Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

AI-Based Mobile Robot for Agricultural Application using Sliding Mode Controller

A Thesis Submitted to School of Graduate Studies, Addis Ababa University, Addis Ababa Institute of Technology, for Partial Fulfillment of the Requirement for Masters of Science in Control Engineering

Zewdu Jemema Guluma

May, 2023

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

AI-Based Mobile Robot for Agricultural Application using Sliding Mode Controller

A Thesis Submitted to School of Graduate Studies, Addis Ababa University, Addis Ababa Institute of Technology, for Partial Fulfillment of the Requirement for Masters of Science in Control Engineering

Zewdu Jemema Guluma

Advisor: Dr. Dereje Shiferaw

May, 2023

Addis Ababa, Ethiopia

Declaration

I, the undersigned, warrant that the work included in this thesis is entirely original, has not previously been submitted for credit at this or any other university and has been properly cited.

Zewdu Jemema Guluma

Name	**Signature**	**Date**

Email: **zedocomoros@gmail.com**

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa, Ethiopia.

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Dereje Shiferew

Advisor	**Signature**	**Date**

Dedication

Completing a master's program and writing a thesis is a collaborative endeavor, rather than an individual effort. It takes a village to achieve such a feat, and I would like to take this opportunity to express my gratitude to the people who have been instrumental in my success.

My utmost gratitude goes to my family and friends who accompanied me throughout my academic journey, as I dedicate this thesis work to them. Their unwavering love, support, and encouragement have been a constant source of motivation for me, and I would not have been able to achieve this without them. Their advice, guidance, and feedback have been invaluable in shaping my research and writing skills, and I am forever grateful for their presence in my life.

I especially appreciate their support towards this thesis work, which is the culmination of my academic journey.

Approval

The undersigned certifies that the thesis prepared by Mr. Zewdu Jemema Guluma, titled **“AI-Based Mobile Robot for Agricultural Application using Sliding Mode Controller,”** and submitted as partial fulfillment for the Degree of Master of Science in Control Engineering, fulfills the regulations of the university and meets the accepted standards for originality, content, and quality.

Signed by Examining Board

Dr. Bisrat Diribsa

Dean, School of ECE	**Signature**	**Date**

Dr. Dereje Shiferaw

Advisor	**Signature**	**Date**

Nebiyu Tenaye

Internal Examiner	**Signature**	**Date**

Dr. Lebsework Negash

External Examiner	**Signature**	**Date**

Dr. Lebsework Negash

Chairperson	**Signature**	**Date**

Abstract

Recent advancements in agricultural robotic systems have greatly enhanced their functionality, usability, and integration into various tasks, particularly in the field of agriculture. The primary goal of designing agricultural robots is to enhance efficiency, save time, and decrease production costs by incorporating controllers, sensors, actuators, and communication systems. These robots have versatile applications and are widely embraced in the agricultural sector in industrialized countries. Extensive research has been dedicated to developing mobile robot platforms tailored for agricultural tasks, including plant health monitoring, pesticide spraying, fruit picking, and harvesting, with the aim of supporting farmers in developing nations like Ethiopia, where approximately 67% of the population is involved in agriculture. The thesis specifically targets fruit harvesting and focuses on the challenging task of modeling, designing, and simulating a mobile manipulator with advanced capabilities for agricultural businesses, making it one of the most difficult undertakings in this field. The study encompasses the presentation of the mobile manipulator's $3D$ design, kinematics, and dynamics. In addition, AI techniques are employed to analyze fruit images, facilitating the accurate detection and determination of fruits. Based on the results, the effectiveness of the training technique has been assessed using an RMSE value of 0.19 and a loss value of 3.6e-02. An SMC utilizes the input generated from the image to govern the mobile manipulator's position. The system's stability and robustness have been assessed by considering uncertainties and variations in mass. When comparing the performance of a designed controller with a PID controller in the presence of uncertainty and parameter variation, it was found that SMC outperformed. According to the evaluation using the ITAE, SMC proves to be more effective, demonstrating a 75% improvement compared to the PID controller. Overall, this research contributes to the development of a robust and intelligent mobile manipulator for fruit harvesting in the agricultural sector, with potential applications to support farmers in countries like Ethiopia.

Keywords: AI, Deep Learning, Computer Vision, YOLOv2, Mobile Robot

Acknowledgments

My first and greatest thanks are to almighty God, for letting me through all difficulties and helped me to complete this thesis.

Next, I would like to acknowledge and express my deepest gratitude to my thesis advisor Dr. Dereje Shiferew for his continuous support, mentorship, enthusiasm, advice, and guidance on this thesis work.

Last but not least, I am grateful for all of the people who have assisted to make my thesis a success by providing inspiration, support, and guidance.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligent
CAD	Computer Adied Design
CNN	Convolution Neural Network
DDMR	Differential Drive Mobile Robot
DOF	Degree of Freedom
IBVS	Image-Based Visual Servoing
IOS	International Organization for Standardization
ITAE	Integral of Time Absolute Error
MM	Mobile Manipulator
NLP	Natural Language Processing
PBVS	Position-Based Visual Servoing
PID	Proportional Integral Derivative
SGDM	Stochastic Gradient Descent with Momentum
SMC	Sliding Mode Controller
ReLUs	Rectified Linear Units
RMSE	Root Mean Square Error
ResNet50	Residual Neural Network
UGV	Unmanned Ground Vehicle
VS	Visual Servoing
VSC	Variable Structure Control
YOLOv2	You Only Look Once version 2

Chapter 1

Introduction

1.1 Background

Various organizations and dictionaries offer definitions for the term ”robot.” According to the International Organization for Standardization (IOS), a robot is described as an ”autonomously operated, re-programmable, multipurpose manipulator having three or more axes” [1]. On the other hand, Merriam-Webster defines a robot as a machine that resembles a human being and performs various challenging human tasks such as walking or conversing [1]. This definition highlights the motivating aspect of robots. Additionally, the US National Bureau of Standards provides different definitions for specific types of robots. For example, an industrial robot is defined as a programmable and multi-functional manipulator used for moving materials, parts, and specialized devices by employing programmed motions to perform a range of tasks. A pick-and-place robot, on the other hand, is a simple robot that transfers items from one location to another by using point-to-point motions [2].

Another type of robot mentioned is a manipulator, which is a mechanism consisting of segments that interlock or slide relative to each other, enabling the robot to grasp and move objects with several degrees of freedom. Manipulators can be controlled remotely either by a computer or by a human operator. An intelligent robot is one that can be programmed to perform tasks based on sensory input. A fixed-stop robot is a robot that has control over its stopping point but not its trajectory. Lastly, a mobile robot is a robot that is mounted on a movable platform [2]. Based on the descriptions provided above, a mobile robot can be defined as a system that possesses complete mobility within its environment. It also exhibits a certain degree of autonomy, limited human interaction and perception capabilities, and the ability to sense and respond to its surroundings.

With all of this potential, the developed mobile robot can be used in the agricultural industry for harvesting and picking fruit. Many countries around the world depend heavily on agriculture. Many academics have made numerous attempts to use technology and mechanical intervention in agriculture. There were attempts to use robotics to support agriculture in the early 1920s. These were crude models, and in order to run the machine, a cable connection was needed. After the 1980s, when developments in computer science and engineering made machine vision guiding practical, ground-breaking projects to use robots in agriculture kept popping up [3].

An agricultural robot is a robot used for various agricultural applications. Their primary use right now in this region is for gathering fruit and harvesting. To detect necessary tasks, allocate them, and ensure that they are completed in accordance with the provided schema, it uses machine vision applications. Fruit picking or harvesting is one of the many uses for machine vision-based mobile robotics in the agricultural sector. Other uses include weed detection and removal, seeding, pesticide spraying, soil analysis, and environmental monitoring to assist farmers. Mobile robots with all of these applications in the agricultural industry require a controller to oversee and carry out the desired action. A controller is a tool that can improve a system's performance by detecting data from a linear or nonlinear system. Stability, effective disturbance rejection, and minimal tracking error are the primary goals when building a controller for a system. Among the different varieties of nonlinear controllers applicable to a nonlinear system, sliding mode controllers are the most powerful nonlinear resilient controllers that have the ability to regulate nonlinear systems when the dynamic parameters of the system are partially unpredictable. With the help of this controller, highly nonlinear systems can be managed, such as mobile robots with manipulators mounted on the platform [4].

The leading goal of this thesis is to design, model, and simulate an AI-based mobile manipulator for agricultural applications using a sliding mode controller, which uses computer vision (to identify, observe, and acknowledge information in the image) and a deep learning algorithm so that the robot can identify and recognize the position of fruit that would be harvested. Visual feedback instructs the mobile robot to move to the target from its home position in the workspace using the computed image from the image processing section. The modeled mobile robot has sufficient DOF to carry out activities of harvesting fruit.

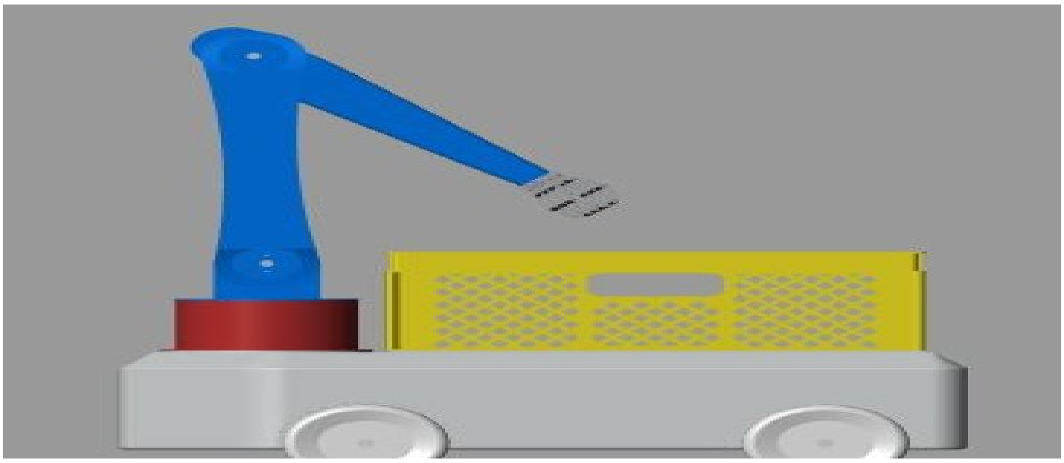


Figure 1.1: Designed Mobile Manipulator

1.2 Problem of Statement

Plowing, seeding, harvesting, and pesticide spraying are all labor-intensive tasks in agriculture, which drive up labor costs and affect the cost of output across agricultural industries. Reduced operational costs for these businesses have been the main focus of the efforts of numerous researchers over the past few decades. A significant issue in the agriculture sector is the presentation, modeling, and design of an autonomous system that executes tasks and replaces human capital. Because crop farming is a time-invariant and nonlinear system, designing and implementing an automation system to carry out this division's process presents challenges in terms of stability, robustness, and controller design. The methods of object detection and fruit identification in the design process also had issues. Developing and deploying a reliable and resilient automation system became a critical measure to reduce and eliminate the operational expenses incurred by the substantial workforce required for agricultural activities.

In order to overcome this issue, the objective of this research is to create an autonomous system that is suitable for agricultural applications, notably collecting fruit. The task at hand involved the design of a dependable and robust mobile manipulator. The Solid-Works software was used to assemble the $3D$ CAD model for the mobile manipulator. The mathematical model was briefly discussed and used in both the design and verification processes. A YOLOv2 deep learning algorithm was used in the design and modeling of an AI-based mobile manipulator for the purpose of detecting and determining the position of the fruit.

Due to its unique characteristics, such as quick response and resilience to the current uncertainty, the sliding mode controller was created to manage high non-linearity and external disturbances. A simulation of the designed system with various payloads on the manipulator, as well as a mobile platform, disturbance rejection, and a small tracking error, was presented. The AI improves the designed model's decision-making abilities with respect to harvesting fruit and proper end-effector positioning through its visual control technique.

1.3 Objective

1.3.1 General Objective

The broad goal of this thesis is to model, design, and simulate an AI-based mobile manipulator for agricultural applications using components of AI and a sliding mode controller.

1.3.2 Specific Objectives

The aforementioned particular goals were accomplished while writing this thesis:

- To create a 3D mechanical representation and develop a mathematical model for a mobile manipulator.
- To identify and determine the position of fruit from the image using AI.
- To design a controller with and without the addition of uncertainty in the system.
- To verify the designed control system using MATLAB and SolidWorks software.
- To assess and contrast the designed controller's performance.

1.4 Scope of the Thesis

A lot of time and money are typically needed to do research in the robotics sector because there are so many sub-fields within the discipline. This study focuses on mobile robots from various robot categories and agricultural applications from their application fields due to their vast applications and classifications. A mobile robot powered by artificial intelligence for agricultural applications was introduced in this study after understanding the main characteristics of the nonlinear system of a mobile robot and AI. The design of mobile robots incorporates a navigation system for usage in a variety of contexts. However, the work in this area is focused on designing and simulating mobile manipulators in situations without obstacles.

Natural Language Processing (NLP), Neural Networks, Deep Learning, Machine Learning, Cognitive Computing, and Computer Vision are artificial intelligence components. These essential components have a wide range of uses in the manufacturing, robotics, military, healthcare, agricultural, and entertainment sectors. Robotics, computer vision, and deep learning are the topics of this study. Position-based visual servoing control techniques, object detection algorithms, and the design and modeling of mobile robots for fruit harvesting are all used to analyze input images. The YOLOv2 object detection algorithm, a single-stage real-time CNN method, was used to calculate and detect the operating position of the mobile manipulator. A sliding mode controller is used to manage disturbances and varying payloads on mobile platforms and manipulators appropriate for Ethiopia's horticulture and agricultural sectors.

1.5 Significant of the Thesis

The modern robot industry has generated a wide range of robot types for a number of tasks. The efficient production of agriculture necessitates autonomous and time-saving technology for picking and harvesting fruit from the plant, especially in light of the fact that robotics are crucial to agricultural operations and production for a variety of uses. Creating an AI-based mobile robot for agricultural applications that can work in any environment without getting off significantly contributes to the industry by lowering the number of workers needed for agricultural activity, notably (harvesting fruit). In addition to this, the thesis has the following significance:

- Increases the quality, productivity, and efficiency of the agricultural sector of horticulture, such as orange farms, during the harvesting season.
- Performs tasks better than a human being and ensures the most accurate result possible since the designed system is based on AI.
- Assists farmers in enhancing productivity and minimizing operational expenses.

1.6 Methodology

The proposed thesis concentrates on studying the literature about the theoretical and structural backgrounds of artificial intelligence, mobile manipulators, and controller methods suitable for mobile manipulators designed for agricultural applications. After having a solid understanding of mobile manipulator and controller design, choose an approach that results in a reliable AI-based sliding mode controller that can be used to precisely regulate the position of a robotic manipulator for picking and harvesting fruit or other objects. Hence, mobile manipulators move automatically from one location to another, are categorized as mobile robots, and have unrestricted movement in the workspace and in the field. The creation of robot platforms and the manipulator installed on those platforms are included in the design processes. Based on the specific design parameters, the $3D$ CAD mechanical design and the mathematical models have been designed. Some of the steps that would be taken are as follows:

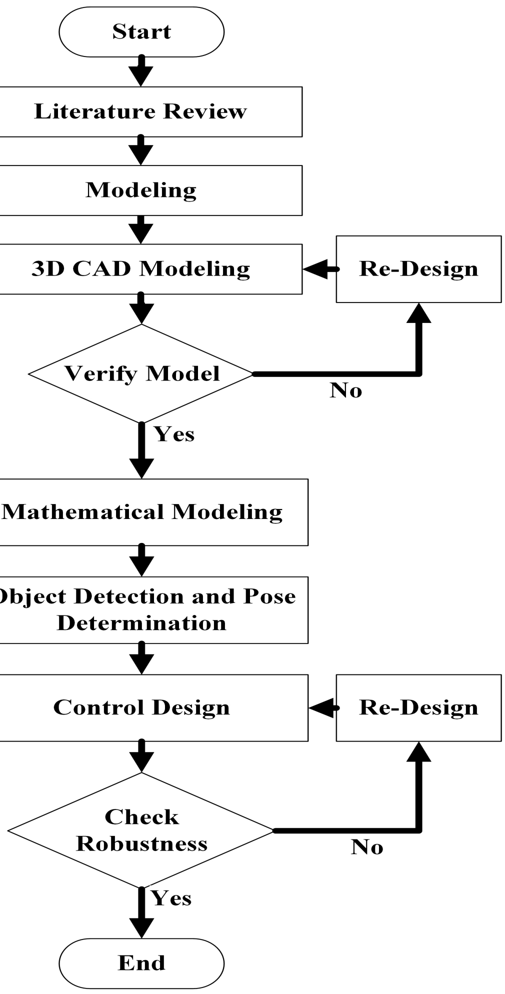


Figure 1.2: Methodology

1.7 Thesis Outline

Under this section, the structural organization of the thesis was briefly explained as follows: The basis of the thesis, the problem statement, the general and specific goals of the thesis, its scope, and the relevance or contributions of the thesis are all discussed in Chapter One. Chapter Two provides concepts and theories, a review of related work, and a unique aspect of this thesis. Chapter $Three$ is the main section of this thesis and presents the overall model of the system, including the mechanical design of the plant by using solid-work CAD design software and a mathematical model that includes the mobile manipulator, kinematic, and dynamic models. Detection and determination of fruit position and a brief description of the controller's design were addressed in Chapter $Four$. Discussions were made based on simulation results and findings from the MATLAB simulation software in Chapter $Five$. The work's conclusion and some potential future work were presented in Chapter Six.

Chapter 2

Review of Literature

2.1 Introduction

In this chapter, some general theories and concepts related to this work have been explained precisely. The first part of this section describes the controller, followed by a definition and elaboration on artificial intelligence (AI) and the algorithm chosen for the specific purpose of the created model, as well as a brief discussion of the mobile manipulator.

2.2 Concept and Theory

Since the physical world is full of complex nonlinear systems, each scenario necessitates a specially designed controller that is stable and robust to parameter variations and external disturbances. It is difficult to develop systematic design procedures to meet control objectives and design specifications due to the complexity of nonlinear feedback control systems. When confronted with such difficulties, it is obvious that a single procedure and method cannot be applied to all nonlinear physical systems. Several nonlinear design tools were proposed to solve these real-world problems [5].

Variable structure control (VSC) has become a reliable method in various domains, offering a robust solution for control challenges. It has been effectively employed in numerous control problems, such as automatic flight control, regulation in chemical processes, space systems, and robotics. Control techniques have also found exciting applications in traditional fields including electrical, mechanical, aerospace, and chemical engineering [6]. In today's highly automated society, it's evident that advanced robotics techniques are widely employed in various agricultural applications.

Just by observing the surroundings, it becomes apparent that these techniques are extensively utilized in the agricultural sector. SMC methodology is a type of VSC scheme that is fundamentally a result of discontinuous control. SMC theory has been applied to various control systems because it has been shown that this nonlinear type of control demonstrates some excellent qualities, such as robustness against large parameter variations and disturbances [7]. Another interesting future of SMC is the relative simplicity of design, control of independent motion, and designing switching functions of state variables or output variables to form sliding surfaces, guaranteeing that when trajectories reach the surfaces, they yield the desired system dynamics. Some benefits of utilizing SMC include quick reaction times, smooth transitional behavior, and the ability to handle uncertainties and external disturbances effectively. In this scheme, SMC was chosen as a nonlinear controller due to its promising features and advantages.

The use of artificial intelligence (AI) in agriculture, specifically in fruit harvesting, has become increasingly popular due to several benefits, including reducing the need for manual labor, increasing efficiency and accuracy, and addressing the decline in fruit-picking capacity caused by the aging rural population. The benefits of AI in agriculture are undeniable, and smart farming tools and vertical farming systems can perform small, repeatable, and time-consuming tasks so farm workers can use their time for more strategic operations that require human intelligence. AI is a vast field that includes a wide array of specializations, including continuous mathematics, logic, perception, probability, reasoning, learning, and action, and it covers everything from small electronic devices to space exploration. Overall, the use of AI in fruit harvesting has promising potential for the future of agriculture and is used in this design to make the model more precise and have decision-making capability in its operation [8, 9].

Researchers in the fields of robotics, natural language processing, computer vision, computational biology, and e-commerce have all noted that AI encompasses a variety of research areas. Machine learning is a type of artificial intelligence that enhances a computer's ability to learn and adapt to new and constantly changing data. It is a subset of AI that focuses on developing algorithms and models that can learn from data and make predictions or decisions based on that learning. Deep learning, on the other hand, is a relatively new field of machine learning research that aims to bring machine learning and AI closer together.

It is based on the study of deep neural networks in the human brain and seeks to mimic the inner-layer functions of the brain while creating knowledge from multiple layers of information processing. Deep learning algorithms discover multiple levels of representation, or a hierarchy of features, with higher-level, more abstract features defined in terms of (or generating) lower-level features. This approach allows for more accurate predictions and decision-making, as well as the ability to learn from unstructured data such as images, audio, and text [10].

Based on their function and the data they utilize to do a particular task, there are many types of robots, as defined in Chapter 1. Mobile robots, also known as autonomous robots, are capable of independently moving between different locations without requiring assistance from human operators. Unlike most industrial robots, which are confined to specific workspaces, mobile robots have the distinctive capability to freely navigate within a designated area in order to accomplish their intended objectives. Thanks to their mobility, they are well-suited for various applications in both structured and unstructured scenarios. Also, mobile robots have multiple uses in the agrarian industry because they are suitable for typical applications with minimal mechanical complexity and energy consumption [11].

