and rotation matrix. Figure below displays structure of the 3-DOF articulated robotic manipulator.

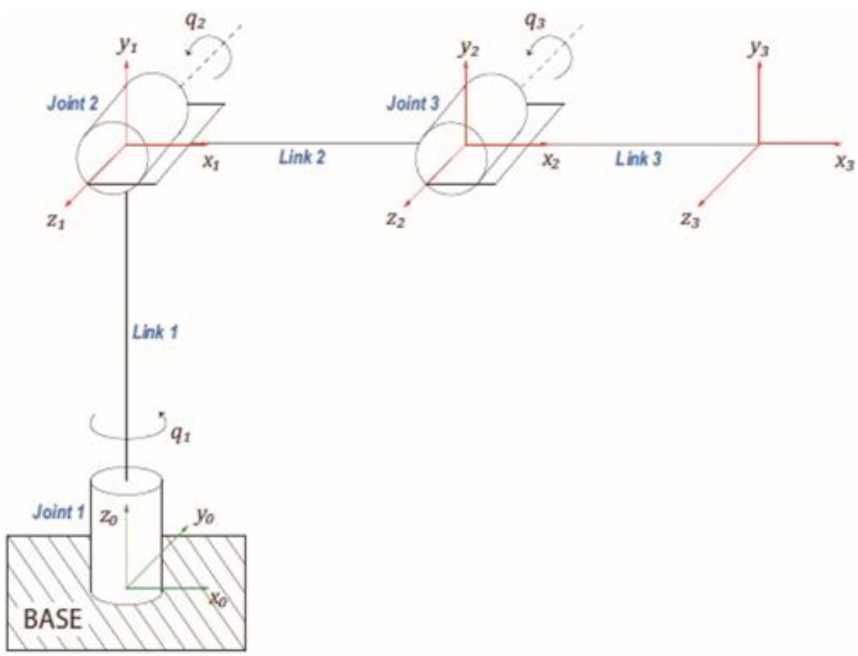


*Figure 3.3: Structure of the 3-DOF articulated robotic arm.* (Ashagrie et al., 2021)

z-is axis considered as axis of rotation, therefore; rotation matrix around z is used. Homogenous transformation matrix from frame 0 to frame 1 is:

$$H_1^0 = \begin{bmatrix} R_1^0 & & & d_1^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

From frame 0 to frame 1, $R_1^0$ and $d_1^0$ represent the rotation matrix and displacement vector, respectively. $R_1^0$, is the product of the new x, y, and z vector projection matrix on the previous Cartesian vector from frame 0 to 1 and the rotation matrix around z. When a rigid body moves without rotating, the projection matrix of the new matrix onto the old matrix is identity. Rotation around the z-axis is:

$$R_z = \begin{bmatrix} C_1 & -S_1 & 0 \\ S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

From frame 0 to frame 1, the following is the projection matrix of new vector on old vector:

$$P_1^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.3}$$

$$R_1^0 = R_z * P_1^0$$

$$R_1^0 = \begin{bmatrix} C_1 & 0 & S_1 \\ S_1 & 0 & -C_1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.4}$$

From frame 0 to frame 1, the displacement vector is as follows:

$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix} \tag{3.5}$$

From frame 0 to frame 1, the homogeneous transformation matrix is as follows:

$$H_1^0 = \begin{bmatrix} R_1^0 & & & d_1^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

A homogeneous transformation matrix between frames one and two is calculated and shown. The second link of the robotic manipulator rotates at the same axis as the first link did. The projection vector matrix is altered from frame 1 to frame 2. Homogeneous transformation matrix is:

$$H_2^1 = \begin{bmatrix} R_2^1 & & & d_2^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

From frame 1 to frame 2, the following is the projection matrix of new vector on old vector:

$$P_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.8}$$

$$R_2^1 = R_z * P_2^1$$

$$R_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The displacement vector from frame 1 to 2 is given as follows:

$$d_2^1 = \begin{bmatrix} l_2 C_2 \\ l_2 S_2 \\ 0 \end{bmatrix} \tag{3.10}$$

From frame 1 to frame 2, the homogeneous transformation matrix is as follows:

$$H_2^1 = \begin{bmatrix} R_2^1 & & & d_2^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_2 & -S_2 & 0 & l_2 C_2 \\ S_2 & C_2 & 0 & l_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

From frame 2 to frame 3, a homogenous transformation matrix is calculated and shown. The third joint of the robotic manipulator shares the same rotational axis as the first and second joints, and the projection vector direction matrix from frame 2 to 3 shares the same matrix as the second joint of the robot. Frame 2 to frame 3 homogenous transformation matrix is:

$$H_3^2 = \begin{bmatrix} R_3^2 & & & d_3^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

From frame 1 to frame 2, the following is the projection matrix of new vector on old vector:

$$P_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

The rotation from frame 2 to frame 3 is calculated using $R_3^2 = R_z * P_3^2$ hence, it is shown as follows:

$$R_3^2 = \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

The displacement vector from frame 1 to 2 is given as follows:

$$d_3^2 = \begin{bmatrix} l_3 C_3 \\ l_3 S_3 \\ 0 \end{bmatrix} \tag{3.15}$$

From frame 2 to frame 3, the homogeneous transformation matrix is as follows:

$$H_3^2 = \begin{bmatrix} R_3^2 & & & d_3^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_3 & -S_3 & 0 & l_3 C_3 \\ S_3 & C_3 & 0 & l_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

This leads to the presentation of a general homogeneous transformation matrix from frames 0 to 3. It's written in compact form.

$$H_3^0 = H_1^0 H_2^1 H_3^2 \tag{3.17}$$

$$H_3^0 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & l_3 C_1 C_{23} + l_2 C_1 C_2 \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & l_3 S_1 C_{23} + l_2 S_1 C_2 \\ S_{23} & C_{23} & 0 & l_3 S_{23} + l_2 S_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

Using the formula of homogenous transformation matrix:

$$H_3^0 = \begin{bmatrix} R_3^0 & & & d_3^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

$$R_3^0 = \begin{bmatrix} C_1C_{23} & -C_2S_{23} & S_1 \\ S_1C_{23} & -S_1S_{23} & -C_1 \\ S_{23} & C_{23} & 0 \end{bmatrix} \tag{3.20}$$

$d_3^0$:- is the displacement vector from frame 0 to 3, which represent, end effector Cartesian space location for robotic arm in forward kinematics.

$$d_3^0 = \begin{bmatrix} l_3C_1C_{23} + l_2C_1C_2 \\ l_3S_1C_{23} + l_2S_1C_2 \\ l_3S_{23} + l_2S_2 + l_1 \end{bmatrix} \tag{3.21}$$

The following are the end effector coordinates:

$$\begin{cases} x_3 = l_3C_1C_{23} + l_2C_1C_2 \\ y_3 = l_3S_1C_{23} + l_2S_1C_2 \\ z_3 = l_3S_{23} + l_2S_2 + l_1 \end{cases} \tag{3.22}$$

To find linear velocities of end effector, derivate equation (3.22) with respect to time.

$$\begin{cases} \dot{x} = -l_2C_1S_2\dot{\theta}_2 - l_3C_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - (l_2S_1C_2 + l_3S_1C_{23})\dot{\theta}_1 \\ \dot{y} = -l_2S_1S_2\dot{\theta}_2 - l_2S_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + (l_2C_1C_2 + l_3C_1C_{23})\dot{\theta}_1 \\ \dot{z} = l_2C_2\dot{\theta}_2 + l_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \end{cases} \tag{3.23}$$

### 3.4.2 Inverse Kinematics

Inverse kinematics is the process of calculating the joint angles needed to position and orient the end-effector to a desired location. Engineers can use inverse kinematics to establish the joint angles necessary to drive a robot to a specified location or carry out a specific task, which is why robot control and trajectory planning are so crucial (Lynch & Park, 2017). Given the position of the end-Cartesian effector's coordinates, the robot manipulator's joint variables being found by using inverse kinematics (Ashagrie et al., 2021).

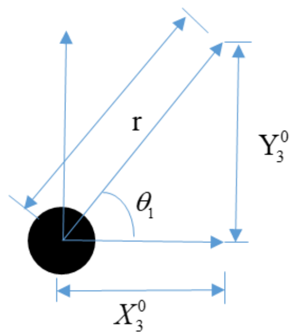


*Figure 3.4: top view frame assignment of 3-DOF robot arm.*

$$\theta_1 = \tan^{-1}(\frac{Y_3^0}{X_3^0}) \tag{3.24}$$

$$r_1 = \sqrt{\left(X_3^0\right)^2 + \left(Y_3^0\right)^2} \tag{3.25}$$

Frame assignment representation of 3-DOF articulated robotic manipulator from side view is illustrated in figure 3. When $\theta_2$ and $\theta_3$ are not at zero position (Endris, 2022).

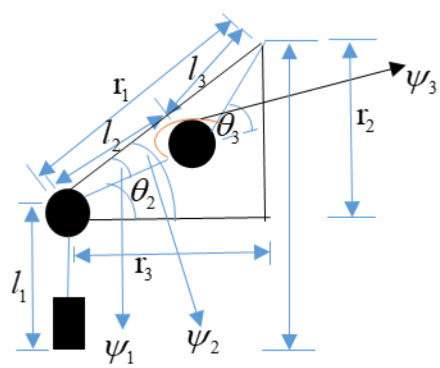


*Figure 3.5: side view frame assignment representation of 3-DOF robotic arm.*

$$\theta_2 = \psi_2 - \psi_1 \tag{3.26}$$

$$\psi_2 = \tan^{-1}\left(\frac{r_2}{r_1}\right) \tag{3.27}$$

Where $r_2 = Z_3^0 - l_1$, and $r_1 = \sqrt{\left(X_3^0\right)^2 + \left(Y_3^0\right)^2}$

Using concept of trigonometric for not right angle triangle the equation becomes:

$$l_3^{\,2} = l_2^{\,2} + r_3^{\,2} - 2l_2 r_3 C\psi_1 \tag{3.28}$$

$$\psi_1 = \cos^{-1}\left(\frac{l_3^{\,2} - l_2^{\,2} - r_3^{\,2}}{-2l_2 r_3}\right) \tag{3.29}$$

Where $r_3 = \sqrt{r_1^{\,2} + r_2^{\,2}}$

$$\theta_3 = \pi - \psi_3 \tag{3.30}$$

$$r_3^{\,2} = l_2^{\,2} + l_3^{\,2} - 2l_2 l_3 C\psi_3 \tag{3.31}$$

$$\psi_3 = \cos^{-1}\left(\frac{r_3^{\,2} - l_2^{\,2} - l_3^{\,2}}{-2l_2 l_3}\right) \tag{3.32}$$

## 3.5 Dynamics of robotic manipulator

The word "dynamics" is used to describe the study of how forces and torques impact the motion of robots and the parts that make up those robots in the field of robotics. Robot dynamics, in particular, entails the mathematical modelling and study of the motion and interaction between the different parts of a robot, such as the joints, linkages, and actuators. This involves researching how external loads, torques, and forces such as those caused by gravity, friction, and other factors affect the robot's movement and behavior. Engineers may build and operate robots to efficiently and successfully carry out specified jobs by having a thorough grasp of the dynamics of the robots (Ajwad et al., 2016). Therefore, the mathematical equations used in this research to characterize the connection between joint torque and robot motion are derived. PMDC motor is used to give input control torque for 3-DOF articulated robotic manipulator but its actuator dynamics is not considered. The dynamics issues with the 3-DOF robot arm are solved using the Lagrange formulation (Van Bach et al., 2021).

The 3-DOF articulated robotic arm's mathematical dynamic equation of motion may be calculated using the Euler Lagrange technique by taking into account the fact that the center of mass for each link is located in the center of the link. Links 1, 2, and 3 for the robotic arm with three degrees of freedom, joint variables are $\theta_1$, $\theta_2$ and $\theta_3$, link lengths are $l_1$, $l_2$ and $l_3$. The masses of the three-degree-of-freedom link are $m_1$, $m_2$ and $m_3$.

The following are the control joint input torque vector, and joint angle vector:

$\tau = [\tau_1\ \tau_2\ \tau_3]^T$ where; $\tau_1$, $\tau_2$ and $\tau_3$ are joint torques for each link respectively. $\theta = [\theta_1\ \ \theta_2\ \ \theta_3]^T$ where; $\theta_1$ $\theta_2$ and $\theta_3$ are the joints' respective varying angles. Table 3.2 displays the physical parameter values of an industrial articulated robotic manipulator with 3 degrees of freedom (Truong et al., 2019).

*Table 3.2: The parameters related to mechanics that were utilized for simulations.*

| Parameters | Notation | Value |
|---|---|---|
| Length of link 1 | $l_1$ | 0.3m |
| Length of link 2 | $l_2$ | 0.5m |
| Length of link 3 | $l_3$ | 0.2m |
| Mass of link 1 | $m_1$ | 5kg |
| Mass of link 2 | $m_2$ | 5kg |
| Mass of link 3 | $m_3$ | 5kg |
| Gravity | g | 9.81m/s$^2$ |

### 3.5.1 Kinetic Energy and Potential Energy

The thesis derives the potential and kinetic energy of an industrial articulated robotic arm with three degrees of freedom for each link of the robot, using the laws of physics. The kinetic energy is expressed based on both the joint variable and angular joint velocity, whereas the potential energy is only expressed based on the joint variable. Additionally, the thesis assumes that the center of mass of an object is positioned at the center of each link of the robotic manipulator (Dereje, 2018).

Potential and Kinetic energy for link 1 can be expressed as:

$$KE_1 = I\frac{\dot{\theta}^2}{2} \text{ , and } PE_1 = 0 \tag{3.33}$$

Where $I$ is inertia, and $\dot{\theta}$ is angular velocity

By multiplying (3.6) and (3.11) HT from frame 0 to frame 2 is obtained.

$$H_2^0 = H_1^0 H_2^0 \tag{3.34}$$

$$H_2^0 = \begin{bmatrix} C_1C_2 & -C_1S_2 & S_1 & l_2C_1C_2 \\ S_1C_2 & -S_1S_2 & -C_1 & l_2S_1C_2 \\ S_2 & C_2 & 0 & l_2S_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

In the (3.35), the position of the end effector of link 2 in Cartesian space is indicated by the displacement vector that extends from frame 0 to frame 2.

$$d_2^0 = \begin{bmatrix} l_2C_1C_2 \\ l_2S_1C_2 \\ l_2S_2 + l_1 \end{bmatrix} \tag{3.36}$$

For link 2, the position is defined in (3.37), and velocity is obtained as follows:

$$\begin{cases} x_2 = l_2C_1C_2 \\ y_2 = l_2S_1C_2 \\ z_2 = l_2S_2 + l_1 \end{cases} \tag{3.37}$$

$$\begin{cases} \dot{x}_2 = -l_2S_1C_2\dot{\theta}_1 - l_2C_1S_2\dot{\theta}_2 \\ \dot{y}_2 = l_2C_1C_2\dot{\theta}_1 - l_2S_1S_2\dot{\theta}_2 \\ \dot{z}_2 = l_2C_2\dot{\theta}_2 \end{cases} \tag{3.38}$$

Hence, velocity squared in kinetic energy is:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2 \tag{3.39}$$

Therefore the kinetic energy and potential energy for the second mass of manipulator is given as:

$$KE_2 = \frac{1}{2}m_2\left(l_2^2C_2^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_1^2\right) \tag{3.40}$$

$$PE_2 = m_2 g l_2 S_2 \tag{3.41}$$

For link 3 position and velocities are obtained as follows:

$$\begin{cases} x_3 = l_3 C_1 C_{23} + l_2 C_1 C_2 \\ y_3 = l_3 S_1 C_{23} + l_2 S_1 C_2 \\ z_3 = l_3 S_{23} + l_2 S_2 + l_1 \end{cases} \tag{3.42}$$

$$\begin{cases} \dot{x}_3 = \left[ -l_2 S_2 \dot{\theta}_2 - l_3 S_{23} \left( \dot{\theta}_2 + \dot{\theta}_3 \right) \right] C_1 - \left( l_2 C_2 - l_3 C_{23} \right) S_1 \dot{\theta}_1 \\ \dot{y}_3 = \left[ -l_2 S_2 \dot{\theta}_2 - l_3 S_{23} \left( \dot{\theta}_2 + \dot{\theta}_3 \right) \right] S_1 + \left( l_2 C_2 + l_3 C_{23} \right) C_1 \dot{\theta}_1 \\ \dot{z}_3 = l_2 C_2 \dot{\theta}_2 - \left[ l_3 \left( \dot{\theta}_2 + \dot{\theta}_3 \right) \right] C_{23} \end{cases} \tag{3.43}$$

Hence, velocity squared in kinetic energy for link 3 is:

$$v_3^2 = \dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2 \tag{3.44}$$

The kinetic energy of the 3rd mass is:

$$\begin{aligned} KE_3 = \frac{1}{2} m_2 (l_2 \dot{\theta}_2^2 + 2 l_2 l_3 C_2 C_{23} \dot{\theta}_2^2 + 2 l_2 l_3 S_2 S_{23} \dot{\theta}_2^2 + \ldots \\ 2 l_2 l_3 C_2 C_{23} \dot{\theta}_2 \dot{\theta}_3 + 2 l_2 l_3 C_2 C_{23} \dot{\theta}_1^2 + l_3^2 \dot{\theta}_2^2 + l_3^2 \dot{\theta}_3^2 + \ldots \\ 2 l_3^2 \dot{\theta}_2 \dot{\theta}_3 + 2 l_2^2 C_2^2 \dot{\theta}_1^2 + l_3^2 C_{23} \dot{\theta}_1^2) \end{aligned} \tag{3.45}$$

$$PE_3 = m_3 g (l_2 S_2 + l_3 S_3) \tag{3.46}$$

Where

$$\begin{cases} C_i = \cos(\theta_i) \\ S_i = \sin(\theta_i) \\ C_{ij} = \cos(\theta_i + \theta_j) \\ S_{ij} = \sin(\theta_i + \theta_j) \\ (i = 1, 2, 3;\ j = 1, 2, 3) \end{cases}$$

### 3.5.2 Euler Lagrange Equation

The system's Euler-Lagrange equation is obtained as follows (Liu et al., 2019):

$$L = KE - PE \tag{3.47}$$

Where; $KE = KE_1 + KE_2 + KE_3$, and $PE = PE_1 + PE_2 + PE_3$

Hence, the Euler Lagrange equation is:

$$\begin{aligned} L = I\frac{\dot{\theta}^2}{2} + \frac{1}{2}m_2\left(l_2^2C_2^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_1^2\right) + \frac{1}{2}m_2(l_2\dot{\theta}_2^2 + 2l_2l_3C_2C_{23}\dot{\theta}_2^2 + \ldots \\ 2l_2l_3S_2S_{23}\dot{\theta}_2^2 + 2l_2l_3C_2C_{23}\dot{\theta}_2\dot{\theta}_3 + 2l_2l_3C_2C_{23}\dot{\theta}_1^2 + l_3^2\dot{\theta}_2^2 + l_3^2\dot{\theta}_3^2 + \ldots \\ 2l_3^2\dot{\theta}_2\dot{\theta}_3 + 2l_2^2C_2^2\dot{\theta}_1^2 + l_3^2C_{23}\dot{\theta}_1^2) - m_2gl_2S_2 - m_3g(l_2S_2 + l_3S_3) \end{aligned} \tag{3.48}$$

The generic equation for a system's motion for any applied torque is:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \dot{\theta}_i} \tag{3.49}$$

$$\begin{aligned} \tau_1 = m_1l_1^2\ddot{\theta}_1 + m_3l_3^2C_{23}^2\ddot{\theta}_1 + 2m_3l_2l_3C_2C_{23}\ddot{\theta}_1 + \left(m_2 + m_3\right)l_2^2C_2^2\ddot{\theta}_1 - \ldots \\ m_3l_3C_{23}S_{23}\left(\dot{\theta}_2 + \dot{\theta}_3\right)\dot{\theta}_1 - \left(m_2 + m_3\right)l_2^2C_2S_2\dot{\theta}_2\dot{\theta}_1 - \ldots \\ m_3l_2l_3\left(S_2C_{23}\dot{\theta}_2 + C_2S_{23}\left(\dot{\theta}_2 + \dot{\theta}_3\right)\right)\dot{\theta}_1 - m_3l_3^2S_{23}C_{23}\dot{\theta}_1\dot{\theta}_2 - \ldots \\ m_3l_2l_3S_2C_{23}\dot{\theta}_1\dot{\theta}_2 - \left(m_2 + m_3\right)l_2^2S_2C_2\dot{\theta}_1\dot{\theta}_2 - \ldots \\ m_3l_2l_3C_2S_{23}\dot{\theta}_1\dot{\theta}_2 - m_3l_3^2S_{23}C_{23}\dot{\theta}_1\dot{\theta}_3 - m_3l_2l_3C_2S_{23}\dot{\theta}_1\dot{\theta}_3 \end{aligned} \tag{3.50}$$

$$\begin{aligned} \tau_2 = \left(m_2 + m_3\right)l_2^2\ddot{\theta}_2 + 2m_3l_2l_3C_3\ddot{\theta}_2 + m_3l_3^2\ddot{\theta}_2 + m_3l_3^2\ddot{\theta}_3 + \ldots \\ m_3l_2l_3C_3\ddot{\theta}_3 + m_3l_3^2S_{23}C_{23}\dot{\theta}_1^2 + m_3l_2l_3S_2C_{23}\dot{\theta}_1^2 + \ldots \\ \left(m_2 + m_3\right)l_2^2S_2C_2\dot{\theta}_1^2 + m_3l_2l_3C_2S_{23}\dot{\theta}_1^2 - m_3l_2l_3S_3\dot{\theta}_3\dot{\theta}_2 - \ldots \\ m_3l_2l_3S_3\dot{\theta}_3^2 - m_3l_2l_3S_3\dot{\theta}_2\dot{\theta}_3 - m_3gl_3S_{23} - \ldots \\ \left(m_2 + m_3\right)gl_2S_2 \end{aligned} \tag{3.51}$$

$$\begin{aligned} \tau_3 = m_3l_3^2\ddot{\theta}_2 + m_3l_2l_3C_3\ddot{\theta}_2 + m_3l_3^2\ddot{\theta}_3 + m_3l_3^2S_{23}\dot{\theta}_1^2 + \ldots \\ m_3l_2l_3C_2S_{23}\dot{\theta}_1^2 + m_3l_2l_3S_3\dot{\theta}_2^2 - m_3gl_3S_{23} \end{aligned} \tag{3.52}$$

The following equations represent the dynamics of the n-degrees of freedom robotic manipulator (Chen, 2018):

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau \tag{3.53}$$

Where $\theta, \dot{\theta}, \ddot{\theta} \in R^3$ are angular position, angular velocity, and angular acceleration vectors respectively for each joint, $M(\theta) \in \mathrm{R}^{3\times 3}$ is matrix of inertia, $C(\theta,\dot{\theta}) \in \mathrm{R}^{3\times 3}$ represents the Coriolis and centripetal matrix, $G(\theta) \in \mathrm{R}^3$ is gravity term, and $\tau \in R^3$ is torque acting on the joints (Truong et al., 2019).

**Property 1:** $M(\theta)$ is symmetric and positive definite matrix (Vasquez & Correa, 2007). It is generally believed to be $M^T(\theta) = M(\theta)$.

