Performance Analysis of Adaptive Non-Singular Fast Terminal Sliding Mode Control for Motion Control of a 3-DOF Delta Parallel Robot

Samuel Chala Dibaba

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 Master of Science Degree in Electrical Power and Control Engineering

(Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

February, 2023

Adama, Ethiopia

Performance Analysis of Adaptive Non-Singular Fast Terminal Sliding Mode Control for Motion Control of a 3-DOF Delta Parallel Robot

Samuel Chala Dibaba

Advisor:
Prof. Sam sun Ma

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 Master of Science Degree in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies
Adama Science and Technology University

February, 2023

Adama, Ethiopia

DECLARATION

I hereby declare that this master thesis entitled ***"Performance Analysis of Adaptive Non-Singular Fast Terminal Sliding Mode Control for Motion Control of a 3-DOF Delta Parallel Robot"*** is my own 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.

Samuel Chala Dibaba	____________________	____________________
Name of student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled ***"Performance Analysis of Adaptive Non-Singular Fast Terminal Sliding Mode Control for Motion Control of a 3-DOF Delta Parallel Robot"*** prepared under my guidance by Samuel Chala Dibaba submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Electrical Power and Control Engineering, postgraduate in control engineering.

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

Prof. Samsun Ma_____	__________________	_________________
Advisor	Signature	Date

APPROVAL PAGE

I the advisors of the thesis entitled ***"Performance Analysis of Adaptive Non-Singular Fast Terminal Sliding Mode Control for Motion Control of a 3-DOF Delta Parallel Robot"*** and developed by Samuel Chala Dibaba, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Prof. Samsun Ma	________________	________________
Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Samuel Chala Dibaba have read and evaluated the thesis entitled ***"Performance Analysis of Adaptive Non Singular Fast Terminal Sliding Mode Control for Motion Control of a 3-DOF Delta Parallel Robot"*** and examined the candidate during open defence. 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
Dr Endalew Ayenew	________________	12/04/2023
Internal Examiner	Signature	Date
Dr Beteley Teka	________________	12-04-2023
External Examiner	Signature	Date

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

________________	________________	________________
Department Head	Signature	Date
________________	________________	________________
School Dean	Signature	Date
________________	________________	________________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

First and forever, all praise and thanks for **God**, who gave me the strength and patience to carry out this work in this good manner.

I owe a debt of gratitude to **Prof. Sam sun Ma**. He has been an excellent advisor, providing sound technical guidance and offering an outstanding example of professional conduct. I am truly lucky to have him as an advisor.

The guidance of **Prof. Kim Seungnam** has also been proven to be valuable during the course of this research. Numerous impromptu discussions with the people associated with him have been useful and have enriched my postgraduate school experience.

Thanks, should also be extended to the Department of Electrical Power and Control Engineering (EPCE), particularly to the Head of the department, **Mr. Mesfin Merga (MSc)** for his support and follow up of my thesis.

Finally, I especially want to thank my family and friends for their love, encouragement, and all sacrifices that they have made on my behalf.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

3D	Three Dimensional
3-DOF	Three Degree of Freedom
ANFTSMC	Adaptive Non-Singular Fast Terminal Sliding Mode Control
CAD	Computer Aided Design
CSMC	Conventional Sliding Mode Control
DOF	Degree of Freedom
EMF	Electromotive Force
EPFL	École Polytechnique Fédérale de Lausanne
FTSM	Fast Terminal Sliding Mode Control
IP	Inverse Position
LSM	Linear Sliding-Mode
NFTSM	Non-Singular Fast Terminal Sliding Mode Control
NTSM	Non-Singular Terminal Sliding Mode Control
PD	Proportional Derivative
PZT	Piezoelectric
RNFTSMC	Robust Non-Singular Fast Terminal Sliding Mode Control
RSS	Revolute-Spherical-Spherical
SMC	Sliding Mode Control
TSM	Terminal Sliding Mode
TSMC	Terminal Sliding Mode Control

ABSTRACT

A delta Robot is a type of parallel mechanism that consists of three arms connected in a common base. Parallel robots have more accuracy because they do not have cumulative joint errors. Further advantages of this robot are low inertia and high stiffness. Furthermore, the delta robot is composed of four main parts such as, base, arm, forearms and end-effector. The arms of the delta robot perform motions in a solitary End of Arm Tooling (EOAT), within a workspace that is dome shaped. This type of robot is well known in the industrial field for its ability to execute minute, precise motions. Delta robots can be called other names in the industrial field, such as spider robots or parallel robots.

The 3-D model of a 3-DOF Delta robot is designed in 3D-CAD (Solidworks). Hence, this modeling environment enables to introduce real but not approximated parameters of the robot and eased computation of large matrixes. Dynamical model with all the kinematic constraints for a robot will simply be found by exporting a 3D-CAD model of the robot to Simscape Multibody. A mathematical model of 3-DOF Delta robot has been carried out, based upon the application of the Newton-Euler equations of motion used in conjunction with the Jacobian matrix to map the inertial and gravitational loadings of the moving platform to the actuators.

This thesis presents an adaptive nonsingular fast terminal sliding mode control algorithm with a modified switch function for a 3-DOF delta robot manipulator with unknown modeling errors and external disturbances. The finite time convergence of the controller is analyzed using Lyapunov stability theory. The algorithm avoids the singularity problem of the traditional terminal sliding mode control and the slow convergence of the traditional TSM control and estimates the upper bound of system uncertainties. A modified switch function is used to achieve precise tracking and reduce chattering in control torque. Finally, the effectiveness of the control method is verified through simulation.

The performance of the controller is tested by tracking a circular trajectory centered at (0, 0, Z_0) on XY- plane with 10 in radius. The simulation result shows that the steady state tracking is reached in less than 2 seconds for a circular trajectory and X and Y rms error of 0.01 in.

Keywords: *3-DOF Delta Robot, 3D CAD (Solidworks) model of Delta Robot, MATLAB/Simulink, Adaptive non-singular fast terminal sliding mode control, finite time convergence, switch function*

CHAPTER 1

INTRODUCTION

1.1 Background of the study

The word robot was introduced Karel Capek a Czech playwright, in his play *"Rossum's Universal Robots".* The term robot is derived from the Slav word *robota* that means executive labour.

Robotic manipulators are mechanical structures that are used in place of humans to handle unsafe or repetitive tasks with high accuracy. They are becoming more common in a wide range of sectors and applications. The most common distinguishing characteristic of the robot is the mechanical structure, so they are classified as serial and parallel manipulators.

1.1.1 3-RSS Delta Parallel Manipulator

The Delta parallel manipulator is a spatial parallel manipulator with three translational DOFs. It consists of a fixed and a moving platform connected through three identical kinematic chains as illustrated in Figure 1.1. The fixed platform houses the actuation subsystem, whereas the moving platform serves as the end-effector. The three actuators are rigidly mounted on the base plate with 120° in between. Each kinematic chain consists of a proximal link and two distal links. The two distal links form a parallelogram that constrains the moving platform to three translational DOFs. Four passive spherical joints (S) form the corners of the parallelogram. An active (actuated) revolute joint (R) connects the proximal link to the fixed platform.

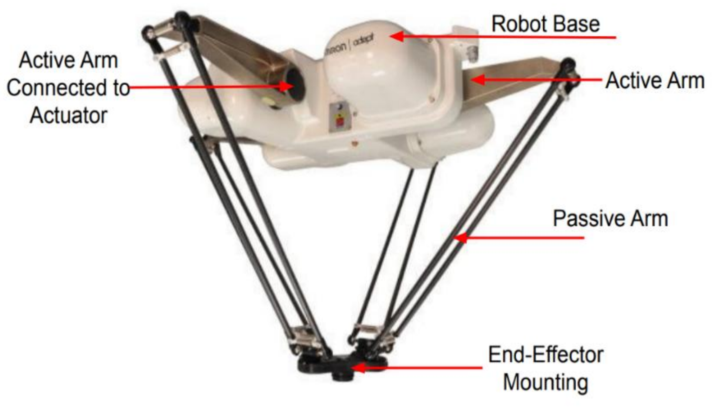


Figure 1. 1: A 3-RSS Parallel Delta Robot Components (*Delta Robot: Working Advantages and Applications - SMLease Design*, n.d.)

These types of robots can work with more than 100 parts per minute, picking, placing, and sorting them much faster than manual labor processes can perform. Their design allows them to perform repetitive tasks very efficiently at a highly consistent rate. A chief benefit of delta robots revolves around the fact that its relatively heavy motors are attached to the robot's frame, permitting the robot's moving parts to be very light by comparison. Figure 1.2 shows the structure of a Fanuc delta robot structure.

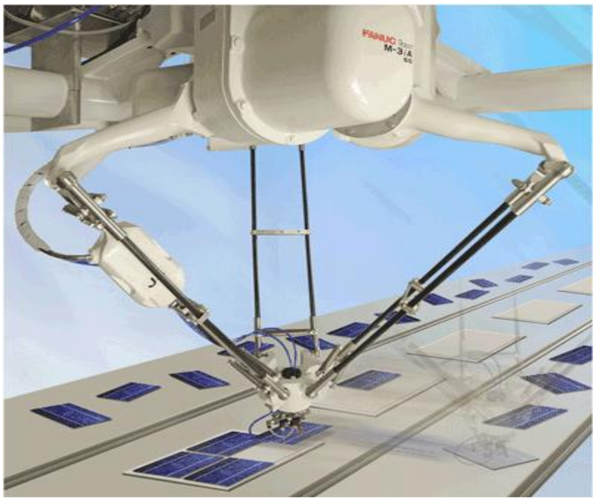


Figure 1. 2: A Delta Robot (Fanuc Delta Robot)(*FANUC Robotics FOOD M-3iA/6S 4-Axis Remote A-Cab R-30Ib - Flexible Automation Supply*, n.d.)

A delta robot is considered a closed kinematic chain manipulator and it possess a great number of advantages when comparing it to serial manipulators. For instance, higher rigidity and small mobile mass, that provides faster acceleration without losing accuracy. Also, delta robots can move heavier payload relative to its body mass. The main drawback of delta robots is their limited range of motion comparing it to a serial manipulator. The other disadvantages of this type of configuration include the fact that the dynamic model is quite complex in nature and there are many instances of singularities that must be mapped out and avoided in order to maintain control of the system.

Industries that take advantage of the high speed of delta robots are the packaging industry, medical and pharmaceutical industry. For its stiffness it is also used for surgery. Other applications include high precision assembly operations in a clean room for electronic components. The structure of a delta robot can also be used to create haptic controllers. More recently, the technology has been adapted to 3D printers. These printers are faster and can be built for about 200 dollars and compete well with traditional Cartesian printers. Figure 1.3 (a) shows the application of delta robot in industry.

(a) FANUC delta robot (b) ABB IRB 360 robot

Figure 1. 3: Commercial Delta Robots (*IRB 360 Series (ABB Robotics) M-1IA/0.5AL FANUC Europe Corporation 6... | Download Scientific Diagram*, n.d.)

Nonlinear dynamical systems suffer from the performance degradation caused by uncertainties and external disturbances. Many nonlinear control schemes have been proposed to improve the control performance of these perturbed systems such as feedback linearization control, back-stepping control, optimal control, intelligent control, and sliding mode control. It is well-known that the sliding-mode control is an important control procedure which has many attractive features such as good transient performance, robustness to parameter variations and insensitivity to disturbances. The basic idea of the sliding-mode control is to drive and maintain the system trajectory on a sliding surface designed a priori in the state space. In conventional linear sliding-mode (LSM) control, linear hyperplane is used as the sliding surfaces. However, LSM control ensures asymptotic convergence of the system states to the equilibrium point, but not in finite-time. To achieve the finite-time convergence and enhance the convergence properties of dynamical systems, terminal sliding-mode (TSM) control and fast terminal sliding-mode (FTSM) control which adopts nonlinear sliding surfaces is developed (X. Shi & Cheng, 2020). Compared to the TSM, the FTSM has faster convergence speed when the states are far from the origin.

However, these control methods suffer from the singularity problem because the terms with negative fractional powers may exist. To overcome this problem, a new type of sliding-mode control techniques called nonsingular terminal sliding-mode (NTSM) and nonsingular fast terminal sliding-mode (NFTSM) are recently developed. Compared to other sliding-mode manifolds, the NFTSM control has finite-time convergence, fast speed when the states are far from the origin, singularity avoidance and chattering reduction. Thanks to such promising features, NFTSM has become increasingly popular for high-precision control in different engineering sectors like space exploration, underwater navigation, machines with multiple motion axes, sophisticated industrial robotic applications such as welding, painting, and object grasping, etc.

Many researchers have investigated the tracking control problem for uncertain nonlinear systems using TSM control and NTSM control. A global NTSM control for rigid manipulators is developed to ensure the elimination of the singularity problem associated with the conventional TSM control. A continuous finite-time control scheme for rigid robotic manipulators is proposed using a NFTSM control. The robustness of the controller is established using the Lyapunov stability theory. Fast and high-precision tracking performance is obtained compared with the conventional TSM method. A fast nonsingular integral terminal sliding-mode control designed by introducing power integral terms which contain a boundary-like structure. The proposed controller can avoid the singularity problem without any constraint and provide faster responses. An NFTSM control which is able to avoid the possible singularity during the control phase is adopted in the robust high-precision control of uncertain nonlinear systems. An NTSM control for nonlinear systems is developed and it is shown that the proposed control strategy can eliminate the singularity while guaranteeing the finite-time reachability of the system. However, all of the above-mentioned work assumes the prior knowledge of upper bound of the system uncertainty.

Unfortunately, the upper bound may not be easily obtained in practical applications due to the complexity of the structure of uncertainties. Therefore, many kinds of controllers which combine the structure of SMC and adaptive control have been proposed to overcome the problem associated with the unknown bounds.

A robust adaptive sliding-mode controller is proposed for a class of uncertain nonlinear multi-input multi-output systems. The upper bounds of the uncertainties are not needed in the procedure for the controller design, and the controller is continuous, which guarantees that the tracking error can converge to a small residual set. A robust adaptive TSM control is proposed to estimate the upper bounds of the system uncertainty where only position and velocity measurements are available. It is also proved that the system uncertainty is related not only to the model properties, but also to the controller structure. By applying this adaptive controller, it is possible to estimate the upper bound of uncertainties and disturbances.

It is worth noting that all the above works mainly focused on the adaptive estimation of the upper bounds of the uncertainties using TSM control and FTSM control but neither of them have considered NFTSM control. It is well known that the main advantage of using NFTSM control is singularity avoidance, fast speed when the states are far from the origin, and strong robustness with respect to system uncertainty and external disturbances. To the best of our knowledge, not many works provide adaptive NFTSM control.

The novelty of the approach presented here can be understood by considering the following points:

1. Unlike the existing robust TSM approaches which are formulated under the assumption that the bound of the system uncertainty and disturbances are usually required to be known in advance, an adaptive parameter-tuning procedure is proposed here to estimate the unknown upper bounds. Therefore, the bound of the lumped uncertainty is unnecessary.
2. Compared with the existing adaptive TSM control which cannot guarantee singularity avoidance, the novel adaptive control law does not exhibit any singularity problem. Furthermore, the convergence rate of the system state is improved when the system state is far away from the equilibrium.
3. Unlike where the acceleration signal is needed for the control design, the proposed approaches need only position and velocity information, which can be easily obtained using appropriate sensors.
4. By employing the proposed controllers, the position and the velocity tracking errors can be stabilized to zero in a short finite-time simultaneously. Also, the chattering effect is eliminated without losing the robustness property and the precision.

1.2 Statement of the problem

Most African countries including Ethiopia had agriculture-based economy with little industrialization. As we know the manufacturing industries in our country are performed by human power for processes that involve repetitive tasks in processes involving pick and placing. This has adverse effect on the efficiency of our manufacturing production. The use of robotics will increase productivity and has the potential to bring more manufacturing production.

One of the newer robot types, the delta has found its place in high-speed applications across the manufacturing industry. These the study of delta robots will have the potential to bring more manufacturing production. Most of our industries can take advantage of the delta robots to achieve the highest production speeds available. A special feature of the delta robot is its housing. The motors are fully enclosed in the base of the robot. Such a construction makes it easy to achieve high IP ratings. This makes delta robots an excellent choice for wash down environments commonly seen in the food and beverage industry.

Traditional SMC can guarantee only asymptotical convergence of system states because of the asymptotical convergence of the linear sliding mode manifold that are generally selected. To solve this problem, the ANFTSMC with global finite-time stability was proposed by inducting nonlinear function in the traditional SMC. The ANFTSMC control can provide fast transient and high precision performance for 3-DOF Delta Robot. ANFTSMC can also resolve the singularity phenomenon near the equilibrium point. This proposal claims to achieve very effective control using the advantages of the ANFTSMC.

1.3 Research Questions

This work intends to answer the following basic research questions.

- ✓ How to model kinematic and dynamic model of 3-DOF Delta Robot.
- ✓ How to generate and track tracking trajectory for the robot.
- ✓ How to design Adaptive Non-Singular Fast Terminal Sliding Mode Control for best controlling and optimizing Delta Robot.
- ✓ How 3-DOF Delta Robot increases the efficiency of production in Ethiopia.
- ✓ How to model the entire system with MATLAB/Simulink

1.4 Significance of the study

My work in this area in general contributes to the growing field of Delta robot systems which gives the ability to have a better understanding of an actual field in the industry and what could be the challenges that will be faced. Specific significance of this thesis are mainly the design and simulation of difficult trajectory tracking controllers for 3-DOF Delta Robot.

Another significance is the development of a dynamical model of the Delta Robot in SolidWorks 3D-CAD which satisfies the kinematic constraints of the Robot. This model opens a new road for further research and to use other control systems and contribute the motion precision of the moving platform.

In the upcoming thesis the singularity will be eliminate, using a non-singular fast terminal sliding mode control. Moreover, adaptive tuning approach is adopted to compensate dynamic uncertainties and actuator faults during the operation of robotic manipulators, and the stability of the proposed control strategy is demonstrated by the sense of the finite-time stability theory.

The Nonsingular Fast Terminal Sliding Mode surface was previously used to control many nonlinear systems, for example, the control of an electrically actuated micro-resonator, the control of a mobile satcom antenna, to solve the problem of ground moving target tracking for fixed wing unmanned aerial vehicle, the trajectory tracking control of uncertain robot manipulators with actuator faults. However, none of the above works have developed adaptive technique combined with NFTSM surface to cope with the unknown upper bound of uncertainties and external disturbances which cause the degradation of the performance of the proposed controller in actual applications. Consequently, the contribution of this thesis is improved using 3-DOF Delta Robot.

1.5 General and specific objectives

1.5.1 Main Objective

The main objective of the thesis will be to Design a control system for trajectory tracking of Delta Robot using an ANFTSMC.

1.5.2 Specific Objectives

- To study kinematics (Both forward and inverse kinematic) and dynamics which are extremely important to employ its advantage.
- To design Adaptive Non-Singular Fast Terminal Sliding Mode Control (ANTSMC) based trajectory tracking controller for a 3-DOF Delta Robot.
- To simulate the developed Adaptive Non-Singular fast Terminal Sliding Mode Control (ANTSMC) based controller using MATLAB SIMULINK.
- To analyze whether the actual circular trajectory could satisfactorily track the desired circular trajectory.
- To compare the simulation results of the SMC and Adaptive Non-Singular Fast Terminal Sliding Mode Controller.

1.6 Limitation of the Study

The thesis is limited to the simulation of the system using MATLAB/Simulink by showing how the different components of the system interact with each other. This is because the implementation of a real system is difficult to procure the components. Along with it takes the additional time and cost incurred in making the actual prototype implementation.

1.7 Outline of the Thesis

In Chapter 2, a description and literature review of parallel manipulators is presented where the manipulator structure and mobility are discussed, and a special case of the manipulator (3-DOF delta robot) is also described.

In chapter 3, the materials and methods used are discussed. Key kinematic relationships of the robot are developed. First the inverse kinematic problem is addressed. Next, the forward kinematic problem is considered.

The kinematic analysis is continued with the development of the Jacobian matrix that provides a transformation from the velocity of the moving platform in Cartesian space to the actuated joint velocities in joint space. The Jacobian is then used to search for conditions that result in singular manipulator configurations where the mobility of the manipulator instantaneously changes. Model for the dynamics of the manipulator is also presented in Chapter 3.

In Chapter 4, Sliding mode control and Adaptive nonsingular fast terminal sliding mode control algorithms with a modified switch function for a 3-DOF delta robot manipulator with unknown modeling errors and external disturbances are developed. The finite time convergence of the controller using Lyapunov stability theory was analyzed.

In Chapter 5, CAD designing of 3-DOF Delta Robot using SolidWorks 2015 as 3D-CAD modeling program to effectively meet the requirement of a 3-DOF Delta Robot system kinematics and dynamics and its integration with Simulink MATLAB tool, SimMechanics. In this chapter the results of simulation are presented.

In Chapter 6 Finally, a conclusion and suggestions for future research are provided.

CHAPTER 2

LITERATURE REVIEW

This chapter involves several research areas, such as parallel manipulators, dynamics of manipulators and mechanisms, dynamics modeling Manipulators in Simmechanics and control of rigid links, etc. This section will review the research literature relative to these research topics.

2.1 Parallel Robot Manipulators

In recent years, Parallel manipulators have been studied extensively and several Parallel mechanisms have been developed by numerous researchers in response to different industrial applications. These Robotic mechanisms are built with a combination of spherical joints and revolution joints on linkages between the base platform and moving platform thus forming closed chain loops to the manipulator. The exploration of the parallel manipulator presented in this thesis is an effort to expand our understanding of the options available for the manipulation of our surroundings.

Parallel manipulators are robotic devices that differ from the more traditional serial robotic manipulators by virtue of their kinematic structure. Parallel manipulators are composed of multiple closed kinematic loops. Typically, these kinematic loops are formed by two or more kinematic chains that connect a moving platform to a base, where one joint in the chain is actuated and the other joints are passive. This kinematic structure allows parallel manipulators to be driven by actuators positioned on or near the base of the manipulator. In contrast, serial manipulators do not have closed kinematic loops and are usually actuated at each joint along the serial linkage. Accordingly, the actuators that are located at each joint along the serial linkage can account for a significant portion of the loading experienced by the manipulator. Whereas the links of a parallel manipulator generally need not carry the load of the actuators. This allows the parallel manipulator links to be made lighter than the links of an analogous serial manipulator. Hence, parallel manipulators can enjoy the potential benefits associated with light weight construction such as high-speed operation and improved load to weight ratios. Some parallel manipulators have also been shown to display outstanding rigidity as a result of the multiple closed kinematic loops (Tahmasebi & Tsai, 1995). The most significant drawback of parallel manipulators is that they have smaller workspaces than serial manipulators of similar size.

Probably the most famous parallel manipulator is the Stewart platform (see Figure 2.1). It was originally designed by (Stewart, 1965) as a flight simulator, and versions of it are still commonly used for that purpose. Since then, the Stewart platform has also been used for other applications such as milling machines (Aronson, 1996), pointing devices (Gosselin & Hamel, 1994), and an underground excavation device.

The Stewart platform has been studied extensively (Griffis & Duffy, 1989; Hunt, 1983; Innocenti & Parenti-Castelli, 1993; Nanua & Waldron, 1989). Generally, the Stewart platform has six limbs, where each limb is connected to both the base and the moving platform by spherical joints located at each end of the limb. Actuation of the platform is typically achieved by changing the lengths of the limbs. While these six-limbed manipulators offer

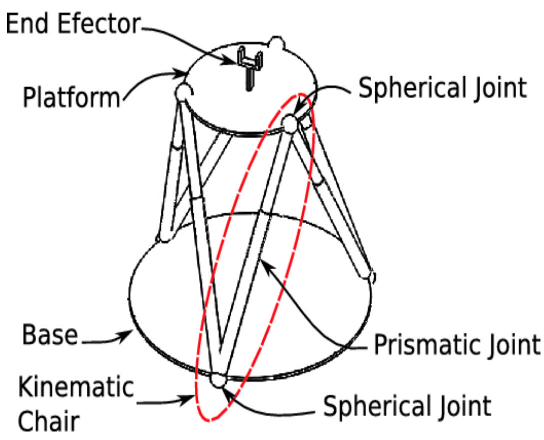


Figure 2. 1: A version of the Stewart platform (Campos & Moratelli, 2020).

good rigidity, simple inverse kinematics, and high payload capacity, they suffer the following disadvantages.

1. Their direct kinematics are difficult to solve.
2. Position and orientation of the moving platform are coupled.
3. Precise spherical joints are difficult to manufacture at low cost.

Moreover, the closed-form direct kinematic solutions that have been reported in the literature have generally required special forms of the Stewart Platform (Griffis & Duffy, 1989; Lin et al., 1994; Nanua & Waldron, 1989). Most commonly, these special forms require concentric spherical joints (Figure 2.2). In these special forms, pairs of spherical joints may present design and manufacturing problems.

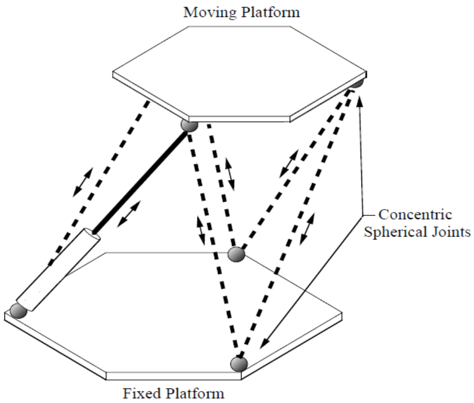
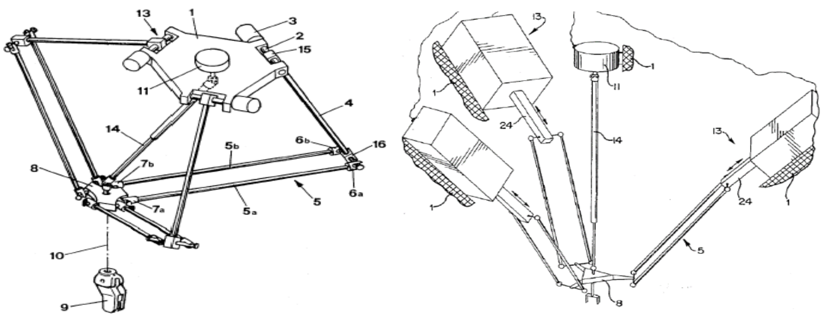


Figure 2. 2: A special form of the Stewart platform with concentric spherical joints (Mechanisms et al., 1956).

2.2 Prior Work on delta parallel Robot

A three degree of freedom parallel manipulator that does not suffer from the first two of the listed disadvantages was invented in the early 1980s by a research team led by (Vischer & Clavel, 1998) at the École Polytechnique Fédérale de Lausanne (EPFL, Switzerland). After a visit to a chocolate maker, a team member wanted to develop a robot to place pralines in their packages. The purpose of this new type of robot was to manipulate light and small objects at a very high speed, an industrial need at that time.

One of the most popular parallel manipulator designs is the DELTA platform, originally developed by Swiss team lead by Reymond Clavel. Driven by three revolute joints located on the base, motion is transmitted through three parallelogram arms to a semi-triangular end piece, called the delta plate. One of Clavel’s models, shown in Figure 2.3 below, has a fourth revolute joint that allows for rotation of the End Effector about the Z axis.



(a) (b)

Figure 2. 3: (a) A Diagram of the Delta Robot from the Original 1990 Patent (b) Clavel's Linear Delta Design (*Delta Robot Original Design [23] | Download Scientific Diagram*, n.d.)

The defining aspect of Delta manipulators are their parallelogram arms. Two long links are connected to the adjoining links by spherical joints at each end, forming a 4-bar mechanism. This setup ensures that the two connected links remain parallel, effectively removing one degree of freedom from the system. By using three 4-bar mechanisms, Delta platforms lock the pitch, roll, and yaw of the Delta plate, such that the end effector has a constant orientation regardless of position.

The Swiss company Demaurex purchased a license for the delta robot and started the production of delta robots for the packaging industry. (Clavel, 1991) presented his doctoral thesis "Conception dun robot parallel rapid 4 degree" and received the golden robot award for his work and development of the delta robot.

Also ABB Flexible Automation started selling its delta robot, the FlexPicker. Researchers from Harvard's Microrobotics miniaturized it with piezoelectric actuators to 0.43 grams for 15 mm x 15 mm x 20 mm, capable of moving a 1.3 g payload around a 7 cubic millimeter workspace with a 5 micrometers precision, reaching 0.45 m/s speeds with 215 m/s^2 accelerations and repeating patterns at 75 Hz.

2.3 Dynamic Modeling of Manipulators

Parallel Robots come in a wide variety of designs and applications ranging from the Stewart platform or Hexapod Parallel Robot which is used in aircraft motion simulators to the Delta Robot, which is used in packaging plants. This endows the fact that there cannot be a conclusive result as to which controller best suits the functionality of all parallel Robot. It is impossible to adequately design any controllers for the parallel Robot without a clear understanding of the dynamic model and the inverse kinematics of the mechanical model. With an understanding of the manipulator dynamics, it is possible to design a controller with better performance characteristics than would typically be found using heuristic methods after the manipulator has been constructed. Moreover, some control schemes such as the computed torque controller rely directly on the dynamics model to predict the desired actuator torque to be used in a feedforward manner.

Several approaches have been used to characterize the dynamics of parallel manipulators. The most common approaches are based upon Lagrange formulations (Miller & Clavel, 1992), the application of Hamilton's Principle (Miller, 1995), and the direct application of the Newton's equations of motion (Guglielmetti & Longchamp, 1994).

The Lagrangian multipliers (Griffiths, 1985; Van Khang, 2011) approach is selected to give a more complete characterization of the manipulator dynamics, with the most significant assumptions being that the mass of each long connecting rod of the upper arms is evenly divided and concentrated at the two joints on the rods and that the short connecting links are massless.

For this research, a simplified dynamics model and the direct application of Newton-Euler (Jalon & Bayo, 1994) equations of motion approach is developed with the approach disregarding friction at the passive joints. This approach assumes that the inertial and gravitational loading of the moving platform and a portion of the upper arm assembly can be mapped back to the actuated joint using the Jacobian matrix. The advantage of this approach is that it is less computationally intensive than the Lagrangian models, while still capturing the character of the manipulator dynamics. This is desirable if the dynamics model is to be used as the basis for a computed torque control scheme.

2.4 Dynamic Modeling of Manipulators using SimMechanics

The process of modeling accurate dynamics of manipulators or mechanisms with link flexibility using SimMechanics does not need to compute forward dynamics and the derivation of differential equations which is quite cumbersome, error prone and time consuming. Sim-Mechanics is a new way for robots modeling with its simple, intuitive and accurate characteristic. SimMechanics toolbox is a block diagram modeling environment for modeling and simulating mechanical systems which use the standard Newtonian dynamics of forces and torques.

The kinematical analyses based on SimMechanics are free from establishing the kinematic model of mechanism. Mechanical systems can be easily represented in a graphical way by connected block diagrams which save time and effort to model. The block set consists of block libraries for bodies, joints, sensors and actuators, constraints and drivers, and force elements. SimMechanics models can be interfaced seamlessly with ordinary Simulink block diagrams. This enables the user to design e.g., the mechanical and the control system in one common environment.

In recent years, many new types of rigid parallel manipulators have been studied and model by using Sim-Mechanics/Simulink. (WOLDU, 2010) analyzed 3-DOF micro-Stewart platform using the Sim-Mechanics, (Lan Wang etc., 2008) analyzed Assistant Robotic Leg using the Sim-Mechanics. Most of the researchers in SimMechanics to simulate their models of mechanism used to analyze dynamics or the workspace of a manipulator (Rahimi et al., 2022)(Patel & Bhatt, 2022; Rahimi et al., 2022).

Few researchers also used SimMechanics to design a controller combining the physical model of rigid manipulator (Makwana & Patolia, 2021; Lee, 2001). (Meng et al., 2021) introduced the method of modeling, simulation inverted pendulum combining Simulink and SimMechanics.

2.5 Related Works on Control System of manipulator Robots

The robot manipulator is a system along with nonlinearity, strong coupling, unknown parameters, and external disturbances, and hence many control algorithms have been developed to reduce the influences of these factors. The robot manipulator control system is also used as a typical system by researchers to test the effectiveness of their control algorithms.

Generally, adaptive control, robust control, and neural network control methods have been used to overcome the uncertainties in the robot control processes. However, some of them need long computational time, and others require the exact robot model to prove the system stability.

Sliding mode control (SMC) has long been recognized as a robust nonlinear control methodology because of its strong stability, and low sensitivity to model parameter variations and external disturbances (Zhao et al., 2022). SMC is designed such that the system trajectory is attracted by a predetermined manifold (s) in the phase plane. Once it reaches (s), the trajectory subsequently slides along it. One limitation of conventional SMC is that tracking errors do not converge to zero in a finite time. In this couple of decades, sliding mode control (SMC) method has become the focus of nonlinear systems with uncertainties. This control method has been applied to robot control systems by several researchers (Saidi & Touati, 2022).

In most applications, the robot must move rapidly from one position to another position or follow a desired trajectory in three dimensional spaces with high precision. In order to perform this task, recently, several control methods have been investigated such as computed torque with PD controller (Kelly et al., 2005), sliding mode controller (Slotine & Li, 1991), (Sciavicco & Siciliano, 2001). The computed torque controller is easy to implement, but it cannot meet the control quality due to uncertainties in the system model and disturbances. The sliding mode controller is robust and can improve control quality.

However, despite the popularity of conventional SMC, the chattering problem remains a major drawback, due to a discontinuous part which makes difficulty to control and reduces quality control discouraging its direct use. One solution is to use a continuous approximation of the switching element (known as the boundary layer) instead of the signum function. For example, (de Jesús Rubio, 2017), applies conventional SMC to a robot manipulator with dead zone nonlinearities, based on this boundary layer method. However, use of the boundary layer method has a critical problem: the system performance highly depends upon the boundary layer thickness.