The agriculture sector requires a lot of human power to perform the production process, particularly in the horticulture sector where fruits like citrus fruits are manually picked from different parts of the fruit trees. However, harvesting fruits on a large scale continues to be both inefficient and costly. To address this, researchers have developed automated robotic picking mechanisms that employ a robotic arm to move a manipulator into the target fruit's picking range before removing the fruit from the tree. The manipulator is guided with the help of a machine vision system that is used to identify the fruit. Automated farming, which uses various technological devices to improve and automate agriculture operations and the crop or livestock production cycle, is transforming modern agriculture. Harvesting robots must handle the products gently, and routine tasks can be automated with robotics technology, reducing labor costs in the agriculture industry [12]. This allows the robots to perceive, comprehend, and appreciate the autonomy and intelligence of their mechanical counterparts. In Chapter 4, the mechanisms of image capture, object detection algorithms, and fruit position determination were explained.

2.3 Related Work

Research frequently focuses on an interactive crop-scale agricultural robot, especially one with dexterous tactile abilities. The introduction of the agricultural robot is geared toward achieving this goal. It has the ability to be utilized for various precision agriculture activities such as weeding, harvesting, spraying, cutting, watering, and other similar tasks. From several types of research that are undergone through the usage of UGV for different agricultural applications in a different environment, a few are reviewed below:

2.3.1 Robot for Agricultural Application

In [13] mechanical weed control in outdoor settings, a mobile autonomous agricultural robot with vision-based perception was demonstrated. The robot has two different vision systems: one that uses gray levels to recognize the structure of crop rows and guide the robot along them, and another that uses colors to distinguish between individual crops and weed plants. A weeding tool is operated by this vision system to get rid of the weeds in the crop row. Verification of the system as a whole demonstrates the interoperability of the subsystems. The main objective of [14] is to present a mobile robotic project that aims to develop agricultural control, navigation, and data acquisition technologies. This project primarily focuses on using the mobile platform for remote sensing of important agronomic parameters across large areas of major crops. By incorporating various sensors and equipment, the mobile manipulator is intended to collect extensive data needed to analyze spatial variations. The proposal is built upon a systematic approach that encompasses previous scientific research, highlighting the key methodologies and technologies utilized in agricultural vehicles and robots.

In [15] the system suggested emphasizes the execution of all farm procedures, particularly plowing and seeding, through the utilization of multiple sensors, a microcontroller such as the H C 05 and H 06 Bluetooth models, and others. Using sensors, the robot locates the planning area, and using its gripper arrangement, it must plant seeds in the matching field. This robot will assist farmers in carrying out their farming operations more precisely. The article by [16] showcased the presentation of a prototype of a compact agricultural robot in their research. The focus was on describing the design, construction process, key features, and evaluation of the robot, which aimed to perform basic tasks in agricultural operations.

An autonomous agrobot for viticulture operations was presented in [17] along with its system architecture design, development, and integration. The main characteristics of this suggested system include the capability to apply chemicals on a crop level, a versatile design consisting of three interconnected units, a flexible operational design that enables automation of harvest, green harvest, and defoliation, as well as the unique ability to personalize all agricultural tasks for customized vineyard practices. In a different publication [18], a robot that is capable of autonomous apple-picking is explained. The main components of this system include a mobile mechanism, a visual system, and a robotic arm with a gripper. In a previous study [19], researchers introduced a strawberry-harvesting robot that consisted of a red-green-blue depth camera, a gripper, and a robotic arm mounted on a wheeled robot.

Another work, [20] focused on using autonomous robots for visually monitoring vineyards. Autonomous robots have proven successful in various industries, such as construction, manufacturing, and agriculture. The motivation behind this research is to reduce labor costs for farmers and improve efficiency in grape production through the use of autonomous robots. The article [21] extensively examines the development of innovative and distinct soft grippers and end effectors for robotic harvesting. The authors took into account the physical attributes of different fruits to determine commonly employed techniques in automated picking for various crops. Leveraging soft grippers for automating direct harvesting processes, that involve twisting, bending, tugging, lifting, or a combination thereof while maintaining gentle contact with the fruits, can provide a significant advantage.

2.3.2 Mobile Robot Based on AI and Sliding Mode Controller

A cherry tomato harvesting robot consisting of different parts, including a stereo vision unit, an end effector, a manipulator, a fruit collector, and a rail vehicle, was introduced in [22]. Furthermore, [23] included details of an independent robot designed for picking kiwifruit; the design concept is that the robotic system consists of a stereo-vision. The machine receives commands through a wireless connection and moves independently by using both GPS and computer vision. In [24] Khare, explored the design and control elements of a newly developed self-governing mobile robot for agricultural fruit harvesting. Their approach involved implementing a robust motion control design that relied on an integral sliding surface.

The study, [25] by Nichols introduced a wall-following robot that derives inspiration from biological systems while utilizing neural networks for its functioning. To ensure stability and accurate trajectory tracking, the researchers incorporated a sliding mode controller into their approach. Moreover, there is an increasing potential for employing machine learning (ML) in the realm of digital agriculture. This includes utilizing robots and sensors to effectively oversee and regulate crops, as well as gather relevant data pertaining to crop growth. In [26] the performance of an adaptive sliding mode controller and a PID controller in reducing mobile robot tracking error is compared. Additionally, a sliding mode higher-order controller for non-holonomic differential drive mobile robots is proposed in a separate paper, which also includes an analysis of its stability.

The work presented in [27] is a novel sliding mode control approach designed for nonholonomic mobile robots in 2D polar coordinates, showcasing the advantages of adaptive and event-triggered methodologies. Furthermore, some studies have merged sliding mode control with other AI techniques like fuzzy logic and convolutional neural networks to enhance performance. In general, existing literature indicates that sliding mode control is widely favored in agricultural applications for controlling mobile robots. Furthermore, artificial intelligence methods, such as deep learning and neural networks (NNs), are frequently employed for tasks like path planning, obstacle avoidance, and navigation.

2.4 Research Gap

In recent decades, numerous studies have dedicated significant efforts to advancing the modeling of mobile robots. However, although these designs have been evaluated in diverse scenarios, they have yet to consider the comprehensive analysis, modeling, design, and simulation of mobile platforms with integrated manipulators. Typically, they incorporate a separate control system for the mobile platform and attach a manipulator to it, which serves their specific purposes. Additionally, their work has primarily focused on mathematical modeling without considering the $3D$ CAD modeling of the physical system and linear control mechanism, even if the nature of mobile manipulators is generally nonlinear. Also, the works of different scholars do not allow the designed system to function and make a decision on its own; the identification and determination of objects to be picked or harvested are not considered.

2.5 Unique Aspect of the Thesis

This work presents a mobile robot that stands out due to its unique characteristics. The robot's design revolves around artificial intelligence (AI), specifically utilizing a deep learning convolutional neural network (CNN) technique to identify fruits. By inputting a captured image, the system processes the data and provides valuable output, enabling the monitoring and control of the end effector's position. The robot operates in real-time, ensuring swift data input operations. It employs sliding mode controllers to achieve rapid response times and incorporates a single-stage real-time CNN model for object detection and position determination. The AI algorithms incorporated in this model and design prioritize data collection and establish guidelines for converting the gathered information into insightful data. Additionally, these algorithms select the most suitable algorithm for achieving desired outcomes and continuously enhance themselves to ensure precise results, particularly in fruit cultivation across various farms within the country.

The process of gathering objects or fruits from a tree is accomplished without causing harm to its branches and leaves. The mobile manipulator distinguishes between fruits and leaves through image capture and collects or harvests the fruits upon detection. Another standout feature of the model is its exceptional ability to generate a comprehensive 3D representation of physical structures through the use of SolidWorks software. This innovative modeling technique encompasses specific design specifications alongside an advanced control system, rendering it exceptionally well-suited for a wide range of applications, both indoors and outdoors, in the realm of agriculture. The primary emphasis of the thesis revolved around the development of an autonomous agrarian system embodied in the form of a mobile manipulator. By combining cutting-edge technology and automation, this groundbreaking system aims to revolutionize the way agricultural tasks are performed. With the assistance of AI algorithms, the robot makes decisions and performs specific tasks like harvesting or picking fruit from the tree based on its surroundings. The research predominantly utilized computer vision, deep learning, and a sliding mode controller to design a mobile robot tailored for fruit harvesting in agriculture.

Chapter 3

Modeling of Mobile Manipulator

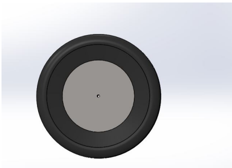
3.1 Introduction

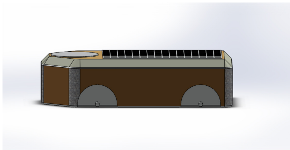
Mobile manipulators consist of two primary subsystems known as the mobile platform and the multi-link manipulator. The mobile platform is responsible for positioning the manipulator in a specific location within the workspace. On the other hand, the manipulator itself is responsible for moving the end effector to carry out tasks such as harvesting objects or fruits. This distinction between the two subsystems is crucial. In the case of a mobile manipulator with a robot arm mounted on a mobile base, the concept of mobile manipulation emerges. It involves coordinating the movement of both bases and robot joints to achieve desired motion of the end effector. Generally, the arm's rotation can be controlled more precisely than the base's direction of travel. As a result, the most common approach to mobile robot manipulation is to drive the base, park it, allow the arm to carry out precise motion tasks, and then drive away afterward [28]. This section focuses on verifying and analyzing the mechanical model of the system, as well as the mathematical modeling that encompasses the kinematics and dynamics of the mobile manipulator.

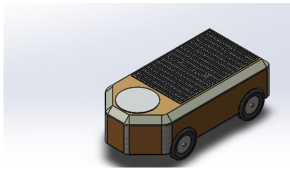
3.2 Mechanical Design of a Mobile Manipulator

The mechanical arrangement of the mobile manipulator was designed using Solid-Work software. This software is utilized for designing a $3D$ model of the mobile manipulator, which consists of a mobile platform equipped with a manipulator. The mobile manipulator's overall structure is composed of the main body of the mobile platform, including the walking device, the wheels, and the driving, power, sensing, and controlling devices.

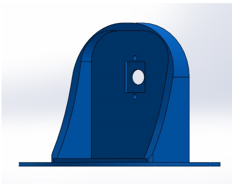
The chassis is attached to the main body's bottom. Two caster wheels are located at the back of the chassis to increase stability. The designed $3D$ model gives details on the fit and functionality of each and every component. The details of each part of the mobile manipulator designed on SolidWork before assembly are shown in figure (3.1).


(a) Wheels

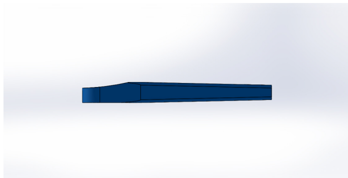

(b) Base

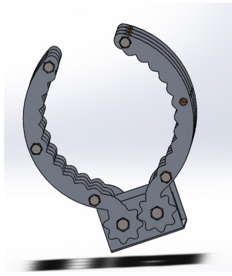

(c) Platform

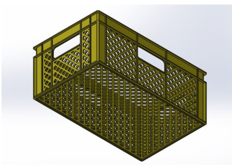

(d) Upper-Base

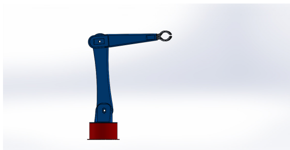

(e) Upper-Base2


(f) Lower-Link


(g) Upper-Link


(h) Gripper


(i) Crate


(j) Manipulator

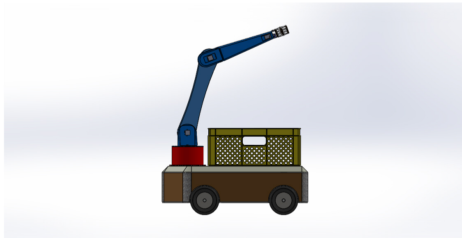

(k) Assembled MM

Figure 3.1: CAD Design of Each Part of the Mobile Manipulator

Figures (3.2) depict a mobile platform with a five-degree-of-freedom manipulator mounted on it, forming a 3D virtual assembly.

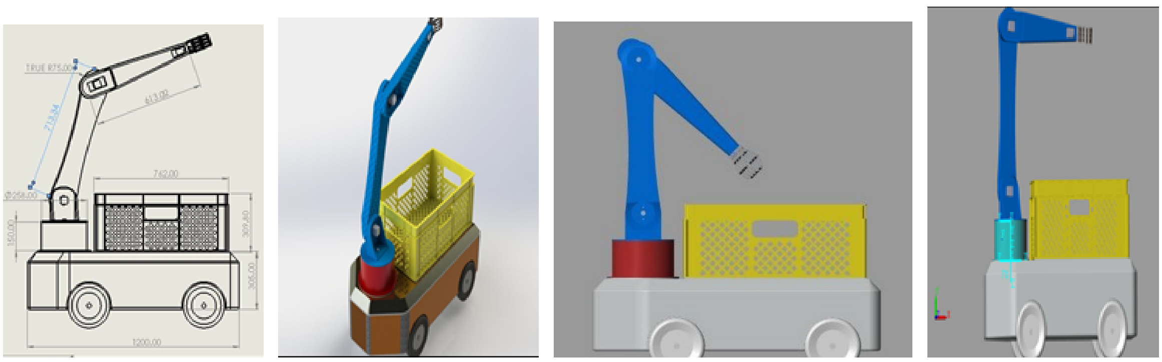


Figure 3.2: Virtual assembly of MM from Solid-Work to Simulink-Simscape

3.3 Model Verification

This section provides an overview of how the mechanical design of a mobile manipulator robot is validated. Firstly, the 3D CAD model of the robot is imported, and subsequently, all the components are virtually assembled within the MATLAB/Simulink/Simscape environment. Within this environment, the dynamic behavior of the kinematic chain is analyzed using the Simscape tool. With SimScape, users can input the forces impacting the mechanism and its intended movement, while also gathering the forces transmitted throughout the entire kinematic chain as output data. The mechanical design of the mobile manipulator that was assembled using SolidWorks was imported into MATLAB/Simulink. Without actuation, the system was verified for different positions, demonstrating the accuracy of the mechanical design. This process helps to ensure that the robot will work as intended without any unforeseen issues arising during its intended use.

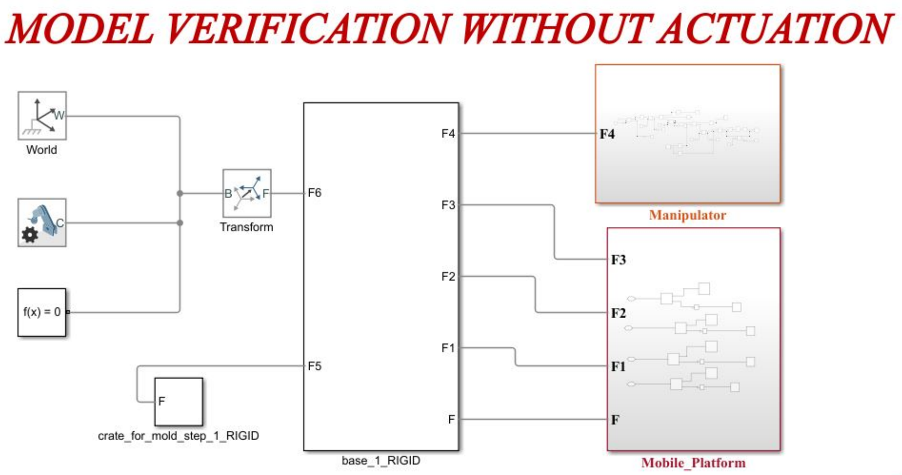
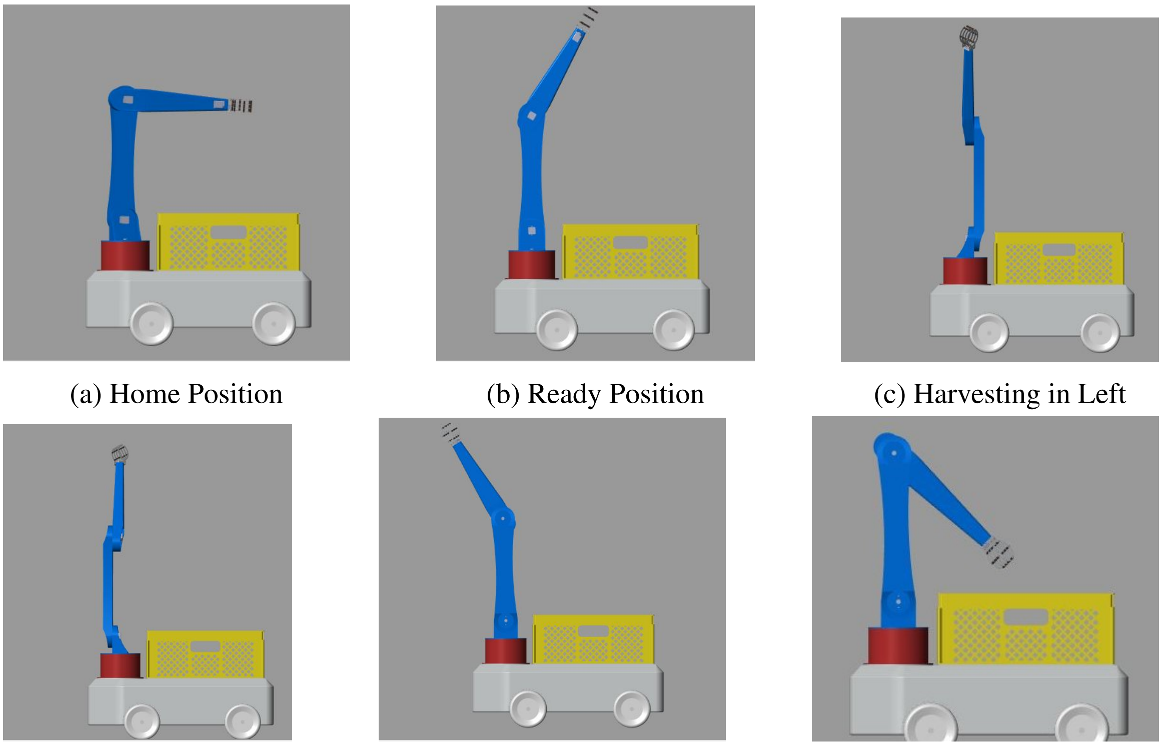


Figure 3.3: Simulink Model of a Mobile Manipulator Without Actuation

Figure (3.4) of the mechanics-explorer shows different positions of the mobile robot with a mounted manipulator. In order to obtain this value, it was hypothesized that the wheels on the mobile platform were not in motion, indicating that the mobile manipulator stayed still and did not shift from its parked position. The robot uses this specific parking position as a starting point for its movements. Even though the robot remains stationary in this position, its manipulator plays a crucial role in harvesting or picking up an object from a desired spot. This shows that while the robot's mobile platform may not be in motion, the manipulator can still carry out important tasks.



(a) Home Position (b) Ready Position (c) Harvesting in Left

(d) Harvesting in Right (e) Harvesting in Front (f) Placing Position

Figure 3.4: Visual Simulation of a Designed MM for Different Scenarios

3.4 Kinematics Analysis of Mobile Manipulator

Kinematic modeling involves analyzing the motion of a mechanical system, disregarding any external forces that may affect its movement. The primary characteristic of the mobile manipulator is its adaptable working environment. The coordinated movement of the manipulator and movable platform causes numerous redundancies. Developing a kinematic controller is crucial for the robot system to simultaneously follow the desired paths of both the end effector and the mobile platform in their workspace coordinates [29].

Sometimes, it is beneficial or even required for the end effector to move through a combination of the base and manipulator motions. This work examines the kinematic analysis of the entire system as described below. To conduct the analysis, define the fixed space frame S, the chassis frame B, the frame at the base of the manipulator O, and the end effector frame E. The position of E in relation to S can be referred to as its configuration.

$$X(q, \theta) = T_{SE}(q, \theta) = T_{SB}(q)T_{BO}T_{OE}(\theta) \tag{3.1}$$

Where: θ represents the joint positions of the five-joint robots,

$T_{OE}(\theta)$ refers to the forward kinematics of the manipulator,

T_{BO} is the fixed offset of point O from point B,

$q = (x, y, \varphi)$ signifies the planar configuration of the mobile base, and when z remains constant, it denotes the height of point b above the floor. The chassis frame B is positioned at the center of the four wheels, with a height of $635mm$ from the floor, and its configuration relative to a fixed space frame S is described by this.

$$T_{SB}(q) = \begin{bmatrix} cos\varphi & -sin\varphi & 0 & x \\ sin\varphi & cos\varphi & 0 & y \\ 0 & 0 & 1 & 0.635 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

The fixed offset between the manipulator's base frame O and the chassis frame B is determined when they are aligned and the base frame is displaced by $390mm$ in the x_B direction and $180mm$ in the z_B direction.

$$T_{BO} = \begin{bmatrix} 1 & 0 & 0 & 0.39 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.18 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

3.4.1 Kinematics Analysis of Mobile Platform

The kinematics models for wheeled mobile manipulators are designed considering both wheel sliding constraints and wheel rolling constraints. The analysis of the kinematics for manipulators is distinct from that of mobile robots with wheels [30]. To maintain simplicity in mathematical details without sacrificing generality, the mobile manipulator in this thesis consists of a firm frame fitted with non-flexible wheels that operate on a flat surface. Different types of steering are employed in the creation of mobile robots; the differential drive steering type is the most prevalent of these, as shown in figure (3.5). It is possible to independently control the wheels on each side of the mobile base. The robot has the ability to move in a straight path, rotate in one spot, track a predetermined route, or perform any other motion by coordinating its two separate speeds.