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{3.54}$$

Where

$$\begin{cases} m_{11} = m_3 l_3^2 C_{23}^2 + 2m_3 l_2 l_3 C_2 C_{23} + \left(m_2 + m_3\right) l_2^2 C_2^2 \\ m_{12} = m_{31} = m_{13} = m_{21} = 0 \\ m_{22} = \left(m_2 + m_3\right) l_2^2 + 2m_3 l_2 l_3 C_3 + m_3 l_3^2 \\ m_{23} = m_3 l_3^2 + m_3 l_2 l_3 C_3 \\ m_{32} = m_3 l_3^2 + m_3 l_2 l_3 C_3 \\ m_{33} = m_3 l_3^2 \end{cases}$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{3.55}$$

Where

$$
\begin{cases}
c_{11} = -m_3 l_3^2 C_{23} S_{23} \left(\dot{\theta}_2 + \dot{\theta}_3\right) - \left(m_2 + m_3\right) l_3^2 C_2 S_2 \dot{\theta}_2 - \ldots \\
\qquad\qquad m_3 l_2 l_3 \left(S_2 C_{23} \dot{\theta}_2 + C_2 S_{23} \left(\dot{\theta}_2 + \dot{\theta}_3\right)\right) \\
c_{12} = -m_3 l_3^2 S_{23} C_{23} \dot{\theta}_1 - m_3 l_2 l_3 S_2 C_{23} \dot{\theta}_1 - \left(m_2 + m_3\right) l_3^2 S_2 C_2 \dot{\theta}_1 - \ldots. \\
\qquad\qquad m_3 l_2 l_3 C_2 S_{23} \dot{\theta}_1 \\
c_{13} = -m_3 l_3^2 S_{23} C_{23} \dot{\theta}_1 - m_3 l_2 l_3 C_2 S_{23} \dot{\theta}_1 \\
c_{21} = m_3 l_3^2 S_{23} C_{23} \dot{\theta}_1 + m_3 l_2 l_3 S_2 C_{23} \dot{\theta}_1 + \left(m_2 + m_3\right) l_2^2 S_2 C_2 \dot{\theta}_1 + \ldots \\
\qquad\qquad m_3 l_2 l_3 C_2 S_{23} \dot{\theta}_1 \\
c_{22} = m_3 l_2 l_3 S_3 \dot{\theta}_3 \\
c_{23} = -m_3 l_2 l_3 S_3 \dot{\theta}_3 - m_3 l_2 l_3 S_3 \dot{\theta}_2 \\
c_{31} = m_3 l_3^2 S_{23} \dot{\theta}_1 + m_3 l_2 l_3 C_2 S_{23} \dot{\theta}_1 \\
c_{32} = m_3 l_2 l_3 S_3 \dot{\theta}_2 \\
c_{33} = 0
\end{cases}
$$

**Property 2:** $\dot{M}(\theta) - 2C(\theta, \dot{\theta})$ is skew symmetric matrix and fulfils (Kuo et al., 2011)

$$x^T \left[\dot{M}(\theta) - 2C(\theta, \dot{\theta})\right] x = 0, \forall x \in R^3 \tag{3.56}$$

Where x is non zero vector.

$$G(\theta) = \begin{bmatrix} 0 \\ -m_3 g l_3 S_{23} - \left(m_2 + m_3\right) g l_2 S_2 \\ -m_3 g l_3 S_{23} \end{bmatrix} \tag{3.57}$$

### 3.5.3 Considered friction

Friction can have a negative impact on the accuracy of position control in robotic manipulators, particularly when the manipulator is operating at slow speeds (Ashagrie et al., 2021). A precise and straightforward friction model can significantly enhance the overall performance of the accuracy and control stability of the robotic manipulator system. The friction model is described in (3.58) (Andr´ & Bittencourt, 2007).

$$F\left(\dot{\theta}\right) = F_v \dot{\theta} + F_s sgn\left(\dot{\theta}\right) \tag{3.58}$$

Where; $F\left(\dot{\theta}\right)$ is Frictional force, $\dot{\theta}$ is joint velocity, $F_v$ is viscous damping coefficient (Ashagrie et al., 2021) and $F_s$ is coulomb friction coefficient. Damping coefficient is 0.003(Ghazanfar Shahgholian & Pegah Shafaghi, 2010) and coulomb friction coefficient is 0.04(Moldoveanu, 2014).

By adding (3.43) and (3.58) the dynamic model of the manipulator considering friction is stated in (3.59).

$$M(\theta)\ddot{\theta}+C(\theta,\dot{\theta})\dot{\theta}+G(\theta)+F\left(\dot{\theta}\right)=\tau \tag{3.59}$$

### 3.5.4 Considered disturbance

When an industrial articulated robot lifts objects of varying masses and moves them from one location to another, this changes the dynamics model of the manipulator. As a result, the robot manipulator becomes more susceptible to load uncertainty and torque vibration. The general dynamic model, which accounts for disturbance uncertainty, is presented in (3.60)

$$M(\theta)\ddot{\theta}+C(\theta,\dot{\theta})\dot{\theta}+G(\theta)+F\left(\dot{\theta}\right)+\tau_d=\tau \tag{3.60}$$

Where; $\tau_d$ is disturbance torque due to payload variation and vibration.

Let $X$ , $\theta$ and $\dot{\theta}$ are state vector, angular position and angular velocity respectively.

$\theta=\left[\theta_1,\theta_2,\theta_3\right]^T$ and $\dot{\theta}=\left[\dot{\theta}_1,\dot{\theta}_2,\dot{\theta}_3\right]^T$ then, state vector can be described as follows:

$$X=\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} \tag{3.61}$$

$$\begin{bmatrix}\dot{X}_1\\ \dot{X}_2\end{bmatrix}=\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} \tag{3.62}$$

Hence, by inserting (3.60) into (3.62) nonlinear state space form can be written as follows:

$$\begin{bmatrix}\dot{X}_1\\ \dot{X}_2\end{bmatrix}=\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix}=\begin{bmatrix}\dot{\theta}\\ M(\theta)^{-1}\left(\tau-\left(C(\theta,\dot{\theta})\dot{\theta}+G(\theta)+F\left(\dot{\theta}\right)+\tau_d\right)\right)\end{bmatrix} \tag{3.63}$$

# CHAPTER FOUR

# CONTROLLER DESIGN

## 4.1 Introduction

Robot manipulators need a nonlinear controller approach for maximum functionality and performance since they are complex systems with nonlinear dynamics. The high-precision motion control of robot manipulators is challenging because of the complex dynamics and uncertainty (Zheng et al., 2019). Numerous control strategies were created and thoroughly researched in order to obtain effective motion tracking (Zhao et al., 2010). In this chapter we will focus on nonlinear control mechanism known as sliding mode control. In a number of applications, this control method has been used; because of its reliability and robustness.

Even though sliding mode controller is quick and reliable, it has certain drawbacks. SMC approaches often use a high switching gain that is bigger than the expected border of uncertainties in order to ensure resilience to uncertainties or disruptions. But this switching term causes chattering phenomena, which can cause issues like overheating and saturation of the robotic mechanical parts. Several techniques were devised to stop chattering (Van et al., 2014; Zheng et al., 2019). Since it is resilient to uncertainties and reduces the chattering phenomena, super twisting sliding mode controller (STSMC) approaches are attractive (Ozer et al., 2016). In this thesis for 3-DOF articulated robotic manipulator super twisting sliding mode controller is designed and optimized using PSO, and compared with PSO optimized PID and PSO optimized SMC.

## 4.2 Super Twisting Sliding Mode Controller Design

To solve the stated control issue using the super-twisting sliding mode method, we first define the formulation of the tracking error dynamics. We want to compel the outputs to have zero tracking errors. Tracking errors are expressed as follows:

$$\begin{cases} e_1 = \theta_1 - \theta_{r1} \\ e_2 = \theta_2 - \theta_{r2} \\ e_3 = \theta_3 - \theta_{r3} \end{cases} \tag{4.1}$$

$$e = \left[ e_1 \ e_2 \ e_3 \right]^T \tag{4.2}$$

Where; $\theta_i$ is i$^{th}$ angular position and $e_i$ is i$^{th}$ tracking error.

Creating a sliding surface that allows the plant that is constrained to the sliding surface to respond in the way that is required for the system is the crucial stage in SMC design. The state variables of the plant dynamics are therefore confined to fulfill a different set of equations that define the so-called switching surface. In considering this, the sliding mode surface of the suggested controller can be created as follows (Al-Dujaili et al., 2020):

$$s = ce + \dot{e},\ c > 0 \tag{4.3}$$

Where c is sliding surface constant; $e$ and $\dot{e}$ are the tracking error and rate of change of tracking error respectively. $e$ and $\dot{e}$ are expressed as follows:

$$e = \theta - \theta_r = \left[\theta_1 - \theta_{r1}\ \ \theta_2 - \theta_{r2}\ \ \theta_3 - \theta_{r3}\right]^T \tag{4.4}$$

$$\dot{e} = \dot{\theta} - \dot{\theta}_r = \left[\dot{\theta}_1 - \dot{\theta}_{r1}\ \ \dot{\theta}_2 - \dot{\theta}_{r2}\ \ \dot{\theta}_3 - \dot{\theta}_{r3}\right]^T \tag{4.5}$$

Where $\theta_1, \theta_2, \theta_3, \theta_{r1}, \theta_{r2}$ and $\theta_{r3}$ are output trajectory of link-1, output trajectory of link-2, output trajectory of link-3, the desired trajectory of link-1, the desired trajectory of link-2 and the desired trajectory of link-3 respectively.

The super-twisting sliding mode algorithm is an exemplary robust control algorithm specifically designed to handle systems with a relative degree of one (Li & Peng, 2021). The switching control law is described in (4.6) as follows using the super-twisting control algorithm.

$$u_{sw} = -\begin{bmatrix} k_{11}\left(\left|s_1\right|\right)^{1/2} sgn\left(s_1\right) \\ k_{12}\left(\left|s_2\right|\right)^{1/2} sgn\left(s_2\right) \\ k_{13}\left(\left|s_3\right|\right)^{1/2} sgn\left(s_3\right) \end{bmatrix} - \begin{bmatrix} k_{21}\int sgn\left(s_1\right) d\tau \\ k_{22}\int sgn\left(s_2\right) d\tau \\ k_{23}\int sgn\left(s_3\right) d\tau \end{bmatrix} \tag{4.6}$$

Set $k_1$ and $k_2$ as: $k_1 = \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{12} & 0 \\ 0 & 0 & k_{13} \end{bmatrix}, k_2 = \begin{bmatrix} k_{21} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_{23} \end{bmatrix}$

Note $k_1$ and $k_2$ are user defined function gain parameters of the STSMC controller satisfying that: $k_{1i} > 0, k_{2i} > 0$ and $k_{3i} > 0$. Where $i$ = 1, 2, 3.

Sgn is a signum function which is defined as:

$$\mathrm{sgn(s)} = \begin{cases} +1 & if\ s(\theta_1, \theta_2, \theta_3) > 0 \\ 0 & if\ s(\theta_1, \theta_2, \theta_3) = 0 \\ -1 & if\ s(\theta_1, \theta_2, \theta_3) < 0 \end{cases} \tag{4.7}$$

The super-twisting SMC law for perturbation and chattering elimination is given by:

$$u_{sw} = -k_1 \left(|s|\right)^{1/2} \mathrm{sgn(s)} + w \tag{4.8}$$

$$\dot{w} = -k_2\, \mathrm{sgn(s)}$$

From previously listed, the dynamic model of uncertain 3-DOF manipulator is written in the form of equation stated below.

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + F = u \tag{4.9}$$

Suppose $M(\theta)\ddot{\theta}$ , $C(\theta, \dot{\theta})$ and $G(\theta)$ are known, $F$ unknown, represents disturbance friction and uncertainty in one, and all states are measured. Assuming that sliding surface is designed as (4.3), and error is defined as (4.4)

By using $\ddot{\theta}$ from (4.9), the time derivative of sliding surface's' is given by:

$$\begin{aligned} \dot{s} &= c\dot{e} + \ddot{e} = c(\dot{\theta} - \dot{\theta}_r) + \ddot{\theta} - \ddot{\theta}_r \\ &= c(\dot{\theta} - \dot{\theta}_r) + M^{-1}(u - C(\theta, \dot{\theta})\dot{\theta} - G(\theta) - F) - \ddot{\theta}_r \end{aligned} \tag{4.10}$$

Define Lyapunov function as

$$V = \frac{1}{2} s^2 \tag{4.11}$$

Differentiating $V$ with inserting (4.9) and (4.10)

$$\dot{V} = s\dot{s} = s\left[c\dot{\theta} + M^{-1}(u - C(\theta, \dot{\theta})\dot{\theta} - G(\theta) - F) - c\dot{\theta}_r - \ddot{\theta}_r\right] \tag{4.12}$$

**Theorem 4.1**. For rigid n-degrees of freedom articulated robotic manipulator (4.9), if the sliding surface is chosen as (4.3), the switching law is as (4.8), and then STSMC control law for n-degrees of freedom manipulator is designed as:

$$u = -k_1\left(|s|\right)^{1/2}\,\mathrm{sgn}(s) + M(\theta)\ddot{\theta}_r + C(\theta,\dot{\theta})\dot{\theta}_r + G(\theta) - \int k_2\,\mathrm{sgn}(s)dt \tag{4.13}$$

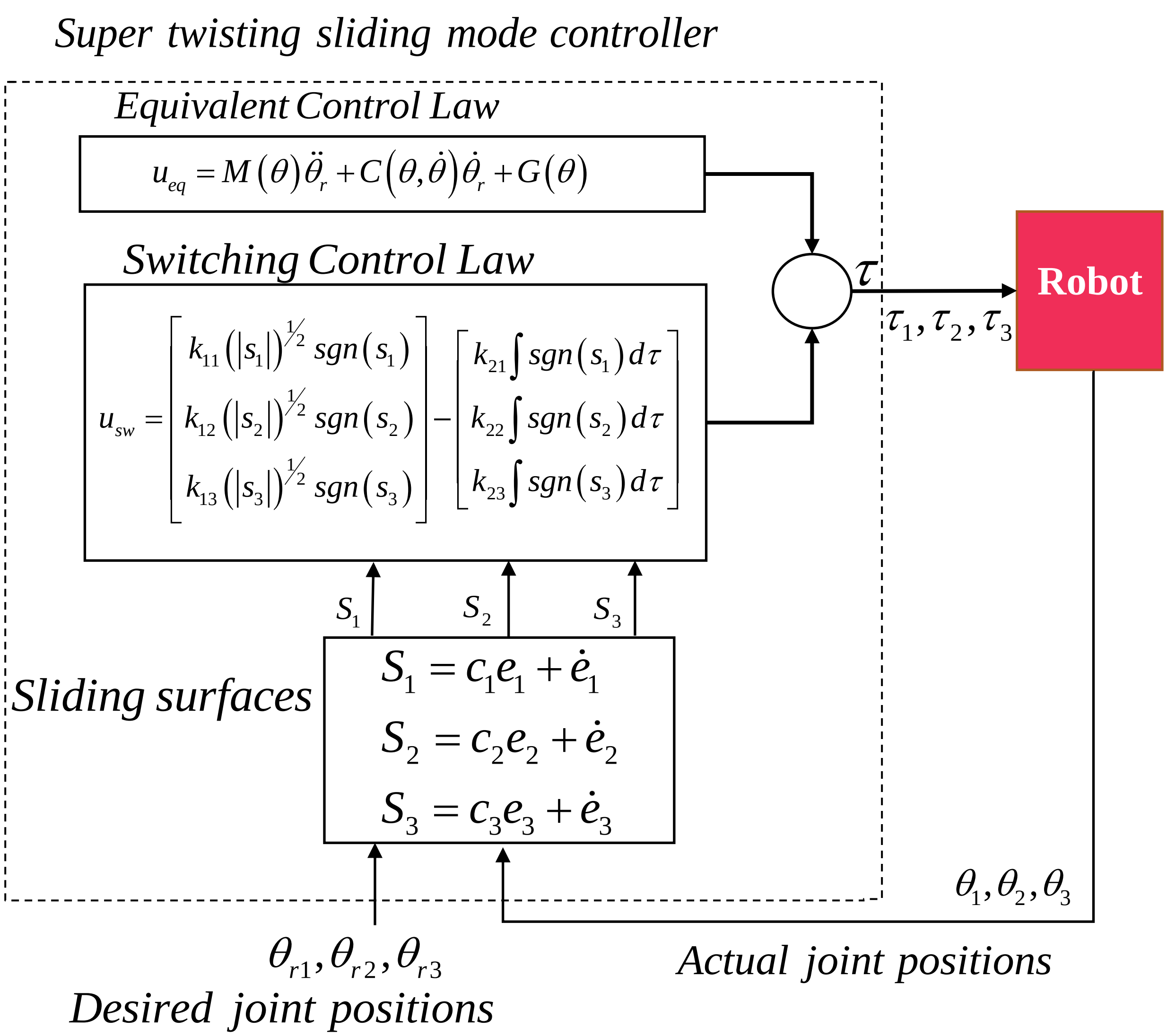


*Figure 4.1: scheme of 3-DOF manipulator controlled by STSMC.*

Where $k_1$ and $k_2$ are user defined parameters. Inserting (4.13) into (4.12) and using Lyapunov direct method the reachability condition is satisfied as:

$$\dot{V}(t) = s\dot{s} = s\left[-k_1\left(|s|\right)^{1/2}\operatorname{sgn}(s) - \int k_2 \operatorname{sgn}(s)\,\mathrm{dt} - F\right]$$
$$= -s\left[k_1\left(|s|\right)^{1/2}\operatorname{sgn}(s) + \int_0^t k_2 \operatorname{sgn}(s)\,dt + F\right] \tag{4.14}$$

$$\Rightarrow -s\left(k_1\left(|s|\right)^{1/2}\operatorname{sgn}(s) + \int_0^t k_2 \operatorname{sgn}(s)\,dt + F\right) < 0 \tag{4.15}$$

$$\Rightarrow -\left(k_1\left(|s|\right)^{3/2} + \int_0^t k_2 |s| dt\right) < 0 \tag{4.16}$$

$$\dot{V}(t) < 0 \tag{4.17}$$

From above equation for all $k_1$ and $k_2 > 0$; it is clear that $\dot{V}$ is globally negative definite (ND). Hence the designed controller is stable.

Thus, it is demonstrated that the suggested ST-SMC ensures that, even in the presence of uncertainties and disturbance, the trajectory tracking error in robotic systems will achieve zero in a finite amount of time.

## 4.3 Sliding Mode Controller Design for comparison

In sliding mode control, the selection of an appropriate sliding surface is crucial, and stability and convergence are guaranteed by utilizing the Lyapunov function. Then suitable sliding mode control law is proposed (Massaoudi et al., 2019).

By using (4.1), (4.2) and (4.3) switching control law for sliding mode control designed as follows:

$$u_{sw} = -\begin{bmatrix} k_1 s_1 \\ k_2 s_2 \\ k_3 s_3 \end{bmatrix} - \begin{bmatrix} \eta_1 \operatorname{sgn}(s_1) \\ \eta_2 \operatorname{sgn}(s_2) \\ \eta_3 \operatorname{sgn}(s_3) \end{bmatrix} \tag{4.18}$$

By setting $k$ and $\eta$ as: $k = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix}, \eta = \begin{bmatrix} \eta_1 & 0 & 0 \\ 0 & \eta_2 & 0 \\ 0 & 0 & \eta_3 \end{bmatrix}$

Sliding mode control law is given by:

$$u_{sm} = -ks - \eta \, \mathrm{sgn(s)} \tag{4.19}$$

**Theorem 4.2**. For rigid n-degrees of freedom articulated robotic manipulator (4.9), if the sliding surface is chosen as (4.3), the switching law is as (4.19), and then SMC control law for n-degrees of freedom manipulator is designed as:

$$u = -ks + M(\theta)\ddot{\theta}_r + C(\theta,\dot{\theta})\dot{\theta}_r + G(\theta) - \eta \, \mathrm{sgn(s)} \tag{4.20}$$

## 4.4 PID controller design for comparison

A PID controller is a device that directs systems towards predetermined set points by utilizing closed-loop control feedback. It operates by directly addressing the error signal, which represents the deviation between the desired set point and the actual output, in order to minimize the disparity and achieve the desired performance. The proportional (P) action, the integral (I) action, and the derivative (D) action are the three actions that make up the PID controller.

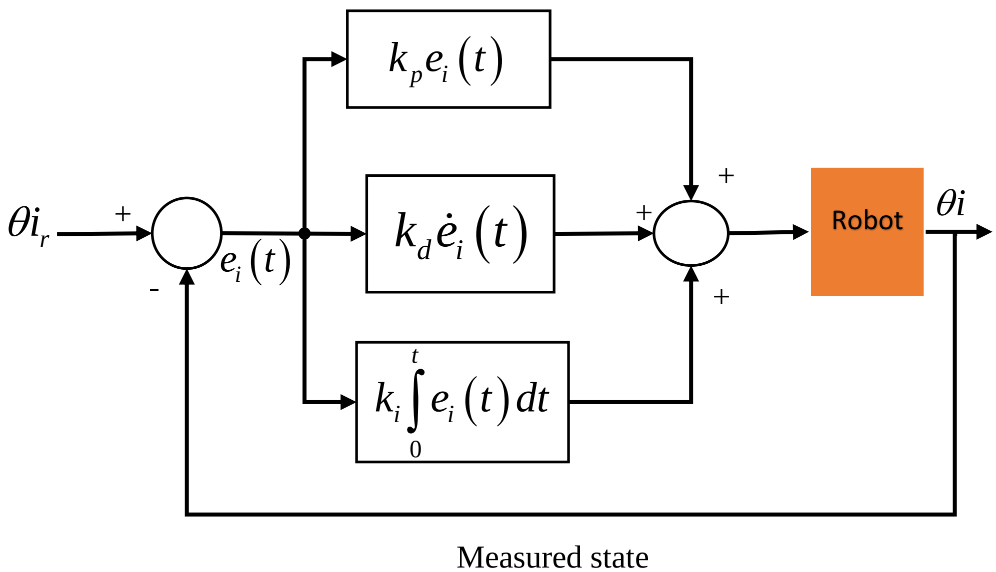


*Figure 4.2: block diagram of closed loop PID controller for robotic manipulator.*

From figure 4.2, the PID control law is stated in (4.21)

$$u_{iPID}(t) = k_p e_i(t) + k_d \dot{e}_i(t) + k_i \int_0^t e_i(t)\,dt \tag{4.21}$$

**Theorem 4.3**. For rigid n-degrees of freedom articulated robotic manipulator (4.9), if the PID control law is chosen as (4.21), then PID control law for n-degrees of freedom manipulator is designed as:

$$M\left(\theta\right)\ddot{\theta}_r + C\left(\theta,\dot{\theta}\right)\dot{\theta}_r + G\left(\theta\right) - k_p e_i\left(t\right) - k_d \dot{e}_i\left(t\right) - k_i \int_0^t e_i\left(t\right)dt \tag{4.22}$$

Where $k_i = \begin{bmatrix} k_{i1} & 0 & 0 \\ 0 & k_{i2} & 0 \\ 0 & 0 & k_{i3} \end{bmatrix}, k_d = \begin{bmatrix} k_{d1} & 0 & 0 \\ 0 & k_{d2} & 0 \\ 0 & 0 & k_{d3} \end{bmatrix}, k_p = \begin{bmatrix} k_{p1} & 0 & 0 \\ 0 & k_{p2} & 0 \\ 0 & 0 & k_{p3} \end{bmatrix}$

## 4.5 Parameter Setting of Proposed Controller

A control system's settings should be optimized for maximum performance and effectiveness. The stability and response to changes of a control system are influenced by its parameters, for example the gains of a PID controller. We can tune the system parameters to satisfy certain needs and raise its overall performance by optimizing these settings. To achieve certain objectives, such as lowering settling time for the system to stabilize or improving its accuracy, optimization involves figuring out the optimal values for these parameters. We can enhance the system's functionality, speed up its response time, and strengthen its resistance to outside influences by optimizing the settings. To sum up, optimizing the parameters of a control system is essential for achieving optimal performance, stability, and efficiency, and ensuring that the system meets the specific requirements of the application.