Another drawback of the sliding mode controller is that it requires the bounds of uncertainties and disturbances being available. The sliding mode control with online learning neural networks has been applied to serial robotic manipulators, in which the system uncertainties and disturbances are estimated by a function approximation technique (Van Khang & Tuan, 2013).

To enhance the convergence performance, terminal sliding mode control (TSMC) has been introduced: See for example references by (S. I. Han & Lee, 2016; Kim et al., 2019). However, TSMC can suffer from singularity problems due to the negative exponent in the sliding manifold (Dinh et al., 2018; Wang et al., 2018; Yu & Zhihong, 2002). Recently, an article on a terminal sliding mode (TSM) controller has been developed by (Yu et al., 2020). It offers some superior properties such as fast finite time convergence and less steady state errors. However, present TSM controller design methods have a common drawback. There exist singularity problems in TSM control (El-Ghazaly et al., 2015). Existing methods to solve the problem in TSM controller design usually adopt an indirect way to avoid the singularity.

To improve the trajectory tracking accuracy with time-varying and nonlinear parameters, adaptive fuzzy sliding mode control (AFSMC) for a 3-DOF parallel manipulator has been developed (Zhang et al., 2020). Here fuzzy inference unit is utilized to modify the gain parameters in real-time by using the state feedback from the task space and the adaptive law is performed to update uncertainties in dynamic parameters. Hence, the control method is insensitive to uncertainties and disturbances and permits to decrease the requirement for the bound of these uncertainties, which validate the effectiveness of the developed control method and exhibit good trajectory tracking performance compared with sliding mode control (SMC) and fuzzy sliding model control (FSMC).

The low tracking accuracy of the manipulator in the presence of model parameter errors and external disturbance is a problem in high accuracy applications. Some researchers had published papers to address this problem. A paper on an adaptive fractional-order non-singular fast terminal sliding mode (AFONFTSM) controller based on a fixed-time disturbance observer (FTDO) was published (R. Shi & Zhang, 2022). The controller is split into three portions. First, a fractional-order non-singular fast terminal sliding mode (FONFTSM) surface is designed to improve the tracking accuracy and convergence speed of the error state. Second, to enhance the performance of the system state in the approaching mode, an adaptive variable exponential power reaching law (AVEPRL) is designed, which changes the coefficient of the exponential term through the size of the system state. Meanwhile, adaptive control is used to modulate one item of the reaching law, which improves the anti-interference of the system. The controller raised in this paper has better tracking precision, faster convergence speed, and stronger robustness. More recently, an article on a cross-coupling contour adaptive nonsingular terminal sliding mode control (CCCANTSMC) which combines adaptive nonlinear terminal sliding mode control (ANTSMC) of joint trajectory tracking and proportion–differentiation control of end-effector contour tracking by introducing the coupling factor between multiple axes based on Jacobian had been proposed (Dachang et al., 2023). The publisher has achieved accurate contour tracking of robotic manipulators with system uncertainties, external disturbance, and actuator faults. The nonsingular terminal sliding mode manifold eliminates singularity completely. The use of an adaptive tuning approach avoids the demand of the prior knowledge of system uncertainties, external disturbance, and actuator faults in practical applications.

An article on robust and adaptive nonsingular fast terminal sliding-mode (NFTSM) control schemes for the trajectory tracking problem are proposed with known or unknown upper bound of the system uncertainty and external disturbances. The developed controllers take the advantage of the NFTSM theory to ensure fast convergence rate, singularity avoidance, and robustness against uncertainties and external disturbances. First, a robust NFTSM controller is proposed which guarantees that sliding surface and equilibrium point can be reached in a short finite-time from any initial state. Then, to cope with the unknown upper bound of the system uncertainty which may be occurring in practical applications, a new adaptive NFTSM algorithm is developed. One feature of the proposed control law is their adaptation techniques where the prior knowledge of parameters uncertainty and disturbances is not needed. However, the adaptive tuning law can estimate the upper bound of these uncertainties using only position and velocity measurements. Moreover, the proposed controller eliminates the chattering effect without losing the robustness property and the precision.

A model-free based adaptive nonsingular fast terminal sliding mode control which comprises three parts: the intelligent PI controller, time-delay estimation, and adaptive sliding mode compensator for a 12 DOF lower limb multi-functional exoskeleton which was realized in Solidworks with the analysis of kinematics has been introduced (S. Han et al., 2018). The proposed control strategy drives tracking errors to a nonsingular fast terminal sliding surface and realizes fast convergence in finite time. The designed adaptive law approximates the upper bound of estimation error in time-delay estimation and thus it reduces the fundamental chattering on the switching manifold. The method has shown an improved tracking performance compared with the existing model free controller in co-simulations.

In this thesis Adaptive nonsingular fast terminal sliding mode control law with adaptive control to estimate the upper bound of uncertainties, and a modified switch function to reduce chattering was presented. An adaptive nonsingular fast terminal sliding mode manifold is proposed which can be used to avoid the singularity problem. The reaching time to the manifold from any initial states and the time to the equilibrium point in the sliding mode can be guaranteed to be finite time. Simulation results have demonstrated the effectiveness of the proposed algorithm. Control algorithms including sliding mode and adaptive non-singular fast terminal sliding mode (ANFTSM) based controllers are implemented for the delta spatial parallel robot. The simulation results show the outstanding features of the adaptive non-singular fast terminal sliding mode-based controller.

CHAPTER 3

MATERIALS AND METHOD

This part aims to describe the methodology to be used in the thesis. In this chapter, the list of materials used, designing of the robot and the modelling of kinematics and dynamics are discussed. Methodology is necessary to correctly understand of the flow of the thesis.

3.1 Description of Materials

The description of materials is a fundamental part in order to describe the methodology. Without the proper description, it is not possible to analyze the integrity of the project. In this research work, software such as MATLAB/2021a, SOLIDWORKS 2021, Microsoft office 2021, and MathType 6.0 equation are used. MATLAB is the computing software which consists of technical toolboxes and Simulink. Microsoft office is used for editing the thesis documentation. MathType 6.0 is used for writing mathematical formulas and equations in Microsoft offices word and PowerPoint presentation tools.

3.2 Method

The method used in this thesis to accomplish the required task is shown in Figure 3.1, depicts the steps of the methodology of work. The approach to trajectory tracking control of the 3-DOF delta robot is classified on the basis of tasks involved as, review of the closely related articles and literatures, derivation of the equations for the modelling and the development of the equivalent CAD model of 3-DOF delta robot using SOLIDWORKS 2021 software, derivation of the equations for the forward and inverse kinematics of the parallel robot, the derivations of control algorithms for the actuation of the delta robot to address the control problem, Programming and doing the Simulink model of the mathematical model and control equations on MATLAB R2021 software, performance analysis and evaluation of simulation.

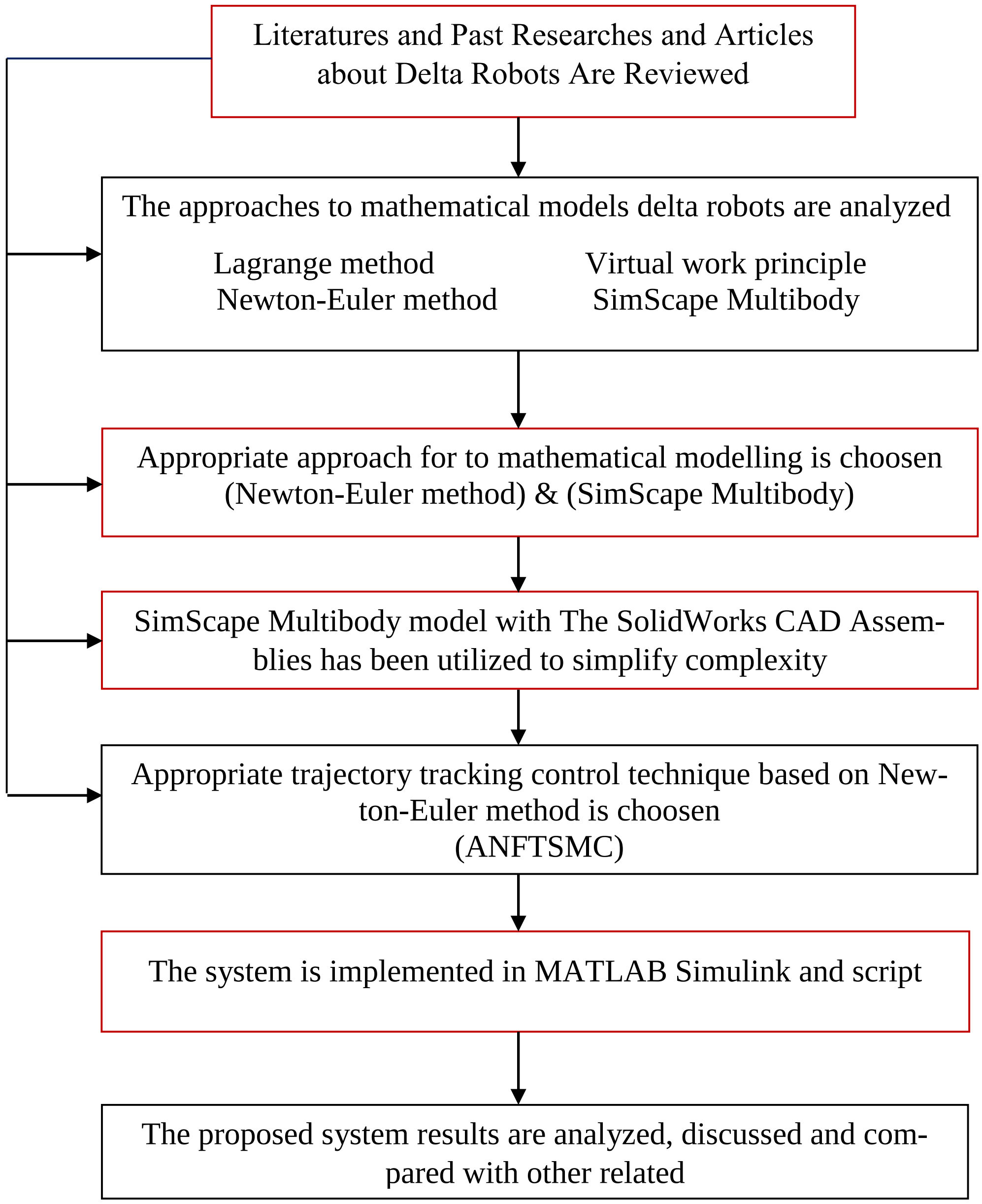


Figure 3. 1: Block diagram of the research method

3.2.1 Block diagram of the system

A block diagram is used to represent a control system in diagram form. In other words, the practical representation of a control system is its block diagram. Each element of the control system is represented with a block. The overall block diagram of a system of 3 DOF Delta Robot is shown in figure below.

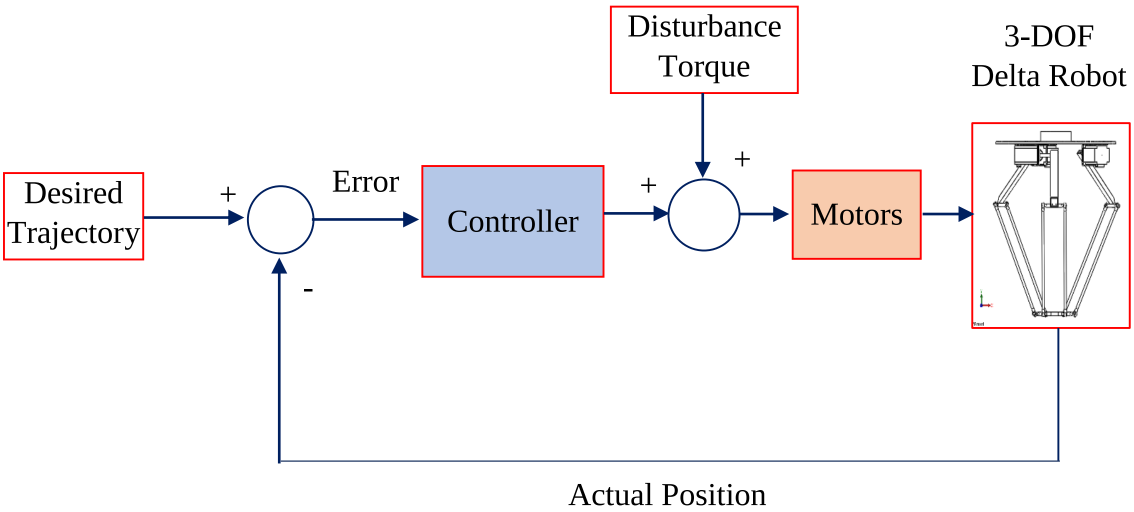


Figure 3. 2: Block diagram of the trajectory tracking control system

3.3 Kinematics of 3-DOF Delta Robot

Delta Robot kinematics deals with the study of the Delta Robot motion as constrained by the geometry of the links. The kinematic analysis is done without regard to the forces or torques that cause or result from the motion. Typically, the study of manipulator kinematics is divided into two parts, inverse kinematics and forward (or direct) kinematics. The inverse kinematics problem involves mapping a known position of the output link of the manipulator to a set of input joint variables that will achieve that position. The forward kinematic problem involves the mapping from a known set of input joint variables to a position of the moving platform that results from those given inputs.

Generally, as the number of closed kinematic loops in the manipulator increases, the difficulty of solving the forward kinematic relationships increases while the difficulty of solving the inverse kinematic relationships decreases.

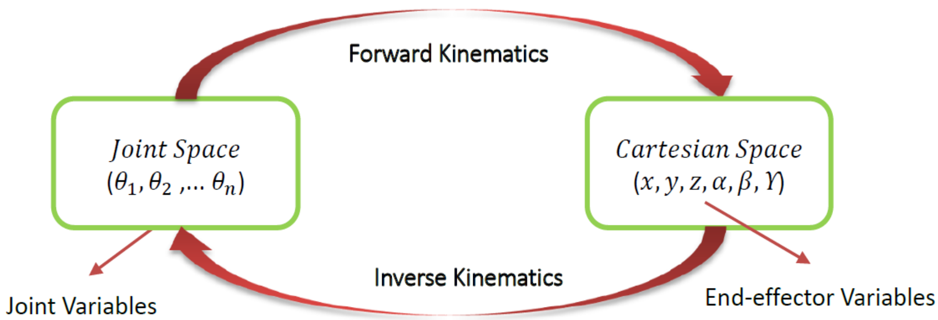


Figure 3. 3: Relations between Forward and Inverse Kinematics

In the following section, we explain the forward kinematics of Delta Robot via geometric constructions at first, then we will obtain the inverse kinematics solution.

3.3.1 Forward Kinematics of Delta Robot

As the forward kinematics is shortly described above, the three joint angles (θ_1, θ_2, θ_3) are inputs and the Cartesian coordinates (x, y, z) are the outputs of the forward kinematics. Joint angles are applied to servo motors and the corresponding Cartesian coordinates are calculated via forward kinematics equations.

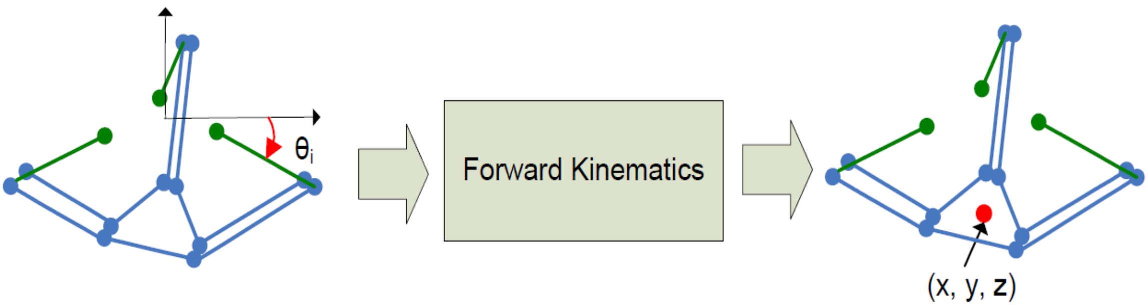


Figure 3. 4: Mapping from Joint Space to Cartesian Space

In general, the forward kinematics of parallel robots are difficult to compute but 3-DOF Delta Robot moves only translational. Therefore, a unique and correct analytical solution could be found in the solution set of the forward kinematics. To construct the forward kinematics of the Delta Robot, let's first designate the physical parameters of the robot below.

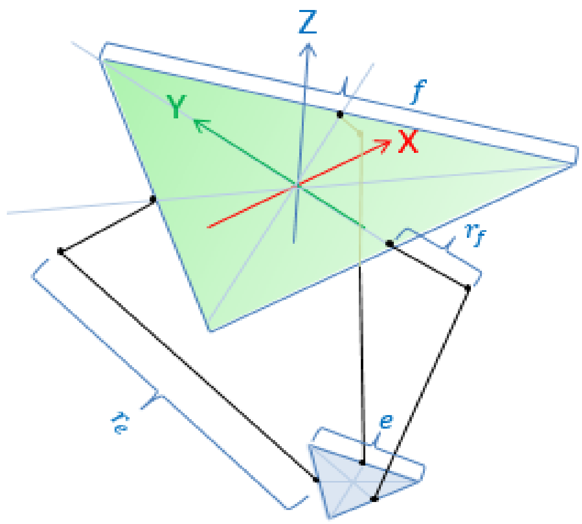


Figure 3. 5: Delta Robot Schematics

Table 3. 1: Physical Dimensions of Delta Robot

Physical Dimensions of Delta Robot (in)		
f	Side of the base plate triangle	27.30
e	Side of the travelling plate triangle	7.79
r_f	Length of the upper arm	14.50
r_e	Length of the lower arm	26.79

A reference will be chosen at the center of the base triangle so z-coordinate of the end-effector will always be negative. It is useful for choosing the unique and correct solution in the forward kinematics solution set.

To obtain the forward kinematics solution, first we need to find the coordinates of points J_1, J_2, J_3 from the joint angles $\theta_1, \theta_2, \theta_3$. While the lower arms are rotating around points J_1, J_2, J_3, we will obtain three spheres with Radius r_e. To intersect these spheres in one point, we move the centers of the spheres from points J_1, J_2, J_3 to points J_1', J_2', J_3'. The points and angles described above are shown in Figure 3.6.

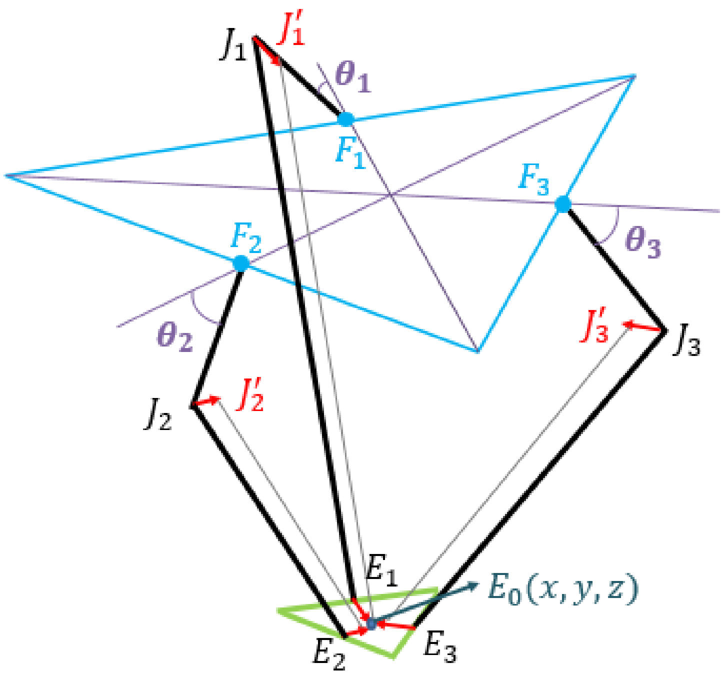


Figure 3. 6: Delta Robot Links, Vertices and Frame Representations

The intersection point of all three spheres is the point $E_0(x, y, z)$ and this intersection point is the Cartesian space solution of the forward kinematics.

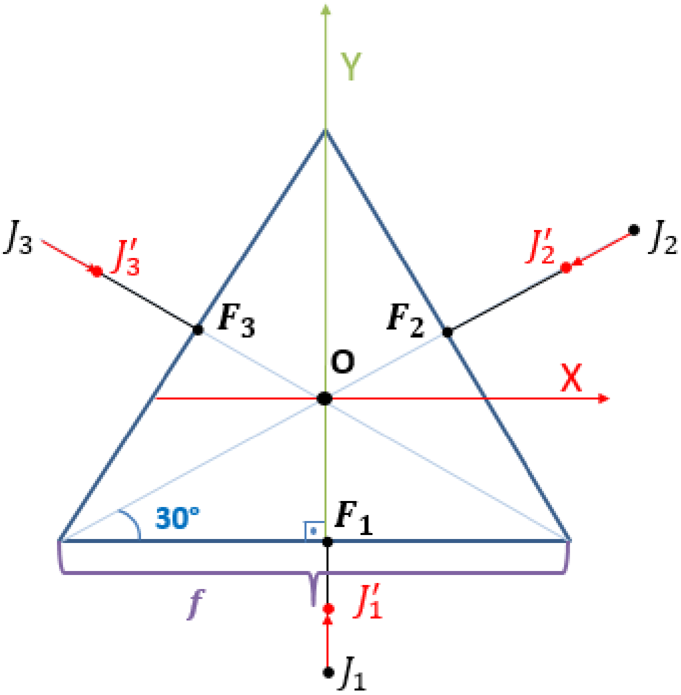


Figure 3. 7: The Triangle of Base Plate

To find the intersection point, the center and the radius of each sphere must be found. The triangle of the base plate is shown in Figure 3.7 and we define the length of the side of this triangle as f before. f is the known parameter so we can calculate the distances between points F_i and the center of the base triangle O.

$$|OF_1| = |OF_2| = |OF_3| = \frac{f}{2}\tan 30 \tag{3.1}$$

Similarly, the distances between points J_i and points $J_i^{'}$ are as follows:

$$|J_1J_1^{'}| = |J_2J_2^{'}| = |J_3J_3^{'}| = \frac{e}{2}\tan 30 \tag{3.2}$$

And the distances between points F_i and points Ji can be obtained as:

$$\begin{aligned} |F_1J_1| &= r_f \cos\theta_1 \\ |F_2J_2| &= r_f \cos\theta_2 \\ |F_3J_3| &= r_f \cos\theta_3 \end{aligned} \tag{3.3}$$

If we combine the Equations (3.1), (3.2) and (3.3), we obtain the centers of the all three spheres with radius r_e respectively below.

$$\left\{\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, r_1\right\} = \left\{\begin{bmatrix} 0 \\ -\left(\dfrac{f-e}{2}\tan 30\right) - r_f\cos\theta_1 \\ -r_f\sin\theta_1 \end{bmatrix}, r_e\right\}$$

$$\left\{\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, r_2\right\} = \left\{\begin{bmatrix} \left(\left(\dfrac{f-e}{2}\tan 30\right) + r_f\cos\theta_2\right)\cos 30 \\ \left(\left(\dfrac{f-e}{2}\tan 30\right) + r_f\cos\theta_2\right)\cos 30 \\ -r_f\sin\theta_2 \end{bmatrix}, r_e\right\}$$

$$\left\{\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}, r_3\right\} = \left\{\begin{bmatrix} \left(\left(\dfrac{f-e}{2}\tan 30\right) + r_f\cos\theta_3\right)\cos 30 \\ \left(\left(\dfrac{f-e}{2}\tan 30\right) + r_f\cos\theta_3\right)\cos 30 \\ -r_f\sin\theta_3 \end{bmatrix}, r_e\right\} \quad (3.4)$$

The equation of a sphere with radius r_e , centered at the below points are given in Cartesian coordinates by the following equations:

$$\begin{aligned} x^2 + (y-y_1)^2 + (z-z_1)^2 &= r_e^{\,2} \\ (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 &= r_e^{\,2} \\ (x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2 &= r_e^{\,2} \end{aligned} \quad (3.5)$$

Let's expand the above three equations by squaring all left side terms and they can be written as:

$$x^2 + y^2 + z^2 - 2y_1y - 2z_1z = r_e^{\,2} - y_1^{\,2} - z_1^{\,2} \quad (3.6)$$

$$x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = r_e^{\,2} - x_2^{\,2} - y_2^{\,2} - z_2^{\,2} \quad (3.7)$$

$$x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = r_e^{\,2} - x_3^{\,2} - y_3^{\,2} - z_3^{\,2} \quad (3.8)$$

To simplify the above equations, substitute $w_i = x_i^{\,2} + y_i^{\,2} + z_i^{\,2}$ into the Equation (3.6), (3.7) and (3.8) and we can rewrite the equations as below:

$$x^2 + y^2 + z^2 - 2y_1y - 2z_1z = r_e^{\,2} - w_1^{\,2} \quad (3.9)$$

$$x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = r_e^{\,2} - w_2^{\,2} \quad (3.10)$$

$$x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = r_e^{\,2} - w_3^{\,2} \quad (3.11)$$

Then, to eliminate $x^2 + y^2 + z^2$, subtract the Equation (3.10) and (3.11) from the first equation in (3.9). As a result, we have the new equation sets respectively:

$$x_2 x + (y_1 - y_2)y + (z_1 - z_2)z = \frac{w_1 - w_2}{2} \tag{3.12}$$

$$x_3 x + (y_1 - y_3)y + (z_1 - z_3)z = \frac{w_1 - w_3}{2} \tag{3.13}$$

Now, to equal the coefficients of x terms, Equation (3.12) multiplies by x_3 and Equation (3.13) multiplies by x_2:

$$x_2 x_3 x + x_3(y_1 - y_2)y + x_3(z_1 - z_3)z = x_3\left(\frac{w_1 - w_2}{2}\right) \tag{3.14}$$

$$x_2 x_3 x + x_3(y_1 - y_2)y + x_3(z_1 - z_3)z = x_3\left(\frac{w_1 - w_2}{2}\right) \tag{3.15}$$

To eliminate $x_2 x_3 x$ terms, subtract Equation (3.14) from the Equation (3.15). We obtain two equations:

$$x = \frac{a_1 z + b_1}{d} \tag{3.16}$$

$$y = \frac{a_2 z + b_2}{d} \tag{3.17}$$

And to simplify the equations, we can describe the new denominator parameter in (3.18):

$$d = x_3(y_2 - y_1) - x_2(y_3 - y_1) \tag{3.18}$$

Finally, we obtain a_1, b_1, a_2, b_2 parameters used in Equations (3.16) and (3.17) as below:

$$a_1 = (z_1 - z_2)(y_3 - y_1) - (z_3 - z_1)(y_2 - y_1) \tag{3.19}$$

$$b_1 = -0.5\left[(w_2 - w_1)(y_3 - y_1) - (w_3 - w_1)(y_2 - y_1)\right] \tag{3.20}$$

$$a_2 = (z_2 - z_1)x_3 - (z_3 - z_1)x_2 \tag{3.21}$$

$$b_2 = 0.5[(w_2 - w_1)x_3 - (w_3 - w_1)x_2] \tag{3.22}$$

Until this point, we have tried to find the expression of x and y in terms of z so we have Equations (3.16) and (3.17). Now, the expressions obtained for x and y are substituted in Equation (3.6) to eliminate x and y :

$$a = (a_1^{\ 2} + a_2^{\ 2} + d_3^{\ 2}) \tag{3.23}$$

$$b = 2(a_1 b_1 + a_2(b_2 - dy_1) - z_1 d^2) \tag{3.24}$$

$$c = b_1^{\ 2} + (b_2 - dy_1)^2 + d^2(z_1^{\ 2} - r_e^2) \tag{3.25}$$

$$az^2 + bz + c = 0 \tag{3.26}$$

As you can see, Equation (3.26) is a second-order linear equation. If the discriminant is bigger than zero, there are two solutions for z .

$$\Delta = b^2 - 4ac > 0 \quad \rightarrow \quad z\pm = \frac{-b \pm \sqrt{\Delta}}{2a} \tag{3.27}$$

As mentioned before, we choose the negative z solution because the origin point is located on the base plate. Therefore, downward direction has a negative sign. The negative solution for z is:

$$z\pm = \frac{-b \pm \sqrt{\Delta}}{2a} \tag{3.28}$$

To complete the forward kinematics solution of the Delta Robot, we substitute z from (3.28) into (3.16) and (3.17). Finally, we have one solution:

$$\{x\ y\ z\}^T \tag{3.29}$$

3.3.2 Inverse Kinematics of Delta Robot

To know the desired position of the end-effector, we need to determine the angle of each joint to set motors in the proper position for picking. This process is known as inverse kinematics, in the following chapter we will explain the inverse kinematic solutions for the Delta Robot.

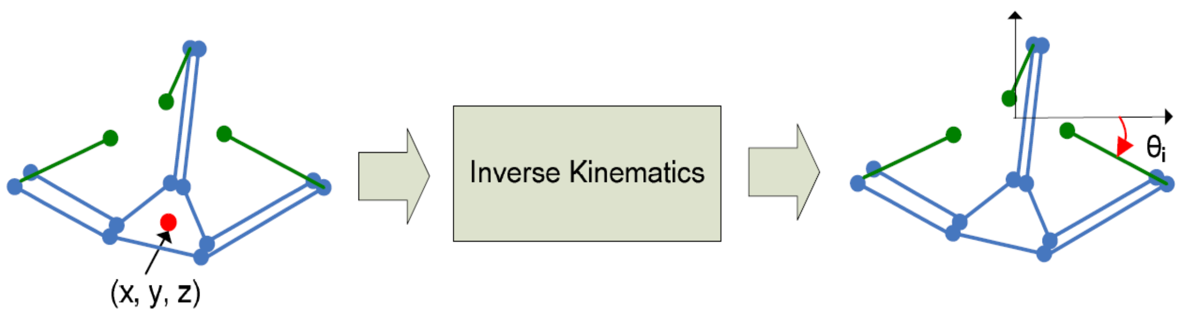


Figure 3. 8: Mapping from Cartesian Space to Joint Space

To construct the inverse kinematics of the delta robot, let's return to Table 3.1 and remember the Delta Robot's physical parameters. Like the forward kinematics section of the Delta Robot, we will move step-by-step.

The F_1J_1 joint can rotate in YZ plane forming circle with center in point F_1 with radius r_f due to the robot design. As opposed to F_1, J_1 and E_1 are called universal joints which mean that $E1\,J1$ can rotate freely relative to E_1, forming a sphere with the center at point E_1 and radius re.

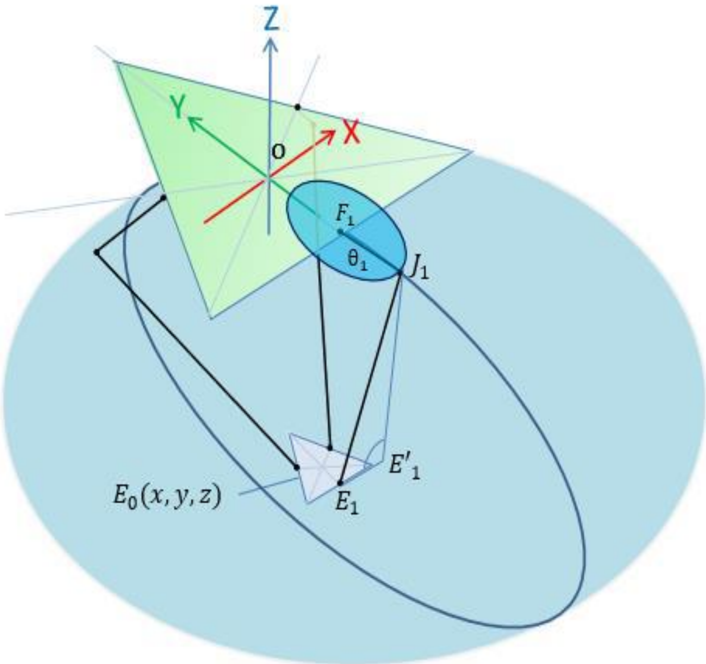


Figure 3. 9: Sample Joint Path

Intersection of this sphere and the YZ plane is a circle with its center at the point E'_1 and radius E'_1J_1, where E'_1 is the projection of the point $E1$ on the YZ plane. The point J_1 can be found now as the intersection of the two circles with known radius' and centers at E'_1 and F_1 , and we can calculate θ_1 angle.