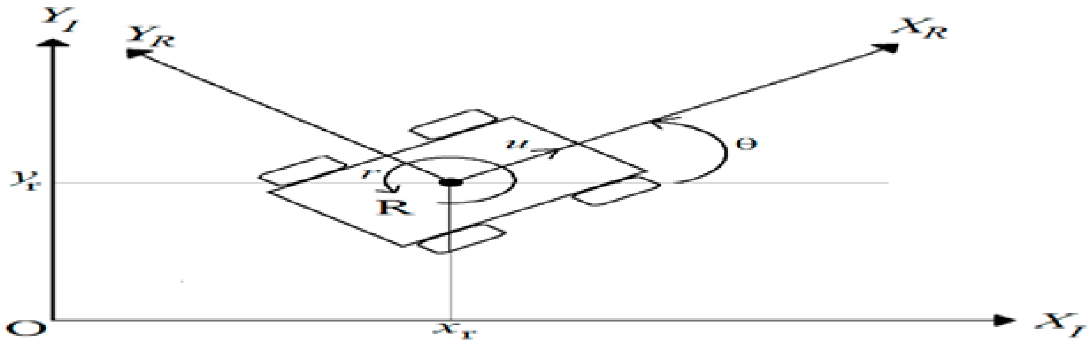


Figure 3.5: Schematic Diagram of Differential Drive Robot

A mobile manipulator designed in this work was a differential drive mobile robot that uses two wheels and two casters or balls on the back to move around. The motion of the robot is controlled by varying the velocities wheels. By making the two wheels spin in opposite directions, the robot can turn in place. The differential drive mobile robot's popularity stems from its ability to offer precise movement without being overly complex in both its design and operation. The kinematic properties of a rigid body moving on a $2-D$ planar surface without slipping are discussed below taking into account the following parameters: The right and left wheels' linear velocities in the reference frame can be expressed as follows, where the driven wheel distance is $2b = 600mm$, side-to-side distance is $2w = 750mm$, and the wheel radius is $r = 117.5mm$.

$$\begin{aligned} V_r &= r\dot{\theta}_r \\ V_l &= r\dot{\theta}_l \end{aligned} \tag{3.4}$$

$\dot{\theta}_r$ represents the angular velocity of the right wheel, while $\dot{\theta}_l$ denotes the angular velocity of the left wheel. The average linear velocity of the wheeled mobile base is:

$$V = \frac{(V_r + V_l)}{2} = \frac{(r\dot{\theta}_r + r\dot{\theta}_l)}{2} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \tag{3.5}$$

The heading angular velocity of the wheeled mobile base can be calculated as the disparity between the angular velocities of its right and left wheels, which amounts to:

$$\dot{\varphi} = \frac{(V_r + V_l)}{2b} = \frac{(r\dot{\theta}_r + r\dot{\theta}_l)}{2b} = \begin{bmatrix} \frac{r}{2b} & \frac{r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \tag{3.6}$$

Hence the mobile base is not having lateral skidding, its velocity is along the x-axis of its own frame, and the y-axis velocity becomes zero. Converging equations (3.5) and (3.6) kinematic model can be written as:

$$\begin{bmatrix} v_x \\ v_y \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \tag{3.7}$$

The velocity of the mobile base in the global coordinate system can be explained in relation to the local coordinate system as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} \frac{rcos\varphi}{2} & \frac{rcos\varphi}{2} \\ \frac{rsin\varphi}{2} & \frac{rsin\varphi}{2} \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \tag{3.8}$$

Then a robot position at some time t is obtained by integration of the velocity kinematic model obtained in Eqn. (3.8). Determination of the robot position for given control variables is forward kinematics:

$$\begin{aligned} x(t) &= \int_0^t V(t)cos(\varphi(t))dt \\ y(t) &= \int_0^t V(t)sin(\varphi(t))dt \\ \varphi(t) &= \int_0^t \omega(t)dt \end{aligned} \tag{3.9}$$

The designed mobile manipulator is of differential drive forward and rotates at the spot at constant velocities. From (3.9) the forward motion can be written as:

$$v_r(t) = v_l(t) = v_r \Rightarrow \theta(t) = 0, v(t) = v_r$$

$$x(t) = x(0) + v_r cos(\varphi(t))$$

$$y(t) = y(0) + v_t sin(\varphi(t))$$

$$\varphi(t) = \varphi(0) \tag{3.10}$$

For rotation motion

$$v_r(t) = -v_l(t) = v_r \Rightarrow \theta(t) = \frac{(2v_r)}{l}, v(t) = 0$$

$$x(t) = x(0), and y(t) = y(0)$$

$$\varphi(t) = \frac{\varphi(t) * (2v_r t)}{l} \tag{3.11}$$

The forward kinematics of the mobile robot is obtained as an equation (3.8). The process of finding the angular velocities of each wheel in a mobile robot based on its position and direction is known as inverse kinematics. The input for this process is information about the robot's position and orientation, while the output is the angular velocities of the fixed wheels. To obtain the inverse kinematics select the joined output $(\dot{y}, \dot{\varphi})$ from equation (3.8).

$$\begin{bmatrix} \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2sin\varphi} & \frac{r}{2sin\varphi} \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \tag{3.12}$$

From equation (3.12) the angular velocity of the differential drive mobile manipulator become:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} \frac{1}{rsin\varphi} & \frac{b}{r} \\ \frac{1}{rsin\varphi} & \frac{-b}{r} \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{\varphi} \end{bmatrix} \tag{3.13}$$

From equation (3.8)

$$\dot{x} = \frac{r}{2\dot{\theta}_r cos\varphi} + \frac{r}{2\dot{\theta}_l cos\varphi} \tag{3.14}$$

Solving for $\dot{\theta}_r$ and $\dot{\theta}_l$ from equation (3.14)

$$\dot{\theta}_r = \frac{2\dot{x}}{rcos\varphi - \dot{\theta}_l}; and \dot{\theta}_l = \frac{2\dot{x}}{rcos\varphi - \dot{\theta}_r} \tag{3.15}$$

By expressing the inverse kinematics of the mobile base as the angular velocities of the mobile robot, which can be rewritten in matrix form, equation (3.14) and (3.15) can be rearranged as:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} \frac{2}{rcos\varphi} & \frac{-1}{rsin\varphi} & \frac{b}{r} \\ \frac{2}{rcos\varphi} & \frac{-1}{rsin\varphi} & \frac{-b}{r} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \tag{3.16}$$

In the preceding subsection, the solution for the zero configuration of the manipulator's end effector frame E in relation to the base frame O is determined.

3.4.2 Forward Kinematics Analysis of the Manipulator

The objective of forward kinematics analysis is to determine the impact of joint variables on the position of the end effector in reference to a coordinate frame, as influenced by the movement of adjacent pairs of links [31]. The difficulty in analyzing forward kinematics lies in accurately establishing the location and alignment of the end effector in relation to a reference coordinate system, based on the joint angles provided. To investigate and examine the issue of forward kinematics for a robotic manipulator equipped with a movable base (i.e., the entire manipulator would be mobile since the arm is mounted on a platform with a wheel), common Denavit-Hartenberg (D-H) conventions and methodologies must be examined and verified. The DH convention makes it less challenging to determine the location and alignment of the end effector in relation to the base [32].

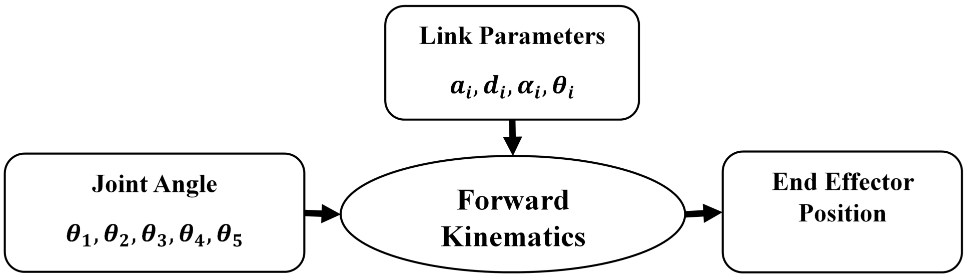


Figure 3.6: Block Diagram of Forward Kinematics Analysis

Denavit-Hartenberg (D-H) Convention

In the realm of robotics, the D-H convention is a well-known and often-used approach for selecting frames of reference. It is advantageous to be meticulous while selecting these frames, even if it is possible to examine the robot's kinematics by arbitrarily attaching a frame to each link. Each homogeneous transformation matrix T_i represents the result of four fundamental transformations according to the D-H convention [32]. This convention also includes the creation of a joint-link parameter table, which enables the identification of the (4×4) homogeneous transformation matrix appropriate for robot manipulators. Therefore, the subsequent step involves utilizing the D-H convention to establish the position and alignment of the manipulator's end effector.

Step 1: A robot consisting of n joints will be comprised of $n + 1$ links as each joint serves to connect two links. Every joint must have an axis assigned to it.

According to the D-H convention, joint i is used to connect link $i-1$ to link i.

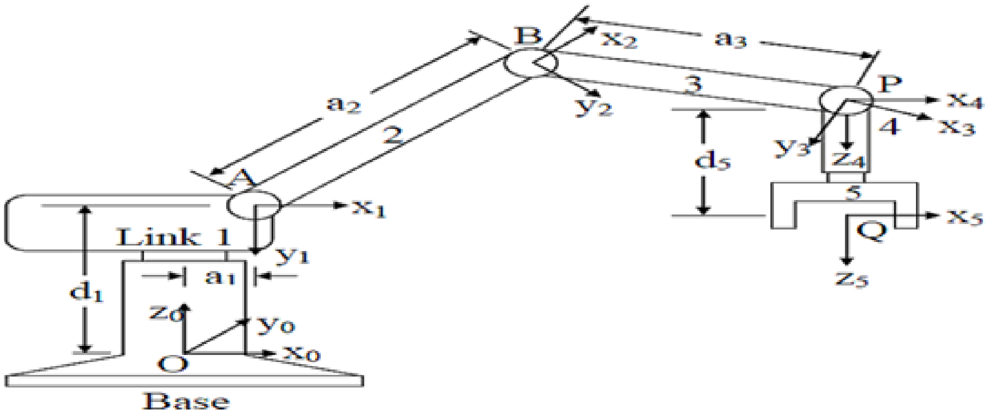


Figure 3.7: Axis Assignation of the Manipulator

Step 2: Establish a set of coordinates for each joint axis that is under control, including the end effector. Commence with the base joint and progress to the other joints, while applying a right-handed coordinate frame. The axis of motion of the (i^{th}) joint is always along $Z_{(i-1)}$ because all joints are revolute.

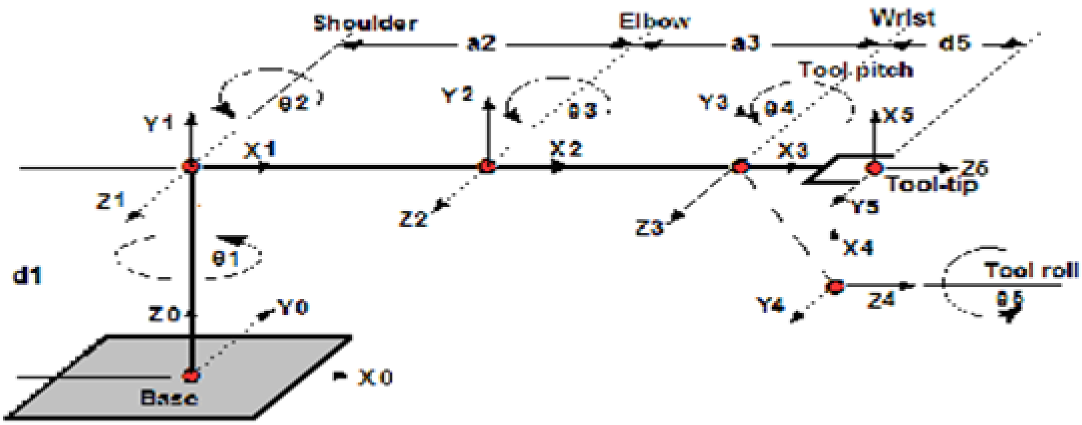


Figure 3.8: Coordinate Frame Assigning
[33]

Step 3: Build the D-H Convention joint-link parameter table.

Table 3.1: DH Parameter

Name of Joint	Joint angle θ	link length a in (m)	link offset d in (m)	offset angle α
Base	$\theta 1$	0	0.33	pi/2
Shoulder	$\theta 2$	0.79	0	0
Elbow	$\theta 3$	0.65	0	0
Wrist Roll	$\theta 4$	0	0	pi/2
Wrist Pitch	$\theta 5$	0	0.30	0

Step 4: Generate five (4×4) homogeneous coordinate transformation matrices using the joint-link parameter of each row of the D-H convention parameter tabulated in step 3. According to the D-H convention, every homogeneous transformation T_i can be expressed as the combination of four fundamental transformations and four independent variables.

$$T_i = Rot_{(z,\theta_i)} \times Trans_{(z,d_i)} \times Trans_{(z,a_i)} \times Rot_{(x,\alpha_i)} \tag{3.17}$$

Where: $Rot_{(z,\theta_i)}$ stands for rotation about Z_i axis by θ_i;
$Trans_{(z,d_i)}$ is the translation along Z_i axis by a distance d_i;
$Trans_{(z,a_i)}$ is translation along Z_i axis by a distance a_i ;
$Rot_{(x,\theta_i)}$ is rotation about X_i axis by α_i.

$$Rot_{(z,\theta_i)} = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & 0 \\ sin\theta_i & cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; Trans_{(z,d_i)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans_{(z,a_i)} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; Rot_{(x,\alpha_i)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha_i & -sin\alpha_i & 0 \\ 0 & sin\alpha_i & cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

Where the four quantities are parameters associated with link_i, θ_i is joint angle of the manipulator; α_i is link twist; d_i is link offset; and a_i, α_i, d_i, and a_i, are constant parameters for a given link and θ_i joint angle is the variable. Therefore, the function $T_i = T_i(\theta_i)$ depends on only one variable. According to the D-H convention, the mobile manipulator's transformation matrices that are homogenous are as follows:

$$T_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

Therefore, the transformation matrix between the end effector and the base frame is a combination of the following individual transformation matrixes and is defined as follows:

$$T_1^0 = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 & 0 \\ sin\theta_1 & 0 & -cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_2^1 = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & a_2cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & a_2sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_3^2 = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & a_3cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & a_3sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_4^3 \begin{bmatrix} cos\theta_4 & 0 & sin\theta_4 & 0 \\ sin\theta_4 & 0 & -cos\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_5^4 \begin{bmatrix} cos\theta_5 & -sin\theta_5 & 0 & 0 \\ sin\theta_5 & cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

Let's break down the forward kinematics of the manipulator into two more manageable sub-problems by dividing the computation at the wrist. After that, allocate terms using a trigonometric identity that can be expressed in the following way:

$$T_5^0 = T_1^0 T_1^2 T_2^3 T_3^4 T_4^5 \tag{3.21}$$

$$\begin{aligned} S_1 = \sin\theta_1; S_2 = \sin\theta_2; S_3 = \sin\theta_3; S_4 = \sin\theta_4; S_5 = \sin\theta_5; \\ C_1 = \cos\theta_1; C_2 = \cos\theta_2; C_3 = \cos\theta_3; C_4 = \cos\theta_4; C_5 = \cos\theta_5; \\ C_{12} = \cos(\theta_1 + \theta_2) = \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2 \\ C_{34} = \cos(\theta_3 + \theta_4) = \cos\theta_3 \cos\theta_4 - \sin\theta_4 \sin\theta_4 \\ S_{12} = \sin(\theta_1 + \theta_2) = \cos\theta_1 \sin\theta_2 + \sin\theta_1 \cos\theta_2 \\ S_{34} = \sin(\theta_3 + \theta_4) = \cos\theta_3 \sin\theta_4 + \sin\theta_3 \cos\theta_4 \end{aligned} \tag{3.22}$$

The transformation matrix of the manipulator mounted on a mobile plat-from in two sub-problems:

$$T_{base}^{wrist} = T_1^0 T_1^2 T_2^3$$

$$T_{wrist}^{tool} = T_3^4 T_4^5$$

$$T_5^0 = [T_{base}^{wrist}][T_{wrist}^{tool}] \tag{3.23}$$

$$T_{base}^{wrist} = T_1^0 T_1^2 T_2^3 = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & C_1(a_3C_{23}+a_2C_2) \\ S_1C_{23} & -S_1S_{23} & -C_1 & S_1(a_3C_{23}+a_2C_2) \\ S_{23} & C_{23} & 0 & d_1+a_3S_{23}+a_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

$$T_{wrist}^{tool} = T_3^4 T_4^5 = \begin{bmatrix} C_4C_5 & -C_4S_5 & S_4 & d_5S_4 \\ S_4C_5 & -S_4S_5 & -C_4 & -d_5S_1 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

The final expression for the forward kinematics of the manipulator is:

$$T_5^0 = \begin{bmatrix} C_1C_{234}C_5+S_1S_5 & -C_1C_{234}S_5+S_1C_5 & C_1S_{234} & C_1(a_2C_2+a_3C_{23}+d_5S_{234}) \\ S_1C_{234}C_5-C_1S_5 & -S_1C_{234}S_5-C_1C_5 & S_1S_{234} & S_1(a_2C_2+a_3C_{23}+d_5S_{234}) \\ S_{234}C_5 & -S_{234}S_5 & -C_{234} & d_1+a_2S_2+a_3S_{23}-d_5c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

Hence the homogeneous transformation matrix of the end effector is given as:

$$T_5^0 = T_{base}^{tool} = T_E = \begin{bmatrix} n_x & O_x & a_x & P_x \\ n_y & O_y & a_y & P_y \\ n_z & O_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

By equating equation (3.26) and (3.27) of the homogeneous transformation matrix, the end effector position obtained with respect to the manipulator base:

$$\begin{aligned} P_x &= C_1(a_2C_2+a_3C_{23}+d_5S_{234}) \\ P_y &= S_1(a_2C_2+a_3C_{23}+d_5S_{234}) \\ P_z &= d_1+a_2S_2+a_3S_{23}-d_5c_{234} \end{aligned} \tag{3.28}$$

and orientation of end-effector also obtained as:

$$O_x = -C_1C_{234}S_5+S_1C_5$$

$$O_y = -S_1 C_{234} S_5 - C_1 C_5$$

$$O_z = -S_{234} S_5 \tag{3.29}$$

According to equation (3.1) the forward kinematics of the mobile manipulator with respect to the world frame is the combination of Eqn. (3.2), (3.3), (3.26), and written as below:

$$X(q,\theta) = T_{SE}(q,\theta) =$$

$$\begin{bmatrix} \sin(\varphi)(C_1S_5 - C_{234}C_5S_1) + \cos(\varphi)(S_1S_5 + C_{234}C_1C_5) & \sin(\varphi)(C_1C_5 + C_{234}S_1S5) + \cos(\varphi)(C_5S_1 - C_{234}C_1S_5) & S_{234}(\cos(\varphi)C_1 - \sin(\varphi)S_1) & x + 0.39\cos(\varphi) + (\cos(\varphi)C_1 - \sin(\varphi)S_1)(a_3C_{23} + a_2C_2 + d_5S_{234}) \\ \sin(\varphi)(S_1S_5 + C_{234}C_1C_5) - \cos(\varphi)(C_1S_5 - C_{234}C_5S_1) & \sin(\varphi)(C_5S_1 - C_{234}C1S_5) - \cos(\varphi)(C_1C_5 + C_{234}S_1S_5) & S_{234}(\cos(\varphi)S_1 + \sin(\varphi)C_1) & y + 0.39\sin(\varphi) + (\cos(\varphi)S_1 + \sin(\varphi)C_1)(a_3C_{23} + a_2C_2 + d_5S_{234}) \\ S_{234}C_5 & -S_{234}S_5 & -C_{234} & d_1 + a_3S_{23} + a_2S_2 + d_5C_{234} + 0.815 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.30}$$

3.4.3 Singularity of Manipulator

Before traveling to inverse kinematic analysis for the robot manipulator mounted on the mobile base, it should be noted to check the singularity issue related to each joint and links of the manipulator. For the case of this work the singular position problem might emerge when frame $X_4Y_4Z_4$ rotates around parallel axes, at $\theta_4 = 90°$, which means that all links of the manipulator would be on the same plane. From the forward kinematic analysis, the transformation matrix from one frame to the next was stated, having that let check the singular position problem as follows:

$$T_5^0 = T_{base}^{tool} = T_E = \begin{bmatrix} n_x & O_x & a_x & P_x \\ n_y & O_y & a_y & P_y \\ n_z & O_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_5^0 = T_1^0 T_1^2 T_2^3 T_3^4 T_4^5 \tag{3.31}$$

Pre-multiply both sides of the equation (3.28) by $(T_1^0)^{-1}$ to verify the singularity of position.

$$\begin{aligned} (T_1^0)^{-1}(T_E) &= (T_1^0)^{-1}[T_1^0 T_1^2 T_2^3 T_3^4 T_4^5] \\ (T_1^0)^{-1}(T_E) &= (T_1^0)^{-1}[T_1^2 T_2^3 T_3^4 T_4^5] = T_5^1 \end{aligned} \tag{3.32}$$

Where:

$$T_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad and \quad [T_1^0]^{-1} = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.33}$$

At frame $X_4Y_4Z_4$ the values of $\theta_4 = 90°$

$$T_3^4 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34}$$

Solving equation (3.29)

$$\begin{bmatrix} C_1n_x + S_1n_y & C_1O_x + S_1O_y & C_1a_x + S_1a_y & C_1P_x + S_1P_y \\ n_z & O_z & a_z & P_z \\ S_1n_x - C_1n_y & S_1O_x - C_1O_y & S_1a_x + C_1a_y & S_1P_x + C_1P_y \\ \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} -C_5S_{23} & S_5S_{23} & C_{23} & d_5C_{23}) + a_3C_{23} + a_2C_2 \\ C_5C_{23} & -S_5C_{23} & S_{23} & d_5S_{23}) + a_3S_{23} + a_2S_2 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

Equating equation (3.32) from both sides, the solution for joint angles $\theta_1, \theta_2, \theta_3$ and θ_5, determined for a given desired transformatianal matrix T_5^0. This implies that there exists no singular position problem in this manipulator in case θ_4 is equal to $90°$. From equation (3.32), solving element-wise $(1, 4), (2, 4), (3, 4)$ get the following solution:

From (1.4) matrix element

$$C_1P_x + S_1P_y = d_5C_{23} + a_3C_{23} + a_2C_2; \tag{3.36}$$

From (2.4) matrix element

$$P_z + d_1 = d_5 S_{23} + a_3 S_{23} + a_2 S_2; \tag{3.37}$$

From (3.4) matrix element

$$\begin{aligned} S_1 P_x - C_1 P_y &= 0; \\ S_1 P_x &= C_1 P_y \\ \theta_1 &= \arctan 2(P_y, P_x) \end{aligned} \tag{3.38}$$

3.4.4 Inverse Kinematics Analysis

Inverse kinematic analysis is the process of determining each joint's values depending on the target position. In other words, rather than plugging in the known variables of the robot into the forward kinematic equations, one would solve the inverse of these equations to determine the required joint values for positioning the robot at the desired location. In practice, the inverse kinematic analysis is of greater significance because the robot controller utilizes these equations to compute the joint values and subsequently move the robot to the desired position [34]. For solving the inverse kinematics analysis of the manipulator, two types of solutions are considered: a closed-form (analytical) solution and a numerical solution. For two reasons, the analytical approach is urged and used in this study. For applications like welding seams, picking, and harvesting whose locations are provided by a vision system, we must quickly compute the inverse kinematic equation once every millisecond. The second argument in favor of closed-form solutions is that, unlike kinematic equations, which frequently have multiple solutions, closed-form solutions allow for the selection of a particular solution from a range of alternatives [32].