### 4.5.1 Particle Swarm Optimization

PSO is a detail optimization algorithm that is inspired by the behavior of swarms of animals, such as flocks of birds or schools of fish. In PSO, a population of candidate solutions, called particles, move around in the search space and communicate with each other to find the optimal solution to a given optimization problem. The algorithm works by iteratively updating the position and velocity of each particle based on its own previous best solution and the best solution found by the swarm so far. The particles move towards promising areas of the search space, guided by their own experience and the knowledge of the best solutions found by the swarm.

**Steps of Particle swarm optimization**

1. Initialization: In the search space, produce a swarm of particles at random. Each particle stands for a potential answer to the optimization issue.
2. Evaluation: Using the optimization problem's objective function, assess each particle's fitness. The solution's quality is reflected in the fitness value.
3. Updating the best solutions: Based on the fitness values, each particle should update both its own best solution and the best solution the swarm has so far discovered.
4. Updating the velocity: Each particle's velocity is updated in accordance with both its own optimal solution and the optimal solution of the swarm. The particle moves in the search space in a direction and to a certain extent based on its velocity.
5. Updating the position: Each particle's position is updated by shifting it in accordance with its velocity.
6. Stopping criteria: Stopping criteria: Verify that a requirement for termination has been satisfied. This can be a maximum for iterations, a cutoff for the fitness value, or a calculation time cap.
7. Repeat: If stopping requirement is not satisfied, return to step 2 and carry out the procedure again until it is.

These phases are repeatedly performed until the PSO algorithm hits a termination condition or converges to an ideal solution. The best answer is the one that has highest in terms of fitness in the search space.

In the Particle Swarm Optimization (PSO) technique, the trajectory of each individual in the search space is adaptively adjusted by modifying their velocity. This adjustment is based on the individual's own flying experience as well as the collective flying experience of other particles in the search space. The position vector and velocity vector of the i$^{th}$ particle in a J-dimensional search space can be expressed as: $x_i = x_{i1}, x_{i2}, ...x_{iJ}$ and $v_i = v_{i1}, v_{i2}, ...v_{iJ}$ respectively. Based on a fitness function defined by the user, suppose the optimal position of each particle, which corresponds to the best fitness value achieved by that particle at a given time, be $p_i = p_{i1}, p_{i2}, ...p_{iJ}$ and the global version of the PSO keeps track of the global best value (gbest), and its location gained. Then, the new velocities and the positions of the particles for the next fitness evaluation are calculated using the following two equations:

$$v_{iJ} = w \times v_{iJ} + c_1 \times rand() \times (p_{iJ} - x_{iJ}) + c_2 \times rand() \times (p_{gJ} - x_{iJ}) \tag{4.23}$$

$$x_{iJ} = x_{iJ} + v_{iJ} \tag{4.24}$$

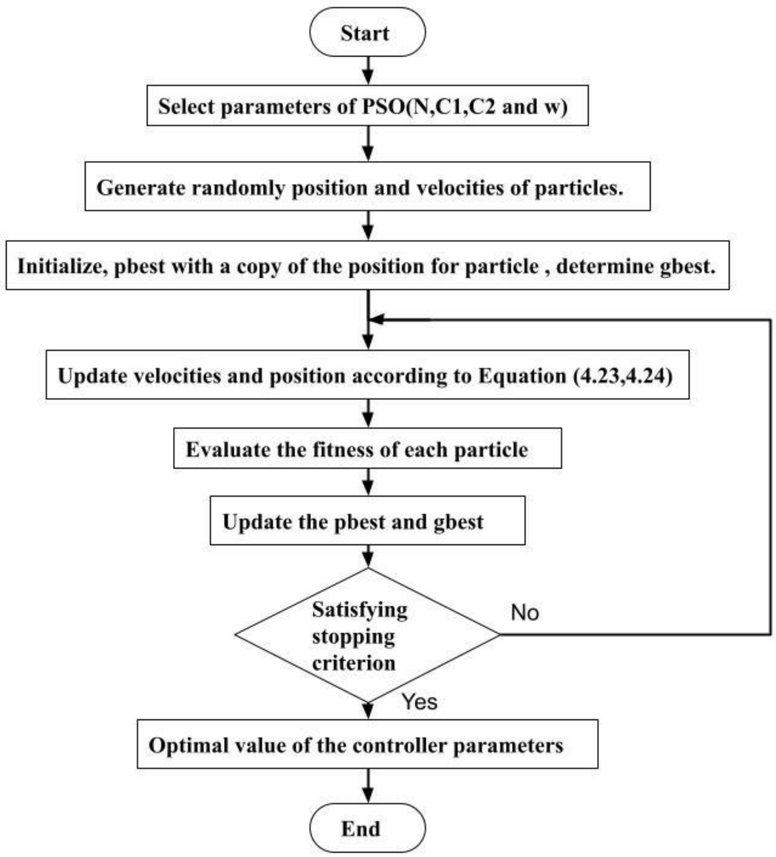


*Figure 4.3 Flow chart of Particle swarm optimization technique.*

Where $N$ is population size, $C_1$ is cognitive weight, $C_2$ is social weight, w is inertia weight $p_{iJ}$ is personal best, and $p_{gJ}$ is global best.

Hence in this thesis the objection function for the best minimization of the error is chosen in (4.25).

$$J = \int_0^t (|e(1)| + |e(2)| + |e(3)| + w \times (|du(1)| + |du(2)| + |du(3)|))dt \tag{4.25}$$

Where $|e(1)|$, $|e(2)|$ and $|e(3)|$ are absolute of error in link1, link2 and link3, and $|du(1)|$, $|du(2)|$ and $|du(3)|$ are the differences between the final desired 'u' and measured 'u'. w is weighting function, and Choosing 0.1, 0.05 and 0.05 for PID, SMC, and STSMC respectively.

The controller parameter values, tuned by PSO, are presented in Table 4.1, Table 4.2, and Table 4.3 for STSMC, SMC, and PID, respectively.

*Table 4.1: Optimal STSMC parameter values.*

| $k_{11}$ | $k_{12}$ | $k_{13}$ | $k_{21}$ | $k_{22}$ | $k_{23}$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|
| 4.2638 | 15 | 18.7927 | 0.10440 | 0.1 | 0.6321 | 5.5 | 5.5021 | 14.7123 |

*Table 4.2: Optimal SMC parameter values.*

| $\eta_1$ | $\eta_2$ | $\eta_3$ | $k_1$ | $k_2$ | $k_3$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|
| 2.0011 | 15.4403 | 17.4387 | 2.5131 | 16.8960 | 1.9549 | 7.0617 | 5.5 | 10.9145 |

*Table 4.3: Optimal PID parameter values.*

| $K_p1$ | $K_p2$ | $K_p3$ | $K_i1$ | $K_i2$ | $K_i3$ | $K_d1$ | $K_d2$ | $K_d3$ |
|---|---|---|---|---|---|---|---|---|
| 21.2747 | 21.2747 | 21.2747 | 0.1 | 0.1 | 0.1 | 4.2631 | 4.2631 | 4.2631 |

# CHAPTER FIVE

# SIMULATION RESULTS AND DISCUSSION

## 5.1 Introduction

In this chapter, a simulation is carried out using MATLAB/Simulink to evaluate the effectiveness of a suggested controller for a three-degree-of-freedom (DOF) industrial articulated robotic manipulator system. The simulation results are then compared with those of an optimized PID controller and optimized SMC with PSO. To test the trajectory tracking performance for three joints, the three controllers are evaluated, and their results are obtained. The optimized STSMC exhibits better performance compared to both the optimized PID and optimized SMC controllers, as indicated by the obtained results.

Uncontrolled dynamic Simulink diagram of the plant is described in figure 5.1. The open-loop dynamic model of the plant simulation was designed using parameters of 3-DOF robotic manipulator, which are stated in Table 3.2.

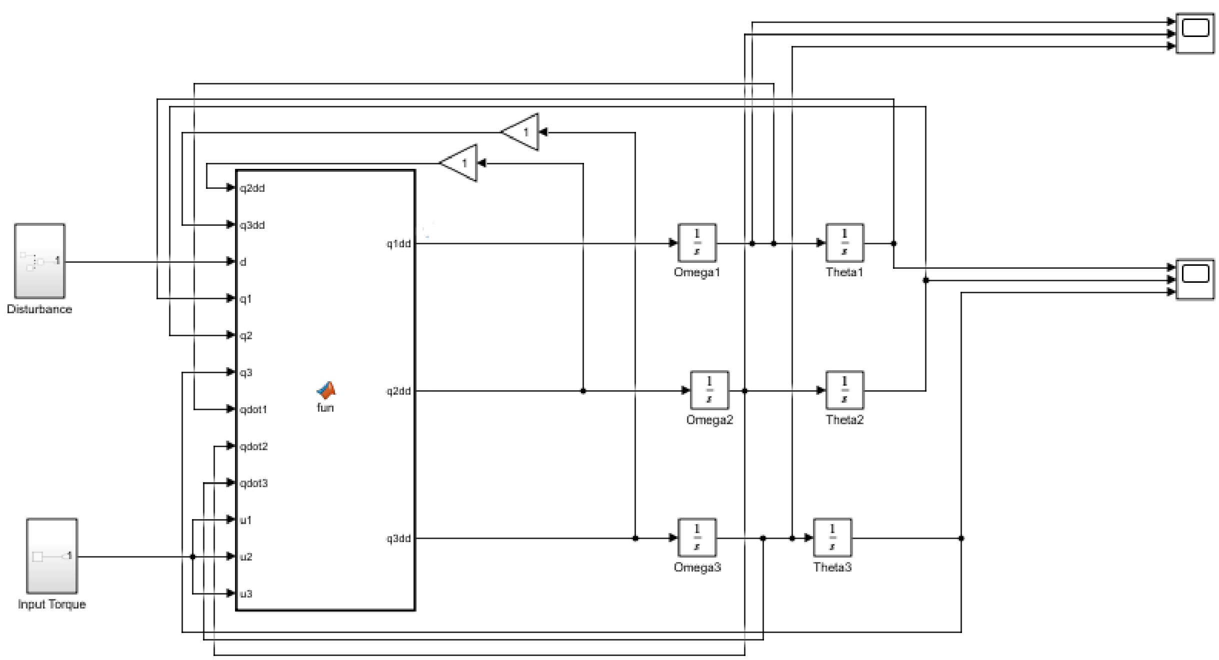


*Figure 5.1: Simulink diagram of open loop system.*

Notations in the figure 5.1 are: $q1=\theta_1, q2=\theta_2, q3=\theta_3$ are joint angle for $1^{st}$,$2^{2nd}$ and $3^{rd}$ joint respectively.

$q1dot = \dot{\theta}_1, q2dot = \dot{\theta}_2, q3dot = \dot{\theta}_3$ are joint velocity for $1^{st}, 2^{2nd}$ and $3^{rd}$ joint respectively.

$u1 = \tau_1, u2 = \tau_2, u3 = \tau_3$ are input torque for $1^{st}, 2^{2nd}$ and $3^{rd}$ joint respectively. 'Fun' is a function of nonlinear dynamic equation of motion for 3-DOF robotic arm.

The open loop nonlinear dynamic model of a 3-DOF articulated robotic manipulator was obtained at a control input of 0 Nm, based on the manipulator's parameters. The results showed that the system is highly nonlinear and unstable, as illustrated in Figure 5.2. Therefore, it is necessary to have a robust controller to stabilize the plant.

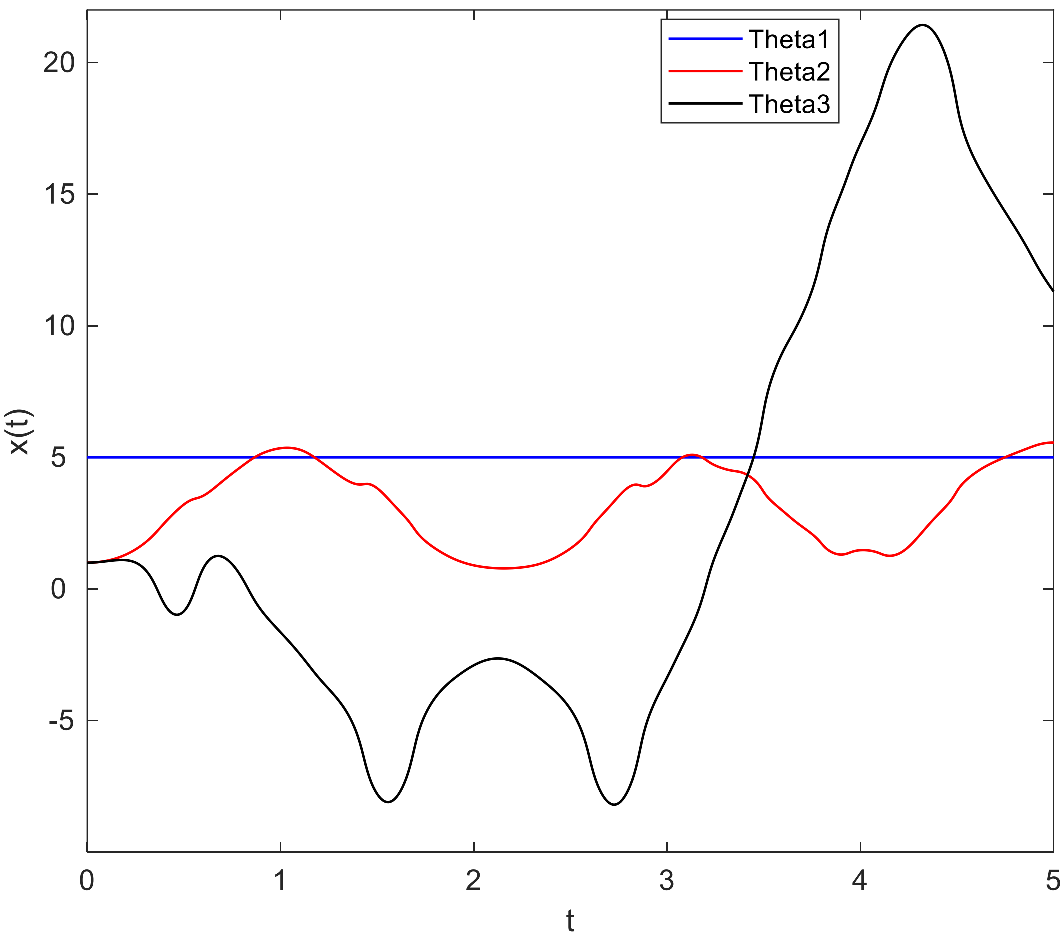


*Figure 5.2: uncontrolled open loop response of the system.*

From the figure5.2, dynamic system of the plant is unstable so it needs a controller like PID, SMC, STSMC, MPC, LQR, LQG etcetera to stabilize it. In this thesis PID, SMC, and STSMC with optimized parameters are utilized to stabilize 3-DOF articulated robotic arm.

## 5.2 Simulation Results

In this section, we present the simulation results obtained by implementing the proposed controller in the dynamic model of our system. The performance of the controller is evaluated based on its ability to track the desired trajectories, minimize the tracking errors, and stabilize the system. The simulations are conducted under disturbance and friction also. Then the results are analyzed in terms of key performance metrics such as tracking accuracy, settling time, overshoot, and robustness, and finally the best controller is chosen for our system. In this thesis three controllers such as PID, SMC and STSMC based on PSO are utilized, and finally compared to each other at the end of this section.

Initial and desired position of joints in radian are listed as follows:

$$\theta_{initial} = \begin{bmatrix} \theta_{1i} \\ \theta_{2i} \\ \theta_{3i} \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 1 \end{bmatrix}$$

$$\theta_{desired} = \begin{bmatrix} \theta_{1d} \\ \theta_{2d} \\ \theta_{3d} \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 3 \end{bmatrix}$$

### 5.2.1 Simulation Result Using STSMC Controller

The limit is established by evaluating the effectiveness of the STSMC controller. Initially, a range of arbitrary values is selected and their effectiveness and optimal gain within that range are assessed. The values are then analyzed, and the upper limit is raised to identify the ideal gain. If the gain is similar to the previous value, the earlier limit is retained to determine the best final optimal value.  Based on these the boundary is chosen for STSMC with PSO as presented in table 5.1

*Table 5.1: Tune parameter boundary of STSMC using PSO.*

| Boundary | $K_{11}$ | $K_{12}$ | $K_{13}$ | $K_{21}$ | $K_{22}$ | $K_{23}$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|---|
| **Lower** | 0 | 0 | 0 | 0.1 | 0.1 | 0.1 | 0 | 0 | 0 |
| **Upper** | 25 | 25 | 25 | 50 | 50 | 50 | 20 | 20 | 20 |

The chosen Swarm Size and Max Iterations are 50 and 40 respectively. The figure 5.3 shows particle swarm optimization function value for STSMC.

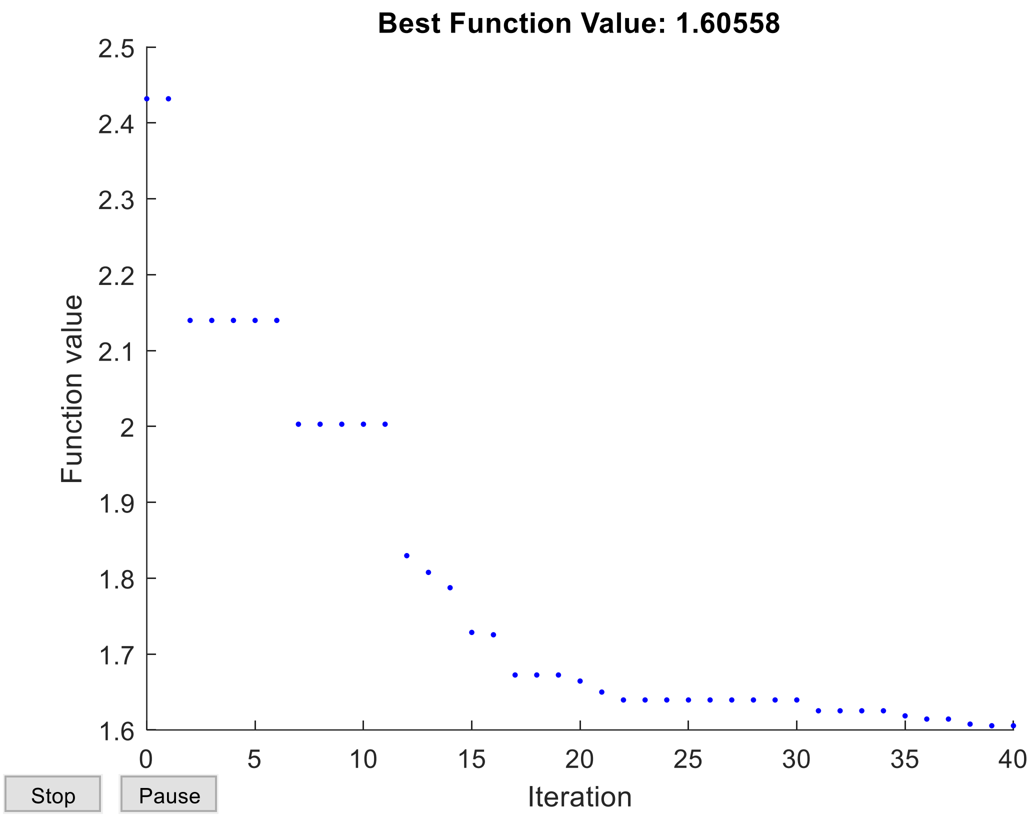


*Figure 5.3: fitness value for STSMC using PSO.*

After tuning, Table 5.2 showcases the optimal values of the STSMC parameters that correspond to the minimum cost function value obtained through particle swarm optimization.

*Table 5.2: Optimal STSMC gain tuned using PSO.*

| $k_{11}$ | $k_{12}$ | $k_{13}$ | $k_{21}$ | $k_{22}$ | $k_{23}$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|
| 4.2638 | 15 | 18.7927 | 0.10440 | 0.1 | 0.6321 | 5.5 | 5.5021 | 14.7123 |

**Position trajectory tracking using STSMC**

The articulation results of a 3-DOF articulated robotic arm were analyzed using a STSMC controller, and the outcomes were presented both in tabular and graphical formats. Figures 5.4, 5.5, and 5.6 illustrate the position tracking of the robotic manipulator. The obtained results reveal that the first link of the manipulator exhibits a raise time of 0.1001s, a settling time of 0.2051s, an overshoot of 0.2914%, and an undershoot of 0%. Similarly, the second link demonstrates a raise time of 0.2967s, a settling time of 0.5095s, an overshoot of 1.42%, and undershoot of 0.37580%. Lastly, the third link shows a raise time of 0.1228s, a settling time of 0.2683s, an overshoot of 0.1605%, and undershoot of 0%.