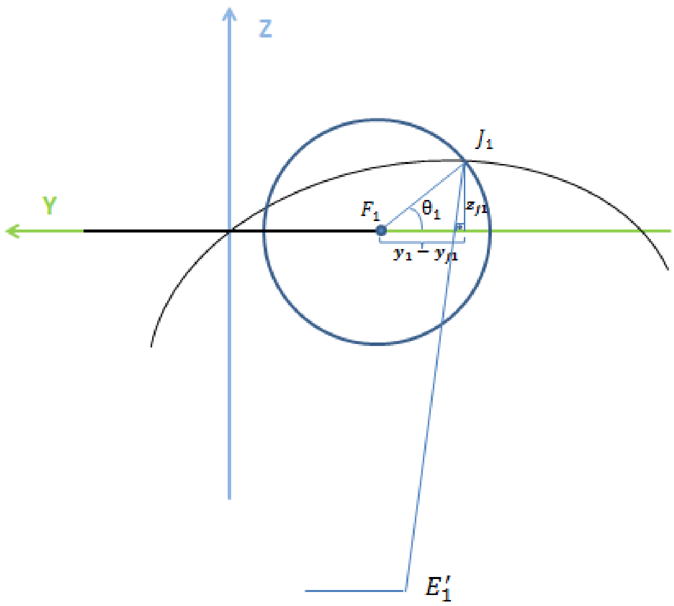


Figure 3. 10: Cartesian Space Representation of Sample Joint

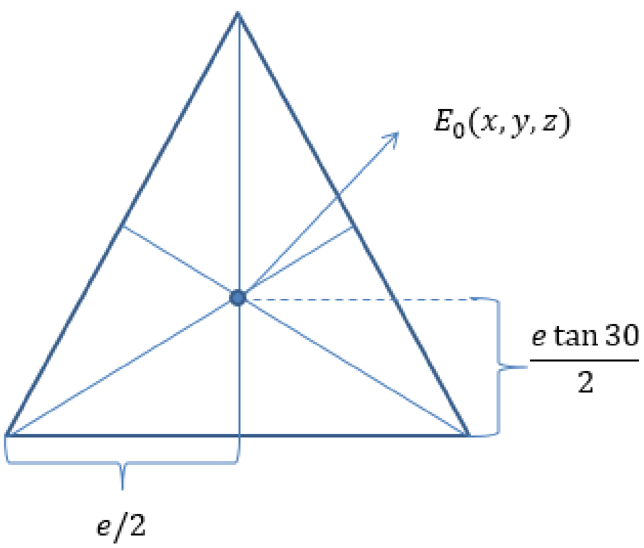


Figure 3. 11: Triangle of the Travelling Plate

So, the corresponding equations can be found and the YZ plane can be seen in the above figure:

$E_0(x, y, z)$ is obtained by subtracting the travelling plate's coordinates from the center of the base triangle's coordinates:

$$EE_1 = \frac{e \tan \pi/6}{2} = \frac{e}{2\sqrt{3}} \tag{3.30}$$

$$E_1\left(x, y - \frac{e}{2\sqrt{3}}, z\right) \quad \rightarrow \quad E_1'\left(0, y - \frac{e}{2\sqrt{3}}, z\right) \tag{3.31}$$

$$EE_1^{'} = x \quad \rightarrow \quad E_1^{'}J_1 = \sqrt{E_1J_1^{\,2} - EE_1^{'2}} = \sqrt{r_e^{\,2} - x^2} \tag{3.32}$$

Coordinates of point F_1 is as follows:

$$F_1(0, \frac{-f}{2\sqrt{3}}, 0) \tag{3.33}$$

Constructing Euclidean equations from Figure 3.10, we obtain:

$$(y_{J1} - y_{E^{'}1})^2 + \left(z_{J1} - z_{E^{'}1}\right)^2 = r_e^{\,2} \;\; \rightarrow \left(y_{J1} - y + \frac{e}{2\sqrt{3}}\right)^2 + (z_{J1} - z)^2 = r_e^{\,2}$$

$$(y_{J1} - y_{E^{'}1})^2 + \left(z_{J1} - z_{E^{'}1}\right)^2 = r_e^{\,2} \;\; \rightarrow \left(y_{J1} - y + \frac{e}{2\sqrt{3}}\right)^2 + (z_{J1} - z)^2 = r_e^{\,2} \tag{3.34}$$

Finally, inverse kinematics solution for the first joint is:

$$\theta_1 = \arctan\left(\frac{z_{J1}}{y_{F1} - y_{J1}}\right) \tag{3.35}$$

For remaining angles $\theta_2, \theta_3,$ let's rotate the coordinate system in the XY plane around z-axis by 120 and -120 degrees.

$$\left.\begin{aligned} x^{'} &= x\cos\left(\frac{2\pi}{3}\right) + y\sin\left(\frac{2\pi}{3}\right) \\ y^{'} &= -x\sin\left(\frac{2\pi}{3}\right) + y\cos\left(\frac{2\pi}{3}\right) \end{aligned}\right\} E_0\left(x, y, z\right) \rightarrow (x^{'}, y^{'}, z^{'}) \tag{3.36}$$

3.3.3 Mobility of Delta Robot

To show the degree-of-freedom of the Delta robot, we can use Kutzback mobility equation which is:

$$M = 6(N-1) - 5J_1 - 4J_2 - 3J_3 \tag{3.37}$$

Table 3. 2: Technical Details for Round Electromagnet

M	**Mobility, # degree of freedom**	
N	**Total number of links, including ground**	
J_1	# One DOF joints	Revolute & prismatic joints
J_2	# Two DOF joints	Universal joints
J_3	# Three DOF joints	Spherical joints

We ignore the four-parallel mechanism by replacing each with a single link instead and we count a universal joint at either end of this virtual link.

$$N = 8, \qquad J_1 = 3, \qquad J_2 = 6 \qquad J_3 = 0 \quad \rightarrow \quad M = 3 - DOF \tag{3.38}$$

3.4 Velocity Kinematics

The kinematics of the Delta robot can be studied by analyzing Figure 3.12, where O represents the center of the fixed platform and A_i three points on the middle of the three sides of the fixed platform. These are the points where the limbs hold onto the platform and correspond to the position of the actuators. P is the center of the moving platform.

Two convenient sets of Cartesian co-ordinate frames, xyz and $x_i y_i z_i$, are defined in such a way that the xy-plane and the $x_i y_i$-plane are the same and coincide with the plane of the fixed platform. Axes z and z_i are perpendicular to the above planes and, therefore, are identical. The angle between x-axis, i.e. Ox, and the x_i-axis, i.e. Ox_i, OA_i or $A_i x_i$, is ϕ_i. OA_i is always parallel to PC_i. Owing to the presence of the rotational joint at OA_i, $A_i B_i$ moves just in the $x_i z_i$-plane. At B_i are passive spherical joints. So $B_i C_i$ can have components along all the three axes, namely $A_i x_i$, $A_i y_i$ and $A_i z_i$. θ_{1i} is the angle between $A_i B_i$ and $A_i x_i$. θ_{2i} is the angle between $A_i B_i$ and the line that results from the intersection of the planes $x_i z_i$ and that of the parallelogram shown on the right half of Figure 3.12. Therefore, this line lies in the $x_i z_i$-plane. θ_{3i} is the angle between $B_i C_i$ and $A_i y_i$. As a result, $b \cos\theta_{3i}$ is the component of $B_i C_i$ parallel to $A_i y_i$ and $b \sin\theta_{3i}$ is the component of $B_i C_i$ in the $x_i z_i$-plane.

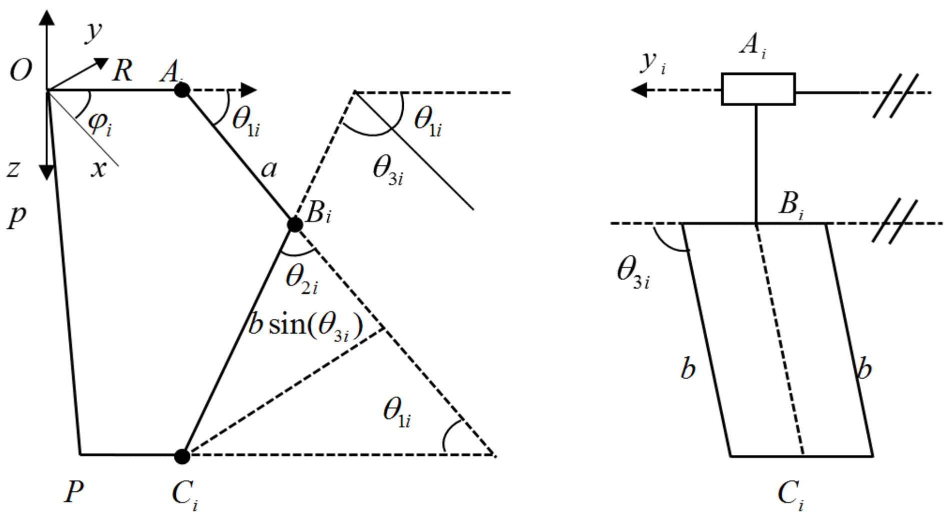


Figure 3. 12: Kinematic description of the i^{th} kinematic chain

The most relevant loop should be picked up for the intended Jacobian analysis. Let $\vec{\boldsymbol{q}}$ be the vector made up of actuated joint variables and $\vec{\boldsymbol{p}}$ be the position vector of the moving platform. Then

$$\vec{\theta} = \theta_{1i} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \quad \vec{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{3.39}$$

The Jacobian matrix will be derived by differentiating the appropriate loop closure equation and rearranging the result in the following form.

$$J_\theta \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} = J_p \begin{bmatrix} \dot{p}_x = v_x \\ \dot{p}_y = v_y \\ \dot{p}_z = v_z \end{bmatrix} \tag{3.40}$$

where v_x, v_y, and v_z are the x, y, and z components of the velocity of the point P on the moving platform in the xyz frame. In order to arrive at the above form of the equation, we look at the loop $OA_iB_iC_iP$. The corresponding closure equation in the $x_iy_iz_i$ frame is.

$$\overrightarrow{\boldsymbol{OP}} + \overrightarrow{\boldsymbol{PC_i}} = \overrightarrow{\boldsymbol{OA_i}} + \boldsymbol{A_iB_i} + \overrightarrow{\boldsymbol{B_iC_i}} \tag{3.41}$$

In the matrix form we can write it as

$$\begin{bmatrix} p_x \cos\phi_i - p_y \sin\phi_i \\ p_x \sin\phi_i + p_y \cos\phi_i \\ p_z \end{bmatrix} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} + a \begin{bmatrix} \cos\theta_{1i} \\ 0 \\ \sin\theta_{1i} \end{bmatrix} + b \begin{bmatrix} \sin\theta_{3i}\cos(\theta_{2i}+\theta_{1i}) \\ \cos\theta_{3i} \\ \sin\theta_{3i}\sin(\theta_{2i}+\theta_{1i}) \end{bmatrix} \tag{3.42}$$

Time differentiation of this equation leads to the desired Jacobian equation as shown in the next section.

3.4.1 The Jacobian Matrices

The loop closure equation (3.41) can be re-written as

$$(\vec{\boldsymbol{p}} + \vec{\boldsymbol{r}}) = \vec{\boldsymbol{R}} + \vec{\boldsymbol{a}}_i + \vec{\boldsymbol{b}}_i \tag{3.43}$$

Differentiating this equation with respect to time and using the fact that $\vec{\boldsymbol{R}}$ is a vector characterizing the fixed platform.

$$\underbrace{\dot{(\vec{\boldsymbol{p}} + \vec{\boldsymbol{r}})}}_{\dot{\vec{p}}} = \dot{\vec{\boldsymbol{a}}}_i + \dot{\vec{\boldsymbol{b}}}_i$$

In this expression, every point on the moving platform has exactly the same velocity. Therefore

$$\dot{\vec{p}} = \dot{\vec{v}} = \dot{\vec{a}}_i + \dot{\vec{b}} \tag{3.44}$$

The linear velocities on the right-hand side of equation (3.44) can be readily converted into the angular velocities by using the well-known identities.
Thus

$$\vec{v} = \overrightarrow{w_{ai}} \times \overrightarrow{a_i} + \overrightarrow{w_{bi}} \times \overrightarrow{b_i} \tag{3.45}$$

The presence of $\overrightarrow{w_{bi}}$ introduces an awkward dependence upon the variables $\dot{\theta}_{2i}$ and $\dot{\theta}_{3i}$.

However, there is a way out. It can be got rid of by taking a scalar product of expression (3.45) with the unit vector $\overrightarrow{b_i}$

$$\hat{b}_i \cdot \left[\vec{v} = \overrightarrow{w_{ai}} \times \overrightarrow{a_i} + \overrightarrow{w_{bi}} \times \overrightarrow{b_i} \right]$$

As the triple product with two identical vectors is zero, what is left is merely.

$$\hat{b}_i . \vec{v} = \overrightarrow{b_i} . \overrightarrow{w_{ai}} \times \overrightarrow{a_i} \tag{3.46}$$

In the component form, the left-hand side of this equation can be written as

$$\begin{aligned} \hat{b}_i . \vec{v} = & [\sin\theta_{3i} \cos(\theta_{2i} + \theta_{1i})][v_x \cos\phi_i - v_y \sin\phi_i] + \cos\theta_{3i}[v_x \sin\phi_i + \\ & v_y \cos\phi_i] + [\sin\theta_{3i} \sin(\theta_{2i} + \theta_{1i})] v_z \end{aligned} \tag{3.47}$$

$$= J_{ix} v_x + J_{iy} v_y + J_{iz} v_z$$

where

$$\begin{aligned} J_{ix} &= \sin\theta_{3i} \cos(\theta_{2i} + \theta_{1i}) \cos\phi_i + \cos\theta_{3i} \sin\phi_i \\ J_{ix} &= -\sin\theta_{3i} \cos(\theta_{2i} + \theta_{1i}) \sin\phi_i + \cos\theta_{3i} \cos\phi_i \\ J_{ix} &= \sin\theta_{3i} \sin(\theta_{2i} + \theta_{1i}) \end{aligned} \tag{3.48}$$

On the right-hand side of equation (3.46), the movement of the joint a is in the $x_i z_i$ -plane. Thus, it only has a component of velocity in this plane. This is the angular velocity about the y axis. Thus

$$\overrightarrow{w_{ai}} = \begin{bmatrix} 0 \\ -\dot{\theta}_{1i} \\ 0 \end{bmatrix} \tag{3.49}$$

The negative sign is just a matter of convention.

Therefore

$$\overrightarrow{\boldsymbol{w}_{ai}} \times \vec{\boldsymbol{a}}_i = \begin{vmatrix} \hat{\boldsymbol{i}} & \hat{\boldsymbol{j}} & \hat{\boldsymbol{k}} \\ 0 & -\dot{\theta}_{1i} & 0 \\ a_{1i} & a_{2i} & a_{3i} \end{vmatrix} = -a_{3i}\dot{\theta}_{1i}\hat{\boldsymbol{i}} + a_{3i}\dot{\theta}_{1i}\hat{\boldsymbol{k}}$$

The right-hand side of equation (3.46) can now be written in its simplified form as

$$\hat{\boldsymbol{b}}_i.(\overrightarrow{\boldsymbol{w}_{ai}} \times \vec{\boldsymbol{a}}_i) = -a\sin\theta_{2i}\sin\theta_{3i}\dot{\theta}_{1i} \tag{3.50}$$

The equations (3.47) and (3.50) can be equated for every value of i

$$\begin{aligned} J_{1x}v_x + J_{1y}v_y + J_{1z}v_z &= -a\sin\theta_{21}\sin\theta_{31}\dot{\theta}_{11} \\ J_{2x}v_x + J_{2y}v_y + J_{2z}v_z &= -a\sin\theta_{22}\sin\theta_{32}\dot{\theta}_{12} \\ J_{3x}v_x + J_{3y}v_y + J_{3z}v_z &= -a\sin\theta_{23}\sin\theta_{33}\dot{\theta}_{13} \end{aligned}$$

which readily implies

$$J_p\vec{v} = J_\theta\dot{\vec{\boldsymbol{\theta}}} \tag{3.51}$$

where

$$J_p = \begin{bmatrix} J_{1x} & J_{1y} & J_{1z} \\ J_{2x} & J_{2y} & J_{2y} \\ J_{3x} & J_{3y} & J_{3z} \end{bmatrix} \tag{3.52}$$

and

$$J_\theta = a \times \begin{bmatrix} \sin\theta_{21}\sin\theta_{31} & 0 & 0 \\ 0 & \sin\theta_{22}\sin\theta_{32} & 0 \\ 0 & 0 & \sin\theta_{23}\sin\theta_{33} \end{bmatrix} \tag{3.53}$$

3.5 The Singularities

The singularities of the manipulator are obtained by encountering the conditions that yield singular Jacobian matrices. The two-part Jacobian helps in the classification of these singularities.

3.5.1 Inverse kinematic singularities

The inverse kinematic singularities are associated with the inverse Jacobian and they arise when.

$$\det(J_\theta) = 0 \tag{3.54}$$

As is well known, the physical significance of this condition becomes obvious if the equation (3.51) is written as follows.

$$\dot{\vec{\boldsymbol{\theta}}} = J_\theta^{-1} J_p \vec{v} = \frac{Adj(J_\theta)}{\det(J_\theta)} J_p \vec{v}$$

Since the velocities cannot be infinitely large, the condition $\det(J_\theta) = 0$ must imply $\vec{\boldsymbol{v}} = 0$ in some direction. Thus, there exist some non-zero $\dot{\vec{\boldsymbol{\theta}}}$ vectors that produce $\vec{v}$ vectors that are zero in some direction, i.e., there are moving platform velocities that cannot be achieved. This happens at the boundary of the workspace. Equation (3.54) implies.

$$\theta_{2i} = 0 \text{ or } \pi \text{ for any of the } i \tag{3.55}$$

$$\theta_{3i} = 0 \text{ or } \pi \text{ for any of the } i \tag{3.56}$$

1) Condition (3.55) corresponds to the configuration when the limb a_i is in the plane of the parallelogram formed by limb b_i for any of the i.
2) Condition (3.56) corresponds to the posture when any of the limbs b_i is parallel (or anti-parallel) to y-axis. As a_i is in the xz-plane, it means that in this configuration $a_i \perp b_i$.

For example, the completely stretched out posture of the robot tends to make all a_i parallel to b_i. On the other hand, a completely contracted position tries to force a_i and b_i anti-parallel to each other. These positions of the manipulator correspond to the lower and upper boundaries of the workspace and have been depicted in Figure 3.13 and 3.14. Note that in practice, a_i and b_i cannot become completely anti-parallel because of the mechanical restrictions imposed by the rotational joints possessing a finite size.

3.5.2 Direct kinematic singularities

The direct kinematic singularities are related to the singularities of the direct Jacobian given in equation (3.52), which is much more complicated than the inverse Jacobian. Recalling that the determinant of a Jacobian vanishes when any row or any column is identically zero, a representative example of a couple of singular configurations is provided next.

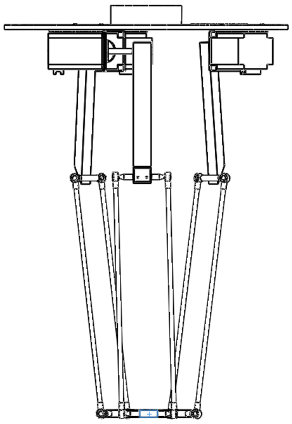


Figure 3. 13: Completely stretched out posture of the robot.

$$\theta_{3i} = 0 \text{ or } \pi \quad \forall i \tag{3.57}$$

$$\theta_{2i} + \theta_{1i} = 0 \text{ or } \pi \quad \forall i \tag{3.58}$$

1) Condition (3.57) implies that the third column of J_p is zero. Physically, the posture of the robot corresponds to when all the limbs b_i are in the plane of the moving platform and in fact lie entirely along y_i-axes.

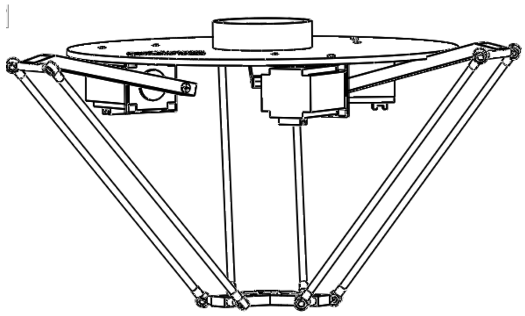


Figure 3. 14: Completely contracted configuration of the robot.

2) Condition (3.58) also implies that the third column of J_p is zero. Physically, the posture of the robot again corresponds to when all the limbs b_i are in the plane of the moving platform. However, they can have both x_i and y_i components non-zero. An example is shown in Fig. 5, where $\theta_{2i} + \theta_{1i} = \pi$. This situation is depicted by a virtual horizontal plane as the actual relative lengths of a_i and b_i prevent reaching that singular configuration.

Conditions (3.56) and (3.57) are harder to visualize and we refrain from displaying their figurative representation.

3.6 Dynamics of Delta robot

Robot dynamic deals with the mathematical formulation of the equations of robot arm motion. These equations describe the dynamic behavior of the manipulator. In this section, the dynamics of the 3-DOF delta robot are considered, where for a given trajectory of the moving platform it is desired to determine the torques applied at the input links by the actuators that will achieve that trajectory.

An understanding of the manipulator dynamics is important from several different perspectives. First, it is necessary to properly size the actuators and other manipulator components. Without a model of the manipulator dynamics, it becomes difficult to predict the actuator torque requirements and in turn equally difficult to properly select the actuators. Second, a dynamics model is useful for developing a control scheme.

3.6.1 Newton-Euler Equations of Motion for Simplified Delta robot

This model of the 3-DOF delta robot dynamics assumes that the fixed base of the manipulator is mounted above the workspace where it operates such that gravity is acting in the positive z-direction as shown in Figure 3.15. This dynamics model also assumes that the mass of each connecting rod, m_b in the upper arm assembly is evenly divided between, and concentrated at, joints B_i and E_i. This assumption can be made without significantly compromising the accuracy of the model since the concentrated mass model of the connecting rods does capture some of the dynamics of the rods. Moreover, the connecting rods should not play a dominant role in the 3-DOF delta robot dynamics because the connecting rods can be made quite light relative to the rest of the robot since these connecting rods don't need to carry any of the load associated with the mass of the actuators.

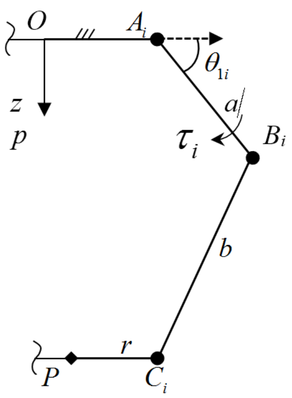


Figure 3. 15: Schematic of the i^{th} leg, showing the orientation of a 3-DOF delta robot and the direction of the applied actuator torque for the dynamic analysis.

Once these assumptions are made, the equation of motion is written by summing the moments about the actuated joint for the i^{th} leg:

$$\sum M_{Ai} = I_A \ddot{\theta}_{1i} + c_d \dot{\theta}_{1i} + \tau^*_{mp,i} \tag{3.59}$$

for $i = 1, 2$ and 3, where $\sum M_{Ai}$ is the sum of the moments applied at joint A_i ; I_A is the mass moment of inertia of the input link, the motor rotor, and one half of the two connecting-rod masses, m_b, that is considered to be concentrated at B_i; $\tau^*_{mp,i}$ is the i^{th} element of τ^*_{mp} which is an array of the intertial loads at joint A_i due to the acceleration of the moving platform and the other half of the connecting rod masses that are considered concentrated at E_i; and c_d is the viscous damping coefficient of the actuator. Note that there are two connecting rods in each parallelogram each with mass m_b. Expressions for I_A and τ^*_{mp} are given by Eqs. (3.60) and (3.61):

$$I_A = I_m + \frac{1}{3} m_a a^2 + m_b a^2 \tag{3.60}$$

$$\tau^*_{mp} = (J^T)^{-1}(3m_b + m_c)\bar{a}_p \tag{3.61}$$

where:

I_m is the mass moment of inertia of the motor rotor,

m_a is the mass of the input link,

m_b is the mass of each rod in link CB,

m_c is the mass of the moving platform and payload,

a is the link length of the input link, and

J is the Jacobian matrix as given in Eq. (4.1), and

$\bar{a}_p$ is the acceleration of the moving platform.

An expression for the resultant moment at joint Ai, due to the motor torques and the gravitational force, for all three legs is given by:

$$\begin{bmatrix} \sum M_{A1} \\ \sum M_{A2} \\ \sum M_{A3} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \frac{1}{2} a\, m_a\, g \begin{bmatrix} \cos\theta_{11} \\ \cos\theta_{12} \\ \cos\theta_{13} \end{bmatrix} + a\, m_b\, g \begin{bmatrix} \cos\theta_{11} \\ \cos\theta_{12} \\ \cos\theta_{13} \end{bmatrix} + (J^T)^{-1} m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.62}$$

where τ_i is the torque applied by the actuator for the i^{th} leg, and $m = 3m_b + m_c$. Substituting Eqs. (3.62) and (3.61) into Eq. (3.59) as written for three legs, and solving for the actuator torques yields:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = -ag(\frac{1}{2}m_a + m_b)\begin{bmatrix} \cos\theta_{11} \\ \cos\theta_{12} \\ \cos\theta_{13} \end{bmatrix} - (J^T)^{-1}m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + c_d\begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} + I_A\begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{12} \\ \ddot{\theta}_{13} \end{bmatrix} + (J^T)^{-1}m\bar{a}_p \quad (3.63)$$

An expression for the acceleration of the moving platform, $\bar{a}_p$ in terms of the manipulator joint angles is given by Eq. (3.64). This is found by differentiating the Jacobian relationship for the manipulator as given by Eq. (4.1).

$$\bar{a}_p = J^{-1}\begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{12} \\ \ddot{\theta}_{13} \end{bmatrix} + \frac{d}{dt}(J^{-1})\begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \quad (3.64)$$

Substituting Eq. (3.64) into Eq. (3.63) and grouping the terms results in:

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

where:

$$\boldsymbol{q} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix} \quad , \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

$$\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{I_A I} + (\boldsymbol{J^T})^{-1} m \boldsymbol{J}^{-1} \quad (3.65)$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = c_d \boldsymbol{I} + (\boldsymbol{J^T})^{-1} m \frac{d}{dt}(\boldsymbol{J}^{-1}) \quad (3.66)$$

$$\boldsymbol{G}(\boldsymbol{q}) = -ag(\frac{1}{2}m_a + m_b)\begin{bmatrix} \cos\theta_{11} \\ \cos\theta_{12} \\ \cos\theta_{13} \end{bmatrix} - (\boldsymbol{J}^T)^{-1}m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.67)$$

and $\boldsymbol{I}$ is the identity matrix: (3.68)

This approach offers the advantage of being less computationally intensive than the Lagrangian approach, while still capturing most of the dynamic characteristics of the 3-DOF delta robot. These characteristics makes it an attractive choice for the dynamics model to be employed in a computed torque control scheme.

3.6.2 Non-Rigid Body Effects

It is important to realize that the derived dynamics equations do not encompass all the effects acting on a manipulator. They include only those forces which arise from rigid body mechanics. The most important source of forces that are not included is friction. All mechanisms are affected by frictional forces. Now days manipulators, in which significant gearing is typical, the forces due to frictional can actually be quite large, perhaps equaling 25% of the torque required to move the manipulator in typical situations. In order to make dynamic equation reflect the reality of the physical device, it is important to include at least the approximation of these forces of friction. A very simple model for friction is viscous friction, in which the torque due to friction is proportional to the velocity of joint motion.

$$\tau_{friction} = V\dot{\theta} \tag{3.69}$$

where V is a viscous-friction constant. Another possible simple model for friction, Coulomb friction, is sometimes used. Coulomb friction is constant except for a sign dependence on the joint velocity and is given by.

$$\tau_{friction} = c\,\mathrm{sgn}(\dot{\theta}) \tag{3.70}$$

where c is a Coulomb-friction constant. The value of c is often taken at one value when $\dot{\theta} = 0$ the static coefficient. But at a lower value, the dynamic coefficient, when $\dot{\theta} \neq 0$.
Whether a joint of a particular manipulator exhibits viscous or Coulomb friction is a complicated issue of lubrication and other effects. A reasonable model is to include both, because both effects are likely:

$$\tau_{friction} = V\dot{\theta} + c\,\mathrm{sgn}(\dot{\theta}) \tag{3.71}$$

It turns out that, in many manipulator joints, friction also displays a dependence on the joint position. A major cause of this effect might be gears that are not perfectly round-their eccentricity would cause friction to change according to joint position. So, a fairly complex friction model would have the form.

$$\tau_{friction} = f(\theta, \dot{\theta}) \tag{3.72}$$

These friction models are then added to the other dynamic terms derived from the rigid body model, yielding the more complete model.

$$\boldsymbol{\tau} = \boldsymbol{M}(\theta)\ddot{\theta} + \boldsymbol{C}(\theta, \dot{\theta})\dot{\theta} + \boldsymbol{G}(\theta) + \boldsymbol{F}(\theta, \dot{\theta}) \tag{3.73}$$

3.6.3 Actuator Dynamics

The active leg of the parallel manipulator is basically composed of DC motor, precision revolute bearing and coupling elements. The model of dc motor is given below.

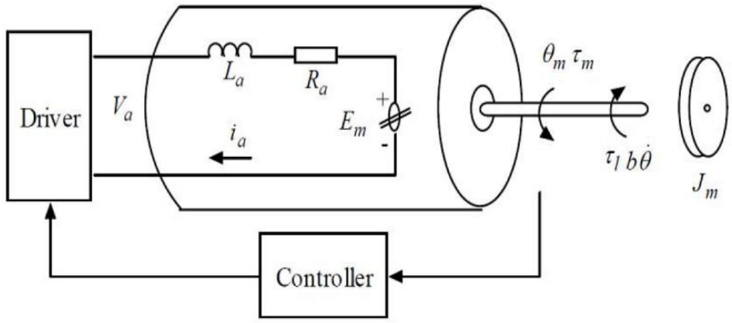


Figure 3. 16: DC motor model (Corapsiz & Kahveci, 2017)

The symbols represent the following variables. θ_m is the motor position in rad, τ_m is the produced torque by the motor in Nm, τ_1 is the load torque, V_a is the armature voltage (V), L is the armature inductance in H, R_a is the armature resistance in Ω, E_m is the reverse EMF(V), i_a is the armature current (A), K_b is the reverse EMF constant and K_m is the torque constant.

$$L\frac{di_a}{dt} + R_a i_a = V_a - E_m \tag{3.74}$$

$$E_m = K_b \frac{d\theta_m}{dt}$$

$$\tau_m = K_m i_a$$

$$\theta_m = K_r \theta_l$$

$$\tau_m = \frac{\tau_1}{K_r}$$

$$\tau_m - \tau_1 = J_m \frac{d^2\theta_m}{dt^2}$$

On the assumption of a rigid transmission and with no backlash the relationship between the input forces (velocities) and the output forces (velocities) are purely proportional. This gives,

$$\theta_m = K_r \theta_l \tag{3.75}$$

Where, constant K_r is a parameter which describes the gear reduction ratio. τ_1 is the load torque at the robot axis and τ_m is the torque produced by the actuator at the shaft axis. Considering the above equation, we can write the motor torque as follows.

$$\tau_m = \frac{\tau_1}{K_r} \tag{3.76}$$

In order to simulate the motion of a manipulator, the dynamic mode of the given manipulator should be utilized. Simulation requires solving the dynamic equation for angular acceleration of the manipulator robot actuator as follows.

$$\ddot{\theta} = M^{-1}(\theta)\left[\tau - C(\theta,\dot{\theta})\dot{\theta} - G(\theta) - F(\theta,\dot{\theta})\right] \tag{3.77}$$

We can then apply any of several known numerical integration techniques to integrate the acceleration to compute future positions and velocities. Given initial conditions on the motion of the manipulator, usually in the form.

$$\theta(0) = \theta_0 \tag{3.78}$$

$$\dot{\theta}(0) = 0$$

CHAPTER 4

CONTROL SYSTEM DESIGN

4.1 Problem statement and preliminaries

4.1.1 Problem Statement

Consider the dynamic equation of 3 DOF delta robotic manipulators as follows.

$$\boldsymbol{M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau(t)+\tau_d(t)} \tag{4.1}$$

$$\boldsymbol{y(t)=q\,,\ y_d(t)=q_d}$$

where, $\boldsymbol{q},\boldsymbol{\dot{q}}$ and $\boldsymbol{\ddot{q}}\in\boldsymbol{R^n}$ are the vector of joint position, joint velocity, and joint acceleration, respectively. $\boldsymbol{M(q)}\in\boldsymbol{R}^{n\times n}$ is a symmetric and positive definite inertia matrix, $\boldsymbol{C(q,\dot{q})}\in\boldsymbol{R}^{n\times n}$ is the matrix of centrifugal and Coriolis forces, $\boldsymbol{G(q)}\in\boldsymbol{R}^n$ is the vector of gravitational forces, $\boldsymbol{\tau}(t)\in\boldsymbol{R}^n$ is the vector of input joint torque and $\boldsymbol{\tau_d}(t)\in\boldsymbol{R}^n$ is the vector of unknown external disturbances.

For practical applications, it is impossible to know the exact dynamic model of the robotic manipulators. Hence, the above dynamic quantities can be expressed as

$$\begin{aligned}M(q)&=M_0(q)+\Delta M(q)\\ C(q,\dot{q})&=C_0(q,\dot{q})+\Delta C(q,\dot{q})\\ G(q)&=G_0(q)+\Delta G(q)\end{aligned} \tag{4.2}$$

where $M_0(q)$, $C_0(q,\dot{q})$, $G_0(q)$ are the nominal values of $M(q)$, $C(q,\dot{q})$, $G(q)$, respectively and $\Delta M(q)$, $\Delta C(q,\dot{q})$, $\Delta G(q)$ are the uncertain parts of $M(q)$, $C(q,\dot{q})$, $G(q)$, respectively. Based on the definition (4.2) the dynamic model of the robotic manipulators is given by

$$M_0(q)\ddot{q}+C_0(q,\dot{q})\dot{q}+G_0(q)=\tau(t)+\rho(q,\dot{q},\ddot{q}) \tag{4.3}$$

where,

$$\rho(q,\dot{q},\ddot{q})=\tau_d(\mathrm{t})-\Delta M(q)\ddot{q}-\Delta C(q,\dot{q})\dot{q}-\Delta G(q)$$

Let define the tracking error $e_1=q-q_d$ and its derivative $\dot{e}_1=\dot{q}-\dot{q}_d$ where q_d is the reference trajectory. Then, the error dynamic of the robotic manipulators with uncertainties and disturbances can be written as

$$\begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = M_0^{-1}(q)\left[-C_0(q,\dot{q})\dot{q} - G_0(q) - M_0(q)\ddot{q}_d + \tau + \rho\right] \end{cases} \tag{4.4}$$

Expression (4.4) can be expressed in the following form.

$$\begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = F_1(e_1,e_2) + B_1(e_1,e_2)\tau + D_1(e_1,e_2) \end{cases} \tag{4.5}$$

where

$$F_1(e_1,e_2) = -M_0^{-1}(q)\left[C_0(q,\dot{q})\dot{q} + G_0(q)\right] - \ddot{q}, \quad B_1(e_1,e_2) = M_0^{-1}(q) \text{ and}$$

$$D_1(e_1,e_2) = -M_0^{-1}(q)\rho = -M_0^{-1}(q)\left[\tau_d(\mathrm{t}) - \Delta M\ddot{q} - \Delta C\dot{q} - \Delta G\right]$$

It is well known that the upper bound of the lumped uncertainty is expressed as.

$$\left|D_1(e_1,e_2)\right| \le \delta_1 = b_0 + b_1\left|q\right| + b_2\left|\dot{q}\right|^2 \tag{4.6}$$

where b_0, b_1 and b_2 are all positive numbers. q and $\dot{q}$ are bounded.