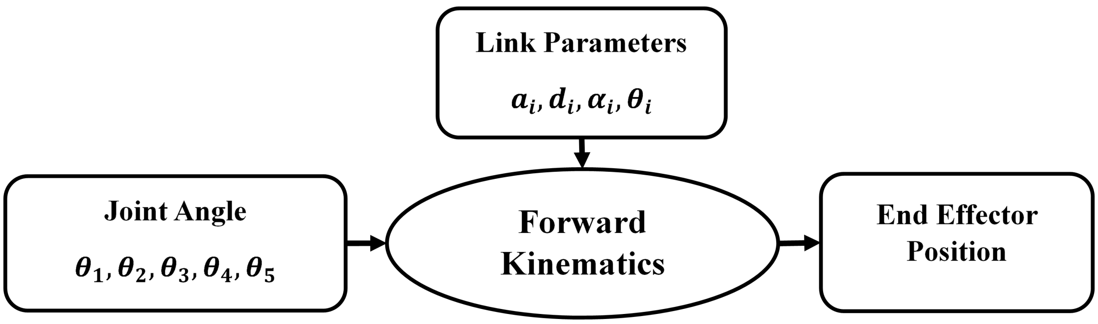


Figure 3.9: Block Diagram of Inverse Kinematics Analysis

Resolving $(T_5^0) = T_E$ by equating individual terms of both matrices of equation (3.26) and (3.27) obtains the solution for the inverse kinematic problem, which is pre-and-post-multiplying the two sides of loop closure equation, by the inverse of the matrix $(T_n)^{-1}$.

$$T_1^0 * T_1^2 * T_2^3 * T_3^4 * T_5^4 = T_5^0 = T_E \tag{3.39}$$

To solve for joint angles pre-multiplying both sides of equation (3.39) by $(T_1^0)^{-1}$.

$$(T_1^0)^{-1} * T_E = T_5^1 \tag{3.40}$$

The inverse matrix of the first frame (T_1^0);

$$[T_1^0]^{-1} = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.41}$$

Substituting Eqn.(3.41) into Eqn.(3.40)

$$\begin{bmatrix} C_1 n_X + S_1 n_Y & C_1 O_X + S_1 O_Y & C_1 a_X + S_1 a_Y & C_1 P_X + S_1 P_Y \\ n_Z & O_Z & a_Z & P_Z + d_1 \\ S_1 n_X - C_1 n_Y & S_1 O_X - C_1 O_Y & S_1 a_X - C_1 a_Y & S_1 P_X - C_1 P_Y \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} C_5 C_{234} & -S_5 C_{234} & S_{234} & d_5 S_{234} + a_3 C_{23} + a_2 C_2 \\ C_5 S_{234} & -S_5 S_{234} & -C_{234} & d_5 C_{234} + a_3 S_{23} + a_2 S_2 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.42}$$

By equating equation (3.42) element-wise, values of variable joint angles obtained as follows: From the $(3, 4)$

$$\begin{gathered} S_1 P_X - C_1 P_Y = 0 \\ \theta_1 = \arctan 2 \left(\frac{P_Y}{P_X} \right) \end{gathered} \tag{3.43}$$

From $(1, 3)$ and $(2, 3)$, θ_{234} can be isolated and found out:

$$\begin{aligned} C_1 a_X + S_1 a_Y &= S_{234} \\ a_Z &= -C_{234} \\ \frac{S_{234}}{C_{234}} &= -\frac{C_1 a_X + S_1 a_Y}{a_Z} \\ \theta_{234} = -\arctan 2 &\left(\frac{C_1 a_X + S_1 a_Y}{a_Z} \right) \end{aligned} \tag{3.44}$$

From the modified homogeneous transformation matrix of equation (3.42), the wrist roll joint variable is obtained by equating both sides of the equation element-wise $(3,1)$ and $(3,2)$ and dividing the first by the second as shown below:

$$\begin{aligned} S_1 n_X - C_1 n_Y &= S_5 \\ S_1 O_X - C_1 O_Y &= C_5 \\ \frac{S_5}{C_5} &= \frac{S_1 n_X - C_1 n_Y}{S_1 O_X - C_1 O_Y} \\ \theta_5 = \arctan 2 &\left(\frac{S_1 n_X - C_1 n_Y}{S_1 O_X - C_1 O_Y} \right) \end{aligned} \tag{3.45}$$

θ_5 is only used for the direction of the end effector and it is independent of the position of the end effector, it can take any value. Equating $(1,4)$ and $(2,4)$ of equation (3.42) element-wise obtain the values of variable angle:

$$\begin{cases} C_1 P_X + S_1 P_Y = d_5 S_{234} + a_3 C_{23} + a_2 C_2 \\ \qquad P_Z + d_1 = d_5 C_{234} + a_3 S_{23} + a_2 S_2 \end{cases} \tag{3.46}$$

Squaring both side of equation (3.46) and solving simultaneously:

$$\begin{aligned} C_1^2 P_X^2 + 2C_1 P_X S_1 P_Y + S_1^2 P_Y^2 + P_Z^2 + 2P_Z d_1 + d_1^2 = \\ d_5^2 (C_{234}^2 + S_{234}^2) + 2a_3 d_5 (S_{234} C_{23} + S_{23} C_{234}) + 2a_2 d_5 (C_3 C_{34} + \\ S_3 S_{34}) + 2a_2 a_3 (C_2 C_{23} + S_2 S_{23}) + a_2^2 (C_2^2 + S_2^2) + \\ a_3^2 (C_{23}^2 + S_{23}^2) \end{aligned} \tag{3.47}$$

Using trigonometric identity and collecting terms together:

$$\sin^2 \theta + \cos^2 \theta = 1$$

$$cos\theta_3 = cos[(\theta_2 + \theta_3) - \theta_2] = C_{23} = C_2 C_{23} + S_2 S_{23}$$

$$2a_3a_2C_{23} = [(C_1^2P_X^2 + 2C_1S_1P_XP_Y + S_1^2P_Y^2 + P_Z^2 + 2P_Z$$

$$d_1 + d_1^2) - d_5^2 - 2d_5S_{234}(a_2 + a_3) - a_2^2 - a_3^2]$$

$$C_3 = \left(\frac{(C_1^2P_X^2 + 2C_1S_1P_XP_Y + S_1^2P_Y^2 + P_Z^2 + 2P_Zd_1 + d_1^2) - d_5^2 - 2d_5S_{234}(a_2 + a_3) - a_2^2 - a_3^2}{2a_3a_2}\right) \tag{3.48}$$

In a similar manner using a trigonometric identity

$sin^2\theta + cos^2\theta = 1; sin\theta = \pm\sqrt{(1 - cos^2\theta}$ and $S_3 = \pm\sqrt{(1 - cos^2\theta_3)}$

$$\theta_3 = \arctan 2\left(\frac{S_3}{C_3}\right) \tag{3.49}$$

By switching up equations (3.46) solve for θ_2

$$\begin{aligned}
C_1P_X + S_1P_Y &= d_5S_{234} + a_3C_{23} + a_2C_2 \\
a_2C_2 &= C_1P_X + S_1P_Y - d_5S_{234} - a_3C_{23} \\
C_2 &= \left(\frac{C_1P_X + S_1P_Y - d_5S_{234} - a_3C_{23}}{a_2}\right) \\
P_Z + d_1 &= d_5C_{234} + a_3S_{23} + a_2S_2 \\
a_2S_2 &= P_Z + d_1 - d_5C_{234} - a_3S_{23} \\
S_2 &= \left(\frac{P_Z + d_1 - d_5C_{234} - a_3S_{23}}{a_2}\right)
\end{aligned} \tag{3.50}$$

From Eqn.(3.50) θ_2 can be obtained as:

$$\begin{aligned}
\theta_2 &= \arctan\left(\frac{S_2}{C_2}\right) \\
\theta_2 &= \arctan\left(\frac{P_Z + d_1 - d_5C_{234} - a_3S_{23}}{C_1P_X + S_1P_Y - d_5S_{234} - a_3C_{23}}\right)
\end{aligned} \tag{3.51}$$

Having values of joint variable for θ_2 and θ_3, from Eqn.(3.51) and Eqn.(3.49) respectively and, subtracting them from θ_{234} the values of joint variable θ_4 were determined:

$$\begin{aligned}
\theta_{234} &= \theta_2 + \theta_3 + \theta_4 \\
\theta_4 &= \theta_{234} - \theta 2 - \theta_3
\end{aligned} \tag{3.52}$$

3.5 Manipulator Jacobian and Workspace

3.5.1 Manipulator Jacobian

The Jacobi-an matrix of the manipulator is described as the matrix that depends on the configuration and DOF and transforms the joint rates in the actuator space to a velocity state in the end-effector space. For some configurations of manipulators, the Jacobi-an matrix may lose its full rank, which means the manipulator loses one or more DOF. Such a condition is called the singular condition [32]. The inverse of the linear mapping can be used to obtain the velocities of the link frame that correspond to specific end-effector linear and angular velocities.

$$\begin{aligned} V_n^0 &= J_v\dot{\theta} \\ \omega_n^0 &= J_\omega\theta \\ \dot{X} = J_n^0\dot{\theta} &= \begin{bmatrix} V_n^0 \\ \omega_n^0 \end{bmatrix} \end{aligned} \tag{3.53}$$

Where V_n^0 is Linear velocity
ω_n^0 is Angular velocity
J_n^0 is manipulator Jacobian
All joints are revolute, and the combined Jacobians n-link manipulator is stated as follows:

$$\begin{aligned} J &= [J_1 J_2 J_3....] \\ J_i &= \begin{bmatrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{bmatrix} \end{aligned} \tag{3.54}$$

The determination of the manipulator's Jacobian is straightforward with the information provided in equation (3.54). Once the forward kinematics are calculated, all the necessary quantities are available. To compute the Jacobian, only the unit vectors Z_i and the origin coordinates $O_1, O_2, ..., O_n$ are required. Upon further consideration, it becomes evident that the coordinates for Z_i in relation to the base frame can be obtained from the first three elements of the third column in T_i^0. Similarly, the coordinates for O_i can be derived from the first three elements of the fourth column in T_i^0. Therefore, assessing the Jacobian only necessitates the third and fourth columns of the transformational matrices T [32].

$$J = [J_1 J_2 J_3 J_4 J_5]$$

$$J_\theta = \begin{bmatrix} Z_0 \times (O_5 - O_0) & Z_1 \times (O_5 - O_4) & Z_2 \times (O_5 - O_3) & Z_3 \times (O_5 - O_3) & Z_4 \times (O_5 - O_4) \\ Z_0 & Z_1 & Z_2 & Z_3 & Z_4 \end{bmatrix} \quad (3.55)$$

From the forward kinematics of the manipulator:

$$T_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_2^0 = \begin{bmatrix} C_1C_2 & -C_1S_2 & S_1 & a_2C_1C_2 \\ S_1C_2 & -S_1S_2 & -C_1 & a_2S_1C_2 \\ S_2 & C_2 & 0 & d_1 + a_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_3^0 = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & a_3C_1C_{23} + a_2C_1C_2 \\ S_1C_{23} & -S_1S_{23} & -C_1 & a_3S_1C_{23} + a_2S_1C_2 \\ S_{23} & C_{23} & 0 & d_1 + a_3S_{23} + a_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_4^0 = \begin{bmatrix} C_1C_{234} & S_1 & C_1S_{234} & a_3C_1C_{23} + a_2C_1C_2 \\ S_1C_{234} & -C_1 & S_1S_{234} & a_3S_1C_{23} + a_2S_1C_2 \\ S_{234} & 0 & -C_{234} & d_1 + a_3S_{23} + a_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_5^0 = \begin{bmatrix} C_1C_{234}C_5 + S_1S_5 & -C_1C_{234}S_5 + S_1C_5 & C_1S_{234} & C_1(a_2C_2 + a_3C_{23} + d_5S_{234} \\ S_1C_{234}C_5 - C_1S_5 & -S_1C_{234}S_5 - C_1C_5 & S_1S_{234} & S_1(a_2C_2 + a_3C_{23} + d_5S_{234}) \\ S_{234}C_5 & -S_{234}S_5 & -C_{234} & d_1 + a_2S_2 + a_3S_{23} - d_5C_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.56)$$

Substituting Eqn. (3.56) into Eqn. (3.55) the Jacobian of the manipulator becomes:

$$J(\theta) =$$

$$\begin{bmatrix} -S_1(a_2C_2 + a_3C_{23} + d_5S_{234}) & -C_1(a_2C_2 + a_3S_{23} - d_5C_{234}) & -C_1(a_3S_{23} - d_5C_{234}) & d_5C_1C_{234} & 0 \\ C_1(a_2C_2 + a_3C_{23} + d_5S_{234}) & -S_1(a_2C_2 + a_3S_{23} - d_5C_{234}) & -S_1(a_3S_{23} - d_5C_{234}) & d_5S_1C_{234} & 0 \\ 0 & a_2C_2 + a_3C_{23} + d_5S_{234} & a_3C_{23} + d_5C_{234} & d_5S_{234} & 0 \\ 0 & S_1 & S_1 & S_1 & C_1S_{234} \\ 0 & -C_1 & -C_1 & -C_1 & S_1S_{234} \\ 1 & 0 & 0 & 0 & -C_{234} \end{bmatrix} \quad (3.57)$$

The Jacobian $J(\theta)$ defines a mapping

$$\dot{X} = J(\theta)\dot{\theta}$$

$$\dot{\theta} = J^{-1}(\theta)\dot{X} \tag{3.58}$$

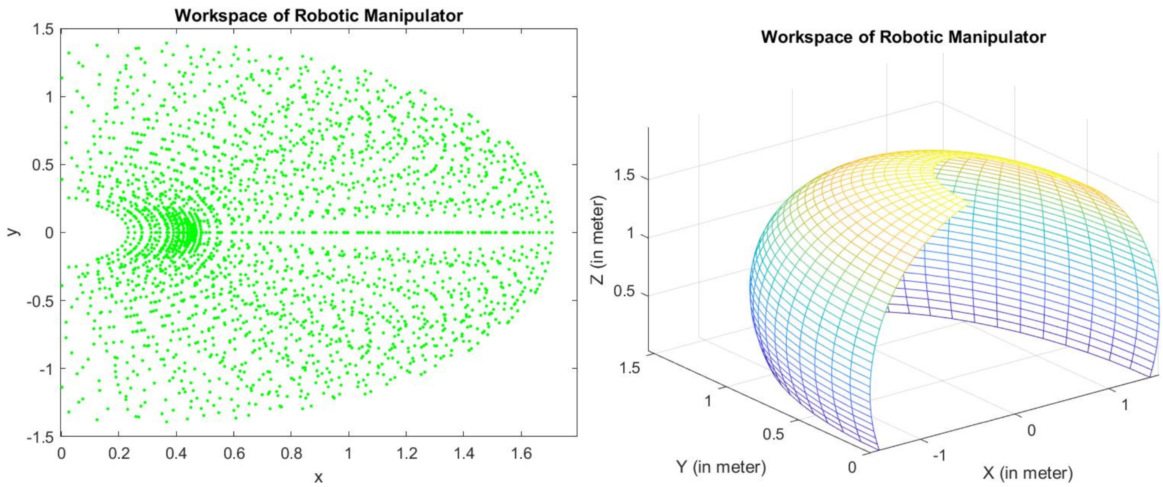
The linear mapping of Eqn. (3.58) exists only for configurations at which the inverse of the Jacobi-an matrix exists, which is a non-singular configuration. In a singular configuration, the manipulator's workspace may not be accessible, and the end effector may not be able to move in certain directions.

Consequently, when the manipulator loses one or more degrees of freedom, the rank of matrix J decreases and multiple columns of J become linearly dependent, resulting in a determinant of the Jacobian being zero. Owing to the fact that most robotic manipulators are designed for tasks that require all degrees of freedom and a reachable workspace despite perturbations, singular configurations are usually steered clear of. In this study, a constant nominal configuration is specified for the manipulator, ensuring that no singularities or joint limitations exist near the nominal configuration. The choice of the nominal configuration can be determined by either maximizing manipulability or considering the available joint range.

3.5.2 Work-Space of the Manipulator

The actual physical space in which a mechanism moves is referred to as task space in the study of mechanisms. This area is frequently used to specify the desired location and orientation of the mechanism's end effector. The precise volume that the end effector may access is referred to as the manipulator workspace. However, it is most straightforward to express the mechanism's articulations in terms of angles and axes displacements. Formally, any combination of variables that, when supplied, determine the position of each point on an object is included in the system's configuration space [35]. The manipulator's workspace is the set of all possible positions and orientations that can be reached by manipulating its joints within their respective limits. When both the position and orientation of the end effector are taken into account, it is called the complete workspace. However, if the end effector's orientation is not considered, then only the achievable workspace is considered.

The nimble working area is a part of the available working area that is accessible regardless of the direction the end effector is in. Nevertheless, it is not possible to create a flawless manipulator due to manufacturing tolerances, assembly mistakes, and other constraints [31]. In this study, the software realized the design specification and the graphs corresponding to the robot manipulator's workspace are shown in figure (3.10). These graphs show a selection of the Cartesian coordinate point successions that were discovered from the robot's articulation positions. It provides an idea of the positions that can be reached within the Cartesian system in this fashion. Most positions that are within a radius can be obtained to control the manipulator positioned on top of the moving platform.



(a) 2D-Workspace of Manipulator (b) 3D-Workspace of Manipulator

Figure 3.10: Workspace of Manipulator

3.6 Dynamic Analysis of Mobile Manipulator

Kinematic and dynamic modeling, as well as model control strategies based on dynamic models, are all part of the overall robot design. In section (3.4), a brief description was provided of the kinematics analysis of a mobile manipulator. In this section, the dynamics analysis of the system was covered, along with the model that depicts the mathematical relationship between the robot's motion, the driving torque, and the load. The thesis describes the design of a mobile robot that comprises a manipulator placed on a mobile platform. This robot offers a workspace that is considerably larger compared to a fixed manipulator. This can be achieved by fusing the flexibility of mobile platforms with the manipulation power of fixed base manipulators mounted on the top of a mobile platform.

The mobile manipulator, which can carry out both jobs at once, has advantages over stationary manipulators, including a larger work area and greater autonomy due to its wireless nature [36]. In order to control the path of the final output tool properly, it is necessary to acquire the integrated dynamic model of mobile manipulators. There are two ways to calculate the connected dynamics of a mobile manipulator: the closed-form Lagrange formulation and the forward-backward-recursive Euler formulation. It is possible to view and manage the mobile manipulator as a unified entity using the Lagrangian method. The system's dynamic equation, which is given below, is produced using this method:
Lagrangian function of the system,

$$\begin{aligned} L(\theta,\dot{\theta}) &= K(\theta,\dot{\theta}) - P(\theta) \\ K(\theta,\dot{\theta}) &= \frac{1}{2}(\dot{\theta})^T M(\theta)\dot{\theta} \\ P(\theta) &= -\sum_{n=1}^{n} M_i g^T T_i^0 r_i^{-i} \end{aligned} \tag{3.59}$$

Where:
$L(\theta,\dot{\theta})$ is the Lagrangian function of the system.
$K(\theta,\dot{\theta})$ is Kinetic energy.
$P(\theta)$ is Potential energy.
Then the dynamic equations:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \tag{3.60}$$

From Eqn. (3.59) and Eqn. (3.60) the vector form representation of the dynamic model for a mobile manipulator, which includes the dynamics of both the mobile platform and the manipulator with some limiting factors, is expressed as a combined dynamic system:

$$B\tau + f\lambda = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \tag{3.61}$$

Where:
$\theta = [\theta_p^T, \theta_m^T] \in \Re^n$
$\theta_p \in \Re^n$ is the vector of generalized coordinates of mobile platform.
$\theta_m \in \Re^n$ is the vector generalized coordinates of manipulator.
$n = n_p + n_m$ is sum of vector of mobile manipulator.
$M(\theta) \in \Re^{n\times n}$ is a positive definite inertia matrix.

$C(\theta, \dot{\theta}) \in \Re^n$ is the centripetal and coriolis force.

$G(\theta) \in \Re^n$ is the gravitational force.

$B(\theta) \in \Re^{n\times k}$ is full range input transformation matrix.

$\tau \in \Re^k$ is the vector control input.

$f \in \Re^n$ is the vector constraint forces.

By adding the constraints into the dynamics equation through the use of Lagrange multipliers λ, Eqn. (3.60) is modified as:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} + \Phi^T(\theta)\lambda \tag{3.62}$$

$$\begin{bmatrix} \lambda_h^T & \lambda_n^T \end{bmatrix}^T$$

Where $\Phi(\theta) \in \Re^{k\times n}$ is set of k velocity constraints.