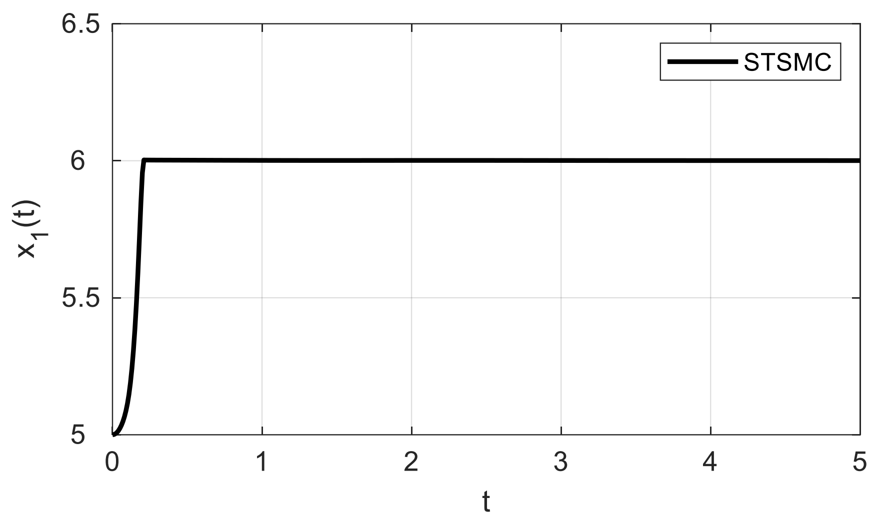


*Figure 5.4: position trajectory tracking of link1 using STSMC.*

*Table 5.3: system performance using STSMC for first link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.1001s |
| **Settling time (s)** | 0.2051s |
| **Overshoot (%)** | 0.2914% |
| **Undershoot (%)** | 0% |

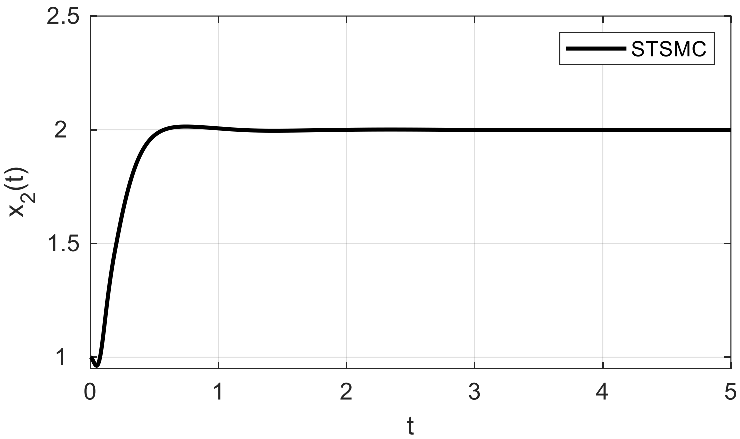


*Figure 5.5: position trajectory tracking of link2 using STSMC.*

*Table 5.4: system performance using STSMC for second link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.2967s |
| **Settling time (s)** | 0.5095s |
| **Overshoot (%)** | 1.42% |
| **Undershoot (%)** | 3.7580% |

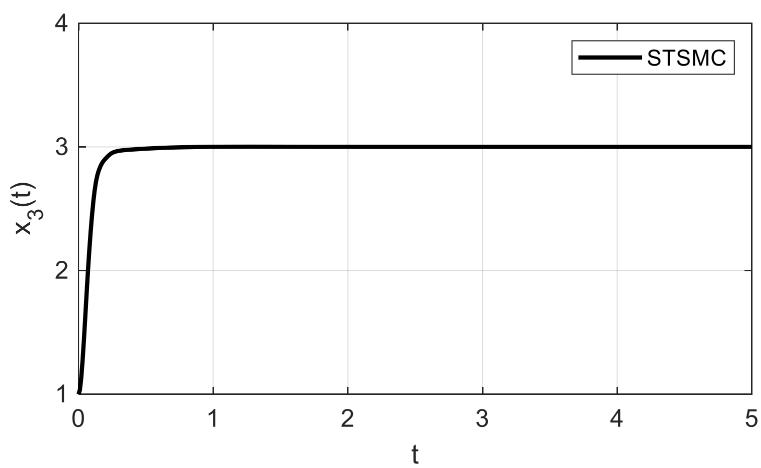


*Figure 5.6: position trajectory tracking of link3 using STSMC.*

*Table 5.5: system performance using STMC for third link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.1228s |
| **Settling time (s)** | 0.2683s |
| **Overshoot (%)** | 0.1605% |
| **Undershoot (%)** | 0% |

**STSMC control input for each input**

Input torques from each actuator using STSMC are illustrated in figures 5.7, 5.8, and 5.9 as follows

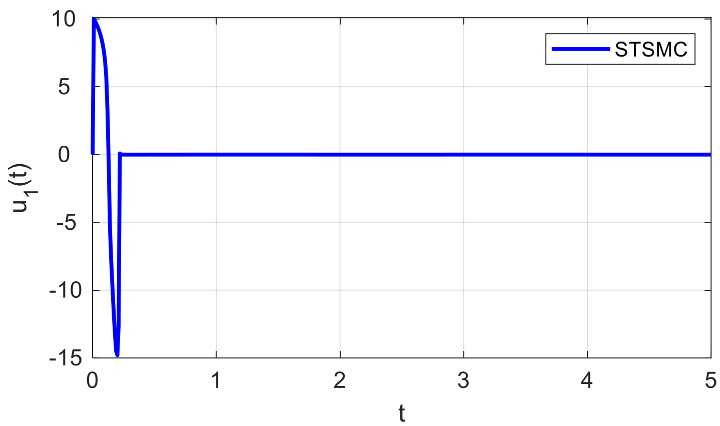


*Figure 5.7: Torque of first link of robot using STSMC.*

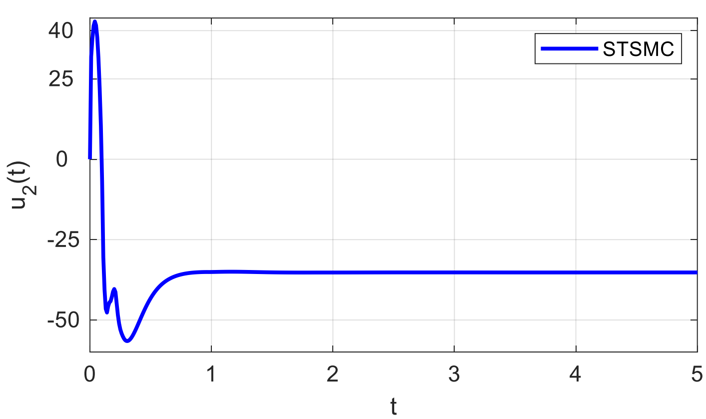


*Figure 5.8: Torque of second link of robot using STSMC.*

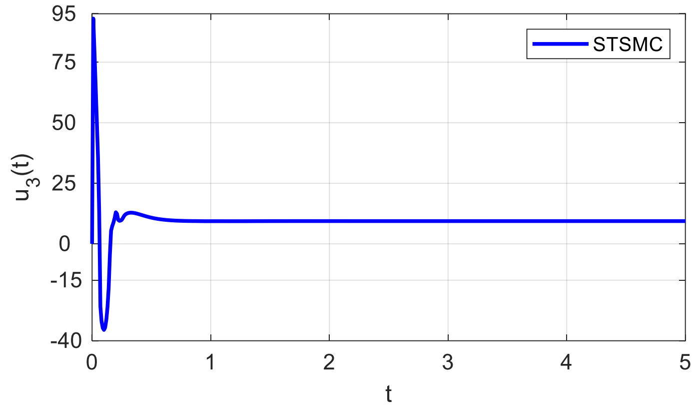


*Figure 5.9: Torque of third link of robot using STSMC.*

**Angular velocities of each joint using STSMC**

The figures labeled 5.10, 5.11, and 5.12 depict the angular velocities achieved by each actuators during the articulation process using STSMC.

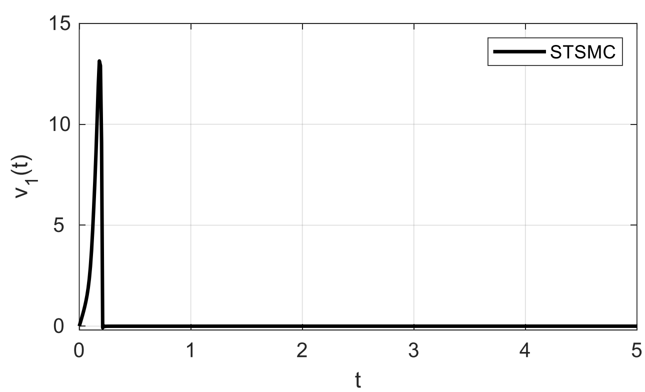


*Figure 5.10: angular velocity of first link of robotic manipulator using STSMC.*

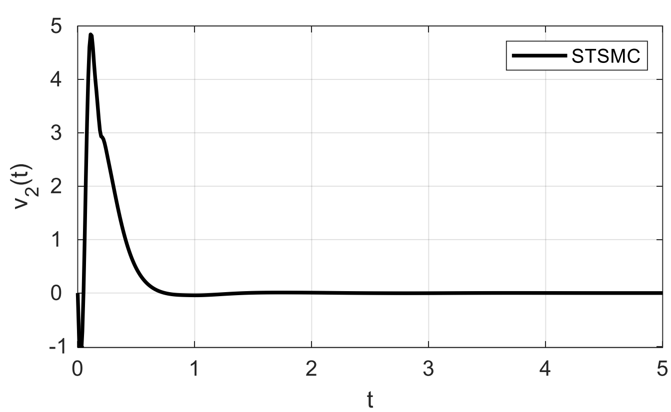


*Figure 5.11: angular velocity of second link of robotic manipulator using STSMC.*

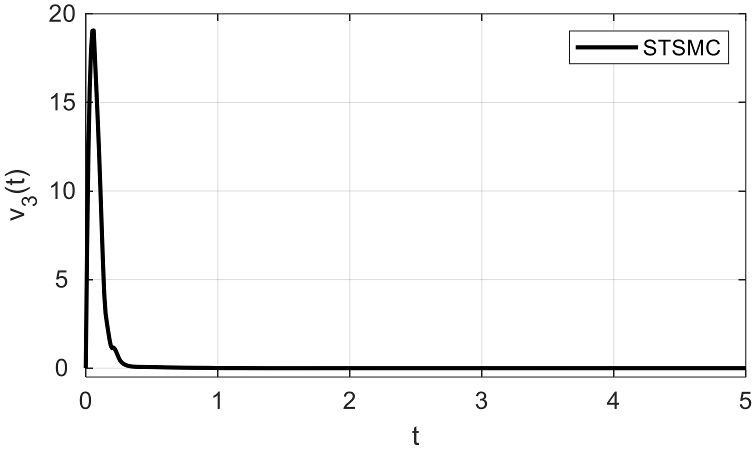


*Figure 5.12: angular velocity of third link of robotic manipulator using STSMC.*

**Angular position tracking errors using STSMC**

Angular position tracking error of each links using STSMC are illustrated in figures 5.13, 5.14, and 5.15 as follows.

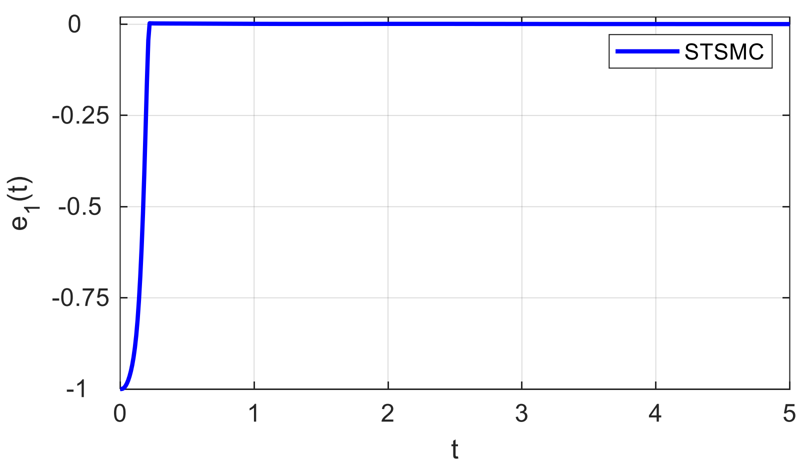


*Figure 5.13: position tracking error of first link of robotic manipulator using STSMC.*

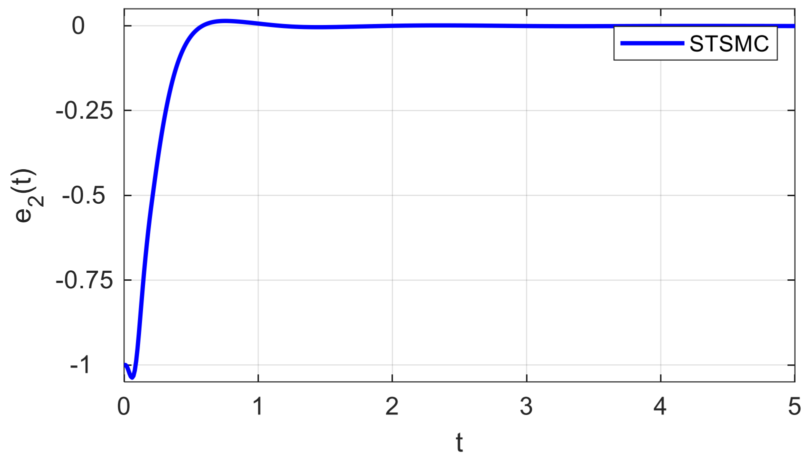


*Figure 5.14: position tracking error of second link of robotic manipulator using STSMC.*

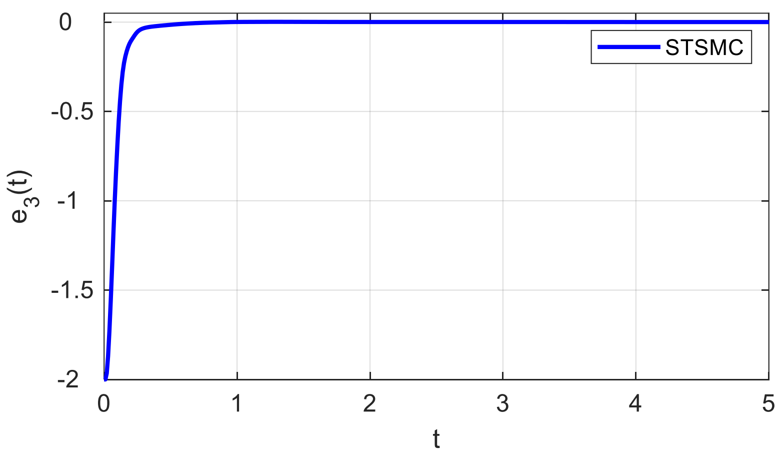


Figure 5.15: position tracking error of third link of robotic manipulator using STSMC.

### 5.2.2 Simulation Result Using SMC Controller

The limit is established by evaluating the effectiveness of the SMC controller. Initially, a range of arbitrary values is selected and their effectiveness and optimal gain within that range are assessed. The values are then analyzed, and the upper limit is raised to identify the ideal gain. If the gain is similar to the previous value, the earlier limit is retained to determine the best final optimal value. Based on these the boundary is chosen for SMC with PSO as presented in table 5.6.

*Table 5.6: Tune parameter boundary of SMC using PSO.*

| **Boundary** | $\eta_1$ | $\eta_2$ | $\eta_3$ | $k_1$ | $k_2$ | $k_3$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|---|
| **Lower** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Upper** | 20 | 20 | 20 | 50 | 50 | 50 | 20 | 20 | 20 |

The chosen Swarm Size and Max Iterations are 50 and 40 respectively. The figure 5.16 shows particle swarm optimization function value for SMC.

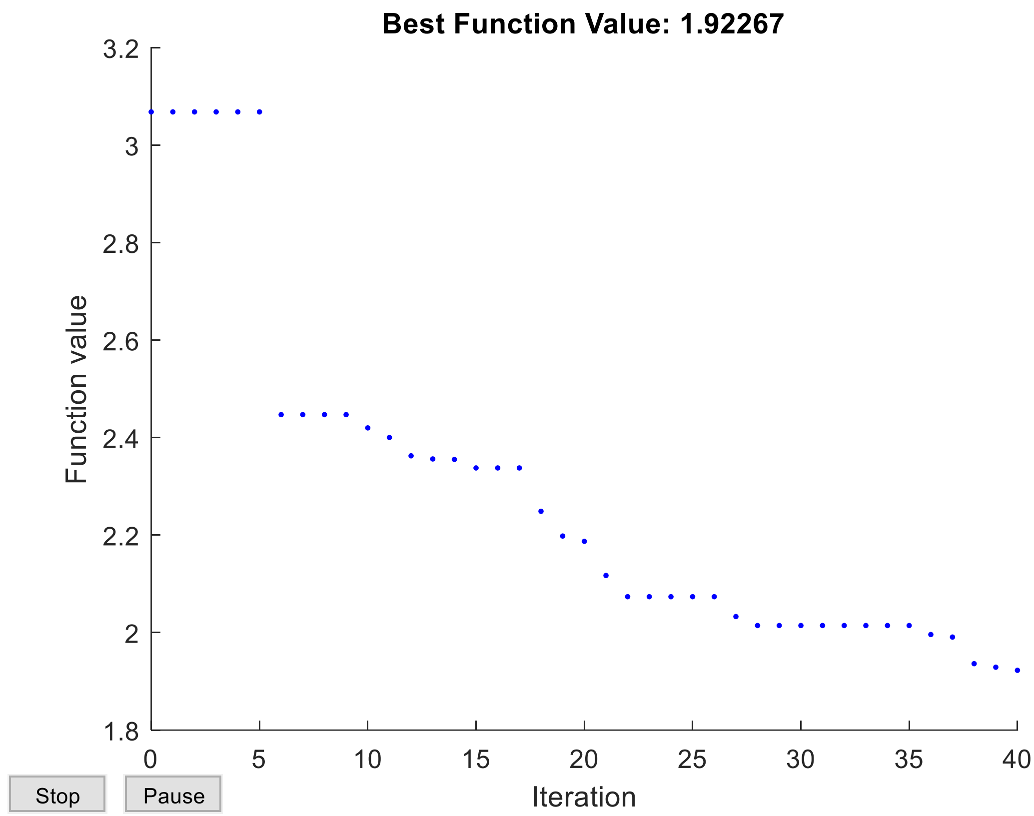


*Figure 5.16: fitness value for SMC using PSO.*

After tuning, Table 5.7 showcases the optimal values of the SMC parameters that correspond to the minimum cost function value obtained through particle swarm optimization.

*Table 5.7: Optimal SMC gain tuned using PSO.*

| $\eta_1$ | $\eta_2$ | $\eta_3$ | $k_1$ | $k_2$ | $k_3$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|
| 2.0011 | 15.4403 | 17.4387 | 2.5131 | 16.8960 | 1.9549 | 7.0617 | 5.5 | 10.9145 |

**Position trajectory tracking using SMC**

The articulation results of a 3-DOF articulated robotic arm were analyzed using a SMC controller, and the outcomes were presented both in tabular and graphical formats. Figures 5.17, 5.18, and 5.19 illustrate the position tracking of the robotic manipulator. The obtained results reveal that the first link of the manipulator exhibits a raise time of 0.1101s, a settling time of 0.2071s, an overshoot of 0.0182%, and an undershoot of 0%. Similarly, the second link demonstrates a raise time of 0.2675s, a settling time of 0.5316s, an overshoot of 0%, and undershoot of 0.3722%. Lastly, the third link shows a raise time of 0.2112s, a settling time of 0.4122s, an overshoot of 0%, and undershoot of 0%.

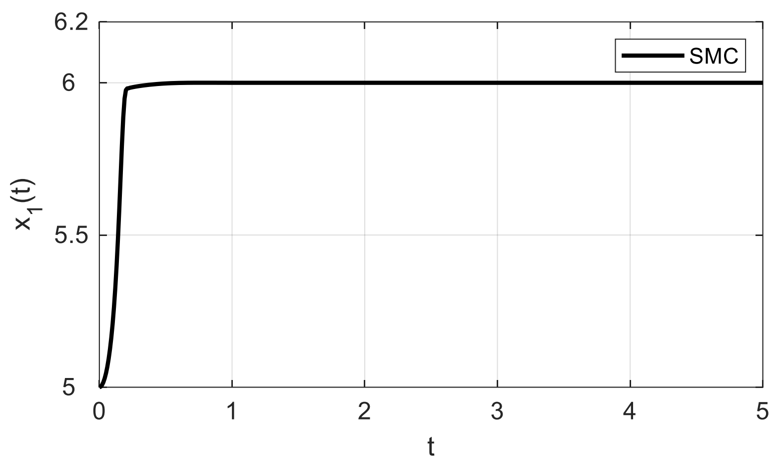


*Figure 5.17: position trajectory tracking of link1 using SMC.*

*Table 5.8: system performance using SMC for first link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.1101s |
| **Settling time (s)** | 0.2071s |
| **Overshoot (%)** | 0.0182% |
| **Undershoot (%)** | 0% |

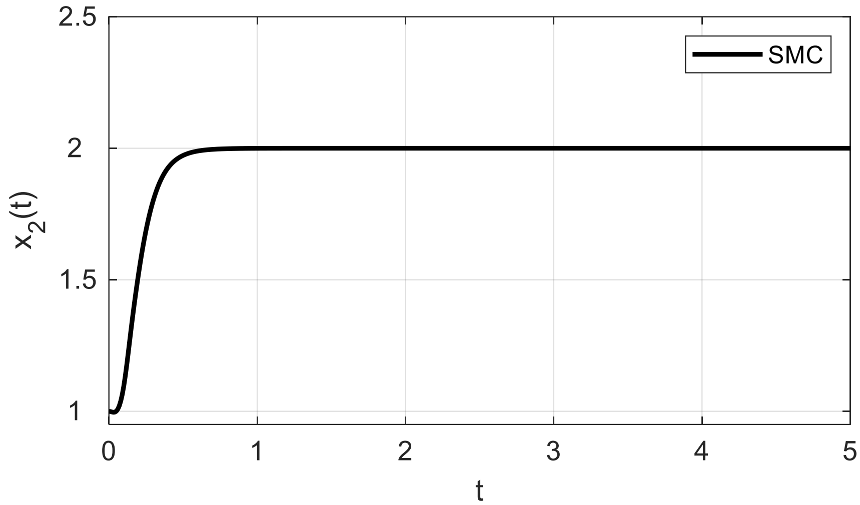


*Figure 5.18: position trajectory tracking of link2 using SMC.*

*Table 5.9: system performance using SMC for second link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.2675s |
| **Settling time (s)** | 0.5316s |
| **Overshoot (%)** | 0% |
| **Undershoot (%)** | 0.3722% |

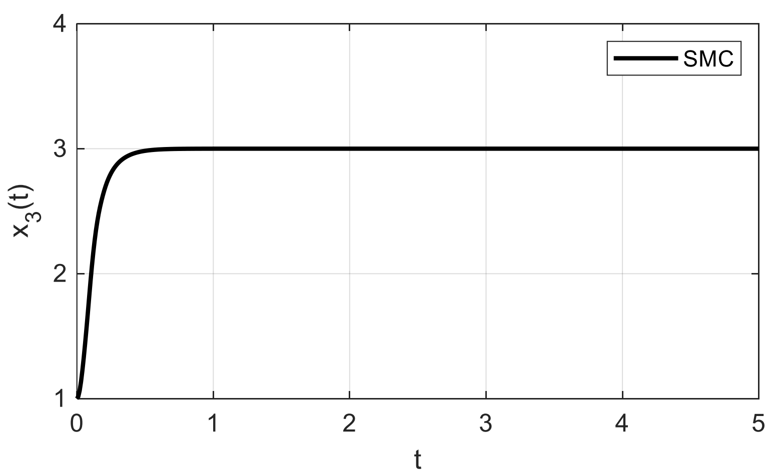


*Figure 5.19: position trajectory tracking of link3 using SMC.*

*Table 5.10: system performance using SMC for third link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.2112s |
| **Settling time (s)** | 0.4122s |
| **Overshoot (%)** | 0% |
| **Undershoot (%)** | 0% |

**SMC control input with Signum function**

In SMC (Sliding Mode Control), one notable issue is chattering, which occurs due to the discontinuous high-frequency switching control signal (sgn(s)), resulting in rapid oscillations around the sliding surface. Chattering can lead to mechanical wear and tear, introduce unnecessary vibrations, and potentially degrade control system performance. In Figures 5.20, 5.21, and 5.22, the torques of the first, second, and third links, respectively, are depicted when employing the signum function.