The objective of this research is to design an appropriate sliding mode controller for the nonlinear system (4.1) based on sliding mode theory, so that the system output $y(t)$ can track its reference output $y_d(t)$ in the presence of uncertainties and external disturbances. In order to achieve this goal, the following assumptions are required:

Assumption 1. The reference output $y_d(t)$ is a twice continuously differentiable function in terms of t:

Assumption 2. The matrix $B_1(e_1,e_2)$ is invertible.

Assumption 3. The uncertainty $D_1(e_1,e_2)$ is usually unknown and assumed to be bounded by a positive function as follows.

$$D_1(e_1,e_2) \le \delta_1 \tag{4.7}$$

where δ_1 denotes the upper bound of the lumped uncertainty.

4.1.2 Preliminaries and notations

Before the control law formulation and the stability analysis, the following notations are used.

The signum function is defined as

$$sign(x) = \begin{cases} 1 & if \ x > 0 \\ 0 & if \ x = 0 \\ -1 & if \ x < 0 \end{cases} \tag{4.8}$$

For a variable vector $x = \left[x_1, ..., x_n\right]^T \in R^n$, we define the fractional power of vectors as

$$\begin{aligned} x^\alpha &= \left[x_1^\alpha, ..., x_n^\alpha\right]^T \\ |x|^\alpha &= \left[|x_1|^\alpha, ..., |x_n|^\alpha\right]^T \end{aligned} \tag{4.9}$$

For the simplicity of expression, we define the vector $\operatorname{sgn}(x)^\alpha \in R^n$ as follows.

$$\operatorname{sgn}(x)^\alpha = |x|^\alpha \operatorname{sgn}(x) = \left[|x_1|^\alpha \operatorname{sgn}(x_1), ..., |x_n|^\alpha \operatorname{sgn}(x_n)\right]^T \tag{4.10}$$

It is easy to verify.

$$\frac{d}{dt}(\operatorname{sgn}(x)^\alpha) = \alpha |x|^{\alpha-1} x \quad \alpha \geq 1 \tag{4.11}$$

4.2 Control System Design for 3-DOF Delta Robot

To achieve the desired trajectory tracking of 3-DOF Delta robot it is essential to develop high performance control algorithm. Various types of sliding mode control systems have been developed for the control of nonlinear dynamic systems. Sliding mode control (SMC) is a systematic and effective approach to robust control and has been used in many applications. To increase the rate of convergence in conventional SMC (CSMC), the so-called Adaptive non-singular fast terminal sliding mode (ANFTSM) control proposed a non-linear sliding surface design which guarantees the reachability of a trajectory tracking system.

Furthermore, the ANFTSM control law can guarantee the exponential stability of the control system, which leads to a faster convergence rate of the tracking error (more specific, in a finite time explicitly determined by the design parameters) in comparison to CSMC.

This chapter gives the design of a conventional sliding mode controller and the ANFTSM controller with the stability proof of the control system.

4.2.1 Conventional Sliding Mode Control

Sliding mode control is robust, which is insensitive to the change of system parameters and disturbances. This can be applied to a nonlinear problem. By using sliding mode control (Slotine & Li, 1991), (Utkin et al., 2017) the system response is divided into two phases: in the first phase the state variable is forced to the sliding surface then it slides on the surface to the origin in the second one.

Here the sliding surface is chosen as the following:

$$s(t) = \dot{e}(t) + \Lambda e(t) = \dot{q} - \dot{q}_d + \Lambda e(t) \tag{4.12}$$

where $e = \tilde{q} = q - q_d$ and Λ is a diagonal matrix with positive elements.

$$\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{n_a}), \quad \lambda_i > 0, \quad i = 1, ..., n_a$$

To find a control law, a Lyapunov candidate function is chosen as

$$V = \frac{1}{2} s^T M(q) s \tag{4.13}$$

Differentiating of V with respect to time one gets

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

Putting $\dot{q}_r = \dot{q}_d - \Lambda e$ and from (4.12) it yields

$$\dot{s} = \ddot{e} + \Lambda \dot{e} = \ddot{q} - \ddot{q}_d + \Lambda \dot{e} = \ddot{q} - \ddot{q}_r,$$

$$\ddot{q}_r = \ddot{q}_d - \Lambda \dot{e} \tag{4.15}$$

and

$$\dot{q} = s + \left(\dot{q}_d - \Lambda e\right) = s + \dot{q}_r, \tag{4.16}$$

Putting (4.15) under consideration of dynamic model into (4.1) one gets

$$\begin{aligned} \dot{V} &= s^T \left| M(q)\ddot{q} - M(q)\ddot{q}_r \right| + \frac{1}{2} s^T \dot{M}(q) s \\ &= s^T \left| u - C(q,\dot{q})\dot{q} - G(q) - M(q)\ddot{q}_r \right| + \frac{1}{2} s^T \dot{M}(q) s \end{aligned} \tag{4.17}$$

Substituting (4.16) into (4.17) and noting the skew-symmetric property of $\left[\dot{M}(q)-2C(q,\dot{q})\right]$ one obtains

$$\dot{V}=s^T\left[u-C(q,\dot{q})\dot{q}-F(q,\dot{q})-G(q)-M(q)\ddot{q}_r\right]-s^TF(q,\dot{q})s$$

If the system model and its parameters are known exactly, the control law will be chosen as follows.

$$\tau_{eq}=M(q)\ddot{q}_r+C(q,\dot{q})\dot{q}_r+F(q,\dot{q})\dot{q}_r+G(q)-K_{pd}s \tag{4.18}$$

By choosing K_{pd} being positive definite matrix results in

$$\dot{V}=-s^TK_{pd}s-s^TF(q,\dot{q})s<0, \forall s\neq 0 \tag{4.19}$$

Because the exact parameters of the systems are not available, so the control law (4.18) is modified as

$$\tau=\hat{M}(q)\ddot{q}_r+\hat{C}(q,\dot{q})\dot{q}_r+\hat{F}(q,\dot{q})\dot{q}_r+\hat{G}(q)-K_{pd}s-K_s\,\mathrm{sgn}(s) \tag{4.20}$$

in which the last term is added to ensure that $\dot{V}$ is negative. Assuming that the difference between the system and the model used for controller is defined by

$$h=\Delta M(q)\ddot{q}_r+\Delta C(q,\dot{q})\dot{q}_r+\Delta F(q,\dot{q})\dot{q}_r+\Delta G(q) \tag{4.21}$$

With

$$\Delta M(q)=\left[M(q)-\hat{M}(q)\right], \Delta\hat{C}(q,\dot{q})=\left[C(q,\dot{q})-\hat{C}(q,\dot{q})\right]$$
$$\Delta F(q,\dot{q})=\left[F(q,\dot{q})-\hat{F}(q,\dot{q})\right], \Delta G(q)=\left[G(q)-\hat{G}(q)\right]$$

is bounded, means $|h_i|\leq h_{0,i}$.

The last term in (4.20) is chosen as

$$\tau_{sld}=-K_s\,\mathrm{sgn}(s),\ \tau_{sld,i}=-K_{s,i,i}\,\mathrm{sgn}(s_i),\ K_{s,i,i}=h_{0,i}+\eta,\eta>0$$

With (17) and (18) Eq. (14) becomes

$$\dot{V}\leq\left[-h-K_{pd}s-K_s\,\mathrm{sgn}(s)\right]=-s^TK_{pd}s-s^T(h+K_s\,\mathrm{sgn}(s)) \tag{4.22}$$

Due to $s_i\,\mathrm{sgn}(s_i)=|s_i|$, Eq. (4.22) becomes

$$\dot{V}\leq-s^TK_{pd}s+\sum|s_i|(h_{0,i}-K_{s,ii})\leq-s^TK_{pd}s-\eta\sum|s_i| \tag{4.23}$$

So, the controller (4.20) with $K_{pd} > 0$ and diagonal matrix K_s having $K_{i,i}^s > |h_{0i}|$ guarantees $\dot{V} < 0$, then $s(t) \to 0$ add $e(t) \to 0$.

Noting that, the controller (4.20) has a non-continuous part $K_s \operatorname{sgn}(s)$. It causes unwanted fluctuations with high frequencies around the sliding surface, which is called "chattering" phenomenon. In order to reduce chattering, the discontinuous function $\operatorname{sgn}(.)$ will be replaced by a smooth function of $\arctan(cs), c \geq 1$.

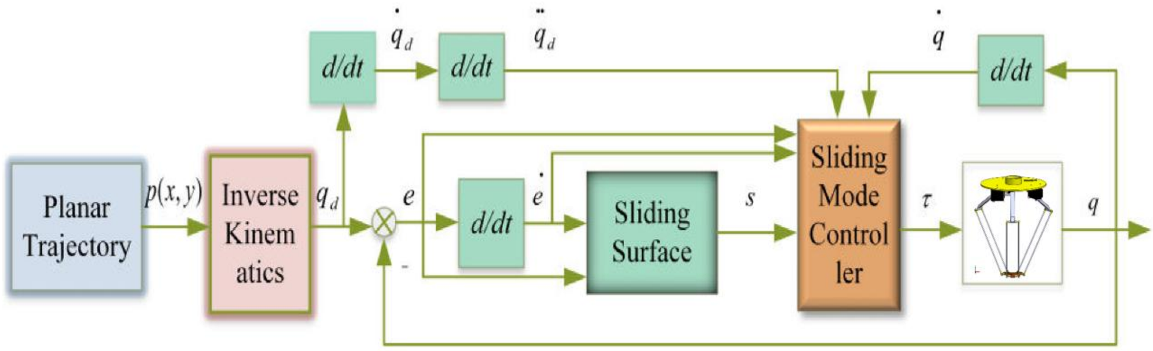


Figure 4. 1: Schematic diagram of the sliding mode control system of the delta robot

4.2.2 The Design Approach of Nonsingular Fast Terminal Sliding Mode Control

The main advantage of using NFTSM is fast convergence, singularity avoidance and strong robustness with respect to system uncertainty and external disturbances. Before developing adaptive control laws for NFTSM first a robust nonsingular fast terminal sliding-mode control will be discussed and analyzed.

A robust nonsingular fast terminal sliding-mode control (RNFTSMC) developed under the assumption that the upper bound of the system uncertainty and external disturbances is exactly known. Since this assumption is difficult to be satisfied in practical applications, an adaptive nonsingular fast terminal sliding-mode control (ANFTSMC) is used to estimate and compensate the unknown upper bounds of the system uncertainty and external disturbances which can increase the robustness of the control system and improve control performance.

4.2.2.1 Robust Nonsingular Fast Terminal Sliding-Mode Control Design

Sliding-mode control design usually consists of the following three steps:

1. The sliding surface design
2. The equivalent control law design
3. The switching control law design

In the first step, an NFTSM surface $s(t)$ is introduced as follows.

$$s(t) = e_1 + k_1 |e_1|^{\alpha} sign(e_1) + k_2 |e_2|^{\beta} sign(e_2) \tag{4.24}$$

where k_1 and k_2 are positive constants, $1 < \beta < 2$ and $\alpha > \beta$, and $sign(.)$ is the signum function.

Remark 1. Using the NFTSM surface (4.24) and for any given initial condition $e_1(0) = e_0 \neq 0$, the system state converges very quickly to $e_1 = 0$ in finite-time. However, when the system state is far from the equilibrium state, the sub item $k_1 |e_1|^{\alpha} sign(e_1)$ dominates $k_2 |e_2|^{\beta} sign(e_2)$ which guarantees a high convergence rate. Moreover, when the system state is close to the equilibrium state, the sub item $k_2 |e_2|^{\beta} sign(e_2)$ guarantees the system convergence in a finite-time.

By using (4.11), the time derivative of the sliding surface (4.24) can be expressed as

$$\dot{s}(t) = e_2 + \alpha.k_1 |e_1|^{\alpha-1} e_2 + \beta.k_2 |e_2|^{\beta-1} \dot{e}_2 \tag{4.25}$$

When the sliding surface $s(t)$ is reached, the system dynamics are equivalent to the following nonlinear differential equation.

$$\beta.k_2 |e_2|^{\beta-1} \ddot{e}_2 + (1 + \alpha.k_1 |e_1|^{\alpha-1}) e_1 = 0 \tag{4.26}$$

After selecting the suitable sliding surface, the next step is to design the equivalent control law. It is obtained by recognizing that $\dot{s} = 0$ which is a necessary condition for the state trajectory to stay on the sliding surface $s = 0$. Thus, substituting the expression (4.5) without any consideration of uncertainty and disturbance into (4.25), we have

$$\dot{s} = e_2 + \alpha.k_1 |e_1|^{\alpha-1} e_2 + \beta.k_2 |e_2|^{\beta-1} (M(q) + B(q)\tau) \tag{4.27}$$

By setting $\dot{s} = 0$, the equivalent control law τ_{eq} is derived as

$$\tau_{eq}(t) = M_0(q)\ddot{q}_d + C_0(q,\dot{q})\dot{q} + G_0(q) - \frac{M_0(q)}{\beta.k_2} |e_2|^{2-\beta} (1 + \alpha.k_1 |e_1|^{\alpha-1}) sign(e_2) \tag{4.28}$$

The equivalent control law $\tau_{eq}(t)$ can make the system state remain on the sliding surface if the dynamic is known exactly.

However, this assumption is difficult to be satisfied in practical applications. In order to satisfy the sliding condition in the presence of uncertainties and external disturbances, a switching control law $\tau_{sw}(t)$ is given by.

$$\begin{aligned}\tau_{sw}(t) &= -M_0(q)\ddot{q}_d\left[k.s+(\delta+\eta)sign(s)\right] \\ &= M_0(q)\ddot{q}_d\left[k.s+(b_0+b_1|q|+b_2|\dot{q}|^2+\eta)sign(s)\right]\end{aligned} \tag{4.29}$$

where η is a small positive constant and k is a positive definite gain matrix. Then, the tracking error converges to zero in finite time.

Thus, the overall control law is designed as

$$\begin{aligned}&\tau(t) = \tau_{eq}(t)+\tau_{sw}(t) \\ &\tau_{eq}(t) = M_0(q)\ddot{q}_d + C_0(q,\dot{q})\dot{q} + G_0(q) - \frac{M_0(q)}{\beta.k_2}|e_2|^{2-\beta}(1+\alpha.k_1|e_1|^{\alpha-1})sign(e_2) \\ &\tau_{sw}(t) = -M_0(q)\ddot{q}_d\left[k.s+(b_0+b_1|q|+b_2|\dot{q}|^2+\eta)sign(s)\right]\end{aligned} \tag{4.30}$$

The main result of the proposed RNFTSMC can be summarized in the following theorem:

Theorem 1. Consider system (4.3) in which the lumped uncertainty satisfies the constraint (4.6). If the NFTSM surface is set as (4.24) and the control law is designed as (4.30), then the system states converge to the sliding surface in a finite time t_r.

Proof:

Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}s^2 \tag{4.31}$$

Differentiating V with respect to time yields

$$\dot{V} = s\dot{s} \tag{4.32}$$

Considering the time derivative of (4.24), (4.32) can be written as follows

$$\dot{V} = s\dot{s} = s\left(e_2+\alpha.k_1|e_1|^{\alpha-1}e_2+\beta.k_2|e_2|^{\beta-1}\dot{e}_2\right) \tag{4.33}$$

By substituting the error dynamic (4.5) and the control law (4.30) in the above equation, the time derivative of V leads to

$$\dot{V} = \beta.k_2 \left|e_2\right|^{\beta-1} \left[D_1(e_1,e_2)s - ks^2 - (\delta+\eta)|s| \right] \tag{4.34}$$

It is easy to verify that.

$$\begin{aligned} \dot{V} &= \beta.k_2 \left|e_2\right|^{\beta-1} \left[\left|D_1(e_1,e_2)\right|.|s| - ks^2 - (\delta+\eta)|s| \right] \\ &= \beta.k_2 \left|e_2\right|^{\beta-1} \left[\left(\left|D_1(e_1,e_2)\right| - \delta \right).|s| - ks^2 - \eta|s| \right] \end{aligned} \tag{4.35}$$

In terms of Assumption (3), we can get.

$$\dot{V} = \beta.k_2 \left|e_2\right|^{\beta-1} \left[D_1(e_1,e_2)s - ks^2 - \eta|s| \right] - \left(\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta \right)|s| \tag{4.36}$$

From (4.6), we can get

$$\dot{V} \le \beta.k_2 \left|e_2\right|^{\beta-1} \left[-ks^2 - \eta|s| \right] \le 0 \tag{4.37}$$

According to the Lyapunov stability theory, the system states converge to the sliding surface $s(t)=0$ asymptotically. To show that this convergence occurs in finite time, (4.37) can be written considering the definition of Lyapunov function (4.31) as

$$\begin{aligned} \dot{V} = \frac{dV}{dt} &\le -2\beta.k.k_2 \left|e_2\right|^{\beta-1} V - \beta.k_2.\eta\left|e_2\right|^{\beta-1}\sqrt{2}V^{1/2} \\ &= -\rho_1 V - \rho_2 V^{1/2} \end{aligned} \tag{4.38}$$

where, $\rho_1 = 2\beta.k.k_2\left|e_2\right|^{\beta-1} > 0$ and $\rho_2 = \beta.k_2.\eta\left|e_2\right|^{\beta-1}\sqrt{2} > 0$

After simple calculation, we get

$$dt \le \frac{-dV}{\rho_1 V + \rho_2 V^{1/2}} = \frac{-V^{1/2}dV}{\rho_1 V^{1/2} + \rho_2} = -2\frac{dV^{1/2}}{\rho_1 V^{1/2} + \rho_2} \tag{4.39}$$

Suppose that the reaching time from the initial state error $e(t) \ne 0$ to $e=0$ is t_r, that is, $V(t_r)=0$. Taking integral of both sides of (4.39), we have

$$\int_0^{t_r} dt \le \int_{V(0)}^{V(t_r)} \frac{-2.dV^{1/2}}{\rho_1 V^{1/2} + \rho_2} = \left[\frac{-2}{\rho_1} \ln\left(\rho_1 V^{1/2} + \rho_2 \right) \right]_{V(0)}^{V(t_r)} \tag{4.40}$$

After simple calculation, one can obtain

$$t_r \le \frac{2}{\rho_1}\ln\left(\frac{\rho_1 V(0)^{1/2}+\rho_2}{\rho_2}\right) \tag{4.41}$$

Therefore, according to the Lyapunov stability criterion, the NFTSM surface $s(t)$ in (4.24) converges to zero at the finite time t_r. On the other hand, if $s(t)=0$ then the output tracking error of the system will also converge to zero in finite time. This completes the Proof.

Remark 2: The first part of the robust control term, $\tau_{eq}(t)$ is used to construct an NFTSM type reaching law that can ensure fast finite-time convergence as the system states are either far away or near the sliding surface. Whereas the second part, $\tau_{sw}(t)$ is adopted to make the system more robust against unmodelled uncertainties and external disturbances.

The characteristics of the proposed RNFTSMC are:

1. The fast finite-time convergence, which can be easily adjusted according to (4.41)
2. The singularity avoidance since the control law (4.30) does not include any negative power
3. The strong robustness against uncertainties and disturbances owing to the NFTSM properties.

4.2.2.2 Adaptive Nonsingular Fast Terminal Sliding-Mode Control Design

It is worth noting that the above-proposed control law can achieve the convergence of the tracking error to zero in a finite time when the upper bounds of the system uncertainty and external disturbances are known. However, this assumption is difficult to be satisfied due to the complexity of the structure of the uncertainties and disturbances. In order to overcome this problem, an adaptive tuning law is used to estimate the parameters of the upper bounds of the system uncertainty δ in expression (4.6). This adaptive tuning method does not require prior knowledge about the upper bound of the system uncertainty. Figure 4.2 shows the control architecture of ANFTSMC. The controller unit calculates the input torques required for actuators (motors) leading the end effector to the target point in minimum time and no error as possible.

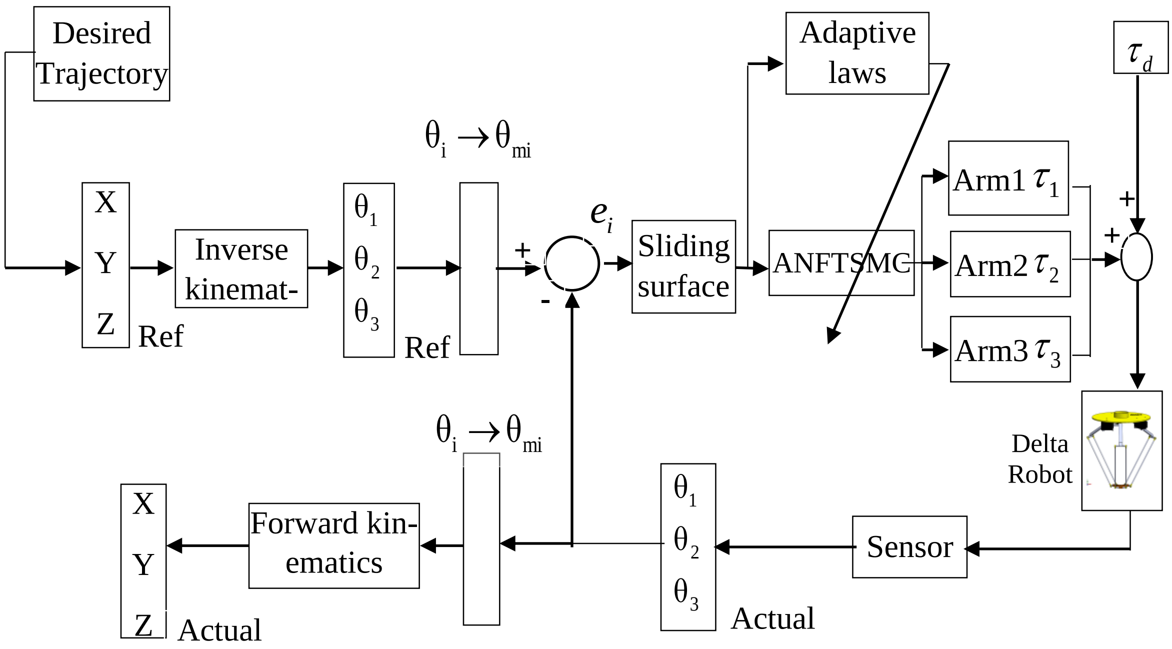


Figure 4. 2: Block diagram of the ANFTSMC trajectory tracking control system

In this section, an ANFTSMC scheme is proposed in order to estimate the unknown upper bounds of the system uncertainties and disturbances. For this purpose, the adaptive switching control law (4.30) can be modified as

$$\tau_{asw}(t) = -M_0(q)\left[k.s + (\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta)sign(s)\right] \tag{4.42}$$

where $\hat{b}_0$, $\hat{b}_1$ and $\hat{b}_2$ be the estimates of b_0, b_1 and b_2, respectively.

Hence, the total control law can be designed as follows:

$$\begin{aligned}
&\tau(t) = \tau_{eq}(t) + \tau_{asw}(t) \\
&\tau_{eq}(t) = M_0(q)\ddot{q}_d + C_0(q,\dot{q})\dot{q} + G_0(q) - \frac{M_0(q)}{\beta.k_2}|e_2|^{2-\beta}(1+\alpha.k_1|e_1|^{\alpha-1})sign(e_2) \\
&\tau_{asw}(t) = -M_0(q)\left[k.s + (\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta)sign(s)\right]
\end{aligned} \tag{4.43}$$

By defining the adaptation error as, $\bar{b}_0 = \hat{b}_0 - b_0$, $\bar{b}_1 = \hat{b}_1 - b_1$ and $\bar{b}_2 = \hat{b}_2 - b_2$ the parameters $\hat{b}_0$, $\hat{b}_1$ and $\hat{b}_2$ are updated by the which are updated by the following adaptive laws

$$\hat{b}_0 = \lambda_0|s|.|e_2|^{\beta-1} \tag{4.44}$$

$$\hat{b}_1 = \lambda_0|s|.|e_2|^{\beta-1}|q| \tag{4.45}$$

$$\hat{b}_2 = \lambda_0|s|.|e_2|^{\beta-1}|\dot{q}|^2 \tag{4.46}$$

where η is a small positive constant, k is a positive definite gain matrix and $\lambda_i;\ i=0,1,2$ are positive constants.

Then, the tracking error converges nearly to zero in finite time. The flowchart diagram of the ANFTSMC scheme is presented in Figure 4.3. The main result of the proposed ANFTSMC is presented in the following theorem:

Theorem 2. Consider system (4.4) in which the upper bound of the lumped uncertainty (4.6) is supposed to be completely unknown. If the NFTSM surface is set as (4.24), the adaptive control law is designed as (4.43) and the adaptation laws are selected as (4.44)-(4.46), then the system states converge to the sliding surface in a finite time.

Proof:

Consider the following Lyapunov function candidate:

$$V=\frac{1}{2}s^2+\beta.k_2\sum_{i=0}^{2}\frac{1}{2\lambda_i}\left(\hat{b}_i-b_i\right)^2 \tag{4.47}$$

Differentiating V with respect to time yields

$$\dot{V}=s\dot{s}+\beta.k_2\sum_{i=0}^{2}\frac{1}{2\lambda_i}\left(\hat{b}_i-b_i\right)\dot{\hat{b}}_i \tag{4.48}$$

Considering the time derivative of (4.24), (4.48) can be written as follows

$$\dot{V}=s\left(e_2+\alpha.k_1\left|e_1\right|^{\alpha-1}e_2+\beta.k_2\left|e_2\right|^{\beta-1}\dot{e}_2\right)+\beta.k_2\sum_{i=0}^{2}\frac{1}{2\lambda_i}\left(\hat{b}_i-b_i\right)\dot{\hat{b}}_i \tag{4.49}$$

By substituting the error dynamic (4.5) and the adaptive control law (4.43) in the above equation, the time derivative of V leads to

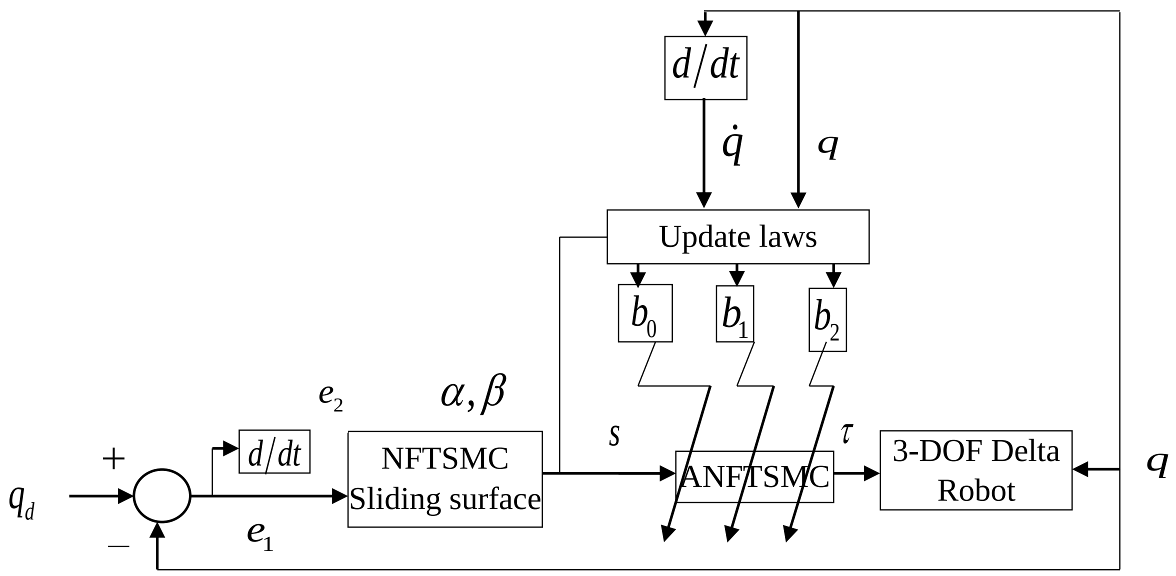


Figure 4. 3: Control flowchart of the proposed ANFTSMC.

$$\dot{V} = \beta.k_2 |e_2|^{\beta-1} \left[D_1(e_1,e_2)s - ks^2 - \left(\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta\right)|e| \right] + \beta.k_2 \sum_{i=0}^{2} \frac{1}{2\lambda_i}\left(\hat{b}_i - b_i\right)\dot{\hat{b}}_i \tag{4.50}$$

Considering the update laws (4.44)-(4.46), it follows

$$\begin{aligned} \dot{V} &= \beta.k_2 |e_2|^{\beta-1} \left[D_1(e_1,e_2)s - ks^2 - \left(\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta\right)|s| \right] \\ &+ \beta.k_2 \left[\left(\hat{b}_0 - b_0\right)|s|.|e_2|^{\beta-1} + \left(\hat{b}_1 - b_1\right)|s|.|e_2|^{\beta-1}|q| + \left(\hat{b}_2 - b_2\right)|s|.|e_2|^{\beta-1}|\dot{q}|^2 \right] \end{aligned} \tag{4.51}$$

Simplifying (4.51) yields

$$\dot{V} = \beta.k_2 |e_2|^{\beta-1} \left[D_1(e_1,e_2)s - ks^2 - \eta|s| \right] - \left(\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta\right)|s| \tag{4.52}$$

From (4.6), we can get

$$\begin{aligned} \dot{V} &\le \beta.k_2 |e_2|^{\beta-1} \left[D_1(e_1,e_2)s - ks^2 - \eta|s| \right] - \left(\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta\right)|s| \\ &\le \beta.k_2 |e_2|^{\beta-1} \left[-ks^2 - \eta|s| \right] \le 0 \end{aligned} \tag{4.53}$$

According to the Lyapunov stability theory, the system states converge to the sliding surface $s(t) = 0$ asymptotically. To show that this convergence occurs in finite time, (4.53) can be written considering the definition of Lyapunov function (4.47) as

$$\begin{aligned} \dot{V} = \frac{dV}{dt} &\le -2\beta.k.k_2 |e_2|^{\beta-1} V - \beta.k_2.\eta |e_2|^{\beta-1} \sqrt{2} V^{1/2} \\ &= -\rho_1 V - \rho_2 V^{1/2} \end{aligned} \tag{4.54}$$

where $\rho_1 = 2\beta.k.k_2 \left|e_2\right|^{\beta-1} > 0$ and $\rho_2 = \beta.k_2.\eta\left|e_2\right|^{\beta-1}\sqrt{2} > 0$

After simple calculation, we get

$$dt \le -2\frac{dV^{1/2}}{\rho_1 V^{1/2} + \rho_2} \tag{4.55}$$

Suppose that the reaching time from the initial state error $e(t) \neq 0$ to $e = 0$ is t_r, that is, $V(t_r) = 0$. Taking integral of both sides of (4.55), we have

$$\int_0^{t_r} dt \le \int_{V(0)}^{V(t_r)} \frac{-2.dV^{1/2}}{\rho_1 V^{1/2} + \rho_2} = \left[\frac{-2}{\rho_1}\ln\left(\rho_1 V^{1/2} + \rho_2\right)\right]_{V(0)}^{V(t_r)} \tag{4.56}$$

After simple calculation, one can obtain

$$t_r \le \frac{2}{\rho_1}\ln\left(\frac{\rho_1 V(0)^{1/2} + \rho_2}{\rho_2}\right) \tag{4.57}$$

Therefore, according to the Lyapunov stability criterion, the NFTSM surface $s(t)$ in (4.24) converges to zero at the finite time t_r. On the other hand, if $s(t) = 0$ then the output tracking error of the system will also converge to zero in finite time. This completes the Proof.

From the Lyapunov stability theory, the system states converge to the sliding surface $s(t)$ asymptotically.

Most of the adaptive work in the literature for controlling robotic systems, the update laws are derived by using well known properties, such as symmetric, anti-symmetric, regression properties. These properties are not required in this work which makes the proposed control laws, applicable to any mechanical system.

Remark 3. Prior knowledge of the upper bounds of the system uncertainty in (4.6) is not required in the ANFTSMC scheme. Adaptive laws in (4.44) -(4.46) are designed to estimate the parameters of the system uncertainty bounds. These estimates are then used in the controller (4.43) to eliminate the effects of the system uncertainty. The ANFTSMC still has an acceptable performance, such as high precision, strong robustness, no singularity, less chattering, and fast finite-time convergence.

Remark 4. The presence of the discontinuous $sign(x)$ function may cause undesirable chattering. To avoid this problem the continuous $\tanh\left(\frac{x}{\varepsilon}\right)$ function is used where $\varepsilon > 0$ a sufficiently small constant is.

From the viewpoint of control engineering, the major difficulties that significantly limit the control performance in real applications are:

i) The knowledge of the system model or the dynamical system upper bound
ii) The presence of the system uncertainties and external disturbances
iii) The feasibility of the control inputs.