λ_h is the constraint force linked to the holonomic constraints and

λ_n corresponds to the constraint force associated with non-holonomic constraints

Algebraic equations, utilizing position variables, can depict holonomic constraints observed in mechanical systems. If these constraints are independent and continuously differentiable, their elimination can be achieved through analytical means. Non-holonomic systems are those where motion is restricted and cannot be expressed as derivatives of generalized coordinate functions with respect to time. Typically, these restrictions can be stated as a non-integrable linear velocity relationship [37]. For a mobile manipulator, there are three kinematic constraints that apply to the mobile platform. The initial constraint necessitates the movement of the mobile platform along the axis of symmetry.

$$\dot{x_w}\sin\varphi + \dot{y_w}\cos\varphi = 0 \tag{3.63}$$

The position coordinates of the center of mass P_w in the inertial system are given by (x_w, y_w). Additionally, there are rolling constraints that prevent the driving wheels from slipping.

$$\begin{cases} \dot{x_w}\cos\varphi + \dot{y_w}\sin\varphi + L\dot{\varphi} = r\dot{\theta}_r \\ \dot{x_w}\cos\varphi + \dot{y_w}\sin\varphi - L\dot{\varphi} = r\dot{\theta}_l \end{cases} \tag{3.64}$$

If the mobile platform satisfies non-holonomic constraints (without slipping), the subsequent constraint is maintained.

$$\Phi(\theta)\dot{\theta} = 0 \tag{3.65}$$

The mobile platform adheres to three constraints: one holonomic and two non-holonomic constraints. These constraints are expressed as follows:

$$\begin{aligned} \varphi &= \frac{r}{2L}(\theta_r - \theta_l) \\ \dot{x_w}\sin\varphi + \dot{y_w}\cos\varphi &= 0 \\ \dot{x_w}\sin\varphi + \dot{y_w}\cos\varphi &= \frac{r}{2}(\dot{\theta}_r + \dot{\theta}_l) \end{aligned} \tag{3.66}$$

Based on the coordination scheme put forward in this study, the mobile platform assumes the task of positioning the manipulator at a designated point within the workspace. Examining the depicted mobile manipulator represented in figure (3.1k), it can be observed that the mobile platform does not exhibit holonomic behavior. To describe the dynamic model of the system the following notation is used:

θ_r : and θ_l : angular displacement of the right and the left driving wheel.

$\theta_1, \theta_2, \theta_3, \theta_4$ and θ_5 joint angles of manipulator.

Φ : refers to the angle at which the mobile platform is heading.

m_p : mass of the mobile platform.

m_w : mass of each driving wheel of a mobile platform.

m_m : mass of the manipulator.

m_1, m_2, m_3, m_4, and m_5 the mass of links- $1, 2, 3, 4$ and, 5.

I_1, I_2, I_3, I_4 and I_5 the moment of inertia of links-$1, 2, 3, 4$ and, 5.

I_w : represents the moment of inertia of each object rotating around its center of mass.

I_d : the moment of inertia of the wheel about the wheel diameter.

P_w : the mid-point of the wheel baseline.

d_1 : refers to the distance that is equal to half the length of the baseline.

d_2 : distance between the center of mass and P_w.

By calculating the derivative with respect to time, the motion formula of the mobile manipulator robot can be expressed as a matrix. Detailed explanations of each element in the matrix are given in the appendix (B).

$$B(\theta)\tau + f\lambda = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{3.67}$$

The dynamic equations of mobile manipulators given above are complex and nonlinear equations. The important fixed variables, like the masses and lengths of the links, offsets of the links, angles of the joints, and inertia moments are seen as factors in established equations using generalized coordinates.

3.7 Actuator Size Determination

One of the most essential choices one will make while building and constructing a robot is the choice of the drive motors. Since it embodies the fundamental collision between the ideal and the real world. When selecting an appropriate actuator for the robot in general, it is important to understand the loads the mobile robot will impose on the motor as well as the performance the motor is capable of giving. Robot joints can be actuated using a variety of actuator types and sizes. They can be categorized as hydraulic, pneumatic, or electric actuators according to their energy consumption. For the work, an electric actuator was employed. The most important step in the mechanical design of serial robot mechanisms is the selection of the actuator for each joint and its size.

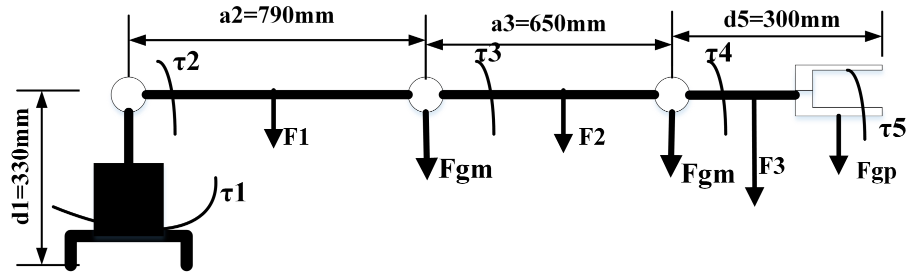


Figure 3.11: Diagram used for Predetermination of Joint Torque

Following are the joint torque calculations for each joint. Applying the general theorem of force and torque calculation:

$$\tau = F \times r \tag{3.68}$$

Where:

$W = F = mg$ is the force holding the mobile robot to the earth's surface because of gravity.

r is assigned for the radius of the wheel and the center of gravity of each link.

To determine the torques for robotic manipulators the following issues were considered:

- Capability of the manipulator to handle an object, in this case, fruit with a maximum mass of $1.5Kg$.
- Length of each links of the manipulator $d_1 = 0.33m, a_2 = 0.79m, a_3 = 0.65m, d_5 = 0.3m$ of total length $l_t = 2.07m$.
- Center of gravity for links in assembly, were considered to be placed in the middle of the segments $[d_1/2, a_2/2, a_3/2]$.
- Maximum mass of each links $m_1 = 1.34Kg, m_2 = 4.57Kg, m_3 = 3.15Kg, m_{EE} = 0.16Kg$ of total mass $M_m = 9.22Kg$.

Considering the pre-condition stated above the torque or actuator size determination for the mobile manipulator designed was calculated starting from the end-effector to the base step by step.

$$\tau_5 = g(m_5\frac{d_5}{2}) = (.16 * 9.81 * .15) = 0.235Nm$$

$$\tau_4 = g(m_p d_5 + m_5 d_5/2] = ((1.5 * 9.81 * .3) + (0.16 * 9.81 * 0.15)) = 4.65Nm$$

$$\tau_3 = g(m_p a_3 + m_3\frac{a_3}{2}) = ((1.5 * 9.81 * .65) + (3.15 * 9.81 * .325)) = 19.61Nm$$

$$\tau_2 = g(m_p(a_2 + a_3) + m_3(a_2 + \frac{a_3}{2}) + m_2\frac{a_2}{2})$$

$$= ((1.5 * 9.81 * 1.44) + (3.15 * 9.81 * 1.115) + (4.57 * 9.81 * 0.395)) = 73.35Nm$$

$$\begin{aligned} \tau_1 &= g(m_p(d_1 + a_2 + a_3) + m_3(d_1 + a_2 + \frac{a_3}{2}) + m_2(d_1 + \frac{a_2}{2}) + m_1\frac{d_1}{2}) \\ &= ((26.04555) + (44.6526675) + (32.503) + (2.12043) = 105.4Nm \end{aligned} \tag{3.69}$$

Next, choose an actuator size for the mobile platform by taking into account the combined weight of the mobile platform, manipulator, and external payload that a mobile robot can generally carry. To determine what size of the actuator is needed for a mobile platform define the following term. The total mass of the mobile manipulator:

$$M_t = M_{load} + M_m + M_p + M_w + M_{crate} = 25 + 9.22 + 59.22 = 93.44Kg \tag{3.70}$$

Where:

$M_{load} = 25Kg$, external load the mobile robot can hold.

$M_m = 9.22Kg$, mass of manipulator mounted on mobile platform.

$M_p + M_w + M_{crate} = 53.4 + 4.32 + 1.5 = 59.22Kg$, a mass of mobile platform, a mass of wheels, and mass of create respectively.

Radius of wheels $r = 0.1175m$;

Desired maximum velocity $v = 0.7m/sec$.

Desired acceleration time $t_a = 1sec$;

Maximum incline angle $\gamma = 10°$.

Considering the worst working surface, concrete is good with surface coefficient $c_{sf} = 0.01$. Hence the mobile robot starts from the initial position and is required to accelerate to reach the desired position to perform the given tasks.

$$\sum F = M_t a \tag{3.71}$$

The summation of force acting on the robot is equal to zero.

$$\sum F = 0 = F - F_g \tag{3.72}$$

Where:

The force that pulls the robot down the incline due to gravity can be represented by the equation $F_g = M_t g \sin\gamma$.

F is the force pushing against the wheel.

Equating Eqn. (3.71) and (3.72)

$$M_t a = F - F_g$$

$$F = M_t a + F_g \tag{3.73}$$

By substituting Eqn. (3.68) the torque of the actuator becomes:

$$\tau = r(M_t a + M_t g \sin\gamma) \tag{3.74}$$

$$\tau = 0.1175[(93.44 * 0.4) + (93.44 * 9.81 * \sin\gamma)] = 23.1Nm$$

As the mobile manipulator is designed with a differential drive mechanism, the torque needed to operate it is distributed between the right and left wheel motors.

$$\tau_r = 11.55Nm \qquad and \qquad \tau_l = 11.55Nm$$

Presently, the majority of robots are driven by rotary electric motors. Industrial robots of significant size typically employ brushless servo motors, while smaller hobby or laboratory robots frequently rely on brushed DC or stepper motors. In this study, the manipulator's joints are actuated using servo motors, and differential drive wheels are equipped with DC motors. The table (3.2) contains comprehensive information on the sizes of actuators for both the manipulator's joints and the platform's wheels.

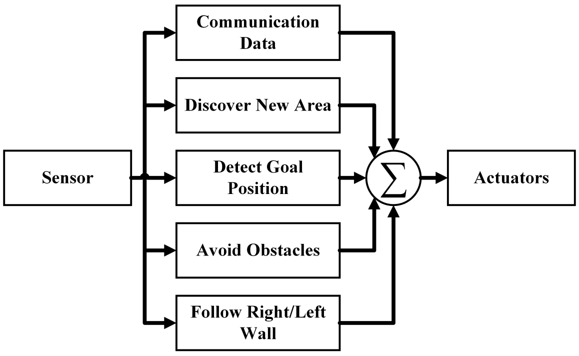
Table 3.2: General Specification of Actuator selected for the Model

Actuator Size Determination for Mobile Manipulator	
Joints	Torque(Nm)
Motor of Base	105.4
Motor of Shoulder	73.35
Motor of Elbow	19.61
Motor of Wrist-Roll	4.65
Motor of Wrist-Twist	0.235
Motor of the Drive Wheel (w_r and w_l)	11.55

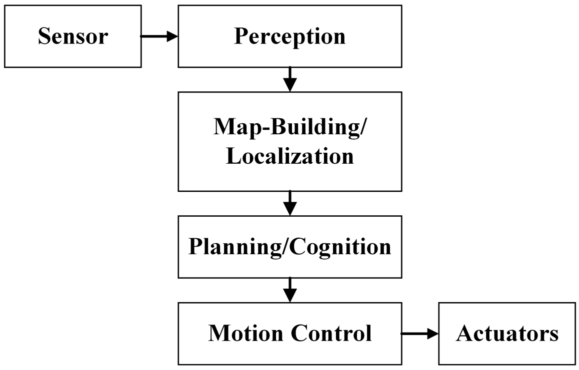
3.8 Mobile Robot Navigation

One of the most demanding skills needed for a mobile robot is navigation. To navigate successfully, one needs to have expertise in four basic constituents of navigation which include perception, localization, cognition, and motion control. In order to extract useful information from its sensors, a robot must interpret its senses during perception, localize itself within its surroundings during localization, think strategically during cognition, and modify its motor outputs during motion control [38]. For mobile robot navigation, there are two techniques: a behavior-based method and a map-based method. The behavior-based school of thought discourages creating a spatial representation for determining location as sensors and effectors produce imprecise signals and limited data. This method fundamentally avoids explicit path planning as well as explicit reasoning about localization and position. The primary advantage of this method is its ability to quickly implement in a specific setting with a small set of target locations. In map-based navigation, the robot actively tries to determine its location by collecting sensor information and adjusting its understanding of its position based on a map of the nearby region. The key advantages of using maps for navigation are as follows: The use of a map-based positional concept allows human operators to easily access the system's understanding of the position.

Additionally, the map serves as a communication medium between humans and robots. This study explores the map-based method for navigating mobile robots. To navigate autonomously, a robot needs to use its sensors to interact with the environment. When conducting a navigation task, a robot must answer three questions: where am I located? what does the world around me look like? and how can I reach my desired destination? The first two questions are related to perception, while the control system addresses the final question.



(a) Behavior Based Approach



(b) Map Based Approach

Figure 3.12: Navigation Approach of Mobile Robot

3.8.1 Perception

Acquiring knowledge about its surroundings is one of the mobile robot's most crucial duties, regardless of its type. This is accomplished by measuring with a variety of sensors and then deriving pertinent information from those measurements. Some sensors are used to measure fundamental factors like the temperature inside a robot's electronics or the rotational speed of its motors. It is also possible to employ more sophisticated sensors to gather further information about the robot's environment or to directly determine its spatial position [38]. The operational range of a sensor that will be utilized to avoid obstacles is an essential feature. It stands to reason that the wider the range, the more early warning the robot will receive. The length of time it will take to cease moving depends on the robot's speed, its wheels or tracks, and the type of surface it is traveling on. The distance between the robot and an obstacle should either exceed the range of the detection sensor, or the robot needs to move slowly enough to ensure that its stopping distance is within the sensor's range [39]. Numerous sensing tools were employed in mobile robot navigation, some of which are described below:

Camera: the purpose of cameras is to gather images of an object being inspected and can be utilized in various applications including factory automation and autonomous systems. The sensors in cameras are specifically built to capture focused light. With advancements in algorithms, modern computing abilities, and desirable qualities of cameras, they have become a preferred sensor for addressing issues related to robotics. Since images are projections of objects within the environment, cameras can be used to locate, identify, and follow objects within their field of view.

Light Detection and Ranging (LIDAR): Lidar functions as an active sensor that emits laser pulses towards a target and calculates the duration it takes for the pulses to bounce back to the sensor receiver. Comparable to radar, Lidar also employs laser pulses. However, Lidar commonly employs shorter wavelengths in the ultraviolet, visible, or near-infrared ranges, distinguishing it from other technologies that operate in different parts of the electromagnetic spectrum. The size of a feature or object that may be imaged is typically just slightly larger than the wavelength [39].

3.8.2 Localization and Mapping

Localization and mapping are crucial in determining the location of robots and their surrounding environment. Localization pertains to identifying where the robots are, while mapping involves visualizing what the area looks like. Both of these perceptions are important in understanding the outward and inward processes of robots moving around. In creating a mobile manipulator designed to perform fruit harvesting in a vast outdoor setting, comprehending the robot's location and the external environment from a random starting point was the main objective. The mobile manipulator needed to learn, interpret, and create an advantageous map while localizing itself within the map. Although technology improvements and computer vision have partially achieved this purpose,

A mobile robot's sensors can only see so far, of course, so to create such a map, the robot must physically explore its surroundings [39]. In order for the robot to observe and analyze its environment, it needs to not just create a map but also do it while it moves and determine its location. The term used for this idea in the field of robotics is "simultaneous localization and mapping" (SLAM), which refers to the simultaneous process of determining a robot's position and creating a map of its surroundings.

This research focuses on analyzing the utilization of a map specifically designed for modern farms. This map is created by dividing the farm area into distinct lines, where crops are planted in alignment with these lines. By adopting this line-based farming approach, robots are able to navigate through predetermined areas and effectively carry out harvesting tasks on the farm.

3.8.3 Cognition/Path Planning

Navigating from a starting point to a goal while avoiding static and moving obstacles can pose challenges for a mobile robot. Different researchers have devised a variety of path-planning strategies for mobile robots, some of which are computationally intensive. A path for the robot can be created with the aid of an overhead camera by taking and analyzing real-time photos [40]. Autonomous robots must be cognitive and should make their own decisions to complete tasks. When an autonomous robot decides what to do, it is important to plan well based on the tasks at hand. Additionally, a collision-free course must be planned while keeping costs like time, energy, and distance to a minimum. Planning an ideal or practical path that will get an autonomous robot from the initial point to the target point while avoiding obstacles in its environment is important. Most of the time, the best-preferred route would be the shortest collision-free path. Path planning refers to the task of finding the optimal route [41].

3.8.4 Motion Control

To answer the question of how the robot moves to a specific location, it is clear that a control system is crucial for navigating the robot. The control system supplies the required velocities as input to enable the robot to reach its intended destination. The challenge in robot control is determining the forces and torques needed for the robotic actuators to achieve the desired position, track a specific trajectory, and perform tasks that meet certain performance requirements. Managing robotics issues in both stationary and mobile environments is more complicated than usual because of the interplay between inertia forces, reaction forces, and the effects of gravity. The following chapter offers a thorough description of fruit detection, position determination, and the controller chosen for the designed mobile manipulator.

Chapter 4

Object Detection and Controller Design

4.1 Object Detection and Position Determination

These days, object detection manifests itself in various ways in the daily lives of human beings. For instance, smartphone users don't have to enter their passwords manually. Smartphone manufacturers have implemented face recognition to enhance the user experience while logging into the device. The smartphone precisely recognizes characteristics from the photograph and determines whether or not the user is the actual owner of the phone. Finding the object or fruit position that was input to the manipulator in the working environment was one of the objectives of this work.

In this instance, two steps were taken to establish the fruit's position in the tree: first, the selected deep learning object detection algorithm (the Yolov2 object detection algorithm) identified and confirmed the fruit as one that should be picked, and second, a position-based visual servoing control mechanism extracted image features to establish the fruit's position. Because vision sensors are increasingly being used in different fields, the following sections of this chapter will specifically address the fundamentals of computer vision and image processing and how they are applied in robotics.

4.2 Computer Vision

Vision is the most powerful sense for humans, providing access to a vast amount of environmental information and enabling deep, intelligent engagement in dynamic surroundings. Therefore, significant effort has been focused on creating robot sensors that can replicate the functions of the human eye.

Visual servoing control, which incorporates visual information in a closed-loop control system, utilizes data from vision sensors such as cameras to create a feedback loop. This control strategy capitalizes on the abundant and cost-effective information provided by machine vision and presents an excellent alternative to robot control. One of the key benefits of visual servoing control is its ability to manage multiple robots efficiently, as it requires fewer sensor data points. This scalability advantage eliminates the need for both internal and external robot sensors and allows for the incorporation of more imaging devices, resulting in expanded functional areas. Furthermore, the utilization of machine vision brings us closer to our ultimate goal of developing robotic systems capable of mimicking human control maneuvers.

The stability and robustness of the closed-loop system are the two critical difficulties that must be assured when the vision and control systems are combined in the so-called visual servoing technique. The main focus of the system's design is the utilization of advanced eye-in-hand camera configurations. This arrangement expands the applicability of the visual servoing technique for controlling mobile manipulators, as it enables the achievement of significant displacement tasks that would otherwise be challenging with a fixed-camera system. The classification of visual servoing techniques is based on the features of the error function, typically resulting in the subdivision of these methods into three distinct types [42, 43].

- Image-Based Visual Servoing (IBVS) is a control strategy that uses images to control robots by measuring the error between the projected desired and actual positions in the $2D$ image plane. This is done via an image Jacobean without reconstructing the target. However, IBVS lacks $3D$ in-depth knowledge of a target.

- Visual servoing with positional data, also known as position-based visual servoing (PBVS), refers to a methodology in robotics that relies on extracting a set of 3D characteristics from visual observations in order to compute the discrepancy or inaccuracy in the Cartesian task space. This approach enables precise control and manipulation of robots by utilizing the position information obtained through vision sensors.

- The process of combined visual steering that involves utilizing both IBVS and PBVS is known as mixed-mode visual steering. In this mechanism, the error function combines measurements from Cartesian data and image-based data.

The research utilized a method of controlling robots using a position-based visual servoing control mechanism with an eye-in-hand camera configuration. This method falls under the category of visual servoing schemes, and the following content will focus on this specific approach.

4.2.1 Position-Based Visual Servoing Control Techniques (PBVS)

The designed system employs a technique known as Position-Based Visual Servoing Control (PBVS) to identify the positioning of fruits in images. In the $3D$ workspace, this control technique directly manages the discrepancy between the desired and actual positions of the end-effector. One may easily and directly control where the end-effector is in relation to the target. The target's attitude with regard to the camera is ascertained by extracting features from the image and combining them with a geometric model of the target. The scheme's shortcomings include the position estimation's susceptibility to inaccurate target model accuracy, camera calibration problems, and image measurement noise.

The main goal of the PBVS strategy is to guide the robot's end effector toward a target object. This is achieved by using a built-in camera and reconstruction algorithm to obtain a 3D position estimate of the object. The process, as shown in figure (4.1), involves utilizing the vision system to determine the end-effector's position relative to the target object or fruit. The estimated position is then sent to a controller for further action.