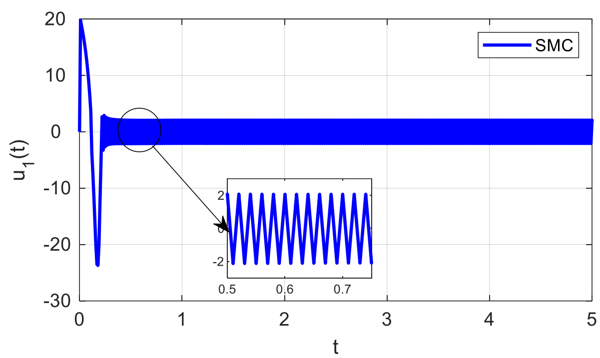


*Figure 5.20: Torque of first link of robot using SMC with signum function.*

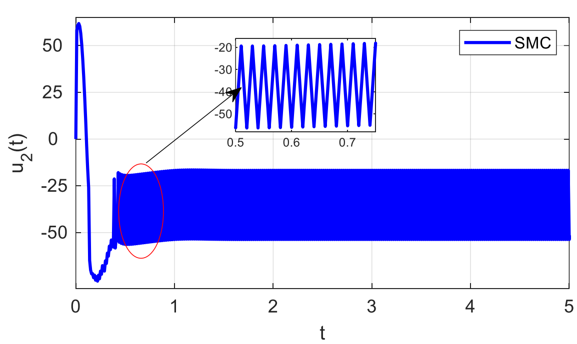


*Figure 5.21: Torque of second link of robot using SMC with signum function.*

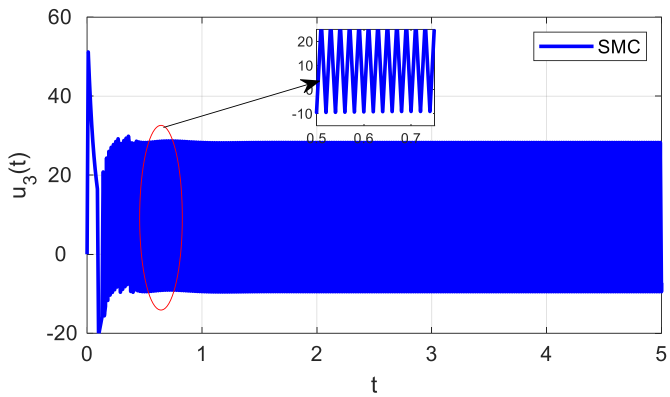


*Figure 5.22: Torque of third link of robot using SMC with signum function.*

**SMC control input with hyperbolic tangent function**

One approach to mitigate chattering is to replace the signum function (sgn(s)), with a continuous approximation that exhibits a high gain within a boundary layer. Examples of such approximations include the saturation function, relay function, and hyperbolic tangent function. Another technique to address chattering is the utilization of higher-order sliding mode control (STSMC). Figures 5.23, 5.24, and 5.25 showcase the torques of the first, second, and third links, respectively, when utilizing the tangent hyperbolic function as an alternative to mitigate chattering.

$$\tanh\left(\frac{s}{\varepsilon}\right)=\frac{e^{s/\varepsilon}-e^{-s/\varepsilon}}{e^{s/\varepsilon}+e^{-s/\varepsilon}} \tag{5.1}$$

Where $\varepsilon$ is positive function which determines the steepness of the tanh function.

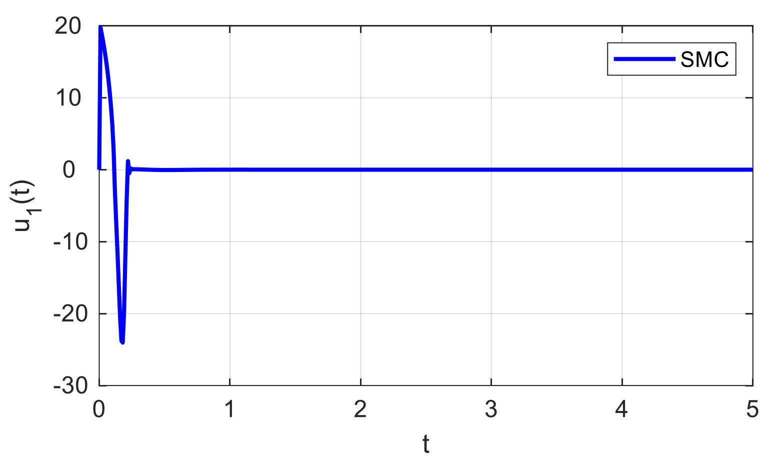


*Figure 5.23: Torque of first link of robot using SMC with hyperbolic tangent function.*

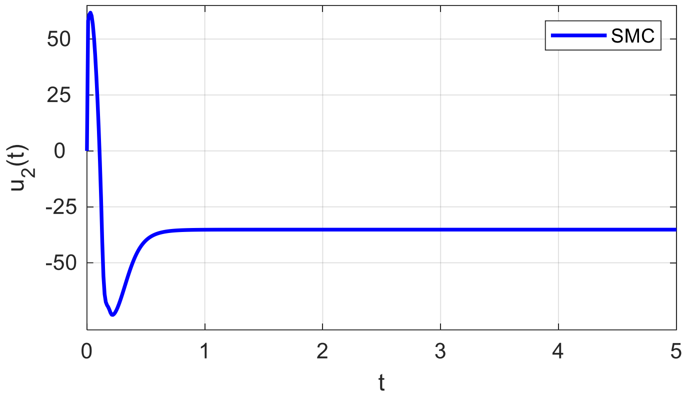


*Figure 5.24: Torque of second link of robot using SMC with hyperbolic tangent function.*

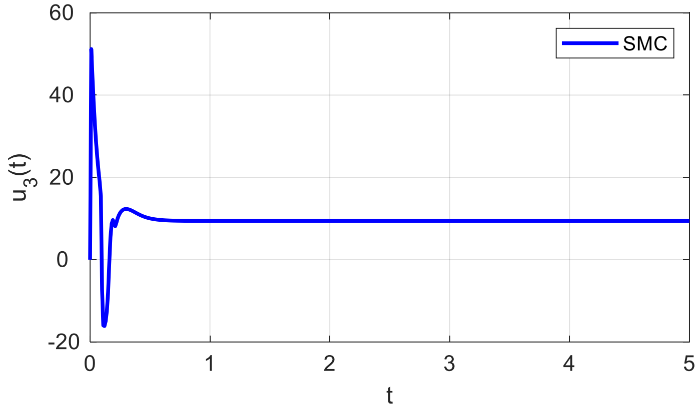


*Figure 5.25: Torque of third link of robot using SMC with hyperbolic tangent function.*

**Angular velocities of each joint using SMC**

The figures labeled 5.26, 5.27, and 5.28 depict the angular velocities achieved by each actuators during the articulation process using SMC.

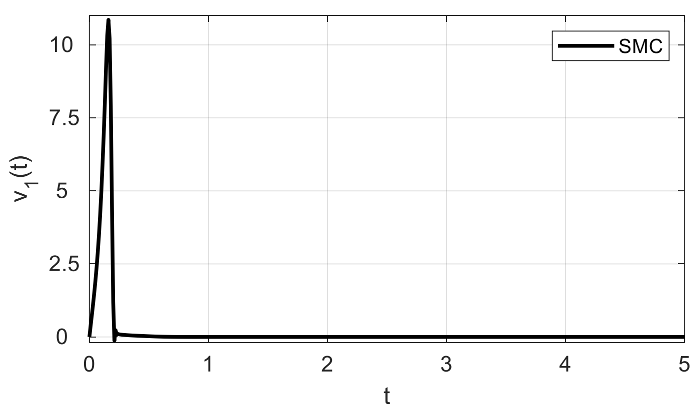


*Figure 5.26: angular velocity of first link of robotic manipulator using SMC.*

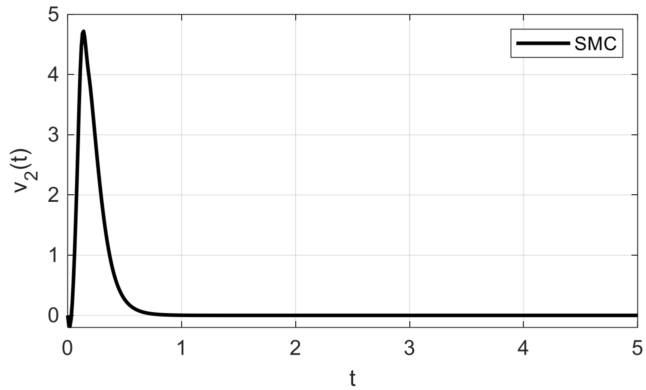


*Figure 5.27: angular velocity of second link of robotic manipulator using SMC.*

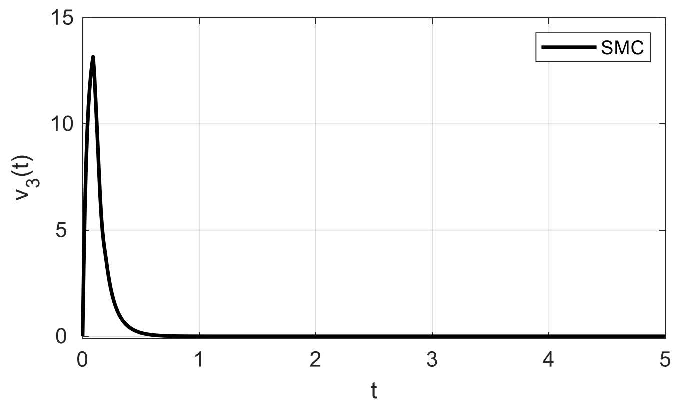


*Figure 5.28: angular velocity of third link of robotic manipulator using SMC.*

**Angular position tracking errors using SMC**

Angular position tracking error of each links using SMC are illustrated in figures 5.29, 5.30, and 5.31 as follows.

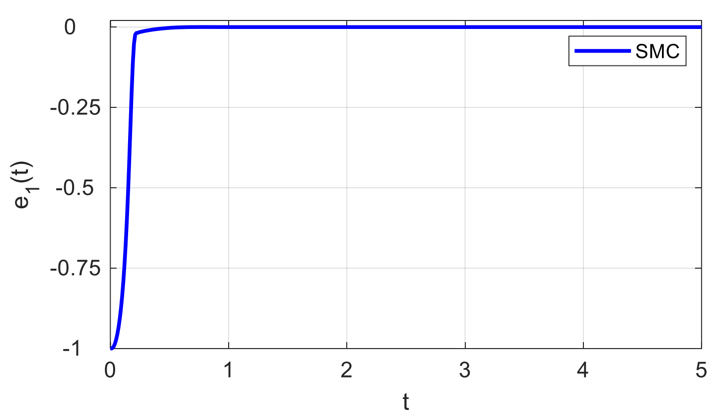


*Figure 5.29: position tracking error of first link of robotic manipulator using SMC.*

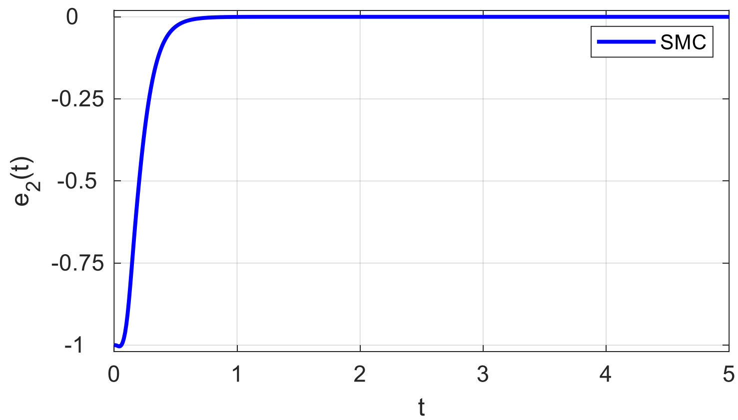


*Figure 5.30: position tracking error of second link of robotic manipulator using SMC.*

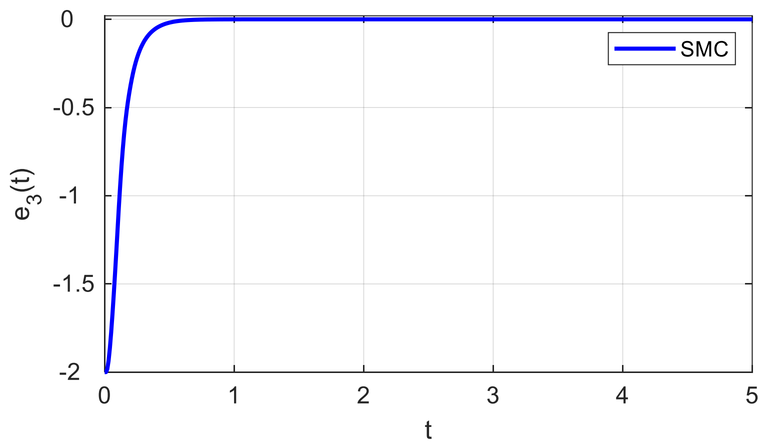


*Figure 5.31: position tracking error of third link of robotic manipulator using SMC.*

### 5.2.3 Simulation Result Using PID Controller

The limit is established by evaluating the effectiveness of the PID controller. Initially, a range of arbitrary values is selected and their effectiveness and optimal gain within that range are assessed. The values are then analyzed, and the upper limit is raised to identify the ideal gain. If the gain is similar to the previous value, the earlier limit is retained to determine the best final optimal value. Based on these the boundary is chosen for proportional, integral and derivative with PSO as presented in table 5.11.

*Table 5.11: Tune parameter boundary of PID controller using PSO.*

| **Boundary** | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|
| **Lower** | 0.1 | 0.1 | 0.1 |
| **Upper** | 30 | 20 | 20 |

The chosen Swarm Size and Max Iterations are 50 and 20 respectively. The figure 5.32 shows particle swarm optimization function value.

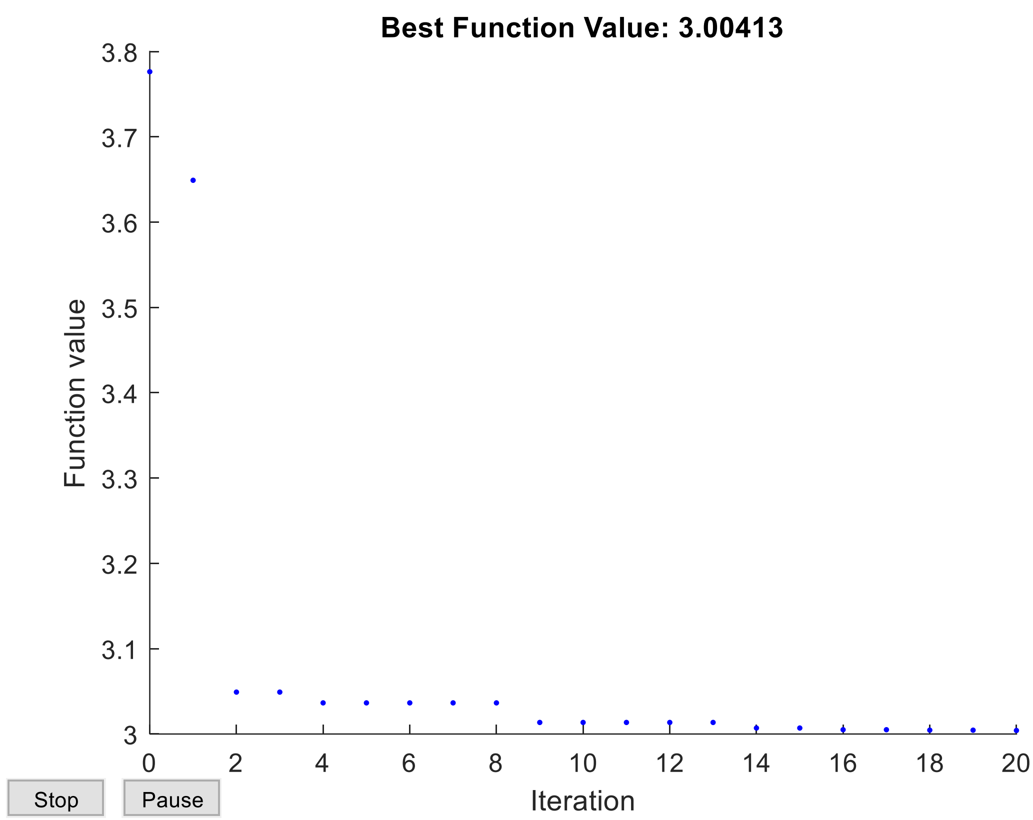


Figure 5.32: fitness value for PID using PSO.

After tuning, Table 5.12 showcases the optimal values of the PID parameters that correspond to the minimum cost function value obtained through particle swarm optimization.

*Table 5.12: Optimal PID gain tuned using PSO.*

| $K_p1$ | $K_p2$ | $K_p3$ | $K_i1$ | $K_i2$ | $K_i3$ | $K_d1$ | $K_d2$ | $K_d3$ |
|---|---|---|---|---|---|---|---|---|
| 21.2747 | 21.2747 | 21.2747 | 0.1 | 0.1 | 0.1 | 4.2631 | 4.2631 | 4.2631 |

**Position trajectory tracking using PID**

The articulation results of a 3-DOF articulated robotic arm were analyzed using a PID controller, and the outcomes were presented both in tabular and graphical formats. Figure 5.33, 5.34, and 5.35 illustrate the position tracking of the robotic manipulator. The obtained results reveal that the first link of the manipulator exhibits a raise time of 0.2260s, a settling time of 0.3543s, an overshoot of 0.1285%, and an undershoot of 0%. Similarly, the second link demonstrates a raise time of 0.3891s, a settling time of 3.2329s, an overshoot of 29.0607%, and undershoot of 3.3491%. Lastly, the third link shows a raise time of 0.3602s, a settling time of 1.2746s, an overshoot of 1.9405%, and undershoot of 0%.

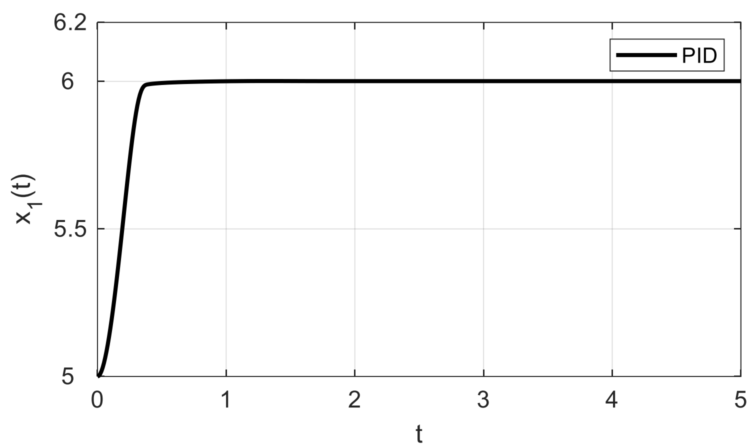


*Figure 5.33: position trajectory tracking of link1 using PID.*

*Table 5.13: system performance using PID for first link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.2260s |
| **Settling time (s)** | 0.3543s |
| **Overshoot (%)** | 0.1285% |
| **Undershoot (%)** | 0% |

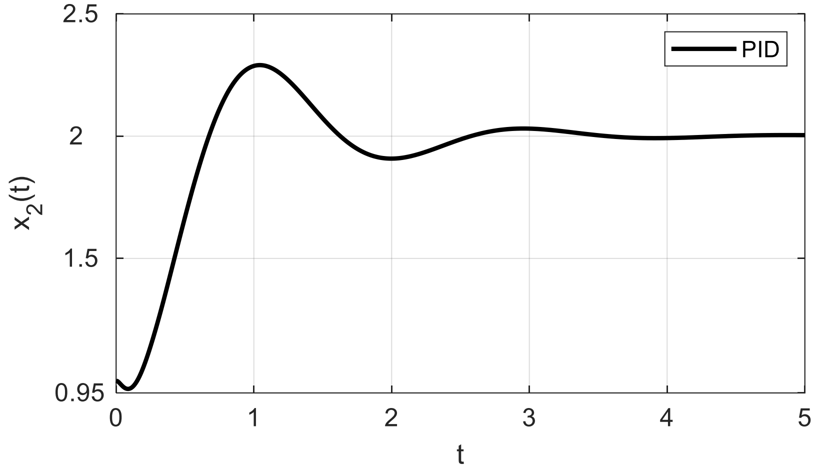


*Figure 5.34: position trajectory tracking of link 2 using PID.*

*Table 5.14 system performance using PID for second link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.3891s |
| **Settling time (s)** | 3.2329s |
| **Overshoot (%)** | 29.0607% |
| **Undershoot (%)** | 3.3491% |

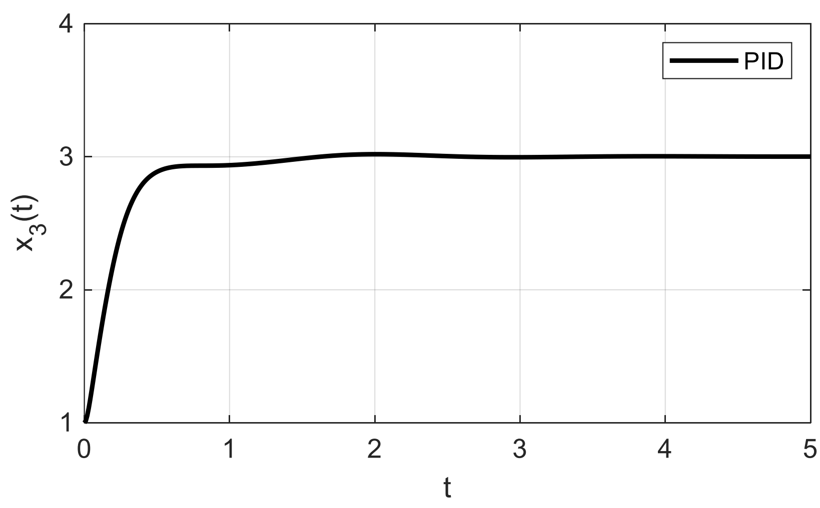


*Figure 5.35: position trajectory tracking of link 3 using PID.*

*Table 5.15: system performance using PID for third link.*

| Performance measurement | |
|---|---|
| **Rise time (s)** | 0.3602s |
| **Settling time (s)** | 1.2746s |
| **Overshoot (%)** | 1.9405% |
| **Undershoot (%)** | 0% |

**PID control input for each link**

The input torques from each actuator using PID are visually represented in figures 5.36, 5.37, and 5.38 as follows.

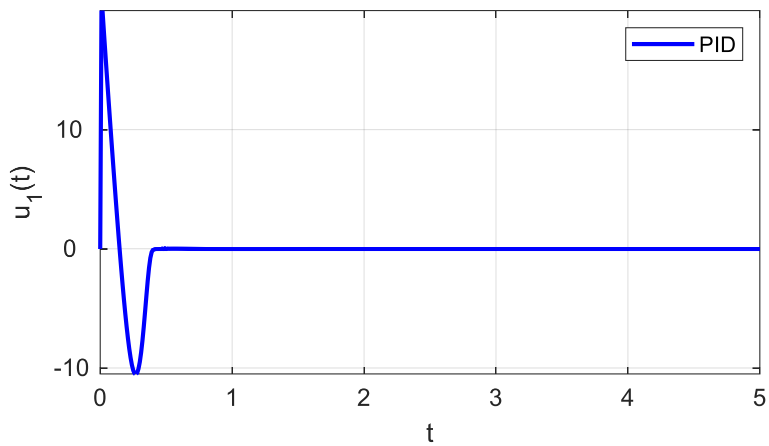


*Figure 5.36: Torque of first link of robot using PID.*

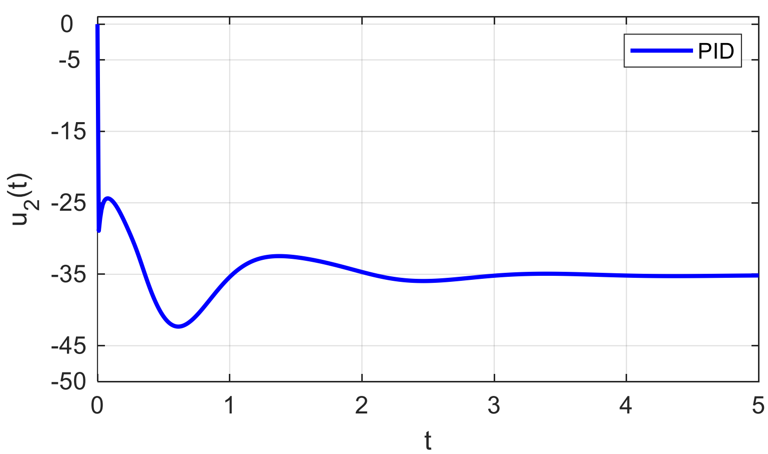


*Figure 5.37: Torque of second link of robot using PID.*

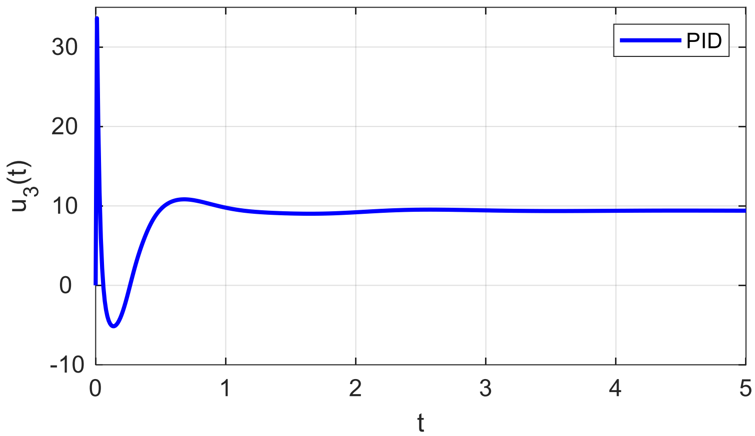


*Figure 5.38 : Torque of third link of robot using PID.*

**Angular velocities of each joint using PID**

The figures labeled 5.39, 5.40, and 5.41 depict the angular velocities achieved by individual actuators during the articulation process using PID.