It is obvious that neither detailed system dynamics, nor prior knowledge of the upper bounds is required in our proposed control scheme. Moreover, good robustness against uncertainties and disturbances is ensured thanks to the efficiency of the NFTSM control. It is also worth noting that for real implementations of the adaptive algorithm in Eqs. (4.24), (4.43) - (4.46) position and velocity states need to be accessible for feedback.

Simulink Multibody tool introduced an easy way to actuate robot joints by joint actuator block. Sensing computed torques, angles, angular speeds, and accelerations are possible by joint sensor. Beside Bod sensor is used to get the Cartesian Coordinates of the moving platform. Thus, it facilitates the overall simulation process.

Hence, the control inputs are feasible and they do not contain harmful oscillations such as chattering. In summary, the proposed controller promises to be highly suitable even for practical applications.

CHAPTER 5

SIMULATION RESULT AND DISCUSSION

5.1 Multi-body Systems

The high development of computer-aided design (CAD) systems, capable of performing strength calculations of the designed mechanisms, allows eliminating full-scale models and experiments. The mechanism designer needs to know how the developed mechanism will behave in practice, that is, to know its kinematic and dynamic characteristics. The data obtained after the corresponding calculations allow us to select the necessary drives, which is relevant for various systems, including mechatronic ones.

To solve the tasks set, it is convenient to use simulation modeling. Simulation modeling is a method of studying objects because a simulating object replaces the object under study. Experiments are conducted with the simulating object and as a result, information about the object under study is obtained. To simulate the movement of various objects, you can use such programs as SolidWorks, Matlab Simulink, and others.
In the process of designing and creating a solid-state model, the SolidWorks CAD system allows you to determine the design parameters that directly affect its dynamics: mass, moments of inertia, the position of the center of mass, etc.

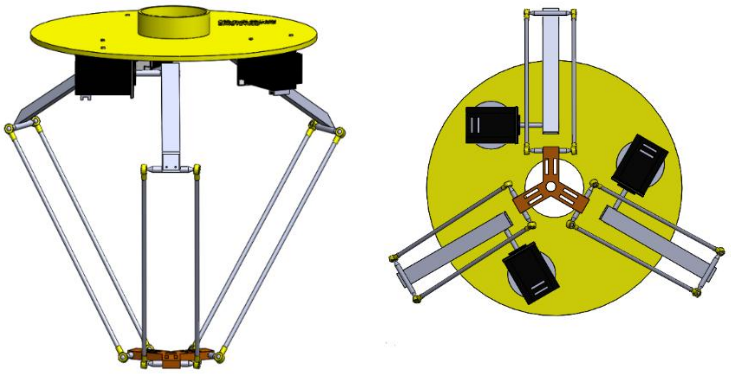
The Simscape software-modeling package is part of the Simulink/MATLAB software package and provides block simulation of complex dynamic systems based on visually oriented programming technology. Simscape Multibody is able to interact with other components of the Simulink/MATLAB library, increasing the simulation capabilities.
The process of simulating a dynamic system in Simulink / MATLAB can be complicated due to the need to determine some parameters: the moment of inertia and the coordinate of each connected element of the system.

To solve this problem, MathWorks has developed a plugin for exporting CAD models, which allows you to create dynamic models in the Simscape Multibody environment based on a three-dimensional model developed in the SolidWorks computer-aided design system using Simscape Multibody Link. Compared to the traditional modeling approach, SimMechanics has several advantages: ease of model construction and parameterization, fast debugging, and flexible measurement and visualization tools.

5.2 3D CAD model of parallel robot

In By using the parametric design and solid modeling technology of Solidworks, the parts of Delta parallel robot can be modeled and assembled quickly, and a perfect visual effect can be obtained as shown in Figure 5.1.

Workspace of SolidWorks, the components the robot were loaded one by one. During this all parts are connected to each other providing the permitted motions between components. Parallelism between fixed and moveable platforms as well as bars of lower arms are also defined. That means the assembled model provides data about permitted motions between components as well as other constrains i.e. distances, angle constrains, parallelisms, etc. On other hand, the 3D CAD model of each part provides information about its geometry, mass center, mass and inertia.



(a) (b)

Figure 5. 1: CAD-model of the manipulator (a) Front view (b) Bottom view.

5.3 Dynamic model of delta robot in Simscape environment

Highly specialized modeling programs do not always allow you to simulate the designed mechanism because they are difficult to master or do not have the ability to repeat the mechanism with the required parameters. The environment of dynamic interdisciplinary modeling of complex technical systems and the main tool for model-oriented design Simulink can help to get out of this situation.

Simulations allow you to test the behavior of the system in critical conditions or emergency scenarios. This reduces the cost of expensive physical prototypes. Simulink also allows you to convert a CAD model to an equivalent Simscape Multibody flowchart. First, we need to build a model of the manipulator in a suitable CAD program (Figure 5.1). For this thesis, the SolidWorks 2021 program was selected.

The conversion is based on the smimport function, which uses the file name of the multi-body XML description as the main argument. The XML file passes the Simscape Multibody program the data needed to recreate the original model or approximate it if the model has unsupported constraints.

The CAD model is translated in two stages: export and import (Figure 5.2). At the export stage, the CAD assembly model is converted to a multi-part XML description file and a set of STEP or STL part geometry files. At the import stage, the part description and geometry files are converted to the SLX Simscape Multibody model and the M-data file. The model gets all the input parameters of the block from the data file.

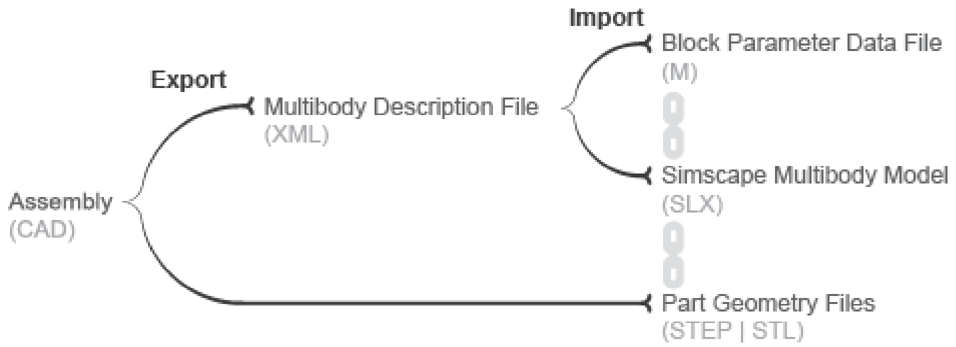
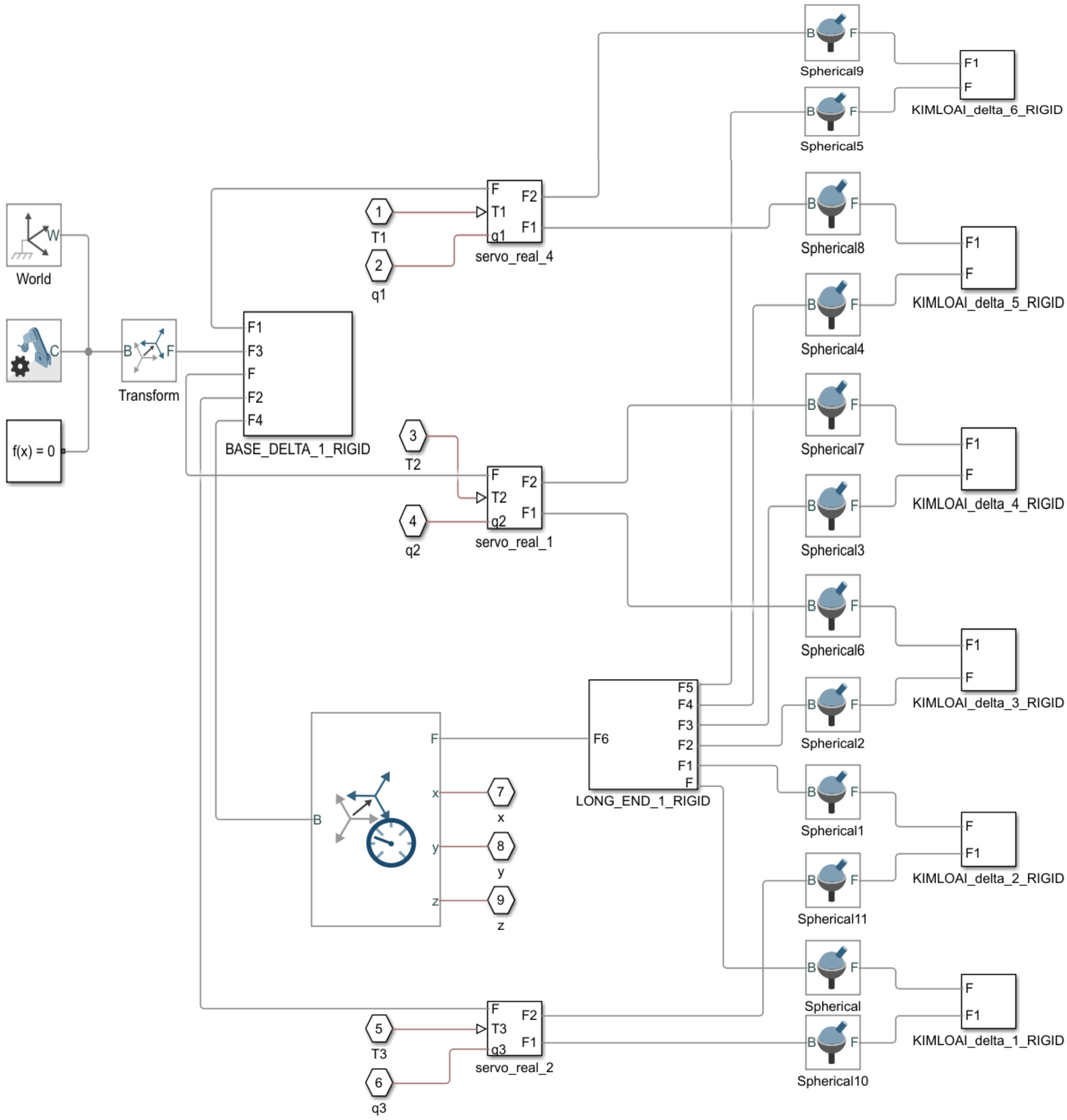


Figure 5. 2: Stages of CAD translation

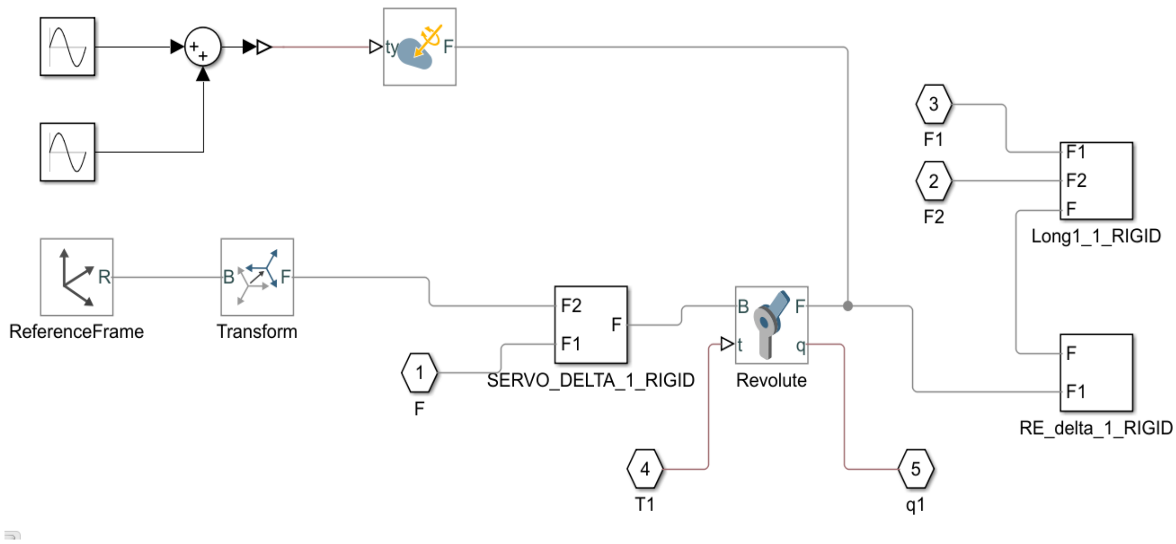
To import model in MATLAB, by typing instruction >> smimport (‘*name of XML file*’) in MATLAB command line the dynamic model of mechanical assembly has been developed (in the formof SLX file). The generated dynamic model is composed of the Simscape library blocks devoted to physical modelling of the systems and it is often realized on one level. The process of converting a CAD model is shown in Figures 5.3. It should also be noted that this method of translation takes into account and affects the characteristics of the materials selected during the construction of the CAD model.

Creating model and build files	Creating and installing an extension for MATLAB and the selected CAD program	Converting files to XML by translator	Import the resulting file to MATLAB and create a model

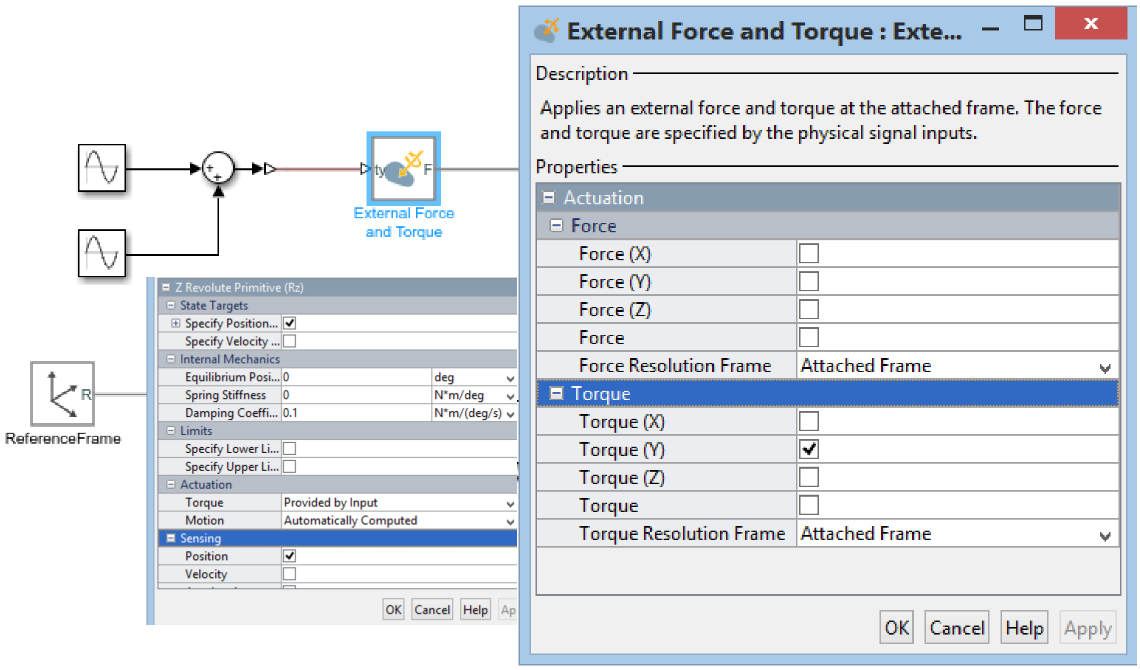
Figure 5. 3: Model conversion process


World
Transform
f(x) = 0
BASE_DELTA_1_RIGID
servo_real_4
T1
q1
T2
q2
servo_real_1
LONG_END_1_RIGID
x
y
z
T3
q3
servo_real_2
Spherical9
Spherical5
Spherical8
Spherical4
Spherical7
Spherical3
Spherical6
Spherical2
Spherical1
Spherical11
Spherical
Spherical10
KIMLOAI_delta_6_RIGID
KIMLOAI_delta_5_RIGID
KIMLOAI_delta_4_RIGID
KIMLOAI_delta_3_RIGID
KIMLOAI_delta_2_RIGID
KIMLOAI_delta_1_RIGID

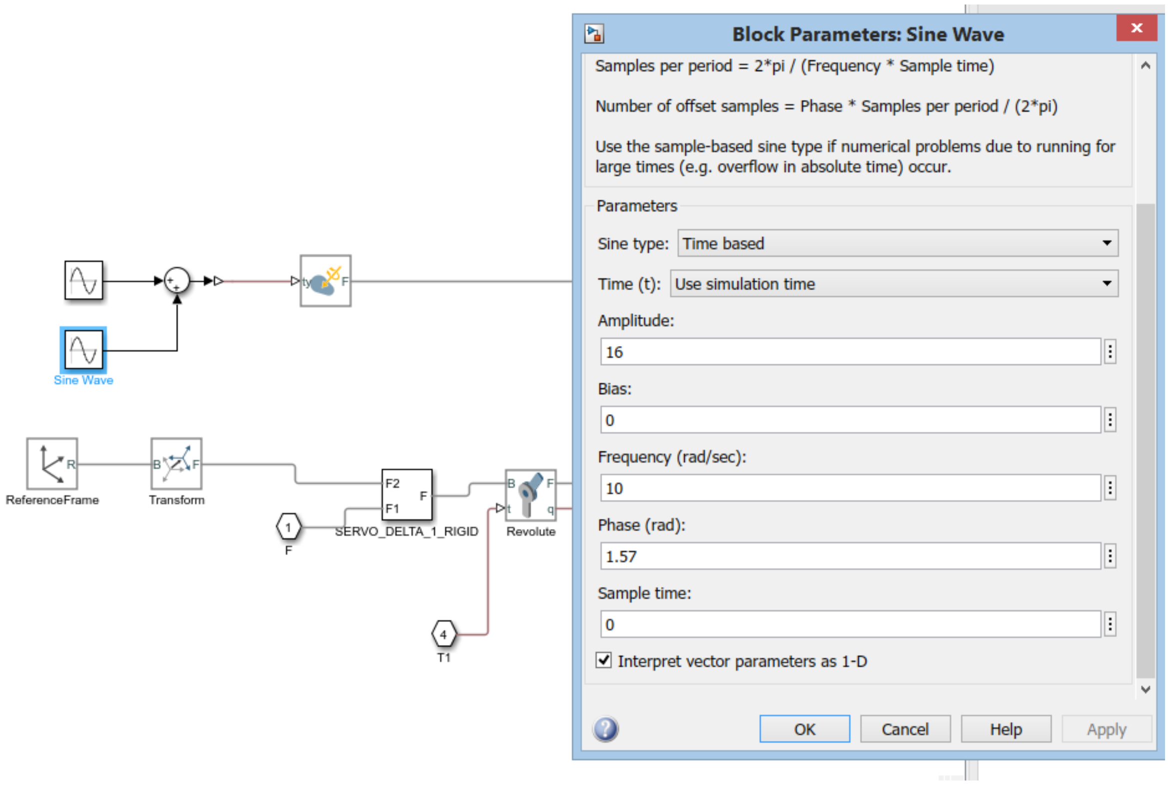
(a)


ReferenceFrame
Transform
F2
F
F1
SERVO_DELTA_1_RIGID
Revolute
F1
F2
F
Long1_1_RIGID
F
F1
RE_delta_1_RIGID
F
T1
q1

(b)


External Force and Torque : Exte...
Description
Applies an external force and torque at the attached frame. The force and torque are specified by the physical signal inputs.
Properties
Actuation
Force
Force (X)
Force (Y)
Force (Z)
Force
Force Resolution Frame
Attached Frame
Torque
Torque (X)
Torque (Y)
Torque (Z)
Torque
Torque Resolution Frame
Attached Frame
OK
Cancel
Help
Apply
External Force and Torque
ReferenceFrame
Z Revolute Primitive (Rz)
State Targets
Specify Position...
Specify Velocity ...
Internal Mechanics
Equilibrium Posi... 0 deg
Spring Stiffness 0 N*m/deg
Damping Coeffi... 0.1 N*m/(deg/s)
Limits
Specify Lower Li...
Specify Upper Li...
Actuation
Torque Provided by Input
Motion Automatically Computed
Sensing
Position
Velocity

(c)



(d)

Figure 5. 4: (a) Simscape Multibody obtained after conversion with few included components (b) Revolute section and disturbance torque (c) and (d) Input torque along y-axis

In real time situation the robot is supposed to be under certain external disturbance force and torque. Depending on the application of the robot the amount and the nature of the disturbance torque may vary. For example, in pick and place application the load mass varies with time in which case the disturbance torque can be approximated as a square wave. When the load is picked the disturbance torque has fixed value and when it is unloaded zero. Though the disturbance can be known or unknown, a bounded sum of two sinusoidal function that could represent the different nature of the disturbances that could possibly encountered have been chosen asymmetrically on each arm. The parameters of these torques are shown in Figure 5.4 (d)

Double clicking on “servo_real” block of the Figure 5.4 (a) the revolute joint and arm with the added disturbance input parameters included is shown in Figure 5.4 (b), (c) and (d). Similarly, the remaining two disturbance inputs are included in the other blocks of Figure 5.4 (a).

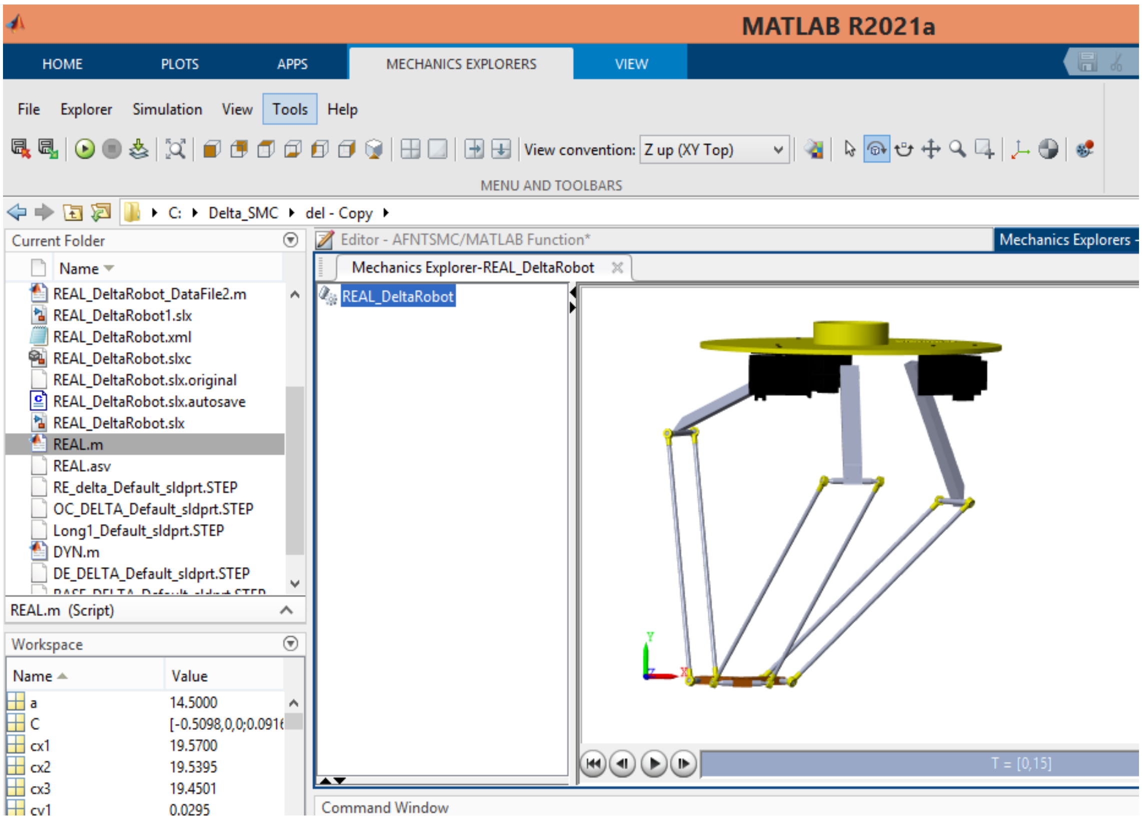


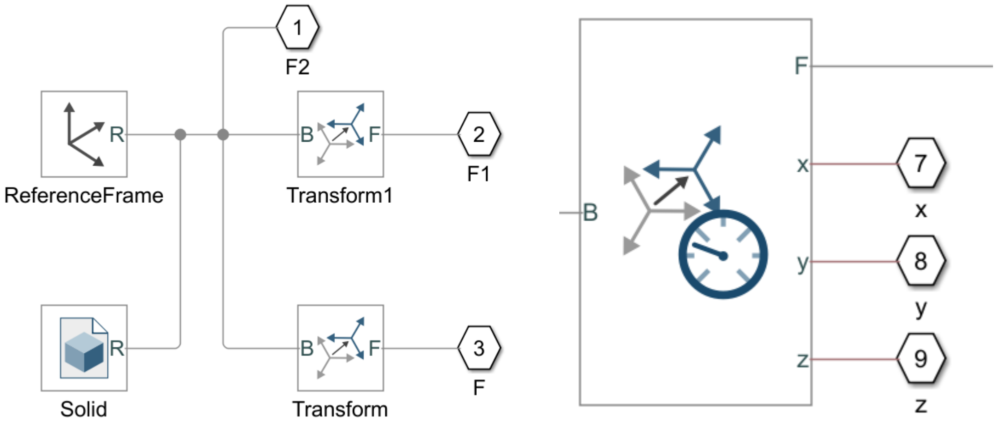
Figure 5. 5: Three-dimensional representation of the transferred model in Simscape Multi-body

In Simulink environment the model can be enhanced and organized systematically on more levels, Figure 5.4.

Blocks from the left side provide following:

- ✓ Block *World* defines unique World coordinate frame,
- ✓ Block *Mechanism Configuration* sets gravity forces and
- ✓ Block *Solver Configuration* defines solver and provides its setting.
- ✓ Block *Transform* provides transformation from World to Global coordinate frames, Figure 5.6.

To measure of rotation angle of three revolute joints in their properties it is specified sensing capability *Position*. Information about rotation angles are fed into *Scope* blocks.



a) b)

Figure 5. 6: Structure of: a) subsystem of rigid body b) block Sensor x, y and z position

Data about mass, coordinate the masscenter, moment of inertia have also taken from 3D CAD model of body. To measure coordinates of point, located in the middle of moveable platform, block *Transform Sensor* is used which can measure the position of the origin and orientation of one coordinate frame with respect to the other.

5.4 Trajectory Generation

For the evaluation of the Delta parallel robot model performance, a circular trajectory in 3 Dimensional spaces was used, as shown by Figure 5.7. In order to obtain a circle as trajectory it was given a cosine input to the x-axis and sine input to the y-axis using constant acceleration motion law. The tracking speed utilized is defined by the angular velocity formula: $\omega = 2\pi f$, where f is the tracking frequency of the end-effector. The frequency implemented is 0.5 Hertz. Radius of the circle trajectory is set to 10 in which is inside the work space and its center is (0, 0, Z_0), where Z_0 is the distance of travelling in z- direction.

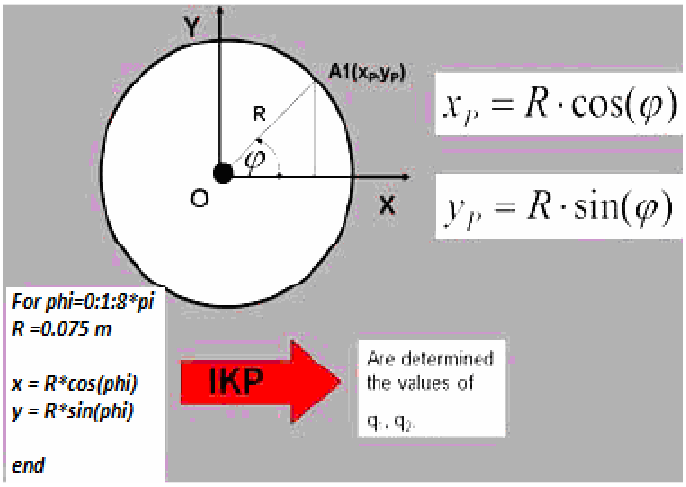
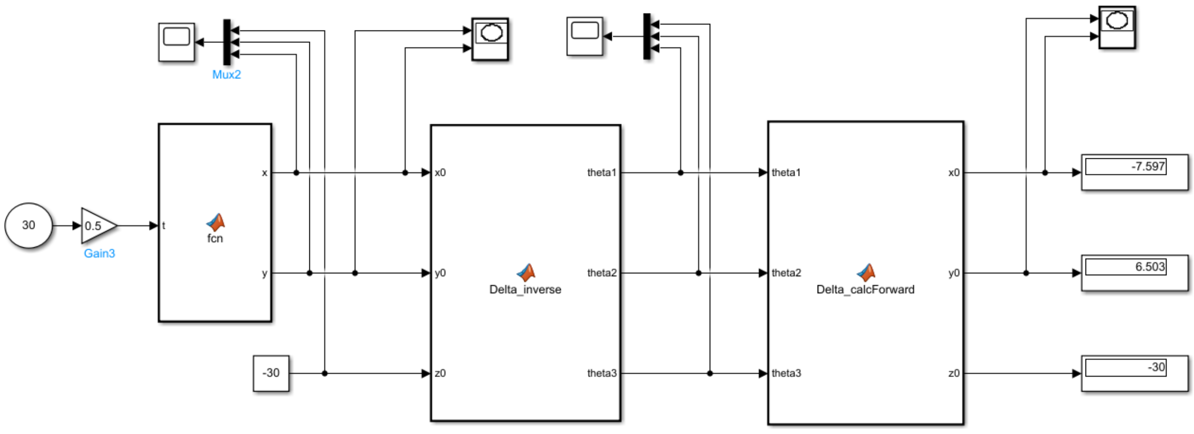


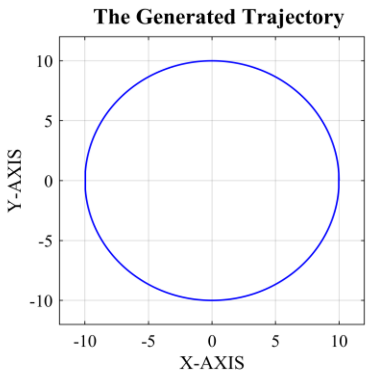
Figure 5. 7: 2D trajectory generation of circle for the delta parallel robot

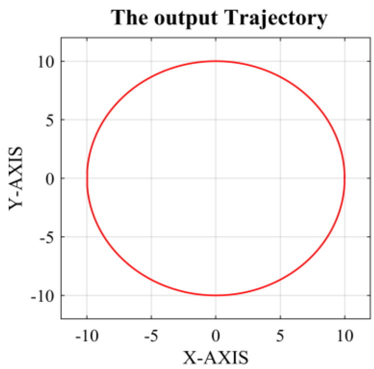
5.5 Kinematics Simulink Model of Delta Robot

The detail derivation of inverse and direct kinematics of Delta robot has been discussed on chapter three section 3.3.1 and 3.3.2. Even though only the inverse kinematics is required to work the simulation of this thesis, forward kinematics also studied and used to proof the kinematics algorism as follows.



(a)





(b) (c)

Figure 5. 8: (a) Simulink model for Kinematics of 3-DOF Delta Robot (b) The generated desired trajectory (c) The output trajectory

5.6 Dynamic Simulink Model Analysis of Delta Robot

Simulink multibody has eased a complex computation which mainly involves large mathematical analysis and singularity in computation. Even if the Newton-Euler based dynamic equation of the delta robot which is derived in chapter 3 is used in the controller design, here the simscape multibody model of the equivalent model is analyzed for the sake of simplicity. In this case we can also observe the benefit of usng simscape multibody model when we deal with complex dynamic models. The essential values for the parameters of the delta robot model are just as mentioned in table 5.1. Figure 5.9 shows the simscape multi-body model of the robot with motion input on matlab Simulink.