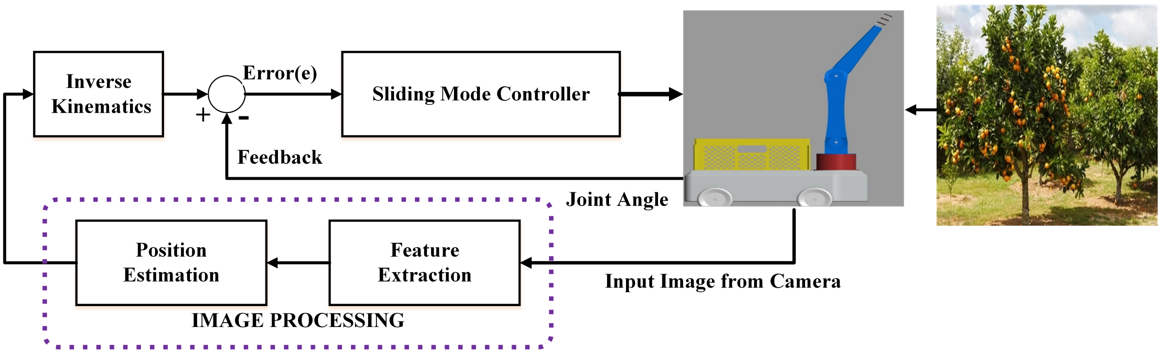


Figure 4.1: Block Diagram of Position Based Visual Control

The position of the robots was controlled using two control loops: the slower inner loop, which implemented motion control through joint control, and the faster outer loop, which used the inverse kinematic block to compute the trajectory for the end-effector based on the captured input image.

4.3 Deep Learning

The emergence of deep learning has brought a fresh dimension to machine learning, aiming to bridge the gap between machine learning and artificial intelligence. This approach enables the processing of vast quantities of data to recognize connections and trends that are often hard for humans to discern. The phrase “deep” pertains to the number of concealed layers composing the neural network, which primarily contributes to its ability to learn. Deep learning is a cutting-edge and widely utilized artificial intelligence (AI) technique that involves employing multilayered neural networks to extensively train the software, enabling it to enhance its proficiency in tasks such as speech and image recognition [44].

Because deep learning architectures possess millions of parameters that can be enhanced, they are often trained on extensive data sets such as ImageNet, PASCAL VOC, and COCO. For training and bench-marking object detection models, these online data sets provide free access to labeled data (images and annotated documents are available for different categories of objects). Deep learning models can be used by transferring the learned traits to different tasks. Transfer learning is a method for “retraining” or “fine refining” a model’s knowledge that was created on an extensive data set using relatively few images.

4.3.1 Convolution Neural Network (CNN)

Convolutional Neural Networks (CNNs), commonly referred to as ConvNets, are composed of multiple layers and are primarily utilized for tasks such as object detection and image processing. Originally known as LeNet, CNNs were developed by Yann LeCun in 1998 with the purpose of recognizing numerals and zip code characters. Over time, CNNs have found widespread applications in anomaly detection, series forecasting, medical image processing, and satellite image identification [45]. The process of performing convolutional operations involves passing the data through several layers to extract features. Rectified Linear Units (ReLUs) are employed in the convolutional layer to rectify the feature map, which is further modified using the pooling layer. The pooling layer typically involves downsampling to decrease the dimensionality of the feature map. Eventually, the output is generated as 2D arrays that have been flattened into a single, continuous vector. This flattened representation is then passed as input to the fully connected layer, also known as the classification layer, which is responsible for identifying and classifying the image [45].

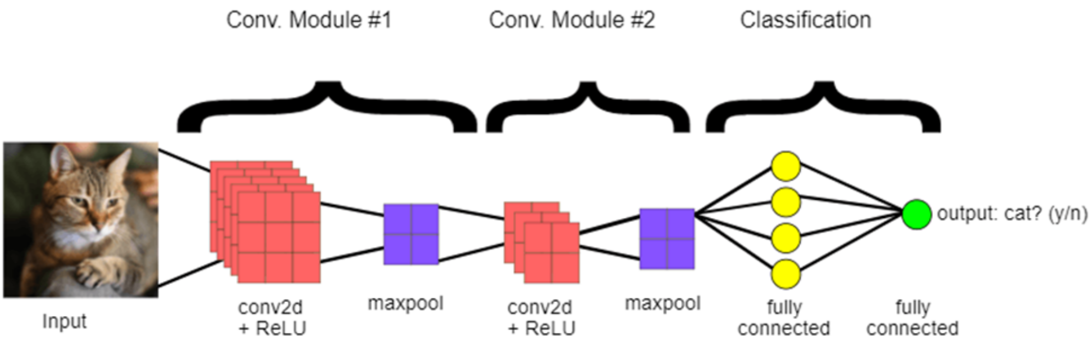


Figure 4.2: The Structure of a Convolutional Neural Network.
[45]

The CNN method is a good choice for quickly and accurately processing image data to label the objects in the input image. The CNN algorithms can be categorized into two groups: two-stage and one-stage techniques. The two-stage techniques, such as R-CNN algorithms, require region proposal identification and a lot of processing power. In contrast, one-stage techniques, like YOLO, Detector-Net, Over-Feat, Corner-Net, and SSD, solely employ convolutional neural network predictions of target categories and locations. One-stage frameworks use a single feed-forward CNN to apply class category prediction and bounding box regression straight to the entire image, without post-categorization or area proposal creation elements. The CNN algorithm that is most appropriate and applied in real-time processing for picking and harvesting fruit is a one-stage detection framework called the YOLO architecture.

4.3.2 YOLOv2 Object Detection Algorithm

The acronym YOLO stands for "You Only Look Once" and was the first genuine one-stage object detector. From picture pixels to bounding boxes and class probabilities, it framed the object detection task as a regression issue, by omitting the region proposal generation and verification processes. As suggested by its name, YOLO predicts detection using features from a whole image based on a limited number of candidate regions. The procedure to detect an object is to divide an input image into $n \times n$ grids. The YOLO algorithm predicts class probabilities, bounding box locations, and an objectiveness score all at once. Since it encodes contextual information about object classes and uses the entire image for prediction, it reduces the prediction of false positives from the background. YOLO is speedy because it bypasses the construction of the region proposal stage.

In comparison to two-stage framework methods like Fast RCNN, YOLO is faster [46]. Additionally, YOLO can detect objects at the same rate as video streaming frame rates. Therefore, as the study involves real-time detection and identification of fruit, it is vital to implement a deep learning technique that combines high accuracy and high speed, similar to YOLO.

YOLOv2 is state-of-the-art and faster than other detection systems across a variety of detection data sets. Furthermore, it can be run at a variety of image sizes to provide a smooth trade-off between speed and accuracy. The foremost intent of the algorithm is to address the regression challenge associated with recognizing objects. This is achieved by utilizing the entire image as the network input. The output layer of the algorithm returns the bounding box position (also referred to as the recognition box or simply box) together with the category that the box belongs to [47]. This deep-learning CNN object detection technique can satisfy the demands of real-time detection and is highly quick. Figure (4.3) illustrates the structure of the YOLOv2 network.

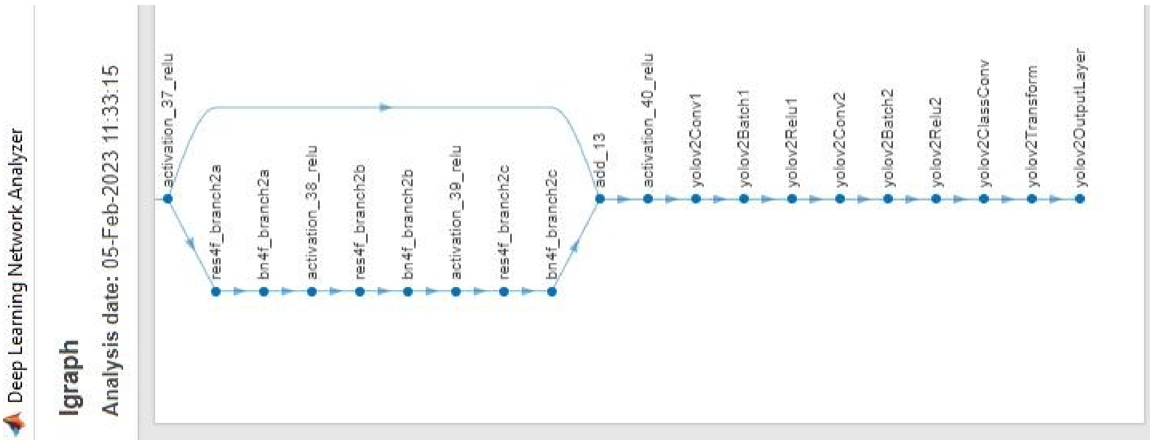


Figure 4.3: Architecture of YOLOv2 Fruit Detection Network

The configuration of the proposed detection model is stated in the table (4.1)

Table 4.1: Configuration of YOLOv2 Object Detector Designed for Mobile Manipulator

Model_Name	Image_Data	Trained	Validation	Test	Learning_Rate	Options
Yolov2	549	80%	10%	10%	0.001	SGDM

The effectiveness of the YOLOv2 deep learning CNN object detector was assessed through its operational efficiency in recognizing and determining the position of orange fruits.

The evaluation was carried out by measuring the Root Mean Squared Error (RMSE) and loss rate, which produced values of 0.19 and 3.6e-02, respectively. The results indicate that the network outputs accurate detections of orange fruits based on these metrics. In the figure (4.4), the RMSE and loss of the detection networks are depicted. The low RMSE value indicates that the predictions made by the network are highly precise. Meanwhile, the low loss rate attests to the network's ability to minimize errors and preserve critical information. Thus, the YOLOv2 deep learning CNN object detector network trained for orange fruit detection and position determination is effective based on these performance measures.

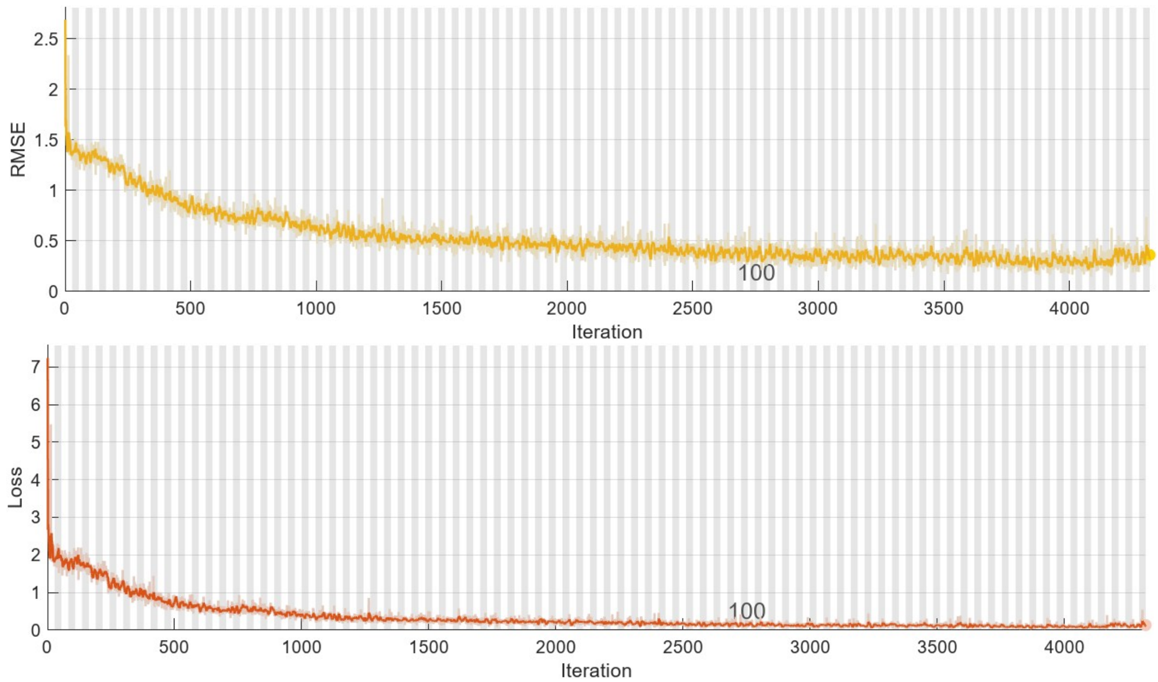


Figure 4.4: Evaluation of the Object Detection Network Using RMSE and Loss for Each Iteration

4.4 Position Determination

An image (such as a photo or video) serves as the input signal for image processing, which produces either an image or a set of variables that are related to the image. In image processing, the image is often represented as a two-dimensional signal denoted as $I(x, y)$, where x and y are the spatial coordinates of the image. At any given pair of coordinates, the amplitude of I is indicative of the intensity of the picture at that specific location.

The objective of this research is to highlight the key operations that aid robotics, namely object detection, and recognition, owing to the presence of common operations across the larger field of image processing. The object "fruit" was recognized and identified using the YOLOv2 object detection method; the $3D$ coordinates of the fruit, $P_f = (X_f, Y_f, Z_f)$, were then used to view its place and calculate its distance from the camera. A precise mapping from the three-dimensional real-world coordinates of the objects or fruit to the two-dimensional coordinates of a pixel in the imaging plane is required. By defining the coordinate system, the position of the object or fruit in reference to the camera coordinate is expressed. The transformation of perspective projection brings the camera coordinate to the physical coordinate [48].

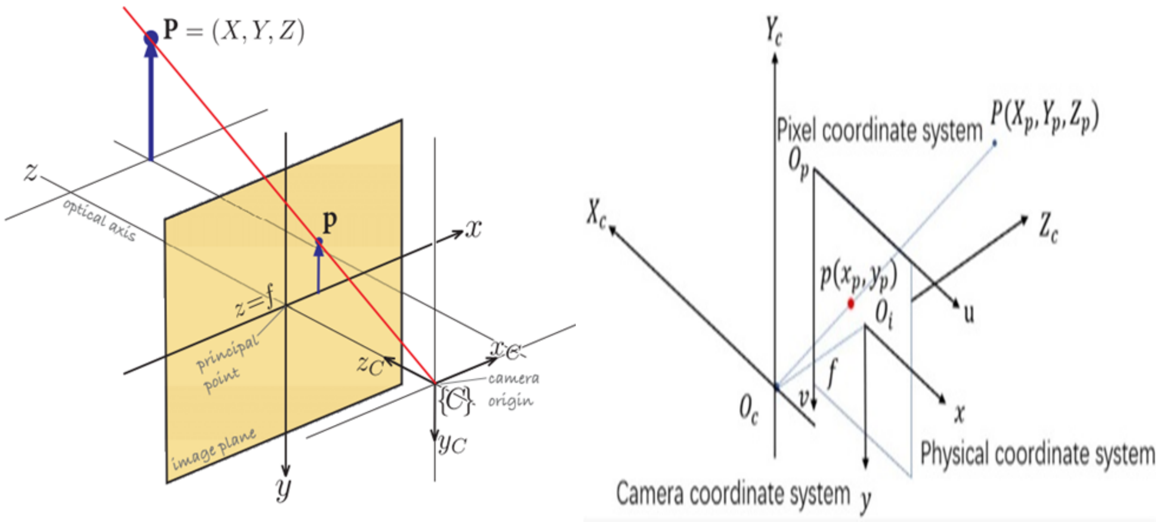


Figure 4.5: Perspective Projection of Image Formation
[48]

Where:

O_i is Physical coordinate of the system;

O_P is Pixel coordinate of the system;

O_C is Camera coordinates of the system;

Assuming that the coordinates of point P in O_C are (X_P, Y_P, Z_P) and the corresponding coordinates projected onto O_i are (x_p, y_p) the relationship of the point p between O_C and O_i are given by:

$$\begin{cases} x_p = f\dfrac{X_P}{Z_P} \\ y_p = f\dfrac{Y_P}{Z_P} \end{cases} \tag{4.1}$$

Where:

f is the focal length of a camera Pixel coordinate system;

O_p is depicted with the origin on the top left vertex of the image;

The u and v are parallel to the x and y axis of O_i;

Let the point (u_p, v_p) in O_p corresponding to the point (x_p, y_p) in O_i can be obtained:

$$\begin{cases} u_p = \dfrac{x_p}{du} + u_0 \\ v_p = \dfrac{y_p}{dv} + v_0 \end{cases} \tag{4.2}$$

Where: du and dv reflect the actual size of a pixel in the u and v axes;

(u_0, v_0) represents translation of the origin of O_i relative to the origin of O_p;

From Eqn. (4.1) and (4.2) the transformed relationship between pixel coordinate and camera coordinate is given by:

$$Z_P \begin{bmatrix} u_P \\ v_P \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} \tag{4.3}$$

Where: $f_x = \frac{f}{du}$ and $f_y = \frac{f}{dv}$

f, u_0, v_0, du, and dv are the intrinsic parameters of the camera obtained from the camera specification;

Z_P is the observed depth value of the depth map;

According to Eqn. (4.3), the spatial coordinate of the point on the object or fruit is given.

$$\begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ Z_i \end{bmatrix} \tag{4.4}$$

Where:

$i = A, B, C, D$ are points on the object or fruit, obtained through the bounding box.

The bounding boxes detected are characterized by four values, namely, (x_center, y_center, width, and height). The center of the bounding box is normalized and represented by x_center and y_center. To normalize the coordinates, take the pixel values of the $x-$ and $y-$ axis, which refer to the center of the box on the respective axis. And by defining the object/fruit diameter d_x, d_y, and d_z along X_C, Y_C and Z_C axes, the diameter of fruit d_x and d_y can be obtained from spatial coordinates of points on fruit and expressed as:

$$d_x = X_C - X_A = f_x(u_C - u_A) = f_x(height - y\_center)$$

$$d_y = Y_D - Y_B = f_x(v_D - v_B) = f_x(x\_center - width) \tag{4.5}$$

In three-dimensional perspective projection, the object's or fruit's diameter d_z couldn't be obtained directly from the image. Hence, the shape of an object or fruit is an ellipsoid, and thus the magnitudes of d_x, d_y, and d_z would be highly correlated by fitting a quadratic polynomial function of d_x and d_y as follows:

$$d_z = a_0 + a_1 dx^2 + a_2 dy^2 + a_3 dx + a_4 dy \tag{4.6}$$

Where:
a_0, a_1, a_2, a_3, a_4 are regression coefficients a polynomial determined by using the least-squares method.
As mentioned in [49] to acquire the regression model for d_z, mentioned in Eqn. (4.6), a grand total of 137 samples of orange fruits were gathered from the orchard, including measurements of the diameters d_x, d_y, and d_z of each fruit were measured by a Vernier caliper (ProSket, PD-151). The quadratic polynomial function fitted for d_z is determined.

$$d_z = 16.0728 + 0.0028dx^2 + 0.0018dy^2 + 0.0264dx + 0.4133dy \tag{4.7}$$

Where the root mean square error (RMSE) is $4.51mm$ and the coefficient of determination $R^2 = 0.940$, indicating a good model for estimating d_z. The point $Q_0(u_q, v_q)$ is the center point of the fruit in a two-dimensional bounding box, which indeed corresponds to the center of the object or fruit surface. The spatial coordinates (X_q, Y_q, Z_q) of Q_0 in Q_C are obtained using Eqn. (4.5), the distance between the camera (eye-in-hand configuration) and the object or fruit d is given by:

$$d = \sqrt{X_q^2 + Y_q^2 + Z_q^2} \tag{4.8}$$

Since the camera coordinate system is not the same as the main fruit coordinate system, the following transformation was done:

$$\begin{bmatrix} P_{X_f} \\ P_{Y_f} \\ P_{Z_f} \end{bmatrix} = \begin{bmatrix} f & 0 & x\_center \\ 0 & f & y\_center \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_q + \frac{d_x}{2} \\ Y_q + \frac{d_y}{2} \\ Z_q + d_z \end{bmatrix}$$

The object/fruit's specific $3D$ location, given by P_X, P_Y, P_Z, can be described based on how P_i is positioned in relation to Q_0 geometrically.

$$\begin{aligned} P_X &= P_{X_c} + P_{X_f} \\ P_Y &= P_{Y_c} + P_{Y_f} \\ P_Z &= P_{Z_c} + P_{Z_f} \end{aligned} \tag{4.9}$$

Eqn. (4.9) indicates the position of the object/fruit detected using an object detection neural network with respect to the manipulator base frame.

4.5 Controller Design

Control systems are used to alter a system's behavior to act in a desired manner over time, and are utilized in numerous application areas in the modern world, eliminating the need for human interaction. When a quantity such as speed, position, altitude, temperature, or voltage needs to behave in a desired way over time, control is utilized. In order to make a system perform in a preferable manner, it is necessary to anticipate the behavior of the relevant quantities over time, particularly their responses to different inputs, and create a controller accordingly. Control theory can be applied to understand how and why feedback control systems function and how to systematically approach different design and analysis problems. When designing a controller for a particular application, the designer takes into consideration the system's stability and simplicity, as well as the speed and smoothness with which the error between the output and the set point is driven to zero.

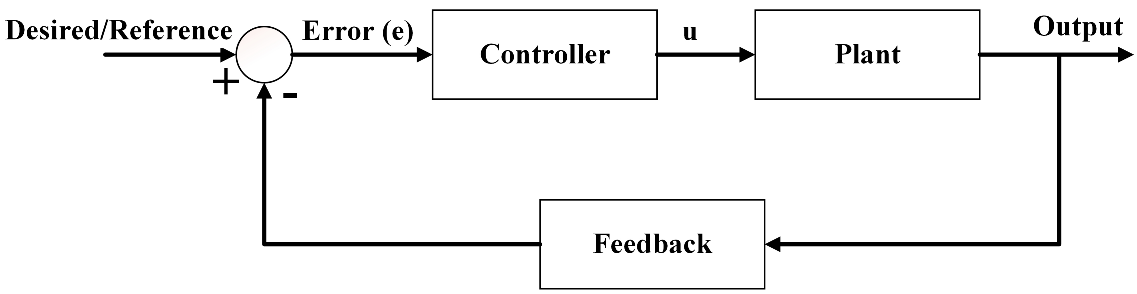


Figure 4.6: Block Diagram of Closed Loop Control System

The reachability criterion, which ensures the existence of the sliding mode, is a crucial need in the literature on sliding modes. Robustness against uncertainty is ensured once sliding has been attained and maintained [6]. In order to ensure that the proper system dynamics are produced, SMC can construct switch functions of state variables or output variables to create sliding surfaces. SMC states that the system has been designed to first navigate and then restrict the system state to remain close to the switching function. The key benefits are that the closed-loop response becomes completely insensitive to a particular class of uncertainty and that the dynamic behavior of the system may be modified by selecting a certain switching function. From a design perspective, sliding mode control is appealing since it allows for direct performance specification [50].