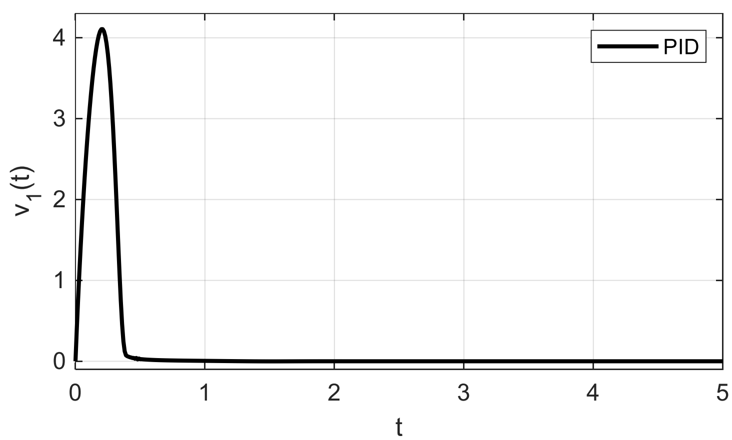


*Figure 5.39: angular velocity of first link of robotic manipulator using PID.*

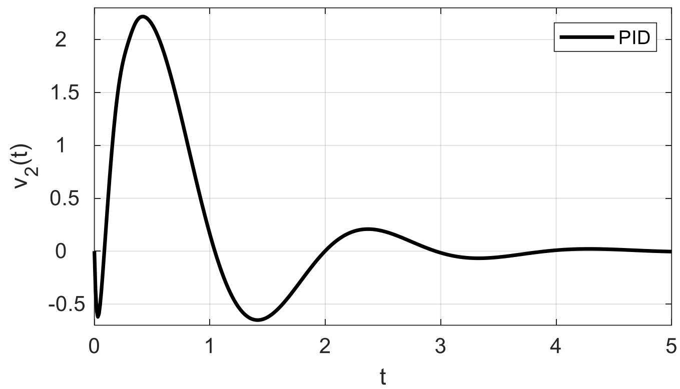


*Figure 5.40: angular velocity of second link of robotic manipulator using PID.*

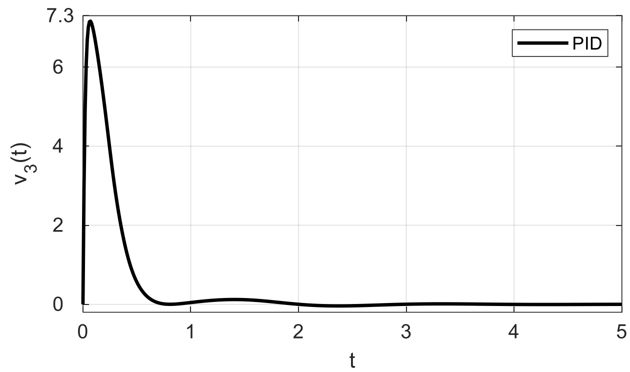


*Figure 5.41: angular velocity of third link of robotic manipulator using PID.*

**Angular position tracking errors using PID**

Angular position tracking error of each links using PIID are illustrated in figures 5.42, 5.43, and 5.44 as follows.

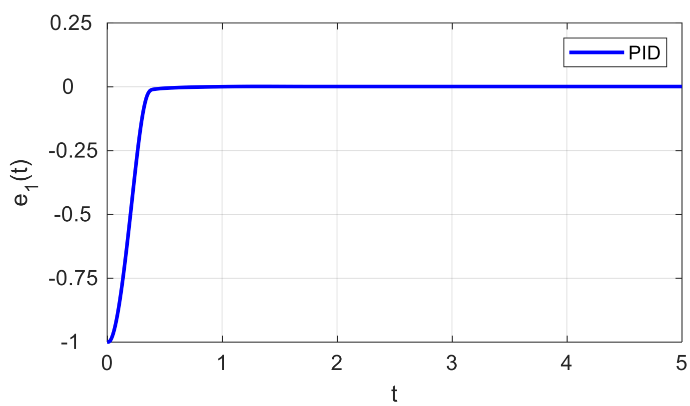


*Figure 5.42: position tracking error of first link of robotic manipulator using PID.*

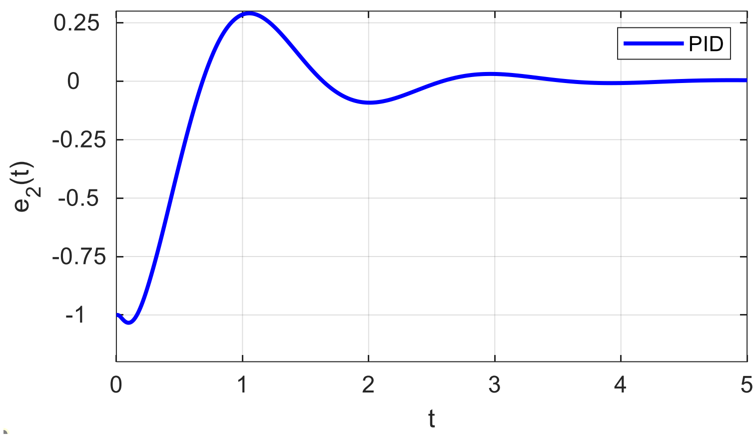


*Figure 5.43: position tracking error of second link of robotic manipulator using PID.*

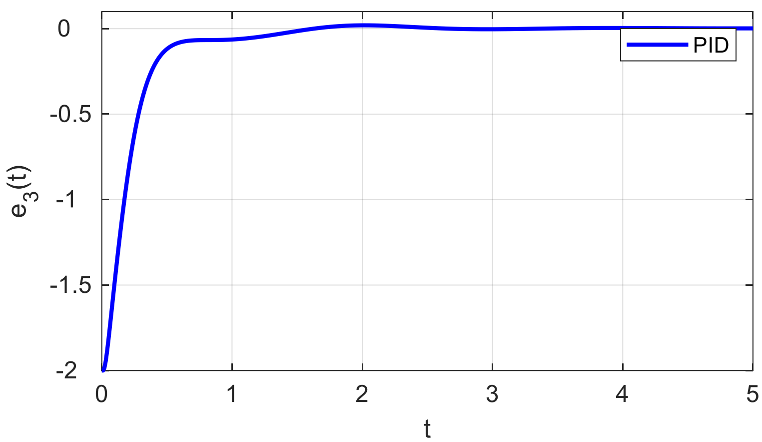


*Figure 5.44: position tracking error of third link of robotic manipulator using PID.*

### 5.2.4 Comparison of PID, SMC, STSMC for 3-DOF robotic manipulator

Certainly! The analysis of the articulation results for a 3-DOF articulated robotic arm was conducted using three different controllers: PID, SMC, and STSMC. The outcomes of this analysis were visually presented through graphical representations. Specifically, Figures 5.45, 5.46, and 5.47 illustrate a comparison of the performance of these three controllers in terms of position tracking for link 1, link 2, and link 3, respectively. These figures provide a visual understanding of how well each controller tracks the desired position for each respective link of the robotic arm.

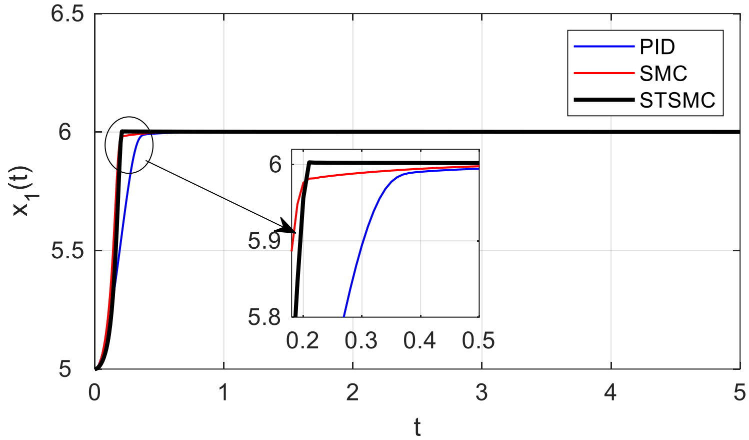


*Figure 5.45: position trajectory tracking of link1 using PID, SMC and STSMC.*

From figure 5.45 STSMC has smallest rise time, and settling time. All PID, SMC and STSMC have 0% undershoot, and SMC has smallest percentage of overshoot. STSMC has more overshoot than both PID and SMC. In general, the result obtained in STSMC has best performance based on time domain specification.

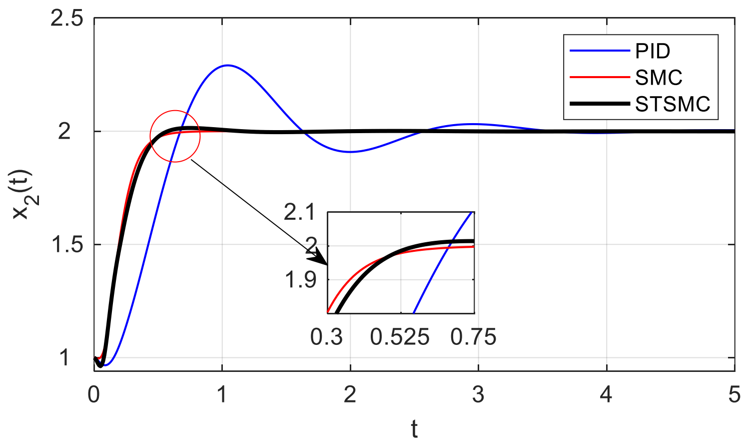


*Figure 5.46: position trajectory tracking of link2 using PID, SMC and STSMC.*

As figure 5.46 shows STSMC has best settling time, and PID has largest percentage of overshoot. SMC has smaller rise time and overshoot than STSMC and PID. Generally, it is seen that STSMC controller has best performance in quickly tracking desired set point with small overshoot and having behavior of critically damped oscillation.

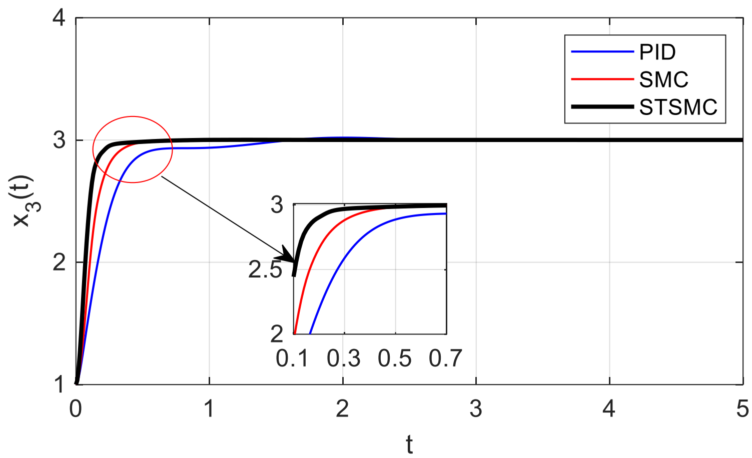


*Figure 5.47: position trajectory tracking of link3 using PID, SMC and STSMC.*

From figure 5.47 STSMC has smallest rise time and smallest settling time. All controllers have 0% of undershoot, and SMC has smallest overshoot. In general, based on time domain specification STSMC has the best performance.

The proposed STSMC controller has ability to track sinusoidal input and ramp function. Figure 5.48 illustrates sine wave tracking using proposed controller. From the result STSMC can handle constraint and minimizes energy for the minimal of objective function, and it has good performance of 3-DOF robotic manipulator. Figure 5.49 shows ramp function tracking of STSMC controller for robotic manipulator without considering disturbance. The simulation is done for 10sec. The red color and blue color are the desired trajectory and trajectory tracked by proposed controller respectively.

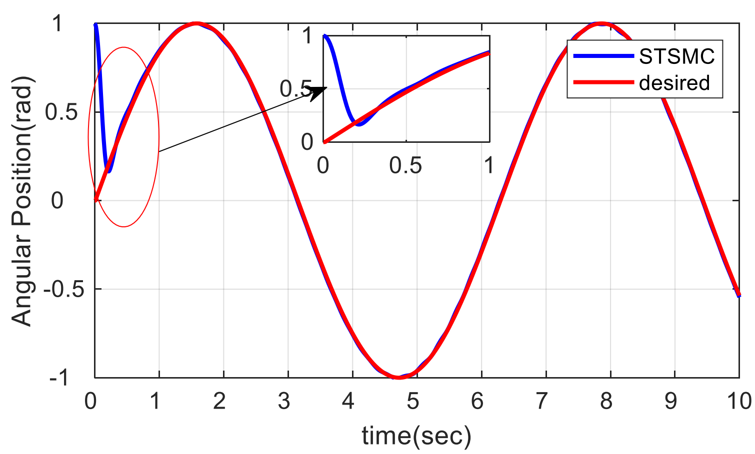


*Figure 5.48: Sine wave input tracking using STSMC for 3-DOF articulated robotic arm.*

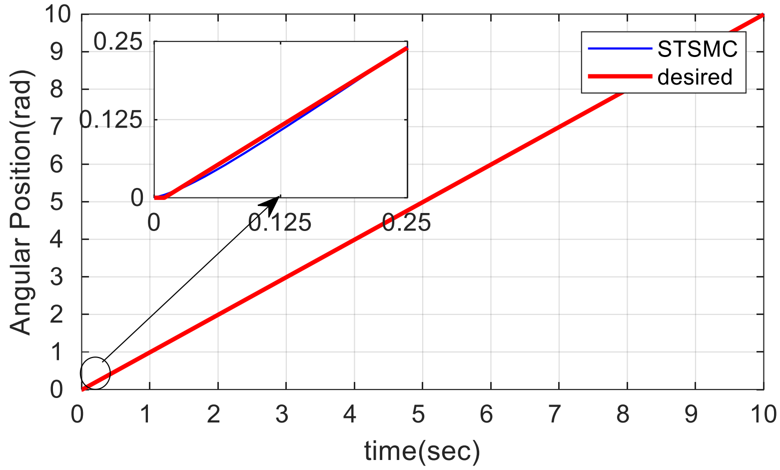


*Figure 5.49: ramp function input tracking using STSMC for 3-DOF articulated robotic arm.*

## 5.3 Disturbance rejection capability of the proposed controller

Given that industrial manipulators operate in hazardous environments, they may encounter external payload and vibration torque during their operation. Hence, this thesis considers the impact of these factors on the robotic manipulator. The testing of the proposed controller in this chapter is focused on its ability to reject the combined effects of payload variation and vibrational torque.

Figure 5.52 illustrates the disturbance torque resulting from the combined of two sources of disturbance uncertainty. Initially there is no payload variation in the system until 2 seconds, and then from 2secs to 2.5 secs payload varying torque is considered in the system model. In this range controller capability of disturbance rejection has seen. Furthermore, the amplitude of payload torque variation is considered to be 10Nm, and amplitude of white noise 0.1Nm is considered for 10 secs.

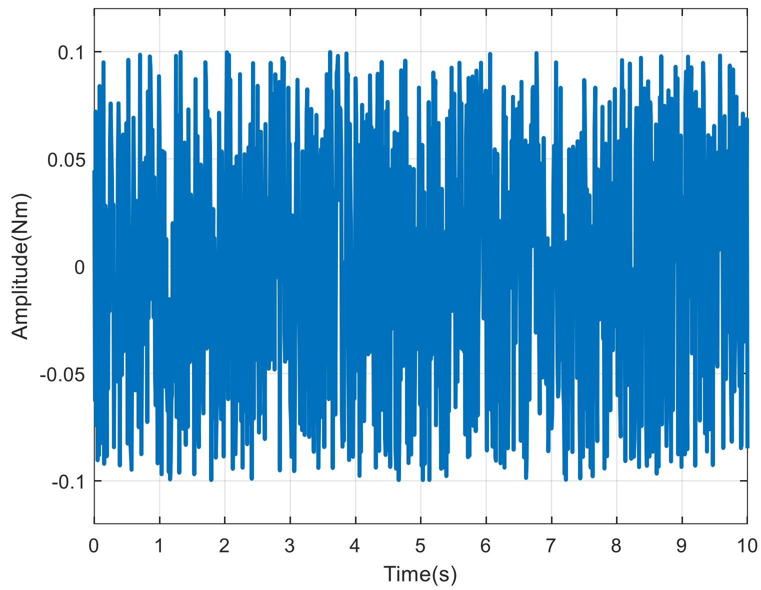


*Figure 5.50: disturbance torque due to vibration.*

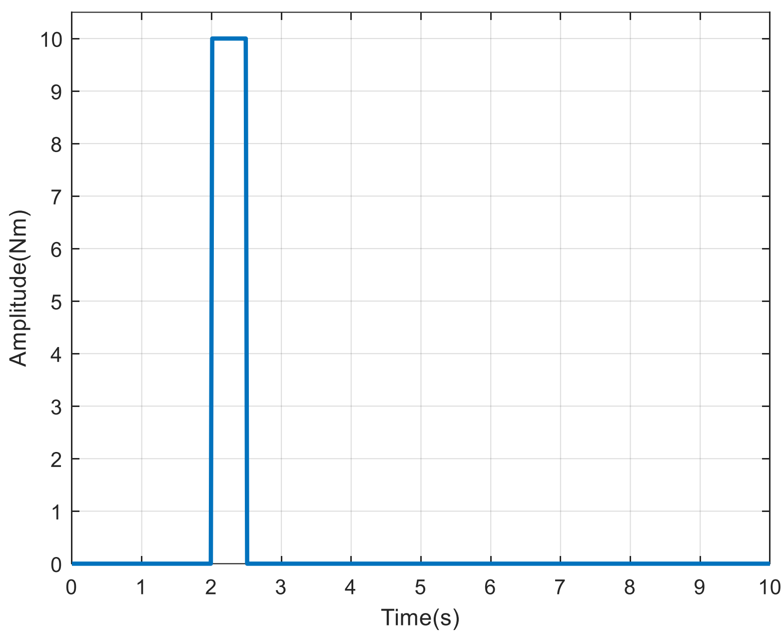


*Figure 5.51: disturbance torque due to payload variation.*

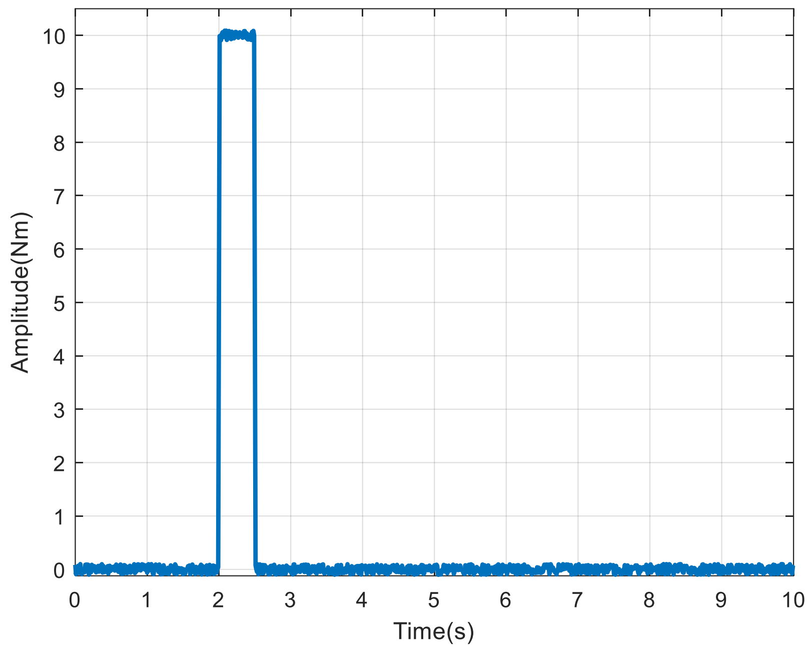


*Figure 5.52: disturbance torques due to combination of vibration and payload variation.*

Figure 5.53, 5.54 and 5.55 illustrates angular position of first link, second link and third link of 3-DOF articulated robotic manipulator respectively using STSMC under disturbance. Considering disturbance in the system increased undershoot of the links when it compared with simulation system without disturbance. In the figure, it can be observed that the proposed controller exhibits fast convergence towards the reference point with minimal undershoot and/or overshoot. Hence, the STSMC controller has good rejection capability and performance.

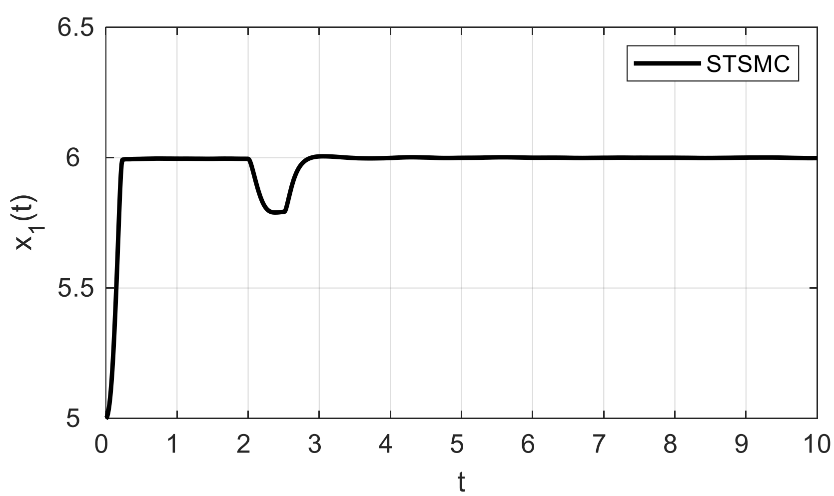


*Figure 5.53: angular position of link 1 under external disturbance using STSMC.*

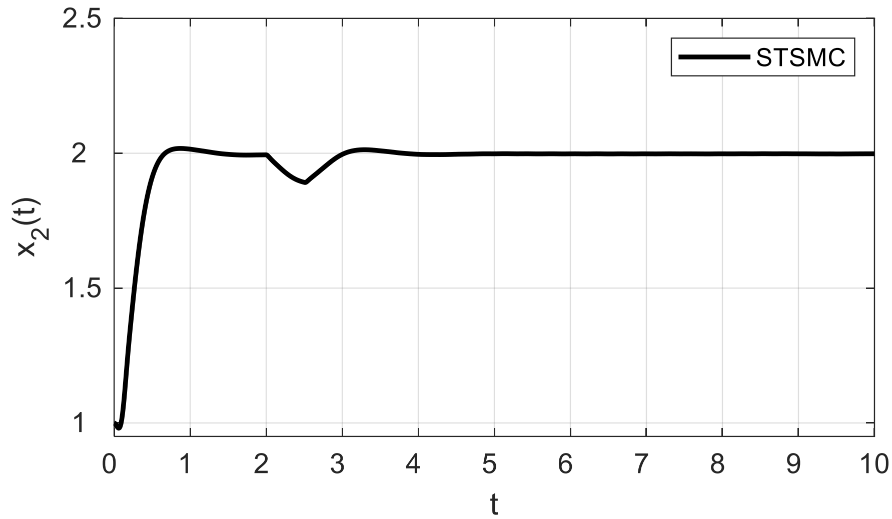


*Figure 5.54: angular position of link 2 under external disturbance using STSMC.*

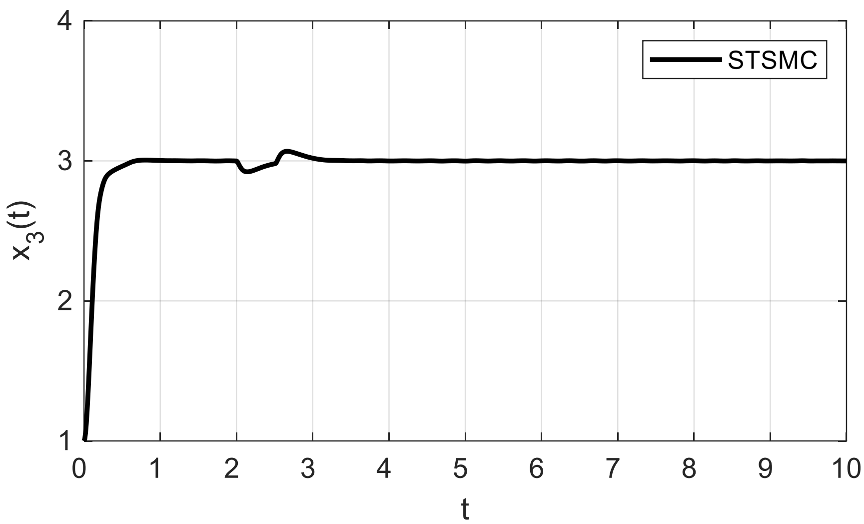


*Figure 5.55: angular position of link 3 under external disturbance using STSMC.*

Figures 5.56, 5.57 and 5.58 depict control input of link1, link2 and link3 respectively under disturbance using STSMC controller.