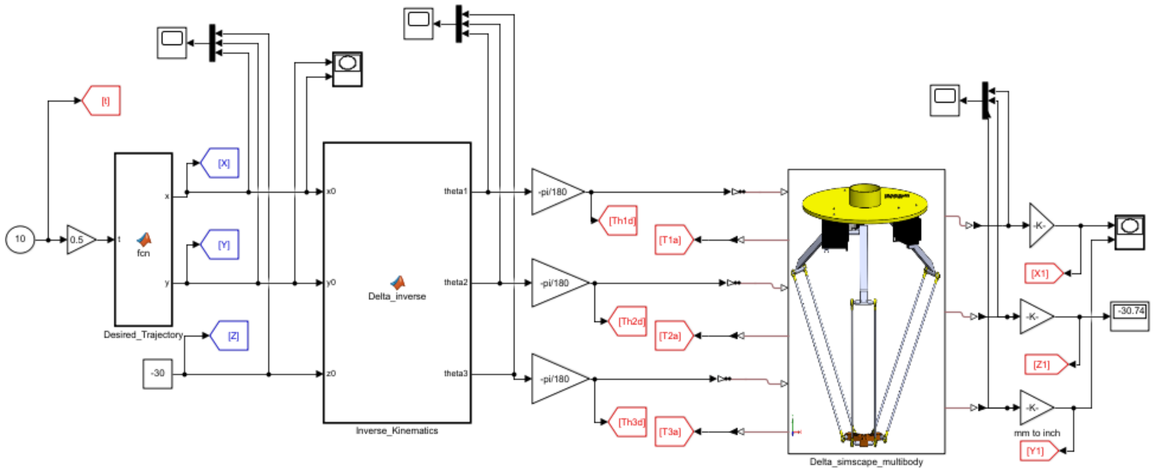


Figure 5. 9: Simulink model for dynamic model of 3-DOF Delta Robot

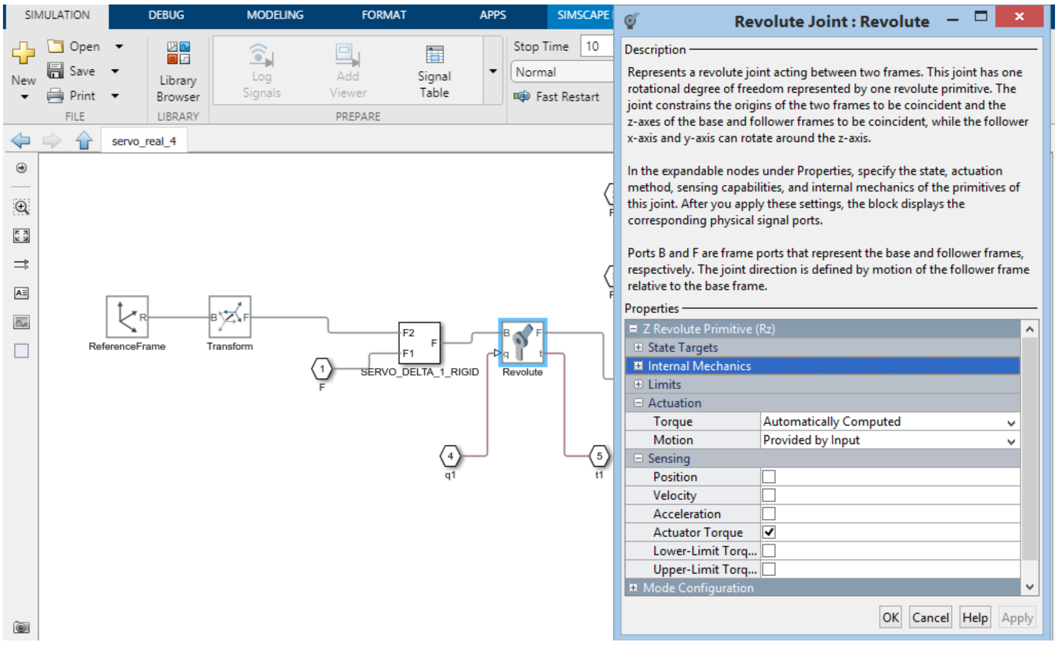


Figure 5. 10: The selected revolute joint actuation feature options on simscape multibody

The model analysis is carried in the absence of disturbance torque. To simulate the dynamic model of Delta robot the actuation option features for torque and motion should be selected appropriately in the simscape multibody model of the robot (in the opposite way to Figure 5.4 (c)). The motion is provided by the input which is a circular path which in turn converted to angular position (Figure 5.15) through delta inverse kinematic calculation. Then the output torque is computed automatically. Figure shows the output computed torque characteristic for the three joints namely joint 1, joint 2 and joint 3.

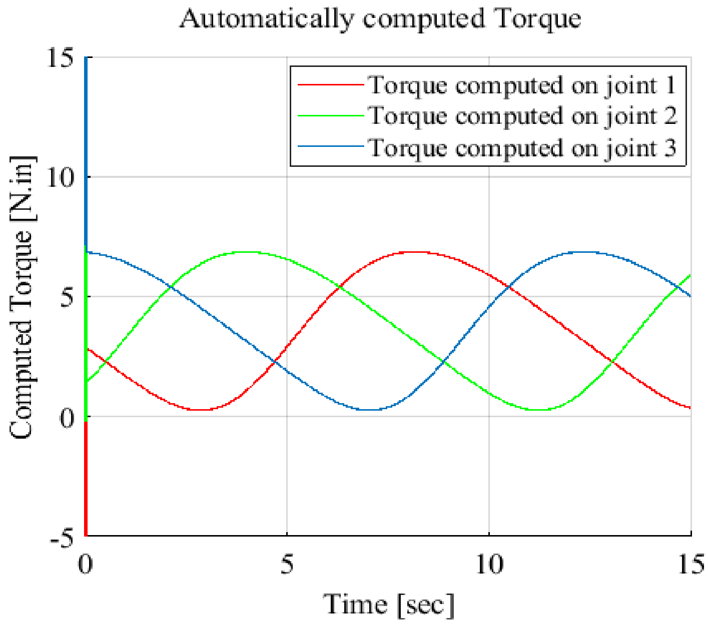


Figure 5. 11: The automatically computed torque for circular input motion

The resulted torque on each actuators has smooth and sinusoidal feature with phase difference of 120^0. The output trajectory obtained from the simscape multibody has similar trajectory. However, the model is responding to the desired trajectory with low accuracy. This is because of the known and unknown model uncertainties. The average error between the two trajectories is about 0.5 in.

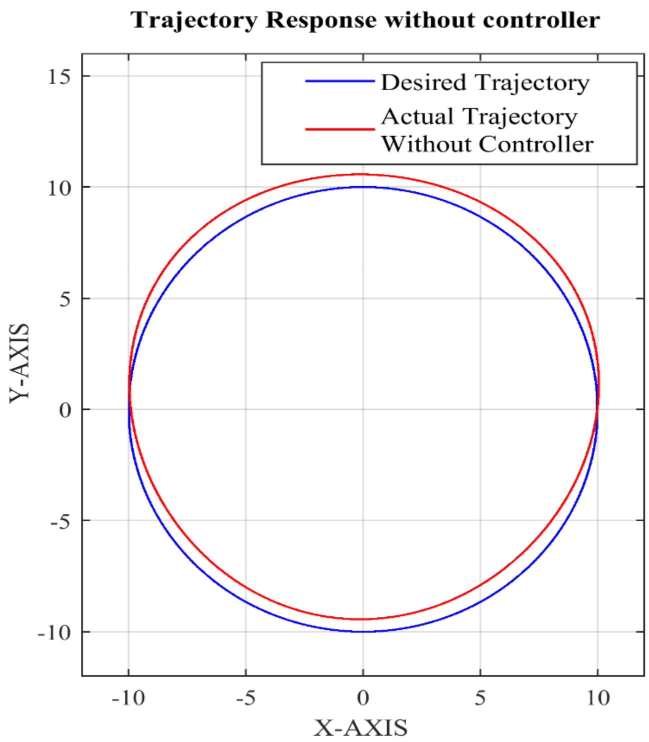


Figure 5. 12: End effector trajectory obtained after the input desired trajectory

5.7 Simulink Block and Designed Control System

There are major sequences of tasks to be followed in order to complete simulation of AFNTSMC control of Delta Robot for circular trajectory tracking. The first step is to build the Solidworks model of the delta robot and generate XML file for the model and export it into Simscape Multibody. And finally, the generated Simscape Multibody model of Delta Robot has been combined with the Simulink model of the control strategy and the rest system.

The block of a Revolute joint in Simscape Multibody primitive has State Targets, Internal Mechanics, limits, Actuation and Sensing fields. Among them Actuation section incorporate torque and motion. Both fields give us the options; none, provided by input and automatically computed. This makes a total of six possible configurations for each joint primitive. From the possible six configuration, Inverse Dynamics and Forward Dynamics are mainly used for Robot's kinematics and dynamics analysis. Motion is determined by input and torque is computed automatically in Inverse Dynamics actuation of revolute joints.

This mode is useful if we know how a joint should move and want to know how much torque or force is required. In case of forward Dynamics, actuation of a revolute joint with a force or torque is specified for the joint primitive and the engine computes the resulting motion. This mode is used in the rest of the analysis in this thesis. MATLAB/ Simulink has been utilized for inverse and forward kinematics, dynamics, Adaptive fast non-singular terminal sliding mode (AFNTSM) controller, design and simulation. Beside the 3-DOF Parallel delta Robot is made to track a circular trajectory constructed in section 5.5 with Simulink's ODE 4 Range-Kutta solver, which is the recommended solver for mechanical systems. Figure 5.13 shows the final setup of the Simulink simulation after exporting the CAD (solidworks) model to Simscape Multibody and integrate it with the rest Simulink block and designed control system.

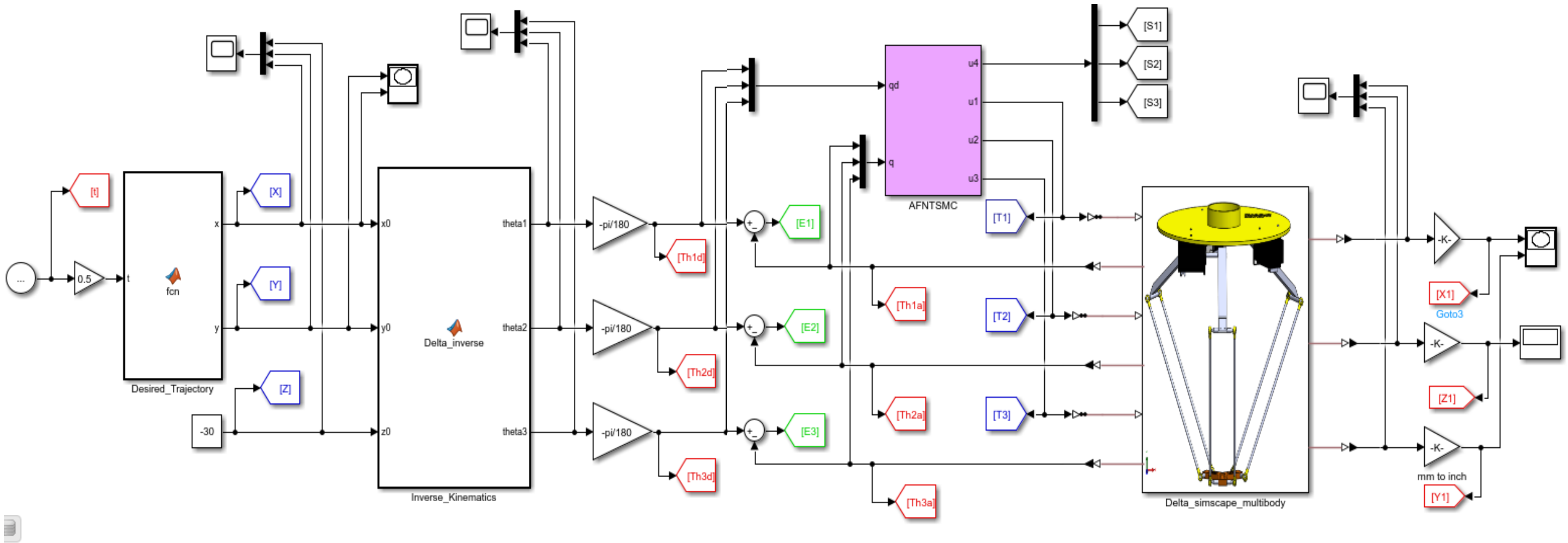


Figure 5. 13: Simulink model of (AFNTSM) controller of Delta Robot for circular path trajectory tracking

In this thesis the desired trajectory is converted into joint angle using inverse kinematic calculation as described in chapter three section 3.3.2 and is applied as input to the controller (ANFTSMC). Then the computed control torque is applied to the robot, forward dynamics mode, so that the motion is automatically sensed from each of the revolute joint primitive. The actual joint angle is then compared with the desired and fed to the designed controller so as to adjust the required control torque accordingly. Simulink multibody has eased a complex computation of the required torque to track or accomplish a given task or motion to be achieved which mainly involves large mathematical analysis and singularity in computation. Whereas in this scenario not only the specific value of torque or motion is required to meet control goal but also the dynamic constants such as: Inertial matrix $M(q)$, the Centrifugal and Coriolis forces matrix $C(q,\dot{q})$ and Gravity $G(q)$ as well. Which in return require computation of Jacobean matrix.

MATLAB displays control results such as joint angle error, required control torque and reference tracking. Whereas the Mechanics Explorer displays the Robot's motion in 3D space during simulation. The entire simulation results are illustrated in Figures below and the simulation is split into two sections.

5.8 Simulation Results

To verify the improved performance of the designed strategy in overcoming the outstanding issues of SMC as well as to demonstrate the effectiveness and applicability of the proposed control method, a 3-DOF Delta Robot as shown in Figure 5.14 is employed and its essential parameters are reported in Table 5.1. The CAD model of the robot is designed using Solidworks (2021 version) and the simulations are performed in the MATLAB (2021 version) simulink environment using ODES 4, Runge-Kutta with a fixed-step size of 0.001s.

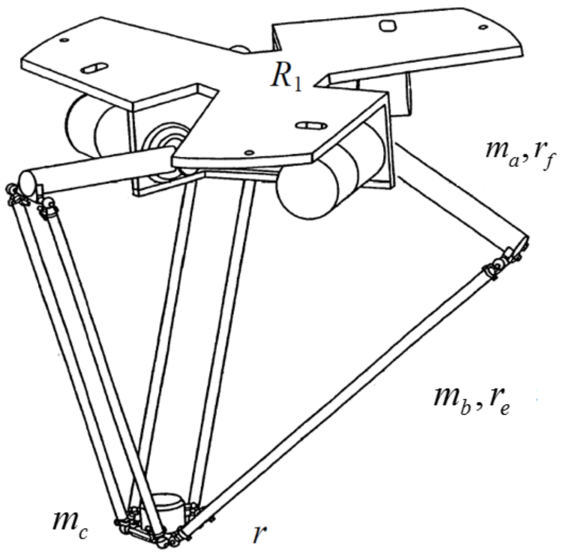


Figure 5. 14: Delta robot model (Quang et al., 2020)

The nominal parameters for the system are selected as:

Table 5. 1: The essential parameters of the delta robot system.

Parameters	**Value**	**Unit**
m_a	0.09	Kg
m_b	0.12	Kg
m_c	0.05	Kg
e	7.79	in
f	27.30	in
r_e	26.79	in
r_f	14.50	in
g	9.81	Kg.m/s
a	14.5	in

The reference trajectories are chosen as

$$p_d = \begin{bmatrix} x_{1d}(t) \\ y_{2d}(t) \\ z_{3d}(t) \end{bmatrix} = \begin{bmatrix} 10\sin(\pi t) \\ 10\sin(\pi t) \\ -30 \end{bmatrix}$$

The external disturbances are assumed to be time-varying as

$$\tau_d(t) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} = \begin{bmatrix} 26\sin(2.5\pi t) + 16\cos(5\pi t) \\ 21\cos(5\pi t) + 21\sin(2.5\pi t) \\ 19\sin(2.5\pi t) + 27\cos(5\pi t) \end{bmatrix}$$

Damping factor is assumed to be 1.

The control strategy, have the corresponding control torque as follows:

$$\tau(t) = \tau_{eq}(t) + \tau_{asw}(t)$$

$$\tau_{eq}(t) = M_0(q)\ddot{q}_d + C_0(q,\dot{q})\dot{q} + G_0(q) - \frac{M_0(q)}{\beta.k_2}|e_2|^{2-\beta}(1+\alpha.k_1|e_1|^{\alpha-1})\tanh\left(\frac{e_2}{\varepsilon}\right)$$

$$\tau_{asw}(t) = -M_0(q)\left[k.s + (\hat{b}_0 + \hat{b}_1|q| + \hat{b}_2|\dot{q}|^2 + \eta)\tanh\left(\frac{s}{\varepsilon}\right)\right]$$

The parameters of the proposed controller are selected as

$$\alpha = 2,\ \beta = 1.8,\ \eta = 1.3,\ k_1 = 9, k_2 = 1, k = 250.$$

The constant values of adaptive update laws are $\lambda_0 = 0.01, \lambda_1 = 0.01, \lambda_2 = 0.01,$ and its initial conditions are selected as $\hat{b}_0(0) = 0, \hat{b}_1(0) = 0, \hat{b}_1(0) = 0,$

Then, the developed adaptive approach in Theorem 2 is utilized to control the system in the presence of completely unknown upper bounds.

5.8.1 Simulation Results for ANFTSMC

Even though the robustness of the designed controller, AFNTSMC, has been proofed in chapter four, the simulation result by adding external disturbance on the model are shown below. The 3-dof Delta Robot inverse kinematics problem is computed given the Cartesian position of the desired path trajectory to find the required actuated revolute joint angels θ_1, θ_2 and θ_3. The values of the joint angles of θ_1, θ_2 and θ_3 are vital in determining the exact position of the end effector. Figure 5.15 shows the required angle generated for the three active joints to track the desired circular path trajectory, which is a circle on X-Y plane centered at $(0, 0, Z_0)$ with 10 in radius.

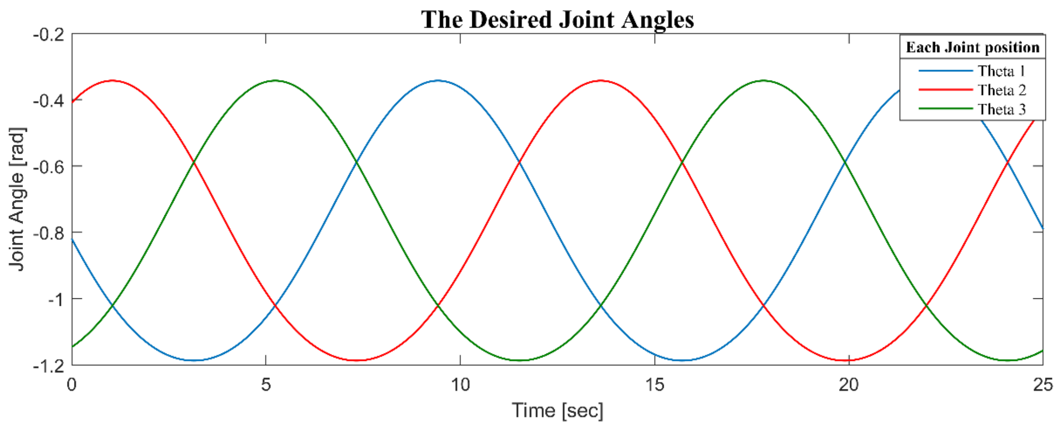
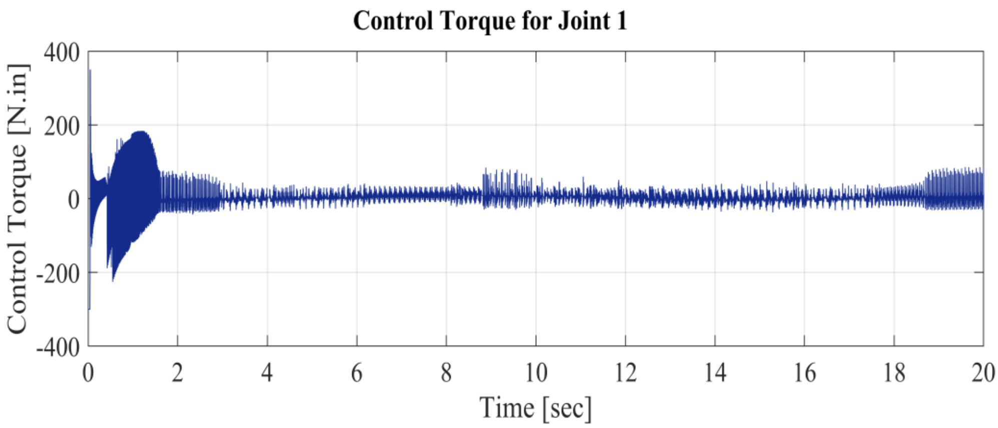
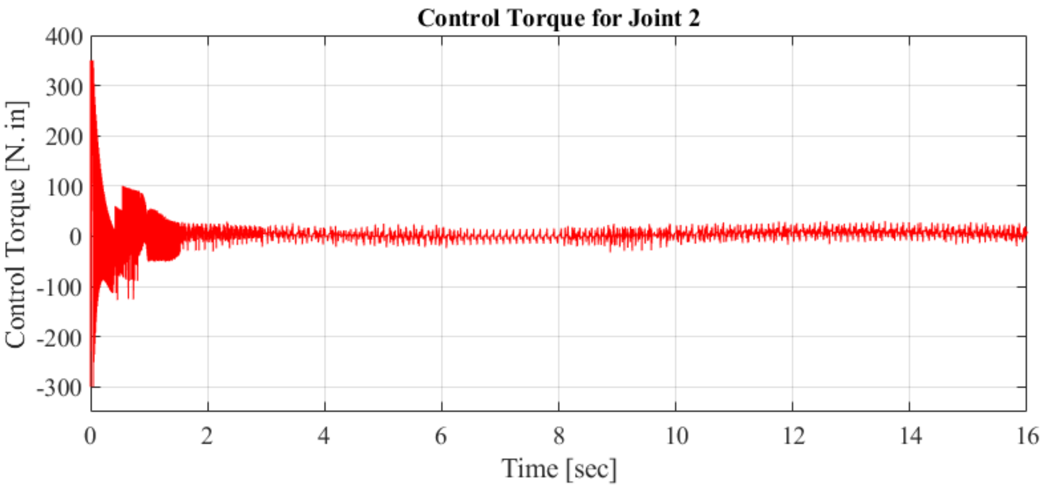


Figure 5. 15: Generated joint angles.

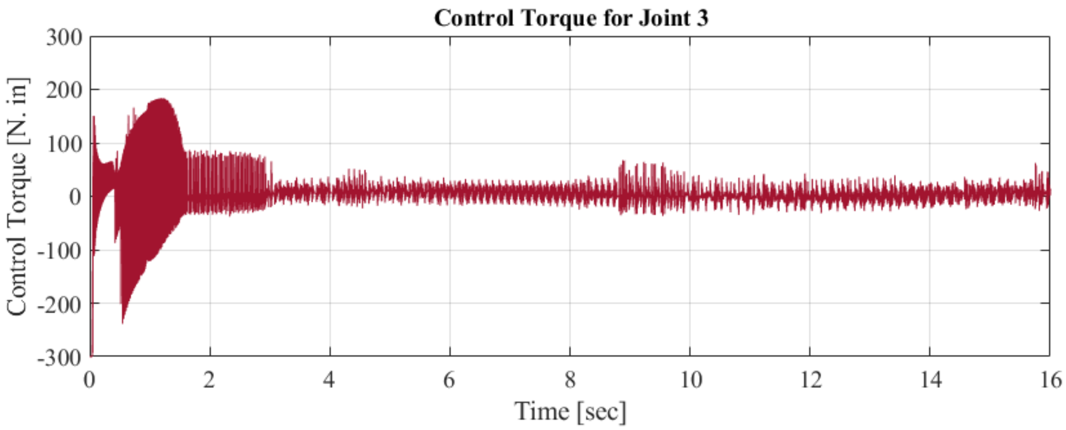
The calculated torque imposed on each active joint of the three DOF Delta Robot and then actual position of the end effector is sensed. Which in return fed back to the ANFTSMC against the desired position to get the controlled torque. If the strategy is effective, this controlled torque should enforce the system to be in stable state and the end effector must track the desired circular trajectory within acceptable time. Figure 5.16 shows the required control torques needed to achieve the desired circular trajectory for each of the three active revolute joints.



(a)



(b)



(c)

Figure 5. 16: Required ANFTSMC Control Torques for the three active revolute joints.

The following set of plots shows the end effector trajectory tracking along with its respective error between its desired and actual position. Beside end effector trajectory error on X-axis and Y-axis are provided to readily determine the severity of the accuracy.

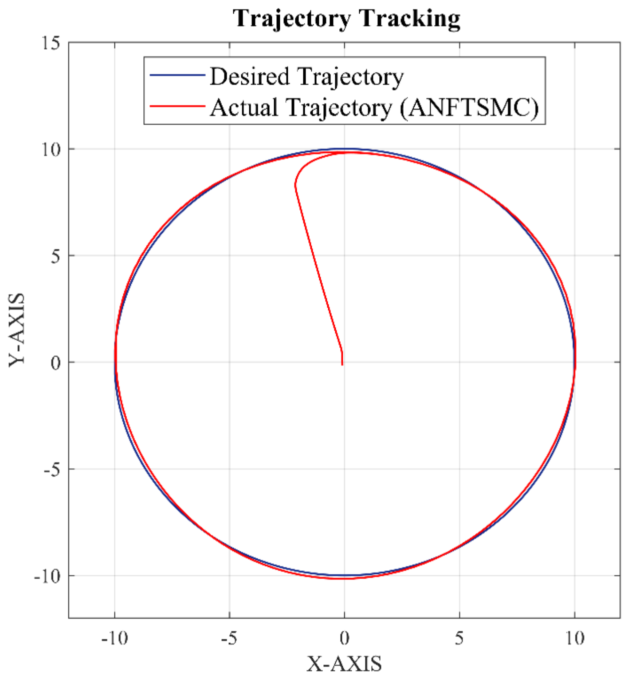


Figure 5. 17: End effector trajectory tracking

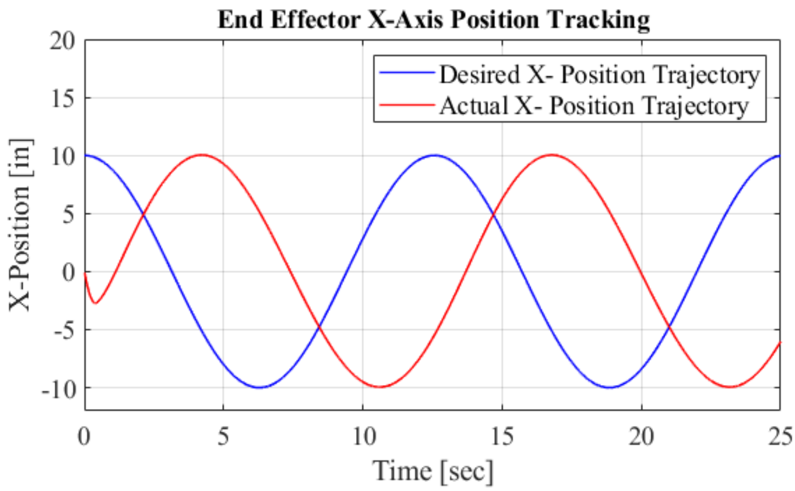


Figure 5. 18: End effector trajectory error on X-axis

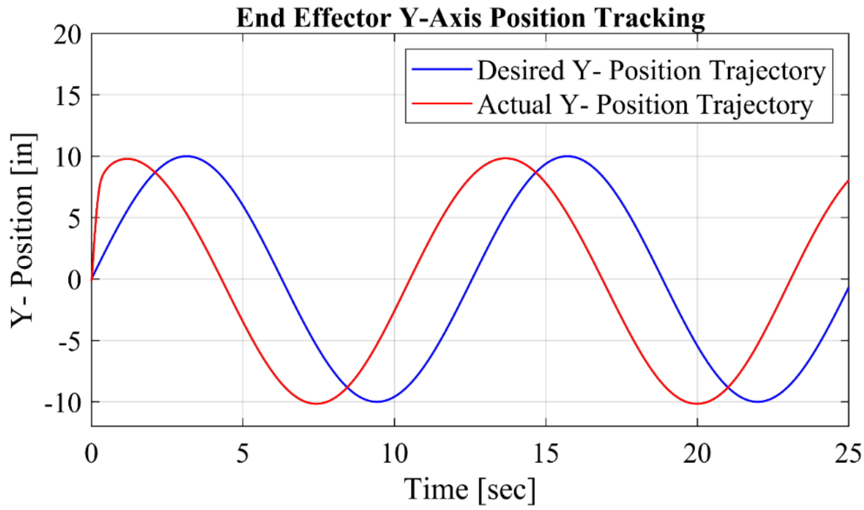


Figure 5. 19: End effector trajectory error on Y-axis

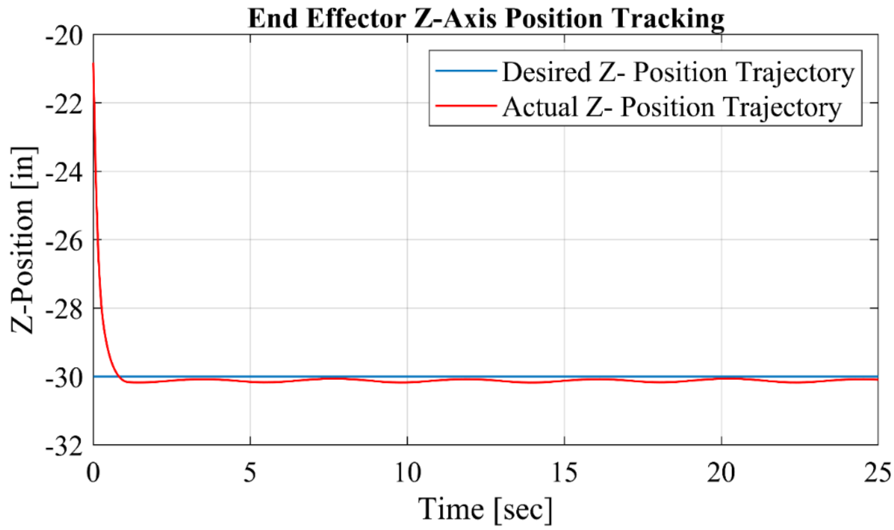
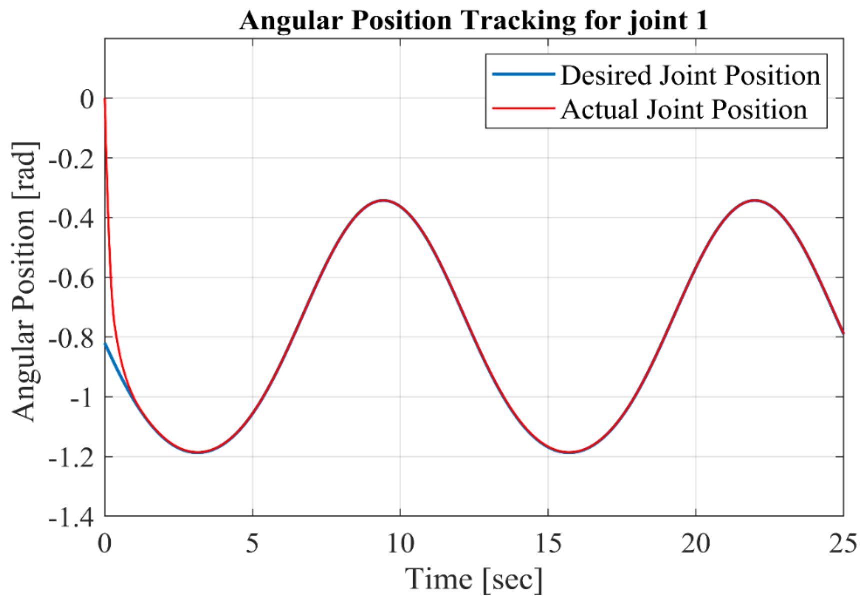
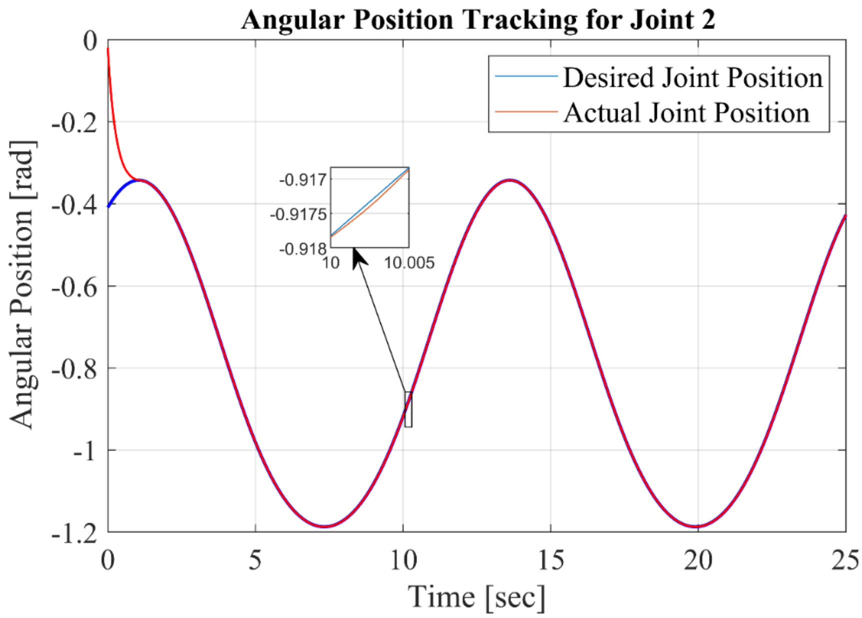


Figure 5. 20: End effector trajectory error on Z-axis

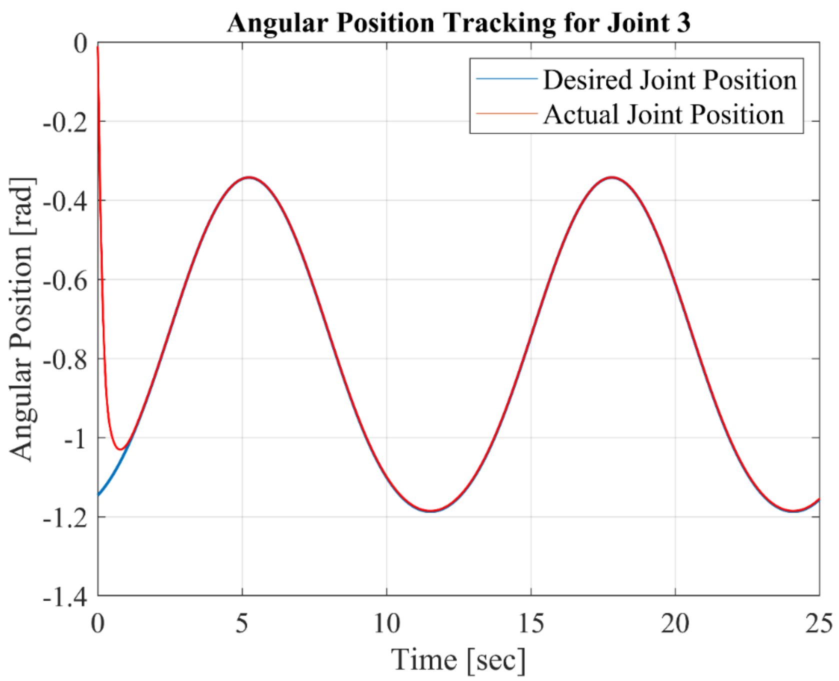
As the Fig 5.21 (a), (b) & (c) below shows the designed ANFSMC controller has made the system insensitive to external disturbance, internal mechanics and parameter variation by absolutely tracking the desired joint angles in less than 2 seconds (at around 1.3 sec). The joint angles for the arms follow the desired trajectory in finite time as required from the design of ANFSMC control law.