4.5.1 Sliding Mode Controller's Design

The foundation of SMC is the introduction of the sliding variable, a custom-designed function. The sliding manifold is defined after the properly specified sliding variable equals zero. When the sliding variable is appropriately designed and the system trajectories are part of the sliding manifold, the performance of the closed-loop system is suitable. The goal of SMC is to direct the system's trajectory to the appropriate sliding manifold and then use control to maintain motion thereafter. This method takes advantage of the sliding mode's key characteristics, including its insensitivity to both internal and external disturbances, its ultimate accuracy, and the finite-time convergence of the sliding variables to zero [50].

Several points of view have been raised to design a controller for a mobile manipulator robot, including stability, path following, robustness to external disturbance, and uncertainties. Designing a control law that provides the desired performance in the presence of disturbances and uncertainties was challenging to achieve the control objective.

This challenge led to intense interest in the development of robust control methods. One specific method for designing a sturdy controller is referred to as the sliding mode control technique. Because of its exceptional features, such as robustness in the face of significant parameter fluctuations and external disruptions, convergence within a limited time, continued adherence to a manifold for all future periods, and reduced-order corrected dynamics, it has become a widely implemented approach to managing various non-linear control systems [51]. SMC is a useful, robust technique to handle sudden and large changes in the system's dynamics and has been applied to many areas, such as robot control, motor control, aircraft and spacecraft control, process control, and power system control. SMC, a method in control theory, is a non-linear control technique that modifies the dynamics of a non-linear system by utilizing a discontinuous control signal to compel the system to slide through a cross-section of its usual behavior.

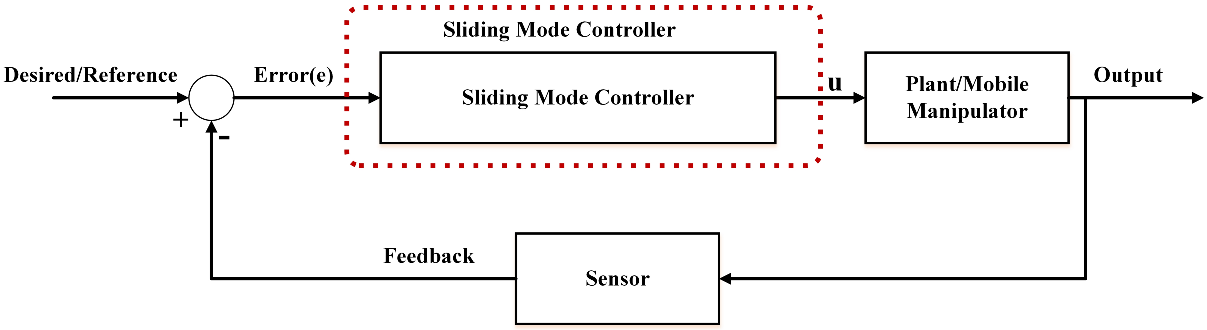


Figure 4.7: Block Diagram of Sliding Mode Controller

A sliding-mode controller can stabilize the trajectory of a system. After the initial reaching phase, the system states slide along the line $S_i = 0$. The reason for selecting the specific surface where $S_i = 0$ is due to its favorable reduced-order dynamics when operating within its boundaries. Specifically, when $S_i = \lambda_i e_i + \dot{e}_i$, the surface is linked to a system represented by $\dot{e}_i = -\lambda_i e_i$, which boasts an exceptionally stable origin that is exponentially so. There are two steps involved in achieving a sliding-mode controller. Initially, it requires the creation of a sliding manifold that can yield the intended system response from the plant. This restricts the state variables of the plant dynamics to comply with another set of equations that are referred to as the switching surface and have been defined below. From the general dynamics equation of a robot.

$$B\tau + f\lambda = M\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{4.10}$$

The selected sliding surface and error for the designed system:

$$S = \lambda e + \dot{e} \qquad \dot{S} = \lambda \dot{e} + \ddot{e}$$

$$e = \theta_d - \theta_a \qquad \dot{e} = \dot{\theta}_d - \ddot{\theta}_a$$

$$\ddot{e} = \ddot{\theta}_d - \ddot{\theta}_a \tag{4.11}$$

Solving for $\ddot{\theta}_a$ from equation (4.11)

$$\ddot{\theta}_a = M^{-1}[\tau - C(\theta_a, \dot{\theta}_a)\dot{\theta}_a - G(\theta_a)] \tag{4.12}$$

where θ_a is the actual joint angle of the mobile manipulator, substituting Eqn. (4.12) into Eqn. (4.11) and solving for the control law that governs the designed system.

$$\dot{S} = \lambda \dot{e} + \ddot{\theta}_d - \ddot{\theta}_a = 0$$

$$\lambda \dot{e} + \ddot{\theta}_d - M^{-1}[B\tau + f\lambda - C(\theta_a, \dot{\theta}_a)\dot{\theta}_a - G(\theta_a)] = 0$$

$$M^{-1}[B\tau] = \lambda \dot{e} + \ddot{\theta}_d - M^{-1}[f\lambda - C(\theta_a, \dot{\theta}_a)\dot{\theta}_a - G(\theta_a)]$$

$$B\tau = M(\theta)(\lambda \dot{e} + \ddot{\theta}_d) + C(\theta_a, \dot{\theta}_a)\dot{\theta}_a + G(\theta) + \Delta u$$

$$U_{eq} = M(\theta)\lambda \dot{e} + M(\theta)\ddot{\theta}_d + C(\theta_a, \dot{\theta}_a)\dot{\theta}_a + G(\theta_a) + \Delta u \tag{4.13}$$

Where $\Delta u = \Delta M(\theta)\ddot{\theta} + \Delta C(\theta, \dot{theta})\dot{\theta} + \Delta G(\theta) + f_r + \delta d_t$ the lumped uncertainty component. Based on the dynamic model, the designed system is governed by an equivalent control law represented as $\tau = u_{eq}$. Designing a control law involves two steps, with the second step relying on the selection of a switching function that compels the trajectories of the system state to attain and glide along the surface. The sliding mode controller, including the equivalent control and the switching control for the mobile manipulator robot, is adopted as follows:

$$U_{SMC} = U_{eq} + U_{dis}$$

$$U_{eq} = M(\theta)(\lambda \dot{e} + \ddot{\theta}_d) + C(\theta_a, \dot{\theta}_a)\dot{\theta}_a + G(\theta_a) + \Delta u$$

$$U_{dis} = -K sgn(S)$$

$$K = [K_1 K_2 K_3 K_4 K_5 K_6 K_7]^T$$

$$U_{SMC} = M(\theta)(\lambda\dot{e} + \ddot{\theta}_d) + C(\theta_a, \dot{\theta}_a)\dot{\theta}_a + G(\theta_a) + \Delta u + Ksgn(S) \tag{4.14}$$

The control component denoted as u_{eq} is employed to achieve the sliding surface from the error states and offset the dynamic attribute.
U_{disc} discontinuous control part used to enforce the state trajectory to maintain a sliding surface within the sliding manifold.
The discontinuous control part U_{disc} includes the $sigmoid$ function selected to attempt to smooth the discontinuity in the conventional $signum$ function to obtain an arbitrarily close but continuous approximation. The reachability criterion, which ensures the existence of the sliding mode, is a crucial need in the literature on sliding methods.

Robustness against uncertainty is ensured once sliding has been attained and maintained. To ensure that the proper system dynamics are produced, SMC can construct switch functions of state or output variables to create sliding surfaces. According to SMC, the system is built to drive and constrain the system state near the switching function. The key benefits are that the closed-loop response becomes entirely insensitive to a particular class of uncertainty and that the system's dynamic behavior may be modified by selecting a specific switching function. From a design perspective, sliding mode control is appealing since it allows for direct performance specification. Lyapunov's stability approach will demonstrate the stability of the control law of Eqn. (4.14).

4.5.2 Stability Analysis

To check the stability of the designed system with the deployed controller, a positive definite function is nominated, called the Lyapunov function. A positive definite function $L_i \geq 0$, and $L_i = 0$ for tracking error $e_i = 0$ and its time derivative $\dot{L_i} \leq 0, \forall t > 0$ and $\dot{L_i(0)} = 0$ for $t = 0$.

$$\begin{cases} L_i = \dfrac{1}{2} S_i^2 \\ \dot{L_i} = S_i \dot{S_i} \end{cases} \tag{4.15}$$

For the control law in Eqn. (4.14), a suggested scalar function $L_i \geq 0$, and $L_i = 0$ if the tracking error $e_i = 0$, therefore, the Lyapunov function L_i selected in Eqn. (4.15) is a positive definite function.

Substituting Eqn. (4.11) into Eqn. (4.15)

$$\begin{aligned} \dot{L}_i &= S_i(\lambda\dot{e} + \ddot{\theta}_d - M^{-1}[B\tau + f\lambda - C(\theta_a, \dot{\theta}_a)\dot{\theta}_a - G(\theta_a)]) \\ \dot{L}_i &= S_i(\lambda\dot{e} + \ddot{\theta}_d - M^{-1}[BU_{SMCi} + f\lambda - C(\theta_a, \dot{\theta}_a)\dot{\theta}_a - G(\theta_a)]) \end{aligned} \tag{4.16}$$

By substituting U_{SMCi} from Eqn. (4.14).

$$\dot{L}_i = S_i(\lambda\dot{e}+\ddot{\theta}_d-M^{-1}[BM(\theta)(\lambda\dot{e}+\ddot{\theta}_d)+C(\theta_a, \dot{\theta}_a)\dot{\theta}_a+G(\theta_a)+Ksgn(S)-C(\theta_a, \dot{\theta}_a)\dot{\theta}_a-G(\theta_a)])$$

$$\dot{L}_i = S_i(\lambda\dot{e}+\ddot{\theta}_d-(\lambda\dot{e}+\ddot{\theta}_d)-M^{-1}[C(\theta_a, \dot{\theta}_a)\dot{\theta}_a+G(\theta_a)+Ksgn(S)-C(\theta_a, \dot{\theta}_a)\dot{\theta}_a-G(\theta_a)] \tag{4.17}$$

$$\dot{L}_i = -KS_i(M^{-1}(\theta)sgn(S)) \leq 0 \tag{4.18}$$

where $i = 1, 2, 3, 4, 5$, and right and left wheel joint variables.

As a result, the suggested Lyapunov function L_i has a derivative that is unambiguously negative, showing that the point $e_i = 0$ is uniformly asymptotically stable for continuous reference and limited K_i. This authorizes the designed mobile manipulators to be stable for all $K_i > 0$. According to the stability analysis, the system is stable for any combination of K_i. However, since the mobile manipulator should respond in a non-oscillatory but not excessively slow manner, it was necessary to determine the controller's parameters. The correct gains for each reference trajectory must be decided through different methods, but trial and error have been used in this case. The control parameter values are given in the table (4.2) for the designed controller.

Table 4.2: Controller Parameter for the Designed Mobile Manipulator

Parameters	Joint_1	Joint_2	Joint_3	Joint_4	Joint_5	Right_Wheel	Left_Wheel
K	150	150	100	100	100	75	75
λ	30	50	10	30	10	15	15

Chapter 5

Result and Discussion

In this chapter, the simulation result of the designed mobile manipulator is presented with a detailed interpretation to show the performance and visibility of the model. The model specification and parameters used for the simulation test have been described in Appendix (A).

5.1 Simulation Result of Object Detection using YOLOv2

5.1.1 Simulation Result of Object/Fruit Detection

To perform the detection of an object or fruit using an AI-based algorithm, the first step is data preparation, collection of images, resizing based on model architecture, and generating "ground Truth" using Image-Labeler or the Ground Truth image labeler app on MATLAB software. The image used for training detection was a token from the "Kaggle data set and COCO data sets", which are available to the public.

The next step was assembling the detection network from the input layer to the output layer, doing training, and testing the trained detector on collected data by using the deep learning CNN algorithm YOLOv2. Figures (5.1) and (5.2) show the sample of input image data and output of the detected image from the YOLOv2 object detection algorithm. Due to seasonal variations, orange trees may produce both orange and green-colored oranges at the same time. As shown in the simulation result, the developed detection algorithm correctly recognizes both colors during identification and position determination.

Figure 5.1: Sampled Input Image used for Fruit Detection Algorithm

Figure 5.2: Detected Sampled Image from Fruit Detection Algorithm

The stated procedure is followed in order to detect and identify the object, in this case, fruit, which includes gathering raw image data, resizing an image to meet algorithm requirements, and dividing the raw data into training, validation, and test data sets in the first stage. A bespoke automation approach is used in the ground truth labeler app of the MATLAB software to scale the image to meet the input standard of the YOLOv2 single-stage object detection algorithm. Figure (5.1) indicates the raw image given as input to a single-stage deep learning, CNN algorithm called the YOLOv2 network.

The fruit is detected, identified, and recognized by the network in real-time captured images. The input image is processed to reveal the detected fruit in Figure (5.2) while the algorithm determines the fruit's position in relation to the camera frame. The object/fruit detection deep learning network provides the $3D$ position of the fruit denoted by P_X, P_Y, and P_Z. The robot controller employs this information to ascertain the joint positions of the manipulator required to reach the desired position and enable the mobile platform to move from the picking or harvesting position to a different position. This movement is initiated by the wheels' angular velocity as soon as there is no fruit in that orange tree.

5.2 Simulink Model of Designed Mobile Manipulator

This section discusses the discussion of the illustrated Simulink model in figure (5.3). The model represents a designed mobile manipulator without actuation and any additional inputs to the system. Furthermore, it is worth noting that the model imported into Matlab from the Solidworks software.

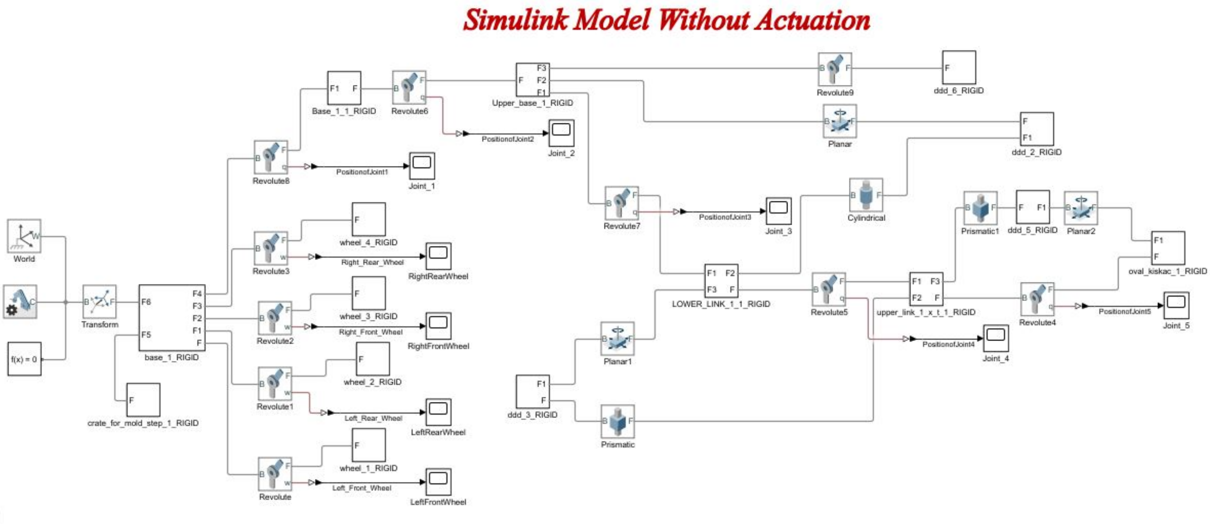
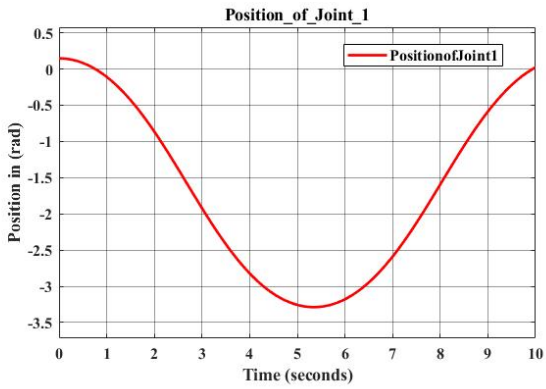


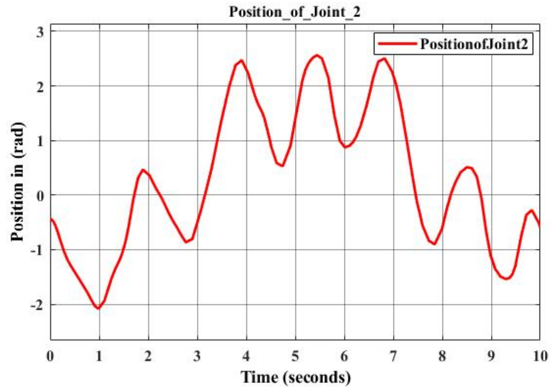
Figure 5.3: Simulink Model of Mobile Manipulator Without Actuation

5.2.1 Simulation Result of Mobile Manipulator without Actuation

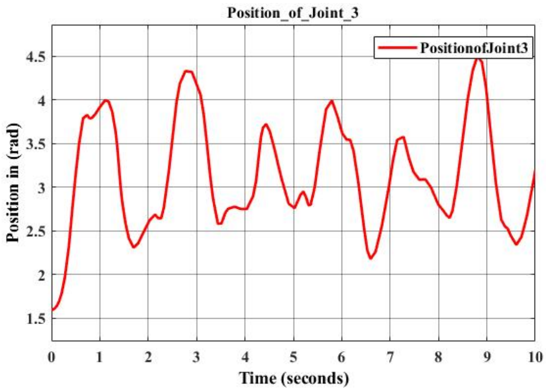
Figure (5.4) exhibits a simulation result of the designed and modeled mobile manipulator without any addition of actuation and desired input from the image. Without the desired input, a mobile platform would not be able to move or perform any tasks that require movement. It would essentially be stationary and unable to respond to any external stimuli or commands. In this scenario, the robot's mounted manipulator can perform tasks, and its joints can be moved about the workspace with flexibility and without control. However, if the wheels' angular velocity is greater than zero, the robot will move, and its linear and angular velocity will depend on the speed of each wheel. The relationship between wheel rotation and robot velocity can be derived by considering the motion of a single wheel and then coupling the two wheels along a single axis of rotation.



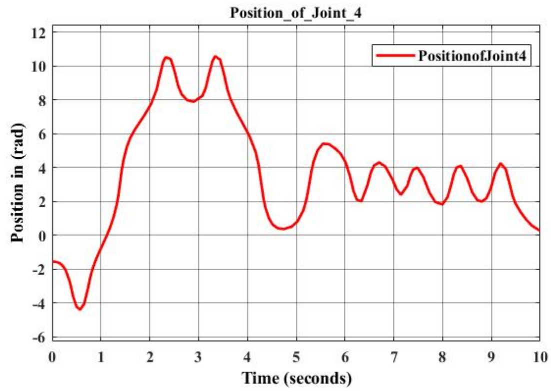
(a) Initial Position of Joint-1



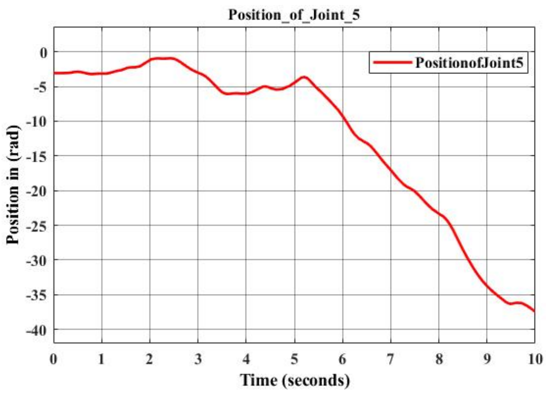
(b) Initial Position of Joint-2



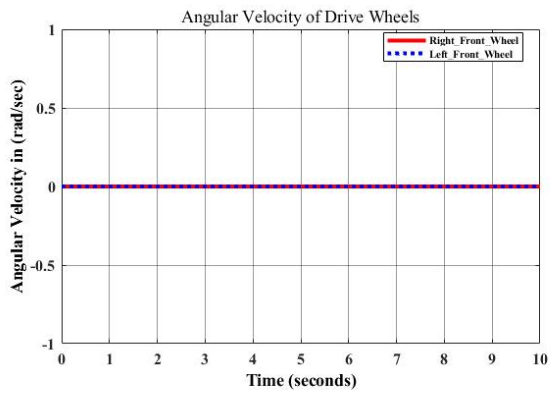
(c) Initial Position of Joint-3



(d) Initial Position of Joint-4



(e) Initial Position of Joint-5



(f) Velocity of Wheels

Figure 5.4: Simulation Result of a Designed Mobile Manipulator without Actuation

The simulation results of the mobile manipulator design, depicted in figures (5.4a), (5.4b), (5.4c), (5.4d), (5.4e), and (5.4f), demonstrate the nonlinearity, instability, and lack of control in both the manipulator joint behavior and the velocity of the differential drive wheels. These outcomes were observed when importing the design from SolidWorks without initially providing any desired input signals. A differential drive robot's movement relies on the angular velocity of its left and right wheels. If the wheels' angular velocity is zero, the robot will remain stationary and its wheels will not be in motion.

5.3 Simulink Model of Mobile Manipulator with Input from Image and Inverse Kinematics

The Simulink model shown in figure (5.5), includes the designed plant with desired input from a real-time captured image and a determined position without deploying a control system. The reference input signal that is used as desired for the designed plant is an image from a camera that is mounted on the end-effector of the manipulator. The camera captures the image and sends it to the image processing algorithm, where the object in the image is detected, and the position of the object in this case, “fruit”, is determined from the image with respect to the camera frame using the perspective projection. The Simulink block diagram also includes the interpolation of the end-effector from the start to the goal position, based on the cubic polynomial.