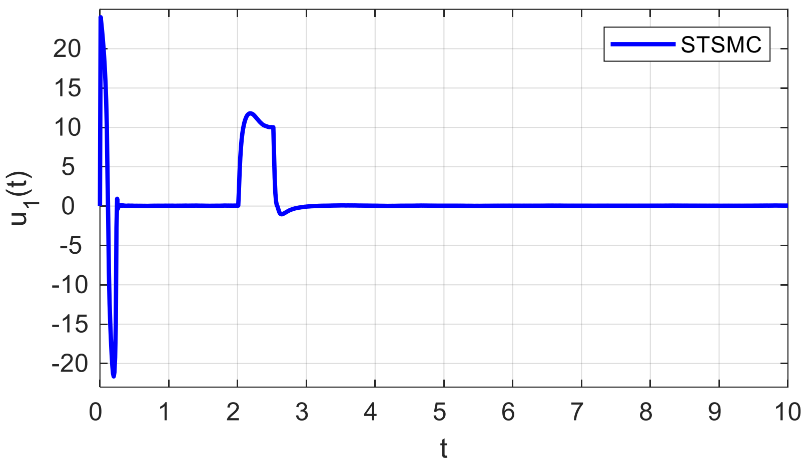


*Figure 5.56: control input of link 1 under external disturbance using STSMC.*

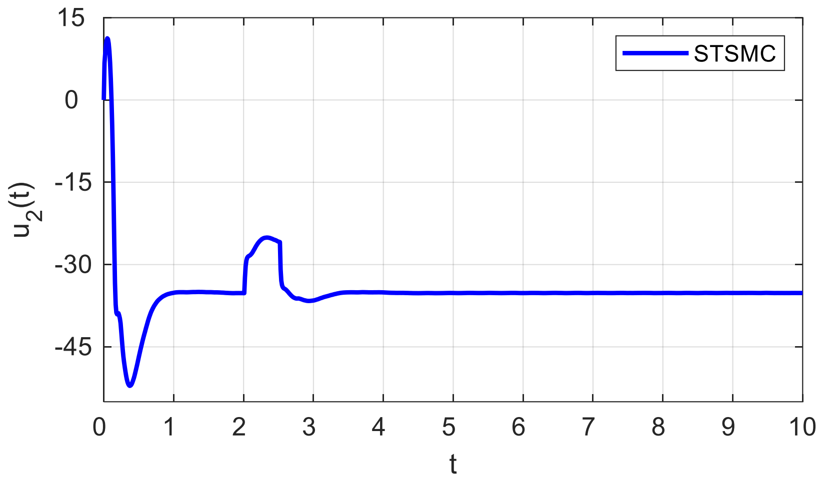


*Figure 5.57: control input of link 2 under external disturbance using STSMC.*

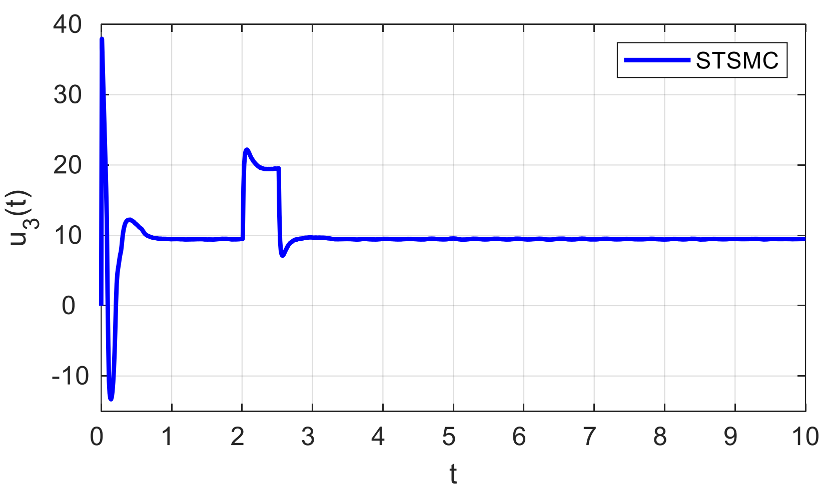


*Figure 5.58: control input of link 3 under external disturbance using STSMC.*

Figures 5.59, 5.60 and 5.61 illustrate articulation error of link1, link2 and link3 respectively under disturbance using STSMC controller.

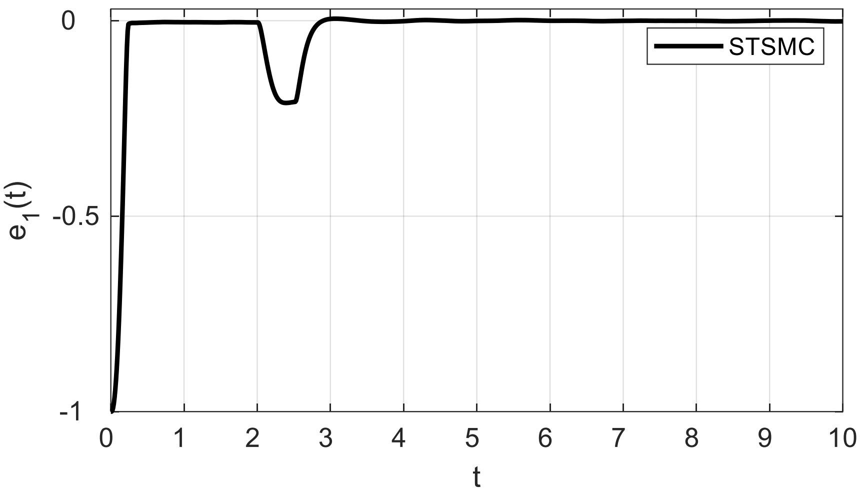


*Figure 5.59: articulation error of link 1 under external disturbance using STSMC.*

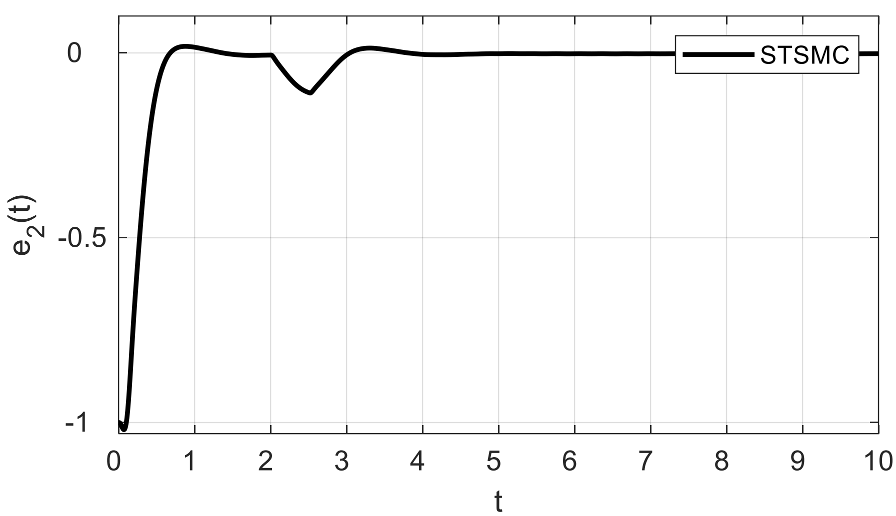


*Figure 5.60: articulation error of link 2 under external disturbance using STSMC.*

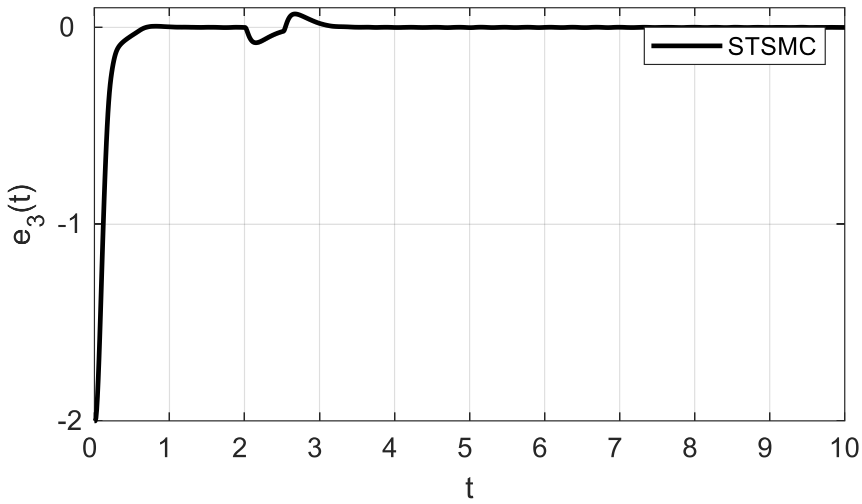


*Figure 5.61: articulation error of link 3 under external disturbance using STSMC.*

Proposed controller has ability to respond to set point change. Figures 5.62, 5.63 and 5.64 shows response of each joint angles to set point change, and figures 5.65, 5.66 and 5.67 depict control input of each joint angles to set point change. Angle of link1 is suddenly changed from 6 to 7 at 5 sec, angle of link2 is suddenly changed from 2 to 2.5 at 5 sec, and angle of link3 is suddenly changed from 3 to 3.5 at 5 sec,

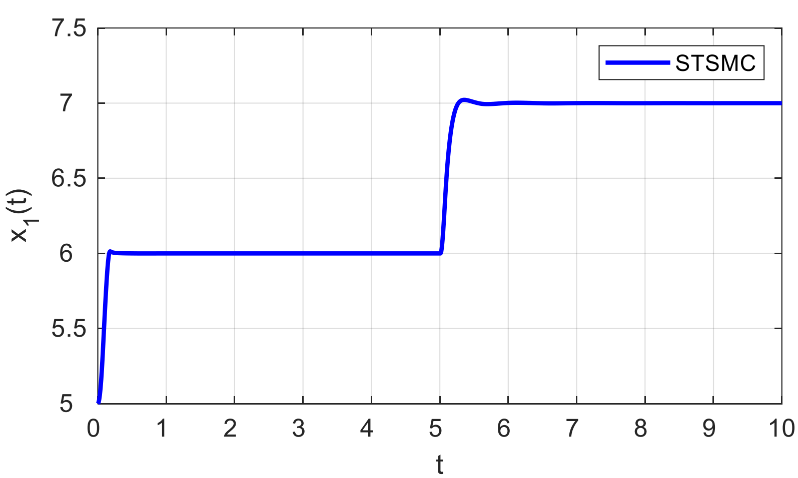


*Figure 5.62: position tracking of link1 to set point change using STSMC.*

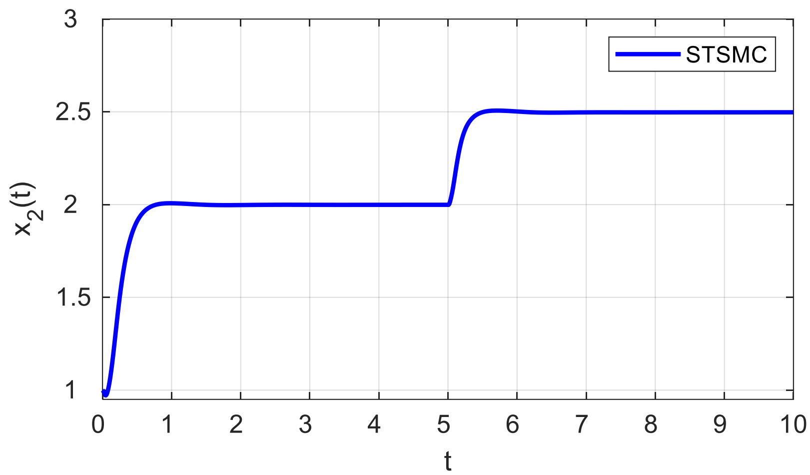


*Figure 5.63: position tracking of link2 to set point change using STSMC.*

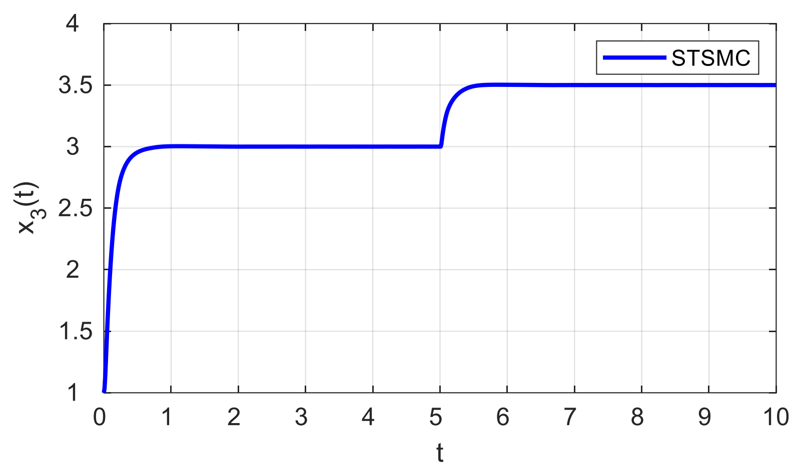


*Figure 5.64: position tracking of link3 to set point change using STSMC.*

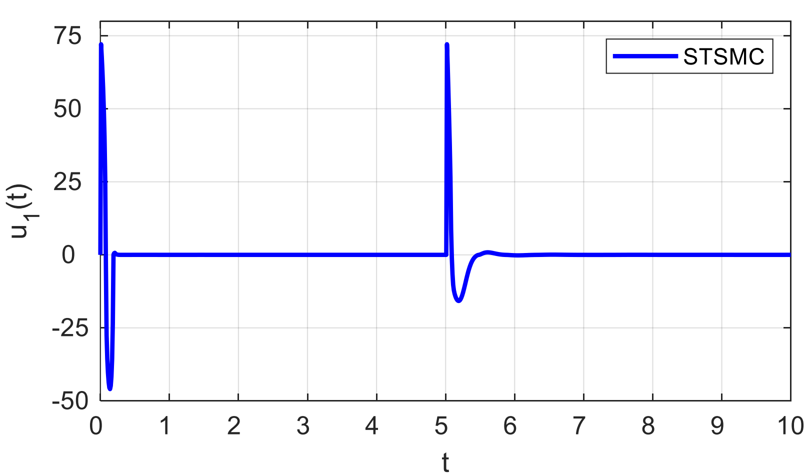


*Figure 5.65: control input of link 1 to set point change using STSMC.*

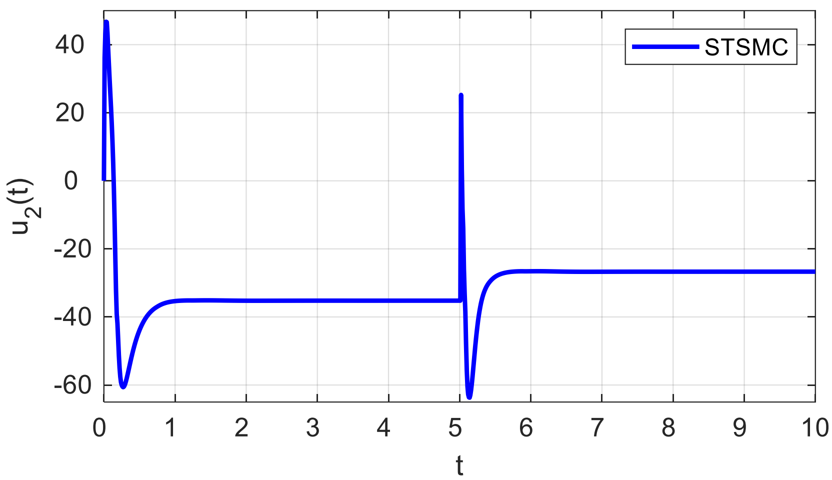


*Figure 5.66: control input of link 2 to set point change using STSMC.*

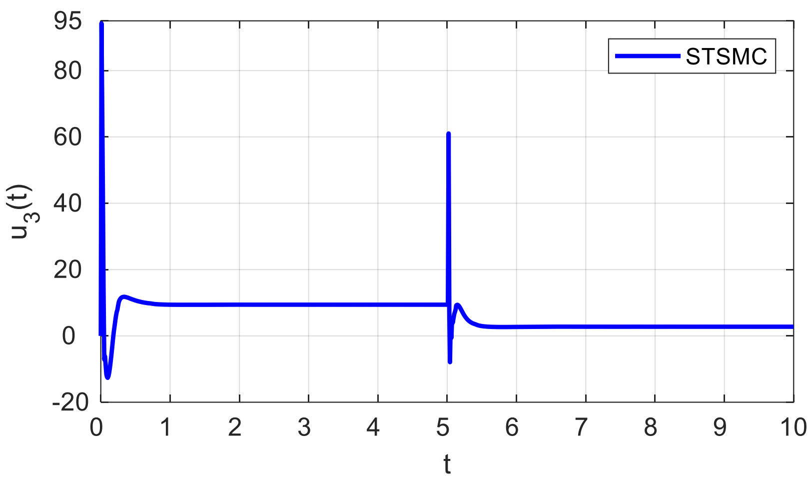


*Figure 5.67: control input of link 3 to set point change using STSMC.*

# CHAPTER SIX
# CONCLUSION AND RECOMMENDATIONS

## 6.1 Conclusions

This thesis introduced the utilization of STSMC for the purpose of tracking a 3-DOF robot arm manipulator. The thesis successfully accomplished the kinematics and dynamics modeling of a three-degree-of-freedom articulated robotic arm. The STSMC Control law is simulated in MATLAB/Simulink, and upon completion of their design, a comparison is made between the performance of the STSMC controller, and that of PID and SMC controllers. Among these controllers, it is determined that the STSMC controller performs the best. The proposed controller, STSMC, exhibits the capability to effectively handle external disturbances in the system, and also tracked the sinewave and ramp input trajectory. Furthermore, the stability of the designed controller is verified using Lyapunov Direct Method. The performance of the designed controller is evaluated against set point changes, demonstrating effective tracking performance.

To optimize system performance and reduce steady state error particle swarm optimization is applied for optimizing controller's parameters. Based on results comparison is performed in terms of minimize error, rapid convergence to set-point, overshoot and undershoot of plant dynamics. For 1st, 2nd and 3rd articulation the rise time of PID, SMC and STSMC are 0.2260s, 0.1101s, 0.1001s; 0.3891s, 0.2675s, 0.2967s; and 0.3602s, 0.2112s, 0.1228s respectively. In second link SMC has smaller rise time than STSMC, however in first link and in third link STSMC has smallest rise time. Settling time for 1st, 2nd and 3rd articulation using PID controller are 0.3543s, 3.2329s and 1.2746s respectively. For 1st, 2nd and 3rd articulation settling time using SMC controller are 0.2071s, 0.5316s, 0.4122s respectively. Using STSMC controller settling time for 1st, 2nd and 3rd articulation are 0.2051s, 0.5095s, 0.2683s respectively.

Based on the results, it can be concluded that the STSMC controller achieves the fastest convergence among the three controllers, considering error minimization, rapid convergence. In general, the proposed STSMC controller demonstrates the fastest convergence among the PID and SMC controllers, as indicated by the settling time. Additionally, STSMC achieves accurate tracking of the reference signal with minimal error and limited overshoot. The proposed Super Twisting Sliding Mode Control (STSMC) with fast convergence and minimal error is well-suited for tasks such as painting, welding, and pick-and-place operations. Its precise control enables accurate positioning and manipulation of objects in industries requiring efficient and reliable robotic arm control.

## 6.2 Recommendations

Here are some recommendation for the future work:

By implementing the control law proposed in this research paper or employing other robust control strategies, it is possible to effectively control robotic manipulators with a high degree of freedom (more than 3-DOF). In future undertakings, it is crucial to incorporate the dynamics of the actuators during real-time implementation. Additionally, it is recommended to validate the control approach through real-time hardware implementation, ensuring its feasibility and effectiveness in practical applications. By considering actuator dynamics and conducting real-time hardware validation, the research can be more applicable and reliable in real-world scenarios.

**Acknowledgments:** This research thesis was funded by Adama Science and Technology University under the grant number of ASTU/SM-R/718/23, Adama, Ethiopia.

# REFERENCES


Abbas, Z. (2018). *Motion Control of Robotic Arm Manipulator Using PID and Sliding Mode Technique*.

Ajwad, S. A., Islam, R. U., Azam, M. R., Ullah, M. I., & Iqbal, J. (2016). *Sliding Mode Control of Rigid-link Anthropomorphic Robotic Arm*.

Al-Dujaili, A. Q., Falah, A., Humaidi, A. J., Pereira, D. A., & Ibraheem, I. K. (2020). Optimal super-twisting sliding mode control design of robot manipulator: Design and comparison study. *International Journal of Advanced Robotic Systems*, *17*(6). https://doi.org/10.1177/1729881420981524

Andr´, A., & Bittencourt, A. C. (2007). *Friction change detection in industrial robot arms*.

Ashagrie, A., Salau, A. O., & Weldcherkos, T. (2021). Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller. *Cogent Engineering*, *8*(1). https://doi.org/10.1080/23311916.2021.1950105

Brunot, M., Janot, A., & Carrillo, F. (2017). State Space Estimation Method for the Identification of an Industrial Robot Arm. *IFAC-PapersOnLine*, *50*(1), 9815–9820. https://doi.org/10.1016/j.ifacol.2017.08.892

Chen, K. Y. (2018). Robust Optimal Adaptive Sliding Mode Control with the Disturbance Observer for a Manipulator Robot System. *International Journal of Control, Automation and Systems*, *16*(4), 1701–1715. https://doi.org/10.1007/s12555-017-0710-1

Corke, P. (2017). *Robotics, Vision and Control* (Vol. 118). Springer International Publishing. https://doi.org/10.1007/978-3-319-54413-7

Darajat, A. U., & Istiqphara, S. (2020). Control of Two-Link Robot Manipulator with Uncertainty Parameter using Self-Tuning Sliding Mode Control. *MECnIT 2020 - International Conference on Mechanical, Electronics, Computer, and Industrial Technology*, 376–380. https://doi.org/10.1109/MECnIT48290.2020.9166643

Dereje, G. (2018). *SLIDING MODE CONTROL OF A TWO DERGEE OF FREEDOM ROBOT ARM USING PERMANENT MAGNET SYNCHRONOUS MOTOR*.