Angular Position Tracking for joint 1
Desired Joint Position
Actual Joint Position
Angular Position [rad]
Time [sec]
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
0
5
10
15
20
25

(a)


Angular Position Tracking for Joint 2
Desired Joint Position
Actual Joint Position
Angular Position [rad]
Time [sec]
-0.917
-0.9175
-0.918
10
10.005
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
5
10
15
20
25

(b)



(c)

Figure 5. 21: Joint position tracking for joint 1(a), joint 2(b) and joint 3(c)

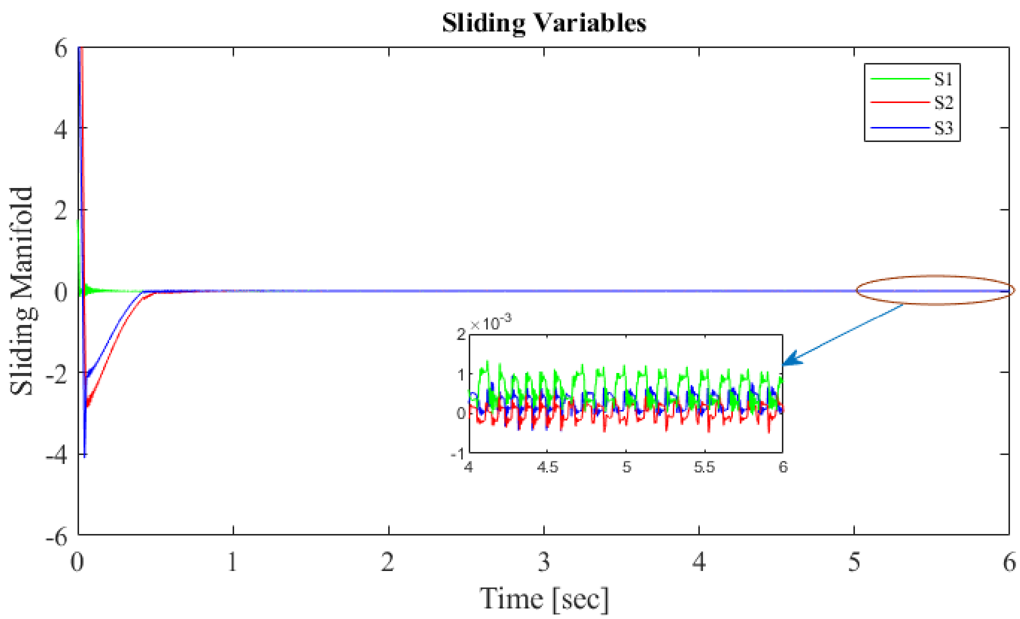


Figure 5. 22: The sliding variables s_1, s_2 and s_3 for each joint

Figure 5.22 shows the sliding variables s_1, s_2 and s_3 for each joint angle. The sliding variables settles to zero before 1 second (at about 0.5 sec.).

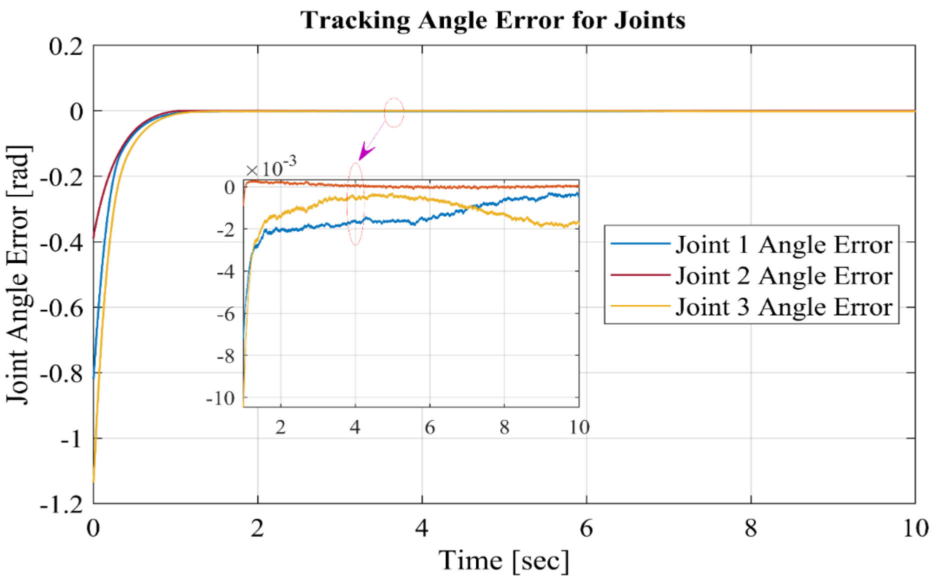


Figure 5. 23: Joint angle errors

Figure 5.23 shows angle error of the three active joints of the Delta Robot in response to the controlled torque.

The joint angle error, end effector trajectory and end effector trajectory error on X and Y-axis figures illustrate that the designed controller, ANFTSMC has made the error to converge in about 1 second with joint angle root mean square error of 0.002 rad and end effector trajectory rms error of 0.01 in on X-axis and Y-axis.

5.8.2 Comparative simulation

Finally, a comparative study between the traditional sliding mode approaches and the proposed adaptive algorithm is presented. Control parameters of different control systems, including SMC and the proposed controller are shown in Table 2.

Table 5. 2: Control parameters of different control systems

Control Strategy	Control Parameters	Value of Control Parameters
SMC	Λ, K_{pd}, K_s	*diag (4,4,4), diag (250,250,250), diag (150,150,250)*
Proposed Control	α , β ,η, k_1 , k_2 , k λ_0 , λ_1, λ_2	1,1.8,1.5,9,1,250,0.01, 0.01, 0.01

Simulation results are shown in Figure 5.24, 5.25 and 5.26. Figure 5.24 exhibits the prescribed path and actual path of end-effector under two different control strategies, including SMC and ANFTSMC.

The end-effector of the robotic system is controlled to follow a circular path. As seen in figure 5.24, it is seen that ANFTSMC seem to provide a good trajectory tracking performance with faster response as compared to CSMC.

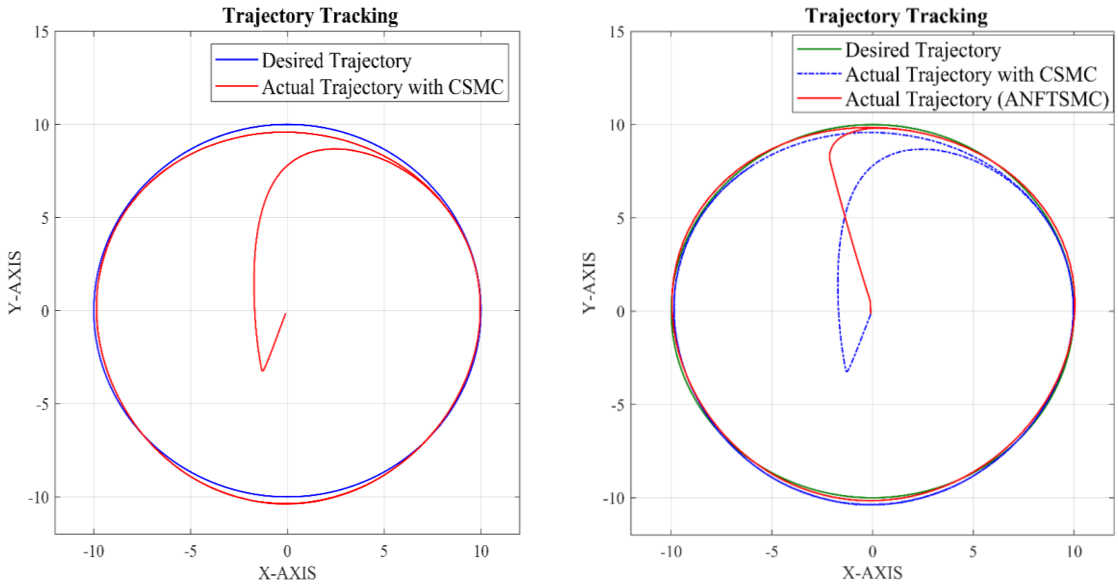
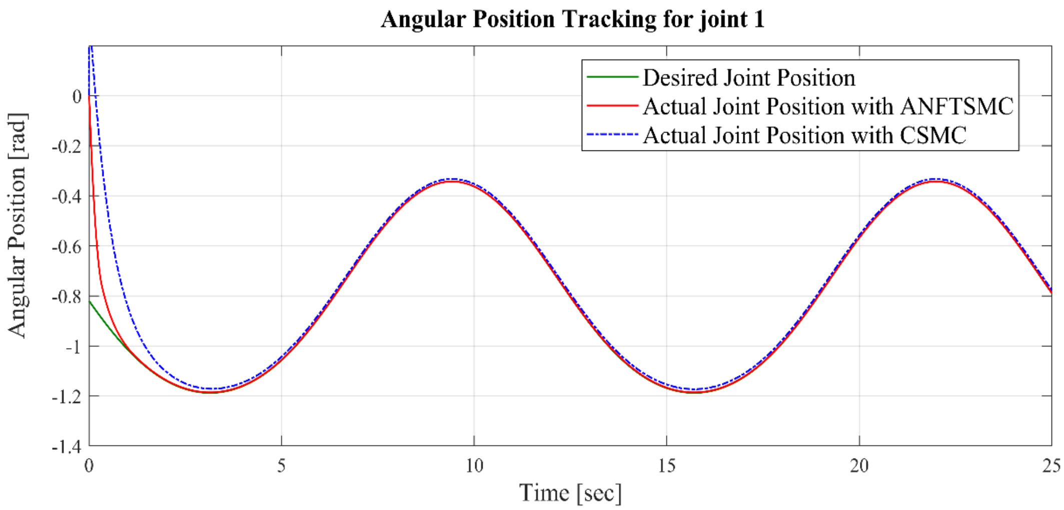
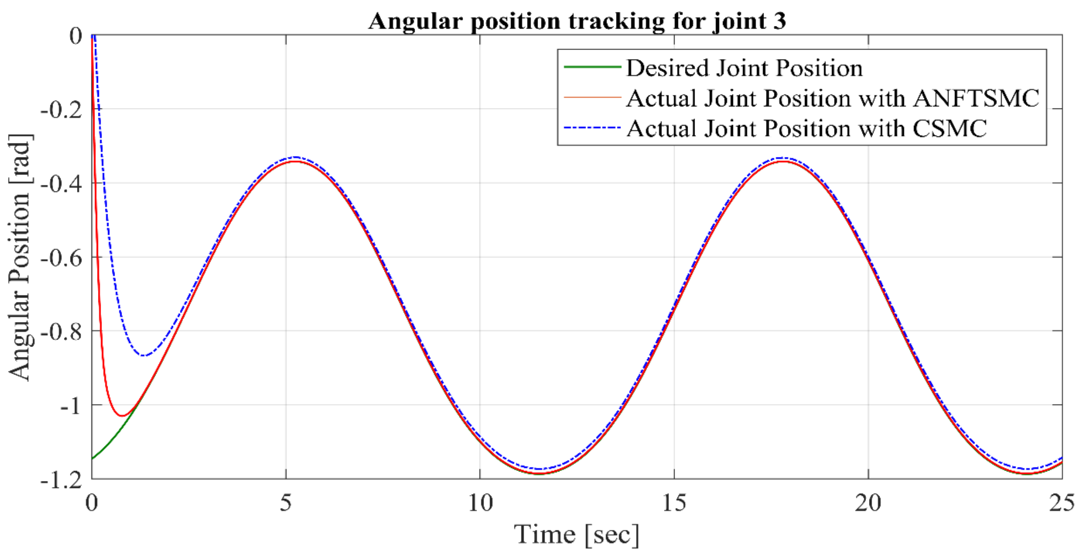


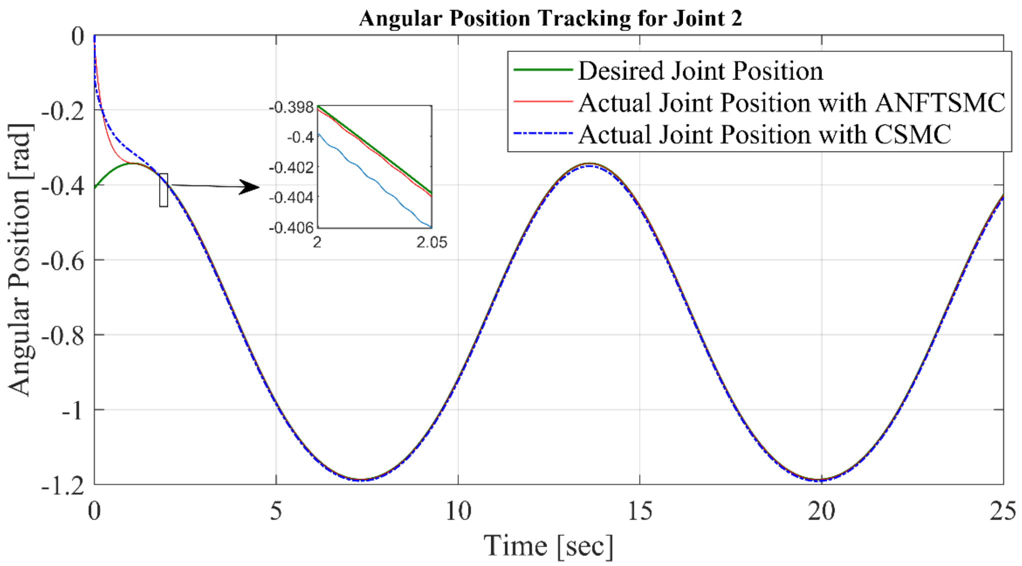
Figure 5. 24: The Desired path and actual path of the end-effector under the two different control methods.



(a)



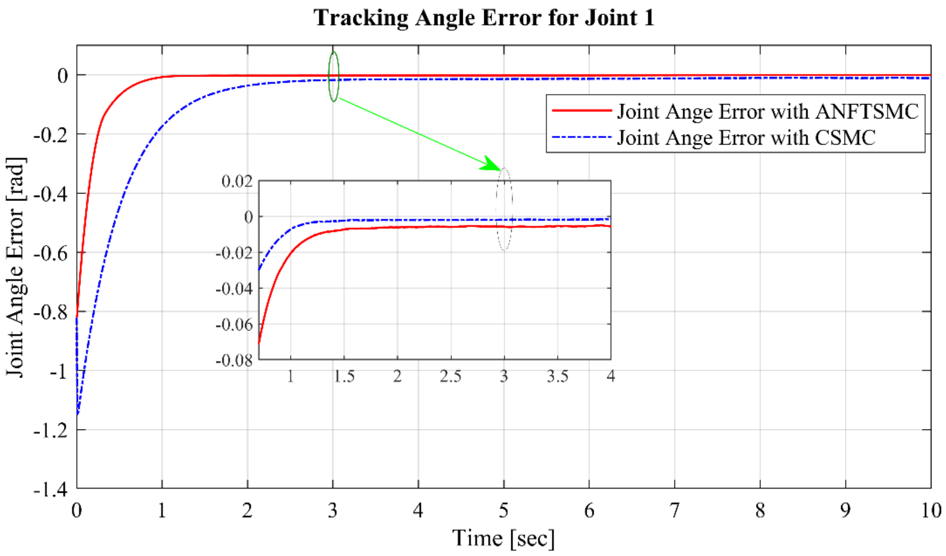
(b)



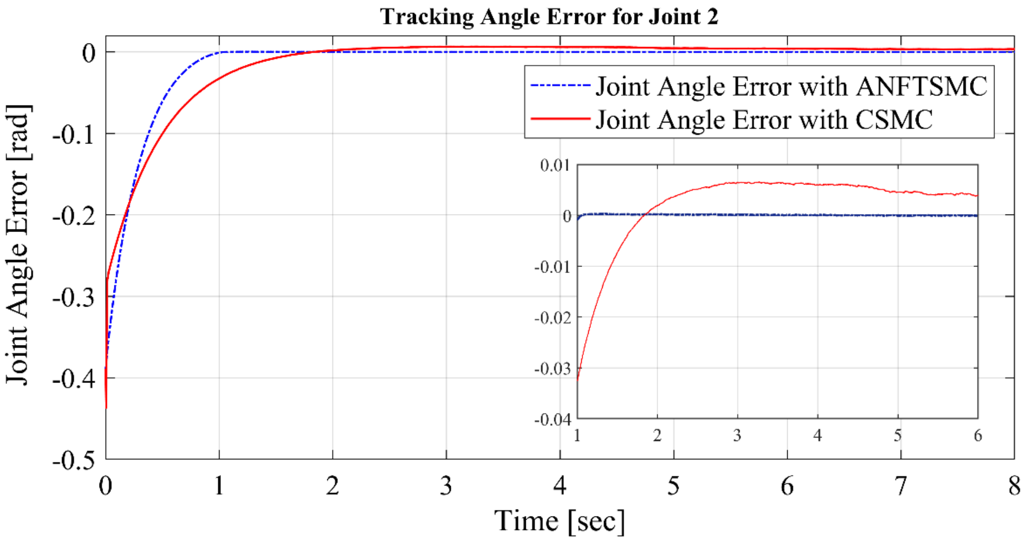
(c)

Figure 5. 25: Joint position tracking for joint 1(a), joint 2(b) and joint 3(c) with ANFTSMC and CSMC

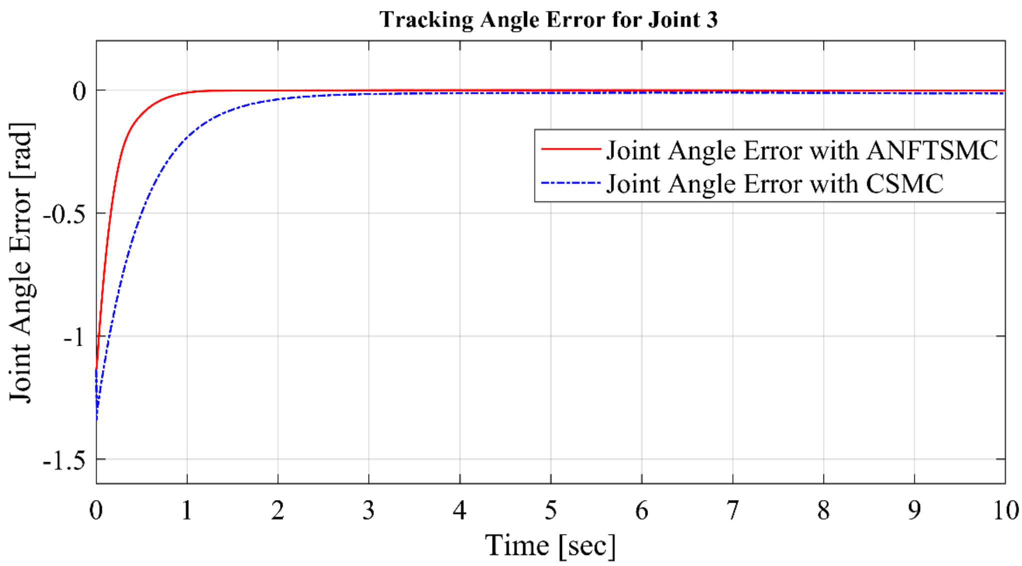
Figure 5.26 exhibit the Angle (position) tracking errors of the end effector. As seen on the figure ANFTSMC ensures fast convergence rate. The error converges faster to zero for ANFTSMC than for CSMC.


Tracking Angle Error for Joint 1
Joint Ange Error with ANFTSMC
Joint Ange Error with CSMC
Joint Angle Error [rad]
Time [sec]

(c)


Tracking Angle Error for Joint 2
Joint Angle Error with ANFTSMC
Joint Angle Error with CSMC
Joint Angle Error [rad]
Time [sec]

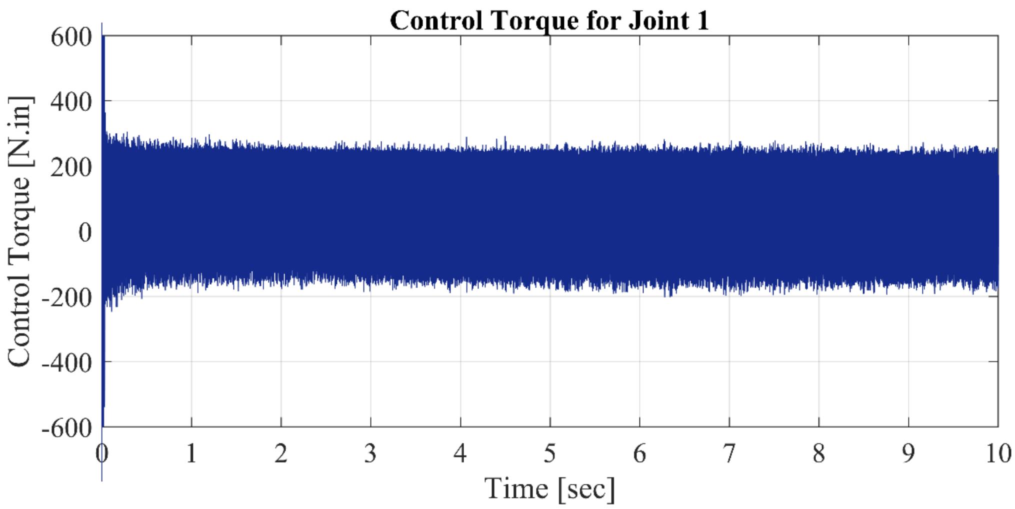
(b)



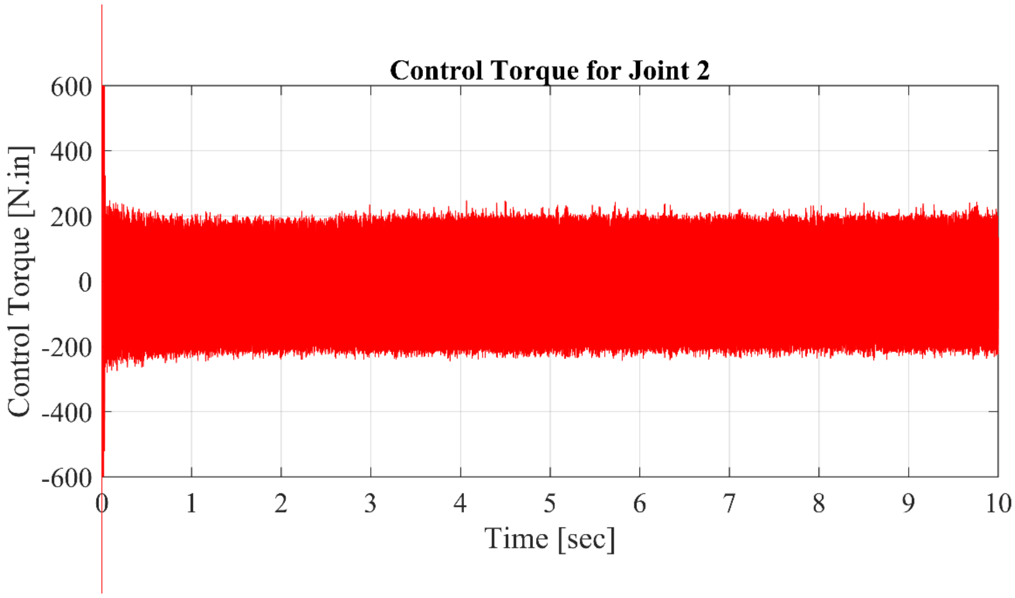
(c)

Figure 5. 26: Joint angle tracking errors for joint 1(a), joint 2(b) and joint 3(c) with ANFTSMC and CSMC

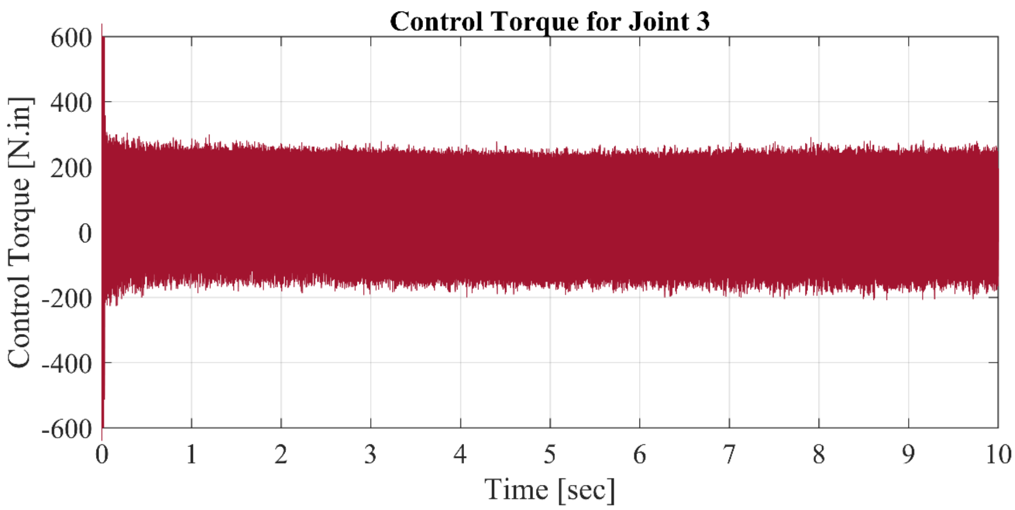
SMC provides the worst tracking performance among the two control methodologies, the tracking errors produced by ANFTSMC are smaller than the tracking errors offered by SMC. Specifically, the proposed control strategy offers the smallest tracking errors compared with.



(a)



(b)



(c)

Figure 5. 27: Control input actions for joint 1(a), joint 2(b) and joint 3(c) with CSMC

The control signals for CSMC control strategy are illustrated in Figure 5.27. In the figure CSMC show a discontinuous control signal with rated torque of 194 N.in and serious chattering. On the contrary, the proposed control strategy seems to show a continuous control signal with impressive small chattering behavior, as illustrated in Figure 5.18.

From the simulation results, it can be concluded that the proposed strategy shows the best performance than CSMC in terms of tracking positional accuracy, small steady state error, fast response speed, and weak chattering behavior.

5.9 The performance of the controller with difficult trajectories

Due to the complexity of manipulator robots, the trajectory tracking task is very challenging. The performance of the proposed controller is well analyzed and discussed for a circular trajectory. However, to evaluate the efficiency of the tracking scheme performance of the proposed controller have been simulated for more difficult trajectories such as number '8' on Simulink to test the performance of the controller. Figures 5.28 shows the matlab code for some trajectories.

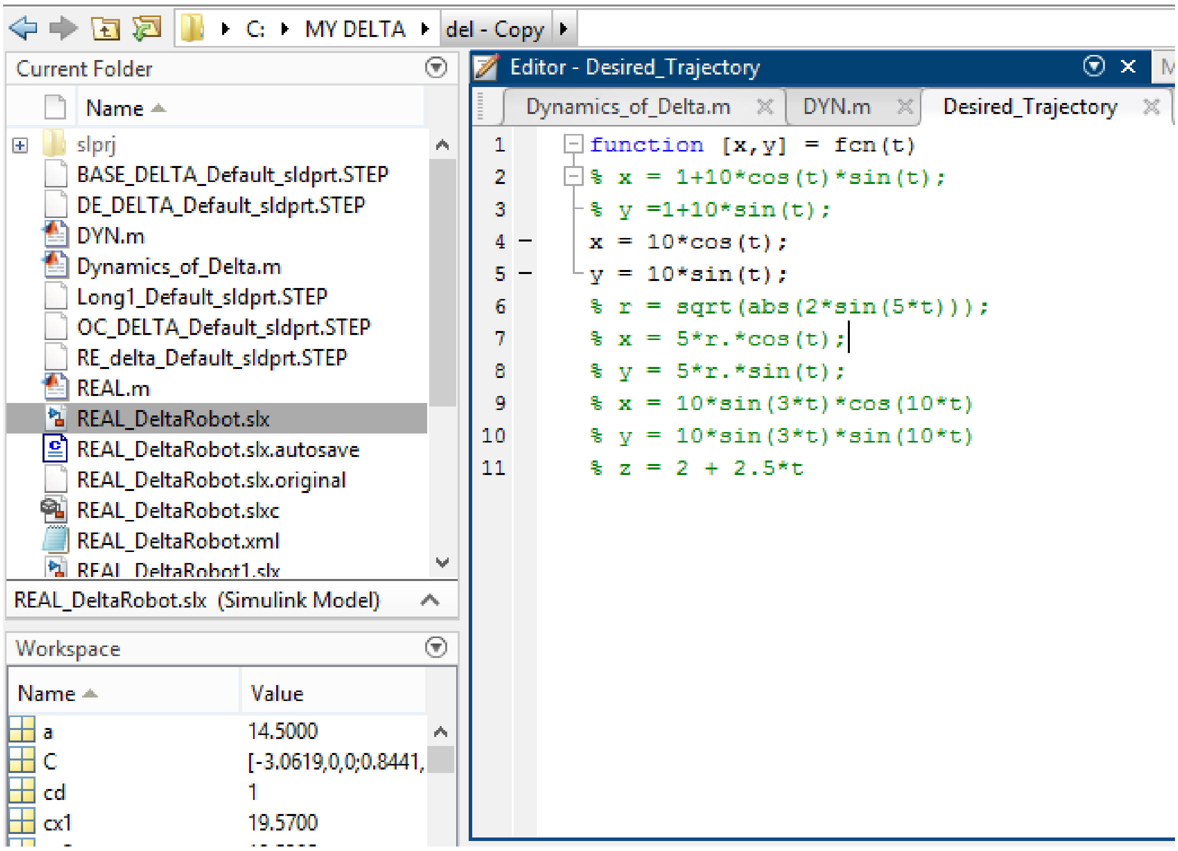


Figure 5. 28: Matlab program for different desired trajectories

Figures 5.29 and 5.30 shows the graphs and the instantaneous virtual motions of the delta robot in 4 different views while it is tracking the desired trajectory. The upper and the lower graph in both figures shows the desired and the actual trajectory respectively. The simulation result shows that the steady state tracking is reached in less than 2 seconds for different trajectories and X and Y rms error of 0.01 in.