Endris, M. H. (2022). *Model Predictive Control of 3-DOF Articulated Robotic Manipulator*.

Gambhire, S. J., Kishore, D. R., Londhe, P. S., & Pawar, S. N. (2021). Review of sliding mode based control techniques for control system applications. In *International Journal of Dynamics and Control* (Vol. 9, Issue 1, pp. 363–378). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/s40435-020-00638-7

Ghaleb, N. M., & Aly, A. A. (2018). Modeling and Control of 2-DOF Robot Arm. In *Online) International Journal of Emerging Engineering Research and Technology* (Vol. 6, Issue 11).

Ghazanfar Shahgholian, & Pegah Shafaghi. (2010). *2010 2nd International Conference on Electronic Computer Technology*.

Gu, E. Y. (2022). *Advanced Dynamics Modeling, Duality and Control of Robotic Systems*.

Kim, J., Jin, M., Park, S. H., Chung, S. Y., & Hwang, M. J. (2020). Task space trajectory planning for robot manipulators to follow 3-d curved contours. *Electronics (Switzerland)*, *9*(9), 1–20. https://doi.org/10.3390/electronics9091424

Kuo, T. C., Huang, Y. J., & Hong, B. W. (2011). *Design of Adaptive Sliding Mode Controller for Robotic Manipulators Tracking Control*.

Liu, S., Peng, G., & Gao, H. (2019). Dynamic modeling and terminal sliding mode control of a 3-DOF redundantly actuated parallel platform. *Mechatronics*, *60*, 26–33. https://doi.org/10.1016/j.mechatronics.2019.04.001

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

Lynch, K. (Kevin M. ), & Park, F. C. (2017). *Modern robotics : mechanics, planning, and control*.

Massaoudi, F., Elleuch, D., & Damak, T. (2019). Robust Control for a Two DOF Robot Manipulator. *Journal of Electrical and Computer Engineering*, *2019*. https://doi.org/10.1155/2019/3919864

Mohammad, A., & Ehsan, S. S. (2008). *Sliding Mode PID-Controller Design for Robot Manipulators by Using Fuzzy Tuning Approach*.

Mohammed, A. A., & Eltayeb, A. (2018). *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE.

Moldoveanu, F. (2014). SLIDING MODE CONTROLLER DESIGN FOR ROBOT MANIPULATORS. In *Bulletin of the Transilvania University of Braşov* • (Vol. 7, Issue 56).

Okubanjo, A., Oluwadamilola, O., Martins, O., & Olaluwoye, O. (2017). Modeling of 2-DOF…. *Futo Journal Series (FUTOJNLS)*, *3*(2), 80–92. www.futojnls.org

Ozer, H. O., Hacioglu, Y., & Yagiz, N. (2016). *High Order Sliding Modes for Vehicle Suspensions via Optimized Super Twisting Algorithm Control of Delay Systems View project*. https://www.researchgate.net/publication/305335702

Piltan, F., Bayat, R., Mehrara, S., & Rahmdel, S. (2012). Design New Control Methodology of Industrial Robot Manipulator: Sliding Mode Baseline Methodology Modern Control Techniques View project SYSTEM IDENTIFICATION View project Design New Control Methodology of Industrial Robot Manipulator: Sliding Mode Baseline Methodology. In *International Journal of Hybrid Information Technology* (Vol. 5, Issue 4). https://www.researchgate.net/publication/264122646

Raza, K., Khan, T. A., & Abbas, N. (2018). Kinematic analysis and geometrical improvement of an industrial robotic arm. *Journal of King Saud University - Engineering Sciences*, *30*(3), 218–223. https://doi.org/10.1016/j.jksues.2018.03.005

Saraf, P., & N, P. R. (2019). *Modeling and Simulation of a Point to Point Spherical Articulated Manipulator using Optimal Control*.

Truong, H. V. A., Tran, D. T., To, X. D., Ahn, K. K., & Jin, M. (2019). Adaptive fuzzy backstepping sliding mode control for a 3-DOF hydraulic manipulator with nonlinear disturbance observer for large payload variation. *Applied Sciences (Switzerland)*, *9*(16). https://doi.org/10.3390/app9163290

Van Bach, N. P., Hai, Q. D., & Trung, T. B. (2021). Optimization of trajectory tracking control of 3-DOF translational robot use PSO method based on inverse dynamics control for surgery application. *Journal of Vibroengineering*, *23*(7), 1591–1601. https://doi.org/10.21595/JVE.2021.21997

Van, M., Kang, H. J., & Shin, K. S. (2014). Novel quasi-continuous super-twisting high-order sliding mode controllers for output feedback tracking control of robot manipulators. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *228*(17), 3240–3257. https://doi.org/10.1177/0954406214526828

Vasquez, R. E., & Correa, J. C. (2007). *KINEMATICS, DYNAMICS AND CONTROL OF A PLANAR 3-DOF TENSEGRITY ROBOT MANIPULATOR*. http://www.asme.org/about-asme/terms-of-use

War Naing, W., Zar Aung, K., & Thike, A. (2018). Position Control of 3-DOF Articulated Robot Arm using PID Controller. In *International Journal of Science and Engineering Applications* (Vol. 7). www.ijsea.com

Zhang, H., Fang, H., Zhang, D., Luo, X., & Zou, Q. (2020). Adaptive Fuzzy Sliding Mode Control for a 3-DOF Parallel Manipulator with Parameters Uncertainties. *Complexity*, *2020*. https://doi.org/10.1155/2020/2565316

Zhao, D., Li, S., Zhu, Q., & Gao, F. (2010). Robust finite-time control approach for robotic manipulators. *IET Control Theory and Applications*, *4*(1), 1–15. https://doi.org/10.1049/iet-cta.2008.0014

Zheng, S., Niu, X., & Peng, C. (2019). ADAPTIVE SUPER-TWISTING-LIKE SLIDING MODE CONTROL with PRESCRIBED PERFORMANCE for ROBOT MANIPULATORS. *Journal of Mechanics in Medicine and Biology*, *19*(8). https://doi.org/10.1142/S0219519419400530

# APPENDIX

```
%Tune the gains of PID using PSO
rng default
options= optimoptions('particleswarm');
options.PlotFcn= 'pswplotbestf';
options.SwarmSize= 50;
options.MaxIterations= 20;
nvar= 3;
LB= [0.1 0.1 0.1];
UB= [20 30 20];
[x,fval]= particleswarm(@EvalObj,nvar,LB,UB,options)

function J= EvalObj(para)
  % PID gains
  kv1= para(1); kv2= para(1); kv3= para(1);
  kp1= para(2); kp2= para(2); kp3= para(2);
  ki1= para(3); ki2= para(3); ki3= para(3);
global Fv Dt MI m11 m12 m13 m21 m22 m23 m31 m32 m33 b11 b12 b13 b21 b22 b23 b31 b32
b33 g1 g2 g3;

  % constants
  m1= 5; m2= 5; m3= 5; l1= 0.3; l2= 0.5; l3= 0.2; g= 9.81;
  % damping constant for PMDM
  bv= 0.003; cv=0.04;

  t= 0; h= 0.01; z= [0;0;0]; ee= [0;0;0]; loop= 1000;
  x= [5;1;1;0;0;0];
  q= [6;2;3];
  e= [x(1)-q(1);x(2)-q(2);x(3)-q(3)];
  % diagonal matrices for PID gains
  ki= diag([ki1 ki2 ki3]);
  kv= diag([kv1 kv2 kv3]);
  kp= diag([kp1 kp2 kp3]);

  buf= [t x(1) x(2) x(3) 0 0 0 e(1) e(2) e(3)];

 for i= 1:loop
   s1= sin(x(1));
   c1= cos(x(1));
   s2= sin(x(2));
   c2= cos(x(2));
   s3= sin(x(3));
   c3= cos(x(3));
   c23= cos(x(2)+x(3));
   s23= sin(x(2)+x(3));
```

```
m21= 0;
m22= (m2+m3)*l2^2+2*m3*l2*l3*c3+m3*l3^2;
m23= m3*l3^2+m3*l2*l3*c3;
m31= 0;
m32= m3*l3^2+m3*l2*l3*c3;
m33= m3*l3^2;

    b11=-m3*l3^2*c23*s23*(x(5)+x(6))-(m2+m3)*l2^2*c2*s2*x(5)-
    m3*l2*l3*(s2*c23*x(5)+c2*s23*(x(5)+x(6)));
    b12=-m3*l3^2*s23*c23*x(4)-m3*l2*l3*s2*c23*x(4)-l2^2*s2*c2*x(4)-
    m3*l2*l3*c2*s23*x(4);
    b13= -m3*l3^2*s23*c23*x(4)-m3*l2*l3*c2*s23*x(4);
    b21=m3*l3^2*s23*c23*x(4)+m3*l2*l3*s2*c23*x(4)+(m2+m3)*l2^2*s2*c2*x(4)+m3*l
    2*l3*c2*s23*x(4);
    b22= -m3*l2*l3*s3*x(6);
    b23= -m3*l2*l3*s3*x(6)-m3*l2*l3*s3*x(5);
    b31= m3*l3^2*s23*c23*x(4)+m3*l2*l3*c2*s23*x(4);
    b32= m3*l2*l3*s3*x(5);
    b33= 0;


g1= 0;
g2= -m3*g*l3*s23-(m2+m3)*g*l2*s2;
g3= -m3*g*l3*s23;
%inertian matrix
M= [m11 m12 m13; m21 m22 m23; m31 m32 m33];
%coriolis and centripital
B= [b11 b12 b13; b21 b22 b23; b31 b32 b33];
%gravitational component
G= [g1; g2; g3];
MI= inv(M);
%frictional torque
Fv=  bv*[x(4);x(5);x(6)]+cv*[sgn(x(4));sgn(x(5));sgn(x(6)];
q= [6;2;3]; qd= [0;0;0]; qdd= [0;0;0];
% error
e= [x(1)-q(1);x(2)-q(2);x(3)-q(3)];
ed= [x(4)-qd(1);x(5)-qd(2);x(6)-qd(3)];
% integration of error
z= z+0.5*h*(e+ee);

ee= e;
u= M*qdd+B*qd+G-kv*ed-kp*e-ki*z;
      x= RK4(@fun,t,x,u,h);
```

```
        t=t+h;
        buf= [buf; t x(1) x(2) x(3) u(1) u(2) u(3) e(1) e(2) e(3)];
    end


    x1= buf(:,2); x2= buf(:,3); x3= buf(:,4);
    u1= buf(:,5); u2= buf(:,6); u3= buf(:,7);
    e1= buf(:,8); e2= buf(:,9); e3= buf(:,10);
    du1= u1-u1(end); du2= u2-u2(end); du3= u3-u3(end);

    w= 0.1;
    ae= abs(e1)+abs(e2)+abs(e3)+w*abs(du1)+w*abs(du2)+w*abs(du3);
    J= 0.5*h*(ae(1)+2*sum(ae(2:end-1))+ae(end));
end

function xdot= fun(t,x,u)
global Fv Dt MI  b11 b12 b13 b21 b22 b23 b31 b32 b33 g1 g2 g3;


xdot(1)= x(2);
xdot(2)= x(3);
xdot(3)= x(4);
xdot(4)= x(5);
xdot(6)= u(1)-(b11*x(4)+b12*x(5)+b13*x(6)+g1+Fv(1)+Dt(1))+u(2)-
(b21*x(4)+b22*x(5)+b23*x(6)+g2+Fv(2)+Dt(2))+ u(3)-
(b31*x(4)+b32*x(5)+b33*x(6)+g3+Fv(3)+Dt(3))
end

%Tuning Parameters of SMC using PSO

rng default
options= optimoptions('particleswarm');
options.PlotFcn= 'pswplotbestf';
options.SwarmSize= 50;
options.MaxIterations= 40;
nvar= 9;
LB= [0 0 0 0 0 0 0 0 0];
UB= [20 20 20 50 50 50 20 20 20];
[x,fval]= particleswarm(@EvalObj,nvar,LB,UB,options)

function J= EvalObj(para)
% SMC gains
mu1= para(1); mu2= para(2); mu3= para(3);
k1=  para(4); k2= para(5); k3= para(6);
C1= para(7); C2= para(8); C3= para(9);

global Fv Dt MI m11 m12 m13 m21 m22 m23 m31 m32 m33 b11 b12 b13 b21 b22 b23 b31 b32
b33 g1 g2 g3;
% constants
m1= 5; m2= 5; m3= 5; l1= 0.3; l2= 0.5; l3= 0.2; g= 9.81;
```

```
 % damping constant for PMDM
 bv= 0.003; cv=0.04;
 t= 0; h= 0.01; s= [0;0;0]; %ee= [0;0;0];


loop= 1000;
x= [5;1;1;0;0;0];
q= [6;2;3];
e= [x(1)-q(1);x(2)-q(2);x(3)-q(3)];

% diagonal matrices for SMC gains
mu = diag([mu1 mu2 mu3]);
k= diag([k1 k2 k3]);
c= [C1 ;C2; C3];

buf= [t x(1) x(2) x(3) 0 0 0 e(1) e(2) e(3)];

for i= 1:loop
    s1= sin(x(1));  c1= cos(x(1));
    s2= sin(x(2));  c2= cos(x(2));
    s3= sin(x(3));  c3= cos(x(3));
    c23= cos(x(2)+x(3));
    s23= sin(x(2)+x(3));

      m11= m3*l3^2*c23^2+2*m3*l2*l3*c2*c23+(m2+m3)*l2^2*c2^2;
      m12= 0;
      m13= 0;
      m21= 0;
      m22= (m2+m3)*l2^2+2*m3*l2*l3*c3+m3*l3^2;
      m23= m3*l3^2+m3*l2*l3*c3;
      m31= 0;
      m32= m3*l3^2+m3*l2*l3*c3;
      m33= m3*l3^2;

      b11= -m3*l3^2*c23*s23*(x(5)+x(6))-(m2+m3)*l2^2*c2*s2*x(5)-
  m3*l2*l3*(s2*c23*x(5)+c2*s23*(x(5)+x(6)));
      b12= -m3*l3^2*s23*c23*x(4)-m3*l2*l3*s2*c23*x(4)-(m2+m3)*l2^2*s2*c2*x(4)-
  m3*l2*l3*c2*s23*x(4);
      b13= -m3*l3^2*s23*c23*x(4)-m3*l2*l3*c2*s23*x(4);
      b21=
  m3*l3^2*s23*c23*x(4)+m3*l2*l3*s2*c23*x(4)+(m2+m3)*l2^2*s2*c2*x(4)+m3*l2*l3*c2*s23
  *x(4);

b22= -m3*l2*l3*s3*x(6);
b23= -m3*l2*l3*s3*x(6)-m3*l2*l3*s3*x(5);
b31= m3*l3^2*s23*c23*x(4)+m3*l2*l3*c2*s23*x(4);
b32= m3*l2*l3*s3*x(5);
b33= 0;

g1= 0;
g2= -m3*g*l3*s23-(m2+m3)*g*l2*s2;
```

```
g3= -m3*g*l3*s23;

%inertian matrix
M= [m11 m12 m13; m21 m22 m23; m31 m32 m33];
%coriolis and centripital
B= [b11 b12 b13; b21 b22 b23; b31 b32 b33];
%gravitational component
G= [g1; g2; g3];
MI= inv(M);
%frictional torque and disturbance
Fv= bv*[x(4);x(5);x(6)];
q= [6;2;3]; qd= [0;0;0]; qdd= [0;0;0];
% error
e= [x(1)-q(1);x(2)-q(2);x(3)-q(3)];
ed= [x(4)-qd(1);x(5)-qd(2);x(6)-qd(3)];
s= c.*e+ed;

u= M*qdd+B*qd+G-mu*tanh(s/0.9)-k*s;
x= RK4(@fun,t,x,u,h);

    t=t+h;
    buf= [buf; t x(1) x(2) x(3) u(1) u(2) u(3) e(1) e(2) e(3)];
 end
   x1= buf(:,2); x2= buf(:,3); x3= buf(:,4);
    u1= buf(:,5); u2= buf(:,6); u3= buf(:,7);
    e1= buf(:,8); e2= buf(:,9); e3= buf(:,10);
    du1= u1-u1(end); du2= u2-u2(end); du3= u3-u3(end);

    w= 0.1;
    ae= abs(e1)+abs(e2)+abs(e3)+w*abs(du1)+w*abs(du2)+w*abs(du3);
    J= 0.5*h*(ae(1)+2*sum(ae(2:end-1))+ae(end));
 end

 function xdot= fun(t,x,u)
 global Fv Dt MI  b11 b12 b13 b21 b22 b23 b31 b32 b33 g1 g2 g3;

xdot(1)= x(2);
xdot(2)= x(3);
xdot(3)= x(4);
xdot(4)= x(5);
xdot(6)=u(1)-(b11*x(4)+b12*x(5)+b13*x(6)+g1+Fv(1)+Dt(1))+u(2)-
(b21*x(4)+b22*x(5)+b23*x(6)+g2+Fv(2)+Dt(2))+ u(3)-
(b31*x(4)+b32*x(5)+b33*x(6)+g3+Fv(3)+Dt(3))
end

%Tuning parameters of STSMC using PSO
rng default

options= optimoptions('particleswarm');
options.PlotFcn= 'pswplotbestf';
```

```
options.SwarmSize= 50;
options.MaxIterations= 40;
nvar= 9;
LB= [0 0 0 0.1 0.1 0.1 0 0 0];
UB= [50 50 50 20 20 20 30 30 30];
[x,fval]= particleswarm(@EvalObj,nvar,LB,UB,options)

function J= EvalObj(para)
   % STSMC gains
   k11= para(1); k12= para(2); k13= para(3);
   k21= para(4); k22= para(5); k23= para(6);
   C1= para(7); C2= para(8); C3= para(9);

global Fv Dt MI m11 m12 m13 m21 m22 m23 m31 m32 m33 b11 b12 b13 b21 b22 b23 b31 b32
b33 g1 g2 g3;

 % constants
 m1= 5; m2= 5; m3= 5; l1= 0.3; l2= 0.5; l3= 0.2; g= 9.81;
 % damping constant for PMDM
 bv= 0.003; cv=0.04;
 t= 0; h= 0.01; s= [0;0;0]; %ee= [0;0;0];
 loop= 1000;
 x= [5;1;1;0;0;0];
 q= [6;2;3];
 e= [x(1)-q(1);x(2)-q(2);x(3)-q(3)];

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %These are values of sliding surface constants

 k1 = diag([k11 k12 k13]);
 k2= diag([k21 k22 k23]);
 c= [C1 ;C2; C3];
 w=[0;0;0];
 temp1=[0;0;0];

buf= [t x(1) x(2) x(3) 0 0 0 e(1) e(2) e(3)];

for i= 1:loop
   s1= sin(x(1));  c1= cos(x(1));
   s2= sin(x(2));  c2= cos(x(2));
   s3= sin(x(3));  c3= cos(x(3));
   c23= cos(x(2)+x(3));
   s23= sin(x(2)+x(3));

   m11= m3*l3^2*c23^2+2*m3*l2*l3*c2*c23+(m2+m3)*l2^2*c2^2;
```

```
    m12= 0;
    m13= 0;
    m21= 0;
    m22= (m2+m3)*l2^2+2*m3*l2*l3*c3+m3*l3^2;
    m23= m3*l3^2+m3*l2*l3*c3;
    m31= 0;
    m32= m3*l3^2+m3*l2*l3*c3;
    m33= m3*l3^2;

    b11= -m3*l3^2*c23*s23*(x(5)+x(6))-(m2+m3)*l2^2*c2*s2*x(5)-
  m3*l2*l3*(s2*c23*x(5)+c2*s23*(x(5)+x(6)));
    b12= -m3*l3^2*s23*c23*x(4)-m3*l2*l3*s2*c23*x(4)-(m2+m3)*l2^2*s2*c2*x(4)-
  m3*l2*l3*c2*s23*x(4);
    b13= -m3*l3^2*s23*c23*x(4)-m3*l2*l3*c2*s23*x(4);
    b21=
  m3*l3^2*s23*c23*x(4)+m3*l2*l3*s2*c23*x(4)+(m2+m3)*l2^2*s2*c2*x(4)+m3*l2*l3*c2*s23
  *x(4);

b22= -m3*l2*l3*s3*x(6);
b23= -m3*l2*l3*s3*x(6)-m3*l2*l3*s3*x(5);
b31= m3*l3^2*s23*c23*x(4)+m3*l2*l3*c2*s23*x(4);
b32= m3*l2*l3*s3*x(5);
b33= 0;

g1= 0;
g2= -m3*g*l3*s23-(m2+m3)*g*l2*s2;
g3= -m3*g*l3*s23;
%inertian matrix
M= [m11 m12 m13; m21 m22 m23; m31 m32 m33];
%coriolis and centripital
B= [b11 b12 b13; b21 b22 b23; b31 b32 b33];
%gravitational component
G= [g1; g2; g3];
MI= inv(M);
%frictional torque and disturbance
Fv=  bv*[x(4);x(5);x(6)]+cv*[sgn(x(4));sgn(x(5));sgn(x(6)];

 % q, dotq, and ddotq
 q= [6;2;3]; qd= [0;0;0]; qdd= [0;0;0];
% error
 e= [x(1)-q(1);x(2)-q(2);x(3)-q(3)];
 ed= [x(4)-qd(1);x(5)-qd(2);x(6)-qd(3)];
 s=c.*e+ed;
 temp=-k2*tanh(s/0.6);
 w=w+0.5*h*(temp+temp1);
 temp1=temp;
 u= M*qdd+B*qd+G-k1*sqrt(abs(s)).*tanh(s/0.6)+w;
 x= RK4(@fun,t,x,u,h);
```

```
    t=t+h;
    buf= [buf; t x(1) x(2) x(3) u(1) u(2) u(3) e(1) e(2) e(3)];
 end
 x1= buf(:,2); x2= buf(:,3); x3= buf(:,4);
    u1= buf(:,5); u2= buf(:,6); u3= buf(:,7);
    e1= buf(:,8); e2= buf(:,9); e3= buf(:,10);
    du1= u1-u1(end); du2= u2-u2(end); du3= u3-u3(end);

    w= 0.1;
    ae= abs(e1)+abs(e2)+abs(e3)+w*abs(du1)+w*abs(du2)+w*abs(du3);
    J= 0.5*h*(ae(1)+2*sum(ae(2:end-1))+ae(end));
 end

 function xdot= fun(t,x,u)
 global Fv Dt MI  b11 b12 b13 b21 b22 b23 b31 b32 b33 g1 g2 g3;

xdot(1)= x(2);
xdot(2)= x(3);
xdot(3)= x(4);
xdot(4)= x(5);
xdot(6)=u(1)-(b11*x(4)+b12*x(5)+b13*x(6)+g1+Fv(1)+Dt(1))+u(2)-
(b21*x(4)+b22*x(5)+b23*x(6)+g2+Fv(2)+Dt(2))+u(3)-
(b31*x(4)+b32*x(5)+b33*x(6)+g3+Fv(3)+Dt(3))
end
```