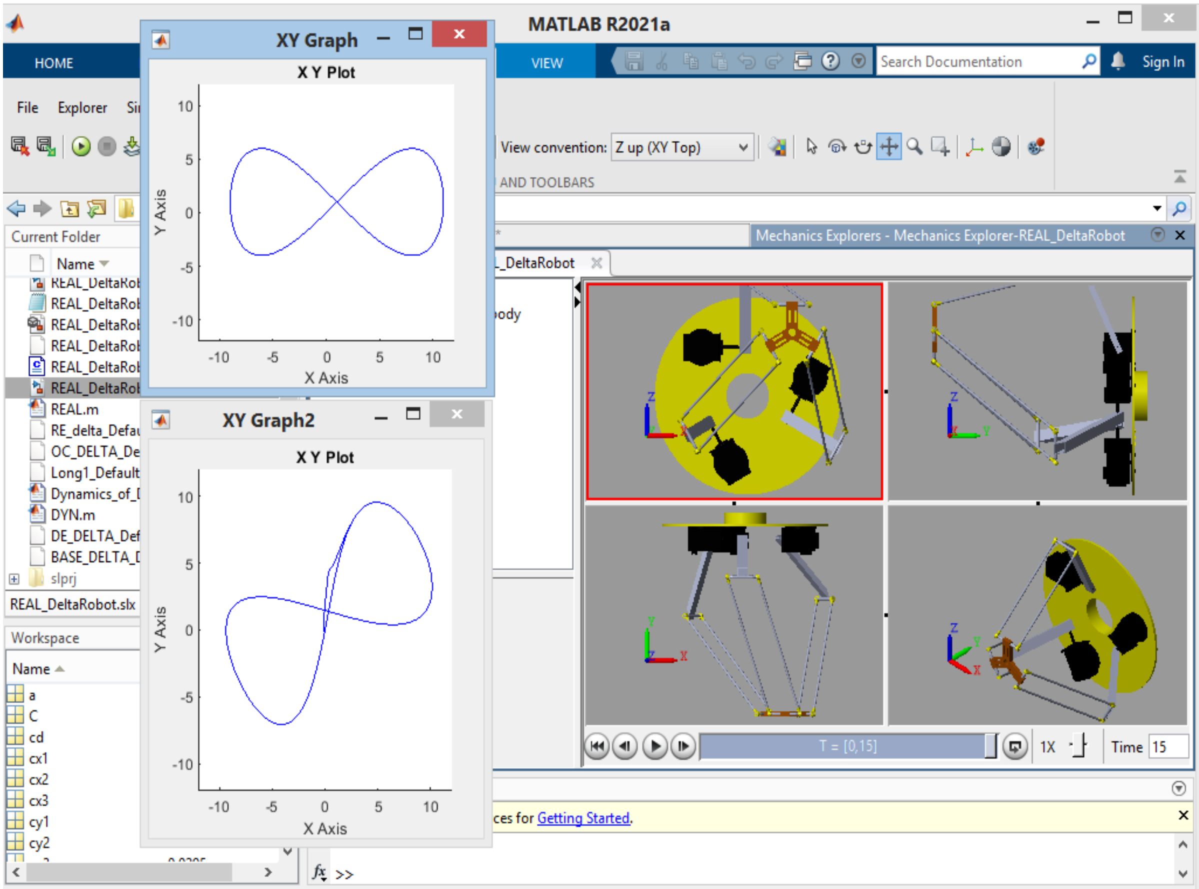


Figure 5. 29: End effector trajectory tracking for number 8

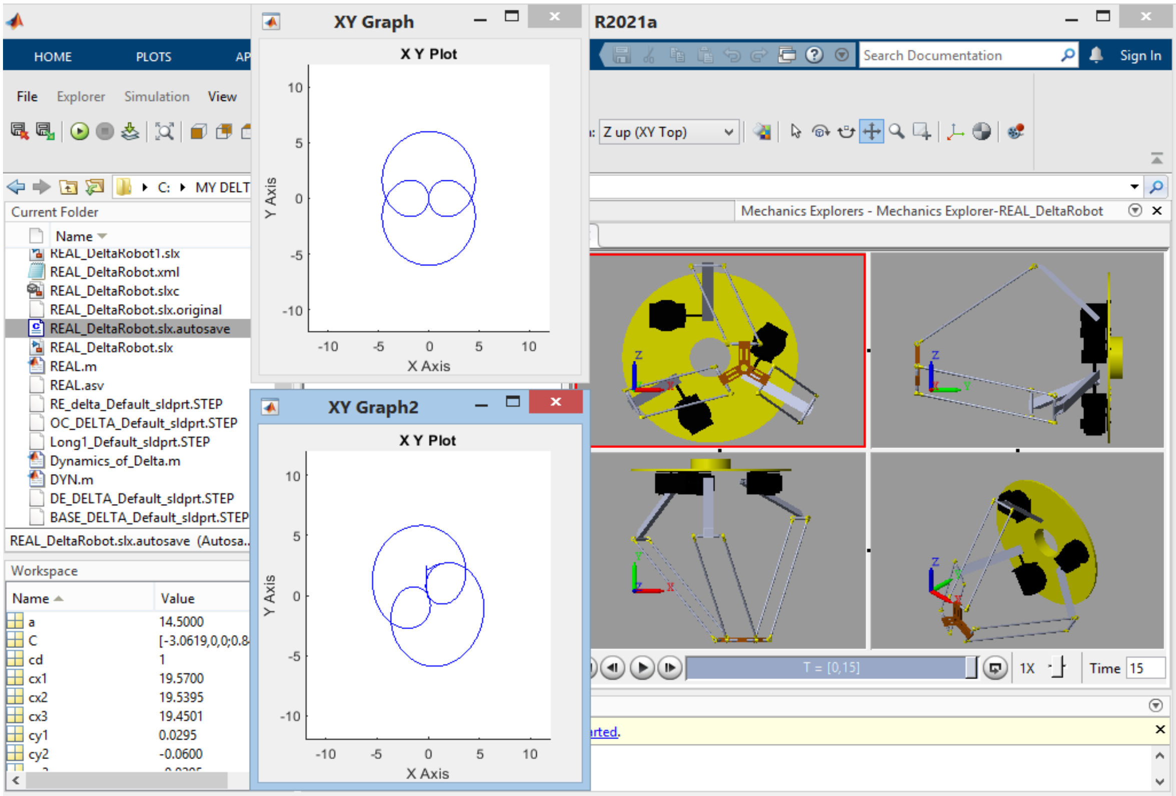


Figure 5. 30: End effector trajectory tracking for more difficult trajectory

5.10 Discussion

The tracking performances under uncertainties and a time-varying external disturbances are illustrated in Figure 5.17-5.21. From the figures, we can see that the system converges to the desired trajectories quickly in a finite time, less than two seconds and achieves good tracking performance. Furthermore, the position tracking errors tend to zero after a short transient due to errors in initial conditions. From Figure 5.16, one can see that the control torque is with a little chattering. Figure 5.22 shows the time responses of the sliding surfaces that confirms the fast convergence of the system. Thus, it is concluded that the ANFTSMC is of high performance, such as high tracking precision, fast response, singularity avoidance and strong robustness to external disturbances and modeling uncertainties. As Figure 5.23 shows ANFTSMC has made the error to converge with joint angle rms error of 0.002 rad and end effector trajectory rms error of 0.01 in on X and Y-axis irrespective of the applied external disturbance torque, internal mechanics and mass uncertainty.

However, the problem of ANFTSMC is its model dependence. In order to apply the designed control strategy, the exact value of the dynamic parameters such as: Inertial matrix $M(q)$, the Centrifugal and Coriolis forces matrix $C(q,\dot{q})$ and Gravity $G(q)$ as well must be known explicitly. Thus, it requires computation of matrixes, Jacobean matrix. So, we can say that somehow it fades one of the benefits can be gained from Simulink Multibody and makes the work a bit challenging.

One of the cons of designing robot model on CAD (Solidworks) environment and export it into Simulink Multibody is to ease a complex computation carried out to obtain the required output and input that can be position, angles, forces, or torque which mainly involves large mathematical analysis and singularity in its computation. Solidworks models can be exported and simulated in the Simulink environment to evaluate forces and torques in mechanical joints, plot accelerations and displacements of each component of the system, run various tests, and acquire needed output without any necessity of motion equations derivation and Simscape physical modeling of the mechanical system.

Simulink Multibody tool introduced an easy way to actuate robot joints by joint actuator block. Sensing computed torques, angles, angular speeds, and accelerations are possible by joint sensor. Beside Bod sensor is used to get the Cartesian Coordinates of the moving platform. Thus, it facilitates the overall simulation process.

The robot (3-DOF Delta robot) real but not approximated parameters have been introduced via CAD dynamic modeling which is one of the advantages of CAD (Solidworks) modeling environment. On the hand, the simulation speed is low. The simulation speed of mathematical dynamic model is more rapid than CAD dynamic model.

Early SMC theory used continuous controllers with a continuous function (e.g., hyperbolic or saturation) in the vicinity of the sliding surface. However, zero tracking error is not ensured within a finite time. Furthermore, usually there is a direct trade-off between chattering and robust performance i.e., chattering reduction may lead to reduced robustness.

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

This thesis is mainly concerned with dynamic modeling and control of a trajectory tracking of a 3-DOF delta robot. The desired circular trajectory, which is a circle on xy-plane centered at (0, 0, -30) with 10 in radius has been tracked. The methodologies of dynamic modeling of a complex multibody dynamic system have been presented using SimScape multibody.

In this thesis, an adaptive control law is proposed for the problem of trajectory tracking under uncertainties and external disturbances. This adaptive law is derived based on the advantages of the NFTSM concept and can drive system states to the origin within a prescribed finite-time. However, not only fast convergence is achieved when the system state is far away from the equilibrium, but also the singularity problem of TSM is avoided and evidence of finite-time convergence was sufficiently demonstrated by Lyapunov theory. A particular attention is paid to the online estimation of unknown upper bound of the system uncertainty which often occur in practical applications. As a result, the upper bounds of the uncertainties are not needed in the procedure of the controller design, and the controller is continuous, which produces a smooth control signal with less chattering phenomenon. To demonstrate the effectiveness of the proposed control strategy, simulations have been carried out in the cases of uncertainties, time-varying external disturbances.

The numerical Simulation in Matlab/ Simulink of a 3-DOF delta parallel robot with rigid links using SimScape Multibody in the cases of uncertainties, time-varying external disturbances has been investigated and examined for Circular Trajectory tracking using ANFTSMC, which provides valuable control of the robot. The simulation result shows that the steady state tracking is reached in less than 2 seconds for a circular trajectory and x and y rms error of 0.01 in.

The developed methods and strategies of dynamic modeling and position control can be extended to other types of parallel manipulators or other multibody dynamic systems with rigid components. In this thesis

From theoretical evidence, simulation results, and comparison with SMC, the proposed control strategy shows a significant improvement in control performance compared to the CSMC methods:

- It offers finite-time convergence and faster transient performance without singularity problem in controlling.
- Inherits the advantages of ANFTSMC in the aspects of robustness against system uncertainties and exterior disturbances as well as its fast convergence.
- It offers the control torque commands with lesser chattering.
- To sum up, the designed controller has proven to be efficient and feasible for trajectory tracking control of robotic systems.

6.1 Recommendation and Future Work

As we all know there are many tasks that require an application of high-speed manipulation. To consider at least one of the long year's manual efforts of most mothers in our country which is making injera. Injera making can be eased if proper manipulator robots are used. The end effector of the robot supported with camera vision can be made to follow a planned spiral trajectory to replace the human arm. Thus, I recommend the implementation of delta parallel robot due to their high speed and stiffness for this application. An adaptive control scheme such as ANFTSMC is used for implementation to account for a variable mass on the end effector. Simulation result shows that, there exists little chattering in control torque. To eliminate this chattering effect the use of delay function or observer-based estimation is recommended.

As many interesting and unsolved problems deserve researchers to invest significant efforts in the future to address, and hence promote the practical application of lightweight delta parallel manipulators. Reviewing the work done in this thesis, one of the important research efforts for future work is that, in the Modeling and Control developed in this thesis, the dynamics of motors and ball screw mechanisms are neglected. To establish more accurate dynamics Model of the delta parallel manipulator, motor dynamics and dynamics from ball screw mechanisms should be included, and the effect of Control system dynamics with other component dynamics should be considered and examined as well.

The other problem is that for high accuracy applications of the delta robot the effect of an inherent vibration should be avoided. Thus, other strategies could be used for suppression of the unwanted vibration, for example, controlling the joint motors with input shapes, vibration suppression with PZT actuators and sensors etc. An important part of our future work will be the experimental evaluation of our method by a real robotic system.

Another possible extension of this work may be the design of a time varying NFTSM control technique to eliminate the reaching phase and optimize the control effort.

REFERENCES

Aronson, R. B. (1996). A bright horizon for machine tool technology. *Manufacturing Engineering*, *116*(3), 57.

Campos, A., & Moratelli, Y. D. (2020). *Graphical Optimization for a Parallel Robot Rotation Based on Platform Initial Orientation BT - Advances in Automation and Robotics Research* (A. Martínez, H. A. Moreno, I. G. Carrera, A. Campos, & J. Baca (Eds.); pp. 40–50). Springer International Publishing.

Cheong, J. Y., Han, S. I., & Lee, J. M. (2013). Adaptive fuzzy dynamic surface sliding mode position control for a robot manipulator with friction and deadzone. *Mathematical Problems in Engineering*, *2013*.

Clavel, R. (1991). *Conception d'un robot parallel rapide á 4 degrés de liberté Ph. D*. thesis, EPFL, Lausanne.

Corapsiz, M. R., & Kahveci, H. (2017). The speed control of DC motor under the load condition using PI and PID controllers. *AIP Conference Proceedings*, *1833*(1), 20116.

Dachang, Z., Pengcheng, H., Baolin, D., & Puchen, Z. (2023). Adaptive nonsingular terminal sliding mode control of robot manipulator based on contour error compensation. *Scientific Reports*, *13*(1), 330.

de Jesús Rubio, J. (2017). Sliding mode control of robotic arms with deadzone. *IET Control Theory & Applications*, *11*(8), 1214–1221.

Delta Robot: Working Advantages and Applications - SMLease Design. (n.d.). Retrieved February 22, 2023, from https://www.smlease.com/entries/automation/delta-robot-working-advantages-applications/

Delta Robot Original design [23] | Download Scientific Diagram. (n.d.). Retrieved February 22, 2023, from https://www.researchgate.net/figure/Fig-10-Delta-Robot-Original-design-23_fig5_340137275

Dinh, T. X., Thien, T. D., Anh, T. H. V., & Ahn, K. K. (2018). Disturbance observer based finite time trajectory tracking control for a 3 DOF hydraulic manipulator including actuator dynamics. *IEEE Access*, *6*, 36798–36809.

El-Ghazaly, G., Gouttefarde, M., & Creuze, V. (2015). Adaptive terminal sliding mode control of a redundantly-actuated cable-driven parallel manipulator: CoGiRo. *Cable-Driven Parallel Robots: Proceedings of the Second International Conference on Cable-Driven Parallel Robots*, 179–200.

FANUC Robotics FOOD M-3iA/6S 4-Axis Remote A-Cab R-30Ib - Flexible Automation Supply. (n.d.). Retrieved February 22, 2023, from https://www.flexibleassembly.com/FI-FOOD-M-3iA-6S

Gosselin, C. M., & Hamel, J.-F. (1994). The agile eye: a high-performance three-degree-of-freedom camera-orienting device. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 781–786.

Griffis, M., & Duffy, J. (1989). A forward displacement analysis of a class of Stewart platforms. *Journal of Robotic Systems*, *6*(6), 703–720.

Griffiths, J. B. (1985). *The theory of classical dynamics*. Cambridge University Press.

Guglielmetti, P., & Longchamp, R. (1994). A closed form inverse dynamics model of the delta parallel robot. *IFAC Proceedings Volumes*, *27*(14), 51–56.

Han, S. I., & Lee, J. (2016). Finite-time sliding surface constrained control for a robot manipulator with an unknown deadzone and disturbance. *ISA Transactions*, *65*, 307–318.

Han, S., Wang, H., & Tian, Y. (2018). Model-free based adaptive nonsingular fast terminal sliding mode control with time-delay estimation for a 12 DOF multi-functional lower limb exoskeleton. *Advances in Engineering Software*, *119*, 38–47.

Hunt, K. H. (1983). *Structural kinematics of in-parallel-actuated robot-arms*.

Innocenti, C., & Parenti-Castelli, V. (1993). *Closed-form direct position analysis of a 5–5 parallel mechanism*.

IRB 360 Series (ABB Robotics) M-1IA/0.5AL FANUC Europe Corporation 6... | Download Scientific Diagram. (n.d.). Retrieved February 22, 2023, from https://www.researchgate.net/figure/IRB-360-Series-ABB-Robotics-M-1IA-05AL-FANUC-Europe-Corporation-6-axis-Robot_fig4_336702692

Jalon, J. G., & Bayo, E. (1994). *Kinematic and dynamic simulation of mechanical systems—the real-time challenge*. Springer-Verlag, Berlin, Germany.

Kelly, R., Davila, V. S., & Perez, J. A. L. (2005). *Control of robot manipulators in joint space*. Springer Science & Business Media.

Kim, J., Jin, M., Choi, W., & Lee, J. (2019). Discrete time delay control for hydraulic excavator motion control with terminal sliding mode control. *Mechatronics*, *60*, 15–25.

Lin, W., Crane, C. D., & Duffy, J. (1994). *Closed-form forward displacement analysis of the 4–5 in-parallel platforms*.

Makwana, M. A., & Patolia, H. P. (2021). Model-based motion simulation of delta parallel robot. *Journal of Physics: Conference Series*, *2115*(1), 12002.

Mechanisms, P., Mechanisms, S. D. P., Mechanisms, S. T. T. P., Mechanisms, S. R. T. P., Mechanisms, O. T. P., Mechanisms, A. P., Mechanisms, T. D. P., Mechanisms, F. P., Redundancy, P. M., Mechanisms, P., & Machine-tools, I. (1956). *parallel kinematics*. 12–49.

Meng, D., Fan, L., Shi, Y., & Cheng, Z. (2021). Virtual simulation model of portable inverted pendulum. *2021 33rd Chinese Control and Decision Conference (CCDC)*, 3789–3793.

Miller, K. (1995). Experimental verification of modeling of DELTA robot dynamics by direct application of Hamilton's principle. *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, *1*, 532–537.

Miller, K., & Clavel, R. (1992). The Lagrange-based model of Delta-4 robot dynamics. *Robotersysteme*, *8*(1), 49–54.

Nanua, P., & Waldron, K. J. (1989). Direct kinematic solution of a Stewart platform. *1989 IEEE International Conference on Robotics and Automation*, 431–432.

Patel, K., & Bhatt, P. M. (2022). Modeling and dynamic analysis of multi-loop RSSR-SS parallel manipulator. *Materials Today: Proceedings*, *66*, 2001–2007.

Quang, N. H., Van Quyen, N., & Hien, N. N. (2020). Dynamic Modelling of 3-RUS Spatial Parallel Robot Manipulator. *Review of Computer Engineering Research*, *7*(1), 20–26.

Rahimi, S., Jalali, H., Yazdi, M. R. H., Kalhor, A., & Masouleh, M. T. (2022). Design and practical implementation of a Neural Network self-tuned Inverse Dynamic Controller for a 3-DoF Delta parallel robot based on Arc Length Function for smooth trajectory tracking. *Mechatronics*, *84*, 102772.

Saidi, I., & Touati, N. (2022). Sliding Mode Control of a 2DOF Robot Manipulator: A Simulation Study Using Artificial Neural Networks with Minimum Parameter Learning. *Recent Patents on Mechanical Engineering*, *15*(2), 241–253.

Sciavicco, L., & Siciliano, B. (2001). *Modelling and control of robot manipulators*. Springer Science & Business Media.

Shi, R., & Zhang, X. (2022). Adaptive fractional-order non-singular fast terminal sliding mode control based on fixed time observer. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *236*(13), 7006–7016.

Shi, X., & Cheng, Y. (2020). Adaptive fast terminal sliding mode (FTSM) control design for quadrotor UAV under external windy disturbances. *2020 International Conference* *on Unmanned Aircraft Systems (ICUAS)*, 512–516.

Slotine, J.-J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199, Issue 1). Prentice hall Englewood Cliffs, NJ.

Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, *180*(1), 371–386.

Tahmasebi, F., & Tsai, L. (1995). On the stiffnes of a novel six-degree-freedom parallel minimanipulator. *Journal of Robotic Systems*, *12*(12), 845–856.

Utkin, V., Guldner, J., & Shi, J. (2017). *Sliding mode control in electro-mechanical systems*. CRC press.

Van Khang, N. (2011). Kronecker product and a new matrix form of Lagrangian equations with multipliers for constrained multibody systems. *Mechanics Research Communications*, *38*(4), 294–299.

Van Khang, N., & Tuan, L. A. (2013). On the sliding mode control of redundant parallel robots using neural networks. *Proc. of the 3th IFToMM International Symposium on Robotics and Mechatronics, Singapore, Research Publishing*, 168–177.

Vischer, P., & Clavel, R. (1998). Kinematic calibration of the parallel Delta robot. *Robotica*, *16*(2), 207–218.

Wang, H., Shi, L., Man, Z., Zheng, J., Li, S., Yu, M., Jiang, C., Kong, H., & Cao, Z. (2018). Continuous fast nonsingular terminal sliding mode control of automotive electronic throttle systems using finite-time exact observer. *IEEE Transactions on Industrial Electronics*, *65*(9), 7160–7172.

WOLDU, Z. G. (2010). *Design and control of a five bar linkage parallel manipulator with flexible arms*.

Yu, X., Feng, Y., & Man, Z. (2020). Terminal sliding mode control–an overview. *IEEE Open Journal of the Industrial Electronics Society*, *2*, 36–52.

Yu, X., & Zhihong, M. (2002). Fast terminal sliding-mode control design for nonlinear dynamical systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *49*(2), 261–264.

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*, 1–16.

Zhao, H., Tao, B., Ma, R., & Chen, B. (2022). Manipulator trajectory tracking based on adaptive sliding mode control. *Concurrency and Computation: Practice and*

Experience, *34*(21), e7051.

REFERENCE LINKS

Trossen Robotic Community *http://forums.trossenrobotics.com/tutorials/introduction-129/delta-robot (ET: 10.06.2015)*

R.L. Williams II, “The Delta Parallel Robot: Kinematics Solutions”, Internet Publication, *www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf, April 2015.*

https://grabcad.com/library/delta-robot-full-assembly-1

APPENDIX

Appendix A: Matlab Code for Direct Kinematics

```
% Direct kinematics of 3-DOF Delta Robot
function [x0,z0,y0]= Delta_calcForward( theta1,theta2,theta3)
t = (f-e)*tand(30)/2;
dtr = pi/180;
theta1=theta1 * dtr;
theta2=theta2 * dtr;
theta3=theta3 * dtr;
z1 = -(t + rf*cos(theta1));
y1 = -rf*sin(theta1);
z2 = (t + rf*cos(theta2))*sind(30);
x2 = z2*tand(60);
y2 = -rf*sin(theta2);
z3 = (t + rf*cos(theta3))*sind(30);
x3 = -z3*tand(60);
y3 = -rf*sin(theta3);
dnm = (z2-z1)*x3-(z3-z1)*x2;
w1 = z1*z1 + y1*y1;
w2 = x2*x2 + z2*z2 + y2*y2;
w3 = x3*x3 + z3*z3 + y3*y3;
% x = (a1*y + b1)/dnm
a1 = (y2-y1)*(z3-z1)-(y3-y1)*(z2-z1);
b1 = -((w2-w1)*(z3-z1)-(w3-w1)*(z2-z1))/2.0;
% z = (a2*y + b2)/dnm;
a2 = -(y2-y1)*x3+(y3-y1)*x2;
b2 = ((w2-w1)*x3 - (w3-w1)*x2)/2.0;
% a*y^2 + b*y + c = 0
a = a1*a1 + a2*a2 + dnm*dnm;
b = 2*(a1*b1 + a2*(b2-z1*dnm) - y1*dnm*dnm);
c = (b2-z1*dnm)*(b2-z1*dnm) + b1*b1 + dnm*dnm*(y1*y1 - re*re);
% The discriminant
d = b*b - 4*a*c;
```

```
if (d < 0)   % non-existing point
x0=0;
z0=0;
y0=0;
else
y0 = -0.5*(b+sqrt(d))/a;
x0 = (a1*y0 + b1)/dnm;
z0 = (a2*y0 + b2)/dnm;
end
```

Appendix B: Matlab Code for Inverse Kinematics

```
% Inverse Kinematics of 3-DOF Delta Robot
function [theta1,theta2,theta3]=Delta_inverse(x0,y0,z0)
theta1 = Delta_calcAnglezy(x0,y0,z0);
x01=x0*cosd(120) + y0*sind(120);
x02=x0*cosd(120) - y0*sind(120);
y01=y0*cosd(120)-x0*sind(120);
y02=y0*cosd(120)+x0*sind(120);
theta2= Delta_calcAnglezy(x01, y01, z0); % rotate coords to +120 deg
theta3= Delta_calcAnglezy(x02, y02, z0); % rotate coords to -120 deg

function [theta]=Delta_calcAnglezy(x0,y0,z0)
e  =  7.79;
f  =  27.30;
re =  26.79;
rf =  14.50;
y1 = -0.5 * 0.57735 * f; % f/2 * tg 30
y0 =y0 - 0.5 * 0.57735 * e; % shift center to edge
% y = a + b*z
a = (x0*x0 + y0*y0 + z0*z0 +rf*rf - re*re - y1*y1)/(2*z0);
b = (y1-y0)/z0;
d = -(a+b*y1)*(a+b*y1)+rf*(b*b*rf+rf);
if (d > 0) % existing point
yj = (y1 - a*b - sqrt(d))/(b*b + 1); % choosindg outer point
zj = a + b*yj;
if (yj>y1)
theta = 180*atan(-zj/(y1 - yj))/pi + 180;
else
theta = 180*atan(-zj/(y1 - yj))/pi ;
end
else
theta=0; %non-existing point
end
```

Appendix C: Matlab Code for Circular Trajectory

```
function [x,y] = fcn(t)
x = 10*cos(t);
y = 10*sin(t);
```

Appendix D: Matlab Code for Jacobian Matrix and Dynamic Constants

```
% Jacobian matrix Generation
e  =  7.79;  % end effector
f  =  27.30; % base
re =  26.79; % lower leg length
rf =  14.50; % upper leg length

ma = 0.09; % mass of upper leg
mb = 0.12; % mass of lower leg
g = 9.81; % gravity
a = 14.50;
mc = 0.05; % mass of moving platform
px = 0.05995;
py = 0.02948;
pz = -17;
q11 = 0.876272316341518;
q12 = 0.449713147816531;
q13 = 0.910373602925376;
cx1 = px + f - e;
cy1 = py;
cz1 = pz;
cx2 = py + f - e;
cy2 = -px;
cz2 = pz;
cx3 = -px + f - e;
cy3 = -py;
cz3 = pz;
q31 = acos(cy1/re);
q32 = acos(cy2/re);
q33 = acos(cy3/re);
q21 = acos((cx1*cx1 + cy1*cy1 + cz1*cz1 - rf*rf - re*re)/(2*re*rf*sin(q31)));
q22 = acos((cx2*cx2 + cy2*cy2 + cz2*cz2 - rf*rf - re*re)/(2*re*rf*sin(q32)));
q23 = acos((cx3*cx3 + cy3*cy3 + cz3*cz3 - rf*rf - re*re)/(2*re*rf*sin(q33)));
```

```
theta = [q11 ,q12, q13 ; q21 ,q22 , q23 ; q31 ,q32 , q33];   %  all angles value
j1x = sin(q31)*cos(q21 + q11);
j2x = cos(q32);
j3x = -1*sin(q33)*cos(q23 + q13);
j1y = cos(q31);
j2y = -sin(q32)*cos(q22 + q12);
j3y = - cos(q33);
jlz = sin(q31)* sin(q21 + q11);
j2z = sin(q32)* sin(q22 + q12);
j3z = sin(q33)* sin(q23 + q13);
jpos = [j1x , j1y , jlz ; j2x , j2y ,j2z ; j3x , j3y , j3z ];
jtheta = rf.* [sin(q21)*sin(q31), 0 , 0 ; 0 , ...
sin(q22)*sin(q32) ,0 ; 0 , 0 ,sin(q23)*sin(q33) ];
J = jtheta\jpos    % Jacobian
J_inv=inv(J);     %inverse of Jacobian
J_inv_T= J_inv.' ;
cd=1;
I = [1 , 0, 0 ;0 , 1 , 0; 0, 0, 1];
m=3*mb+mc;
% Calculation of inertia matrix
Im=15*I;
m = m*I;
IA = Im + (1/3 * ma*a*a + mb*a*a)*I;
M = IA + J_inv_T*m*J_inv

% Calculation of the centrifugal and Coriolis forces
p_dotdot = [-1039.19 ; -0.51 ; -29.95];
q_dot = [-1004.16 ; -0.13 ;-0.29];
q_dotdot = [-3.03 ; -0.0061 ; -3.01];
j_dot_inv = (p_dotdot - (J\q_dotdot))/q_dot;
C = cd*I + J_inv_T * m * j_dot_inv

% Calculation of gravitational force
G = -(0.5*ma+mb)*g *a*[cos(q11); cos(q12); cos(q13)] - J_inv_T * m*[0 ; 0 ; -g]
```

Appendix E: Matlab Code for ANFTSMC Block (Used to Simplify Calculation of Terms in Control Law, Sliding Manifold and Adaptation Law)

```
function [t_term ,s,adapt]= Sliding_Manifold(q,qdot,e1,e2,sg1,sg2)
k1=9;
k2=1;
A=2;
B=1.8;
eta=1.3;
l=0.01;
q=[q(1) q(2) q(3)];
qdot =[qdot(1) qdot(2) qdot(3)];

e1=[e1(1) e1(2) e1(3)];
e2=[e2(1) e2(2) e2(3)];
sg1=[sg1(1) sg1(2) sg1(3)];
sg2=[sg2(1) sg2(2) sg2(3)];
I=[1 0 0;0 1 0;0 0 1];

s=e1'+k1*diag(sg1)*abs(e1').^A + diag(sg2)*k2*abs(e2').^B;
t_term=-1/(B*k2)*diag(sg2)*(((I+A*k1*diag(abs(e1).^(A-1))))*(abs(e2').^(2-B)));

b0=l*norm(s)*norm(e2)^(B-1);
b1=l*norm(s)*norm(e2)^(B-1)*norm(q);
b2=l*norm(s)*norm(e2)^(B-1)*norm(qdot)^2;
% b0=9.5; b1=2.2; b2=2.8;
adapt=b0+b1+b2+eta;
```

Appendix F: Conventional sliding mode controller Design

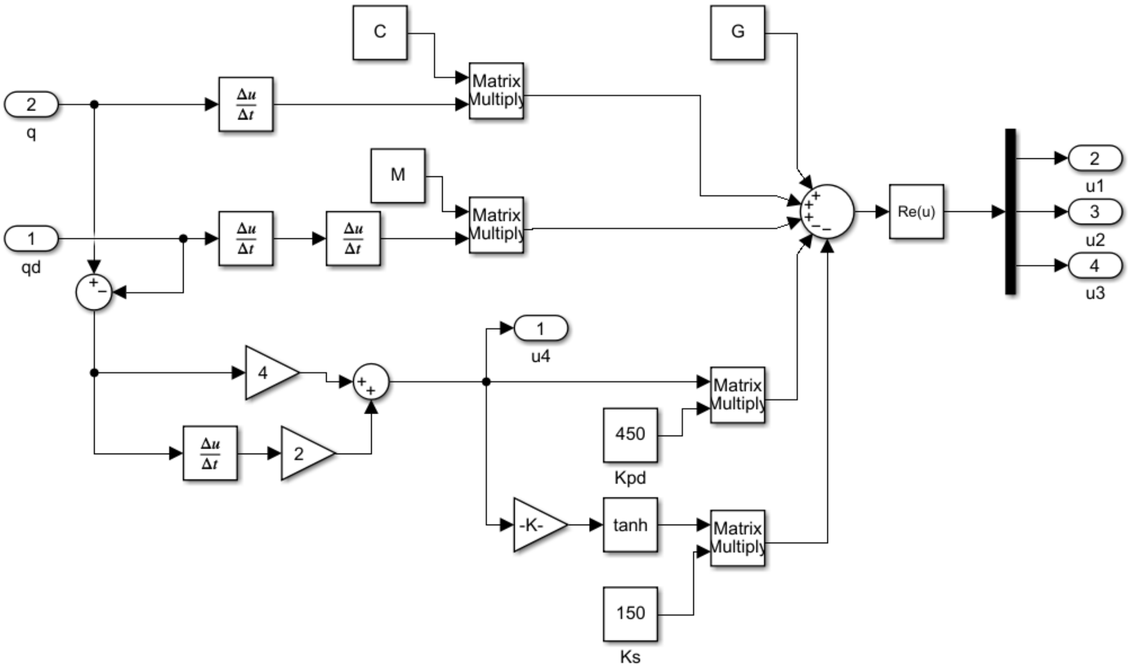


Figure 5. 31: Conventional sliding mode Simulink model

Appendix G: Adaptive Fast Non-Singular Terminal Sliding Mode Controller Design

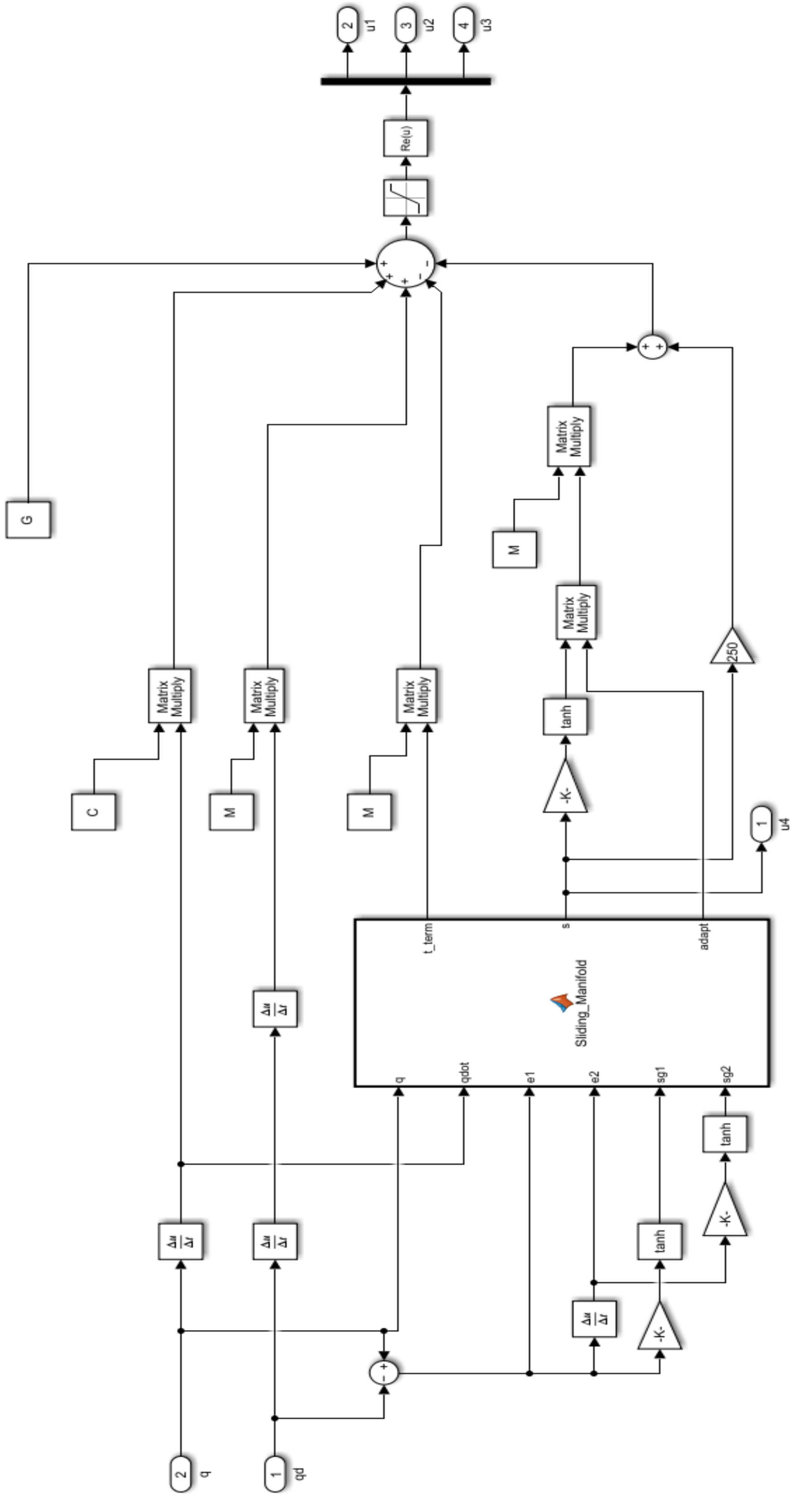


Figure 5. 32: Adaptive fast non-singular terminal sliding mode Simulink model