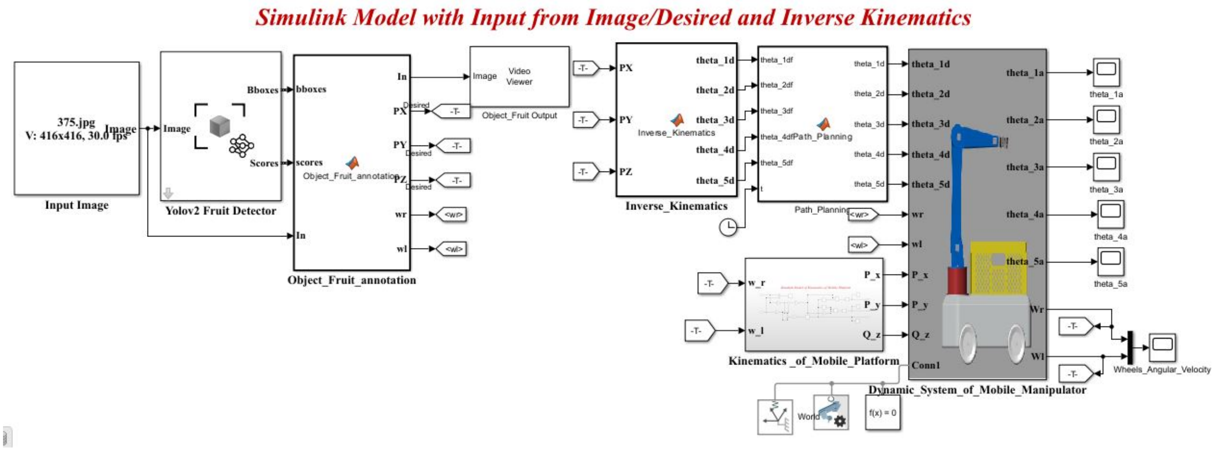
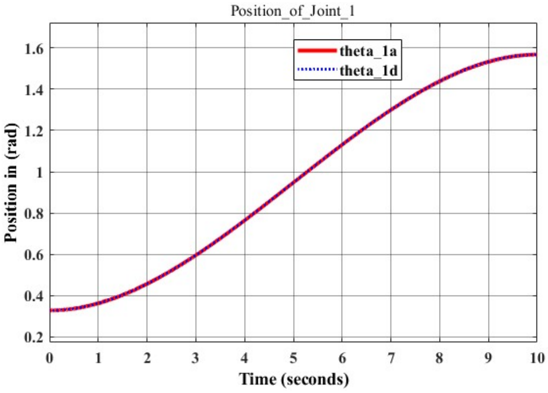


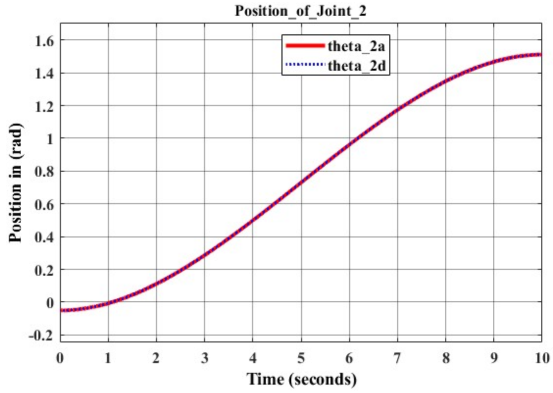
Figure 5.5: Simulink Model of System with Input from Image and Interpolation

5.3.1 Simulation Result of Mobile Manipulator with Input from Image and Inverse Kinematics

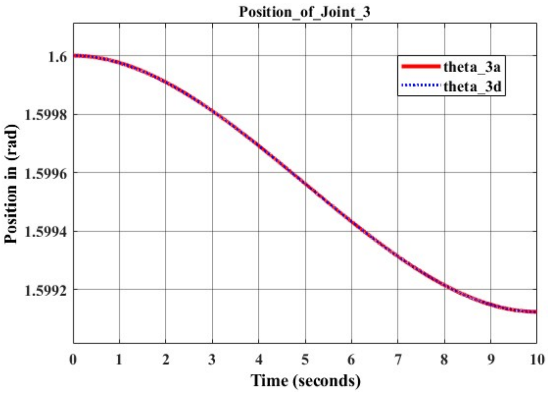
Mobile manipulators that are designed and intended for grabbing the entire fruit off a tree do so without obstructing the movements of the platform or manipulator mounted on it. In order to park at the site of the fruit harvest and allow the manipulator to start up and complete the desired task, the mobile platform drove forward along the $x-$axis. The mobile manipulator can be moved about by computed desired input joint angle of the manipulator and angular velocity of the mobile platform by using the inverse kinematic subsystem and input from acquired images as desired or as a reference.



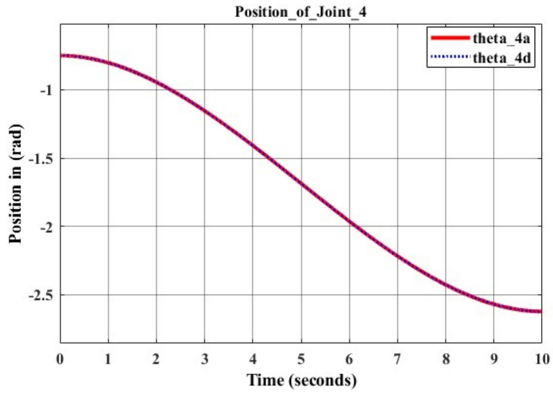
(a) Joint-1



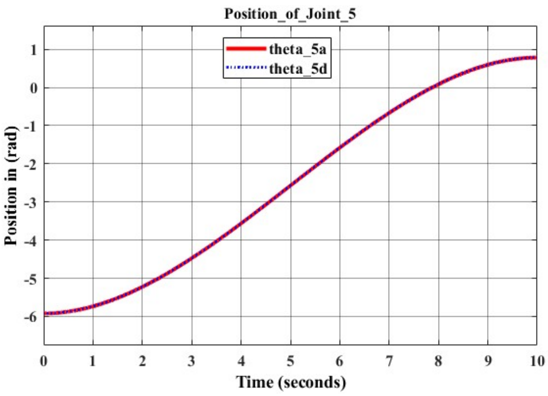
(b) Joint-2



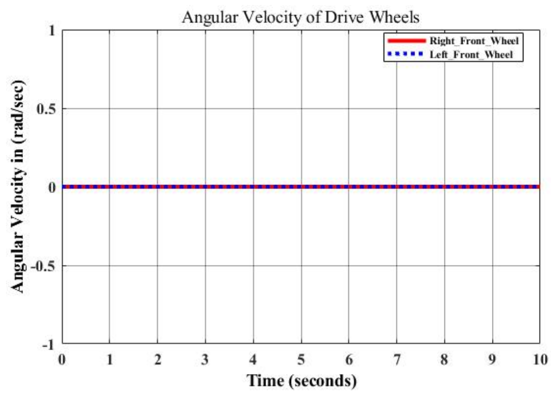
(c) Joint-3



(d) Joint-4



(e) Joint-5



(f) Velocity of Wheels

Figure 5.6: Position of a Designed MM with Desired from Image and Inverse Kinematics

The joint movement instruction is used as the jumping-off point for an investigation of the output of the kinematic model. Both a starting position and a desired joint position are used in this. The test is chosen to begin at the desired joint position, and it is also chosen to arrive at or end at angles specified in radians. These figures can be used to interpolate the desired joint angles from the home position. Based on the simulation outcome joints of a portable manipulator, which can be observed in figure (5.6a), (5.6b), (5.6c), (5.6d), and (5.6e), track the desired position with a cubic path.

Figure (5.6f), demonstrates the angular velocity of the drive wheels was zero, which indicates that the mobile platform is at the parking stage, which allows the manipulator to perform manipulation tasks of harvesting fruit. To make the position of each joint stable and move to the desired values obtained from a captured image, the PID controller is first applied, then the sliding mode controller is designed, and the result is presented in the coming sections.

5.4 Simulink Model of Mobile Manipulator with PID

A PID controller is a widely used control technique in robotics applications that is used to maintain the desired position or trajectory of the robot by adjusting the speed and direction of the robot. The result of the PID controller was visualized using MATLAB's plotting functions. This helps to understand the behavior of the system and verify the design of the controller. Figure (5.7) depicts the Matlab Simulink model that was used to simulate a PID controller.

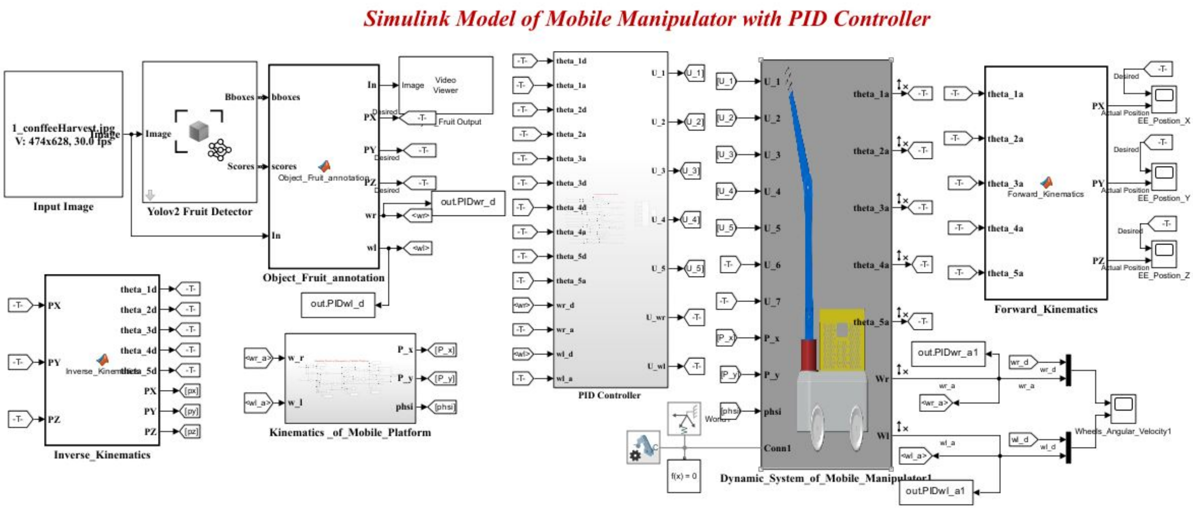
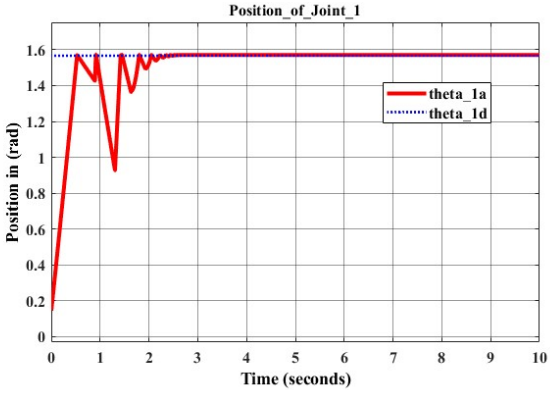


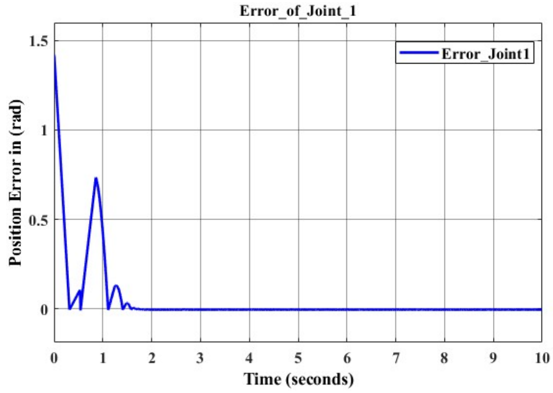
Figure 5.7: Simulink Model of Mobile Manipulator with PID

5.4.1 Simulation Result of Mobile Manipulator with PID Controller

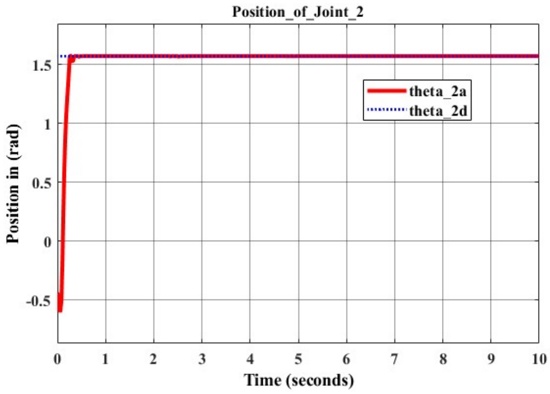
The simulation results of the mobile manipulator using a traditional PID controller are presented in figure (5.8). The purpose of the simulation is to evaluate the performance of the designed system. The simulation result shows the plant position tracking and position tracking error for the manipulator's joints using a traditional PID controller.



(a) Joint-1

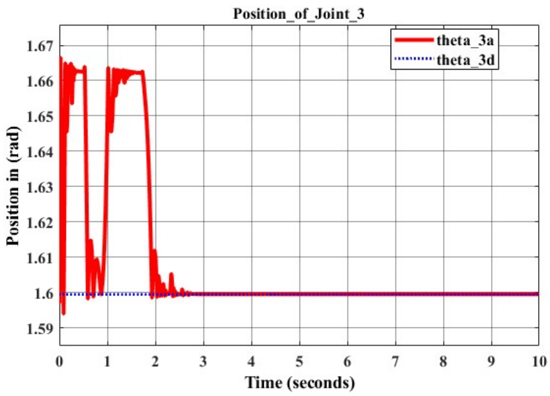


(b) Error of Joint-1



(c) Joint-2

(d) Error of Joint-2

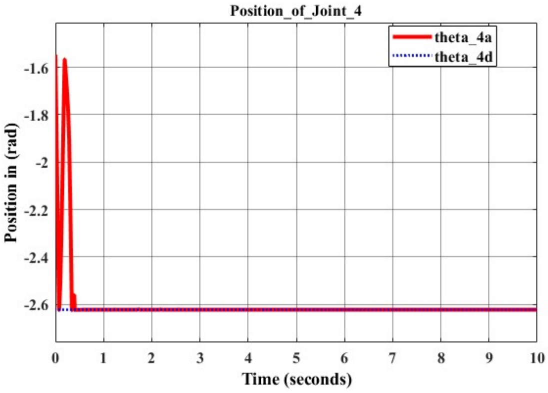


(e) Joint-3

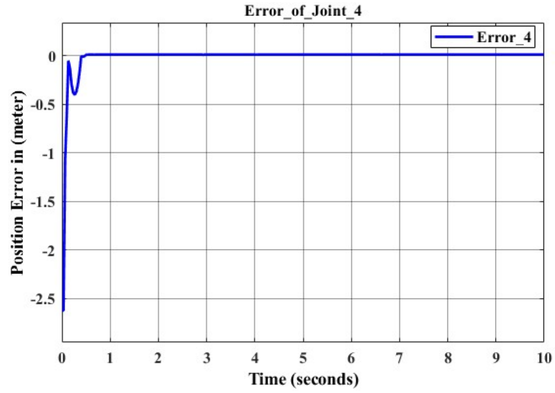
(f) Error of Joint-3

Figure 5.8: Position and Tracking Error of Joints of a Designed MM with PID Controller

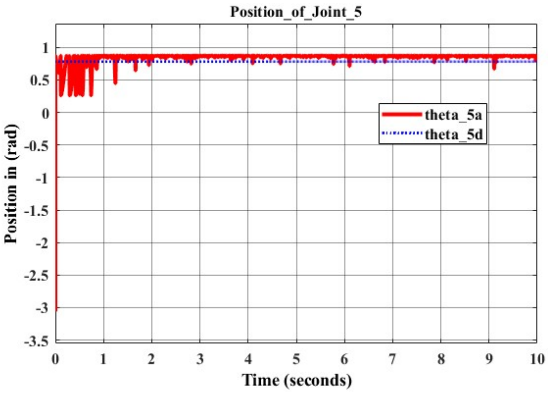
Figures (5.8a), (5.8c), and (5.8e) demonstrate the target tracking of joints one, two, and three of the manipulator, respectively. On the other hand, figures (5.8b), (5.8d), and (5.8f) show the tracking error for joints one, two, and three, respectively. Based on the observation from these plots, it can be concluded that the plant has some overshoot for joints one and three. The mobile platform stays in the packing slot and manipulation is performed during this time. Plots shown in figure (5.9) illustrate the joint positions for the fourth and fifth manipulator joints, along with the end effector position and platform angular velocity.



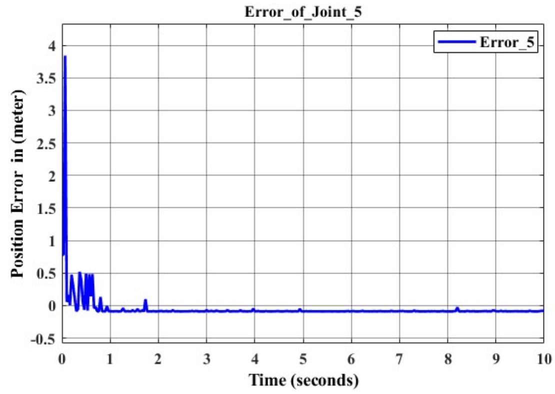
(a) Joint-4



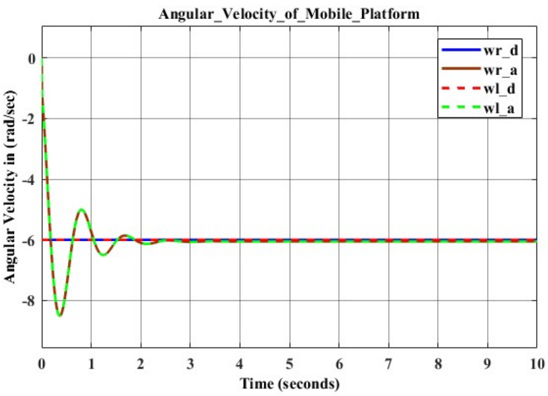
(b) Error of Joint-4



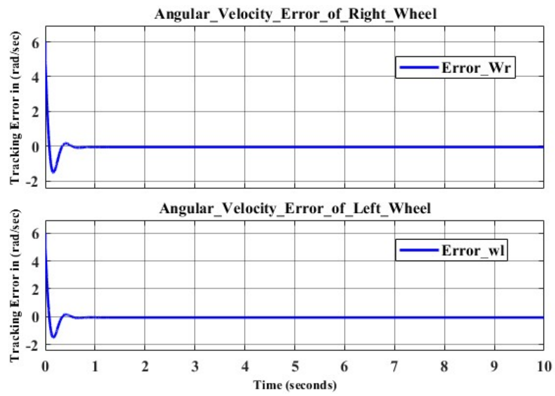
(c) Joint-5



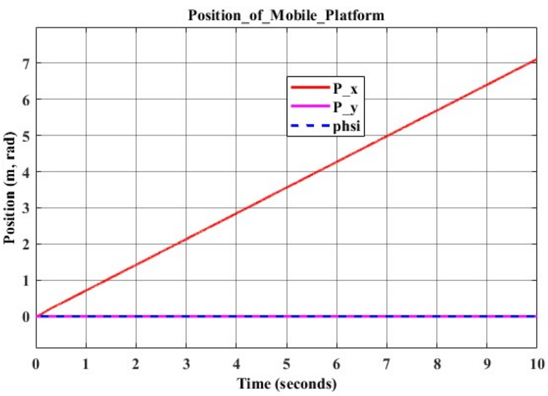
(d) Error of Joint-5



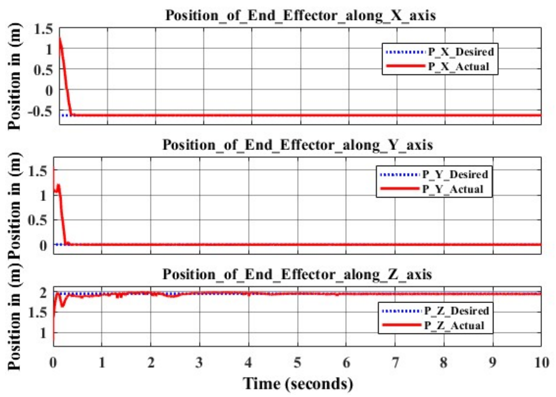
(e) Angular Velocity of Wheels



(f) Tracking Error of Wheels



(g) Position of Mobile Platform



(h) Position of End effector

Figure 5.9: Position and Velocity of a PID-Controlled Designed Mobile Manipulator

The figures (5.9a) and (5.9c) display the simulation results that demonstrate the desired position tracking of joints four and five of the manipulator. The position tracking error for these joints is illustrated in figures (5.9b) and (5.9d), respectively. The position of the manipulator end effector was indicated in the figure (5.9h) using a PID controller. Once the manipulation task is completed, the mobile platform moves on to the next task space. During this movement, the controller actuates the angular velocity of the wheels to track the desired angular velocity, and the result is shown in figure (5.9e). In Figure (5.9g), the motion of the mobile platform in the forward $x-$ direction is depicted. The simulation results reveal that PID controllers exhibit effective control at a steady state and are relatively simple to design and implement. Nevertheless, they lack robustness against disturbances and uncertainties and are not suitable for controlling highly nonlinear systems. To address these limitations and control a highly nonlinear mobile manipulator under the presence of disturbances and uncertainties, Sliding Mode Control was developed. The SMC controller was then compared with the PID controller to determine which one performed better, with the aim of selecting the superior controller for the intended application.

5.5 Simulink Model of Mobile Manipulator with SMC

To make the designed plant stable, a sliding mode controller was deployed, and the Simulink model in figure (5.10) shows the incorporated desired input from vision or a processed captured image, a designed model, and a sliding mode controller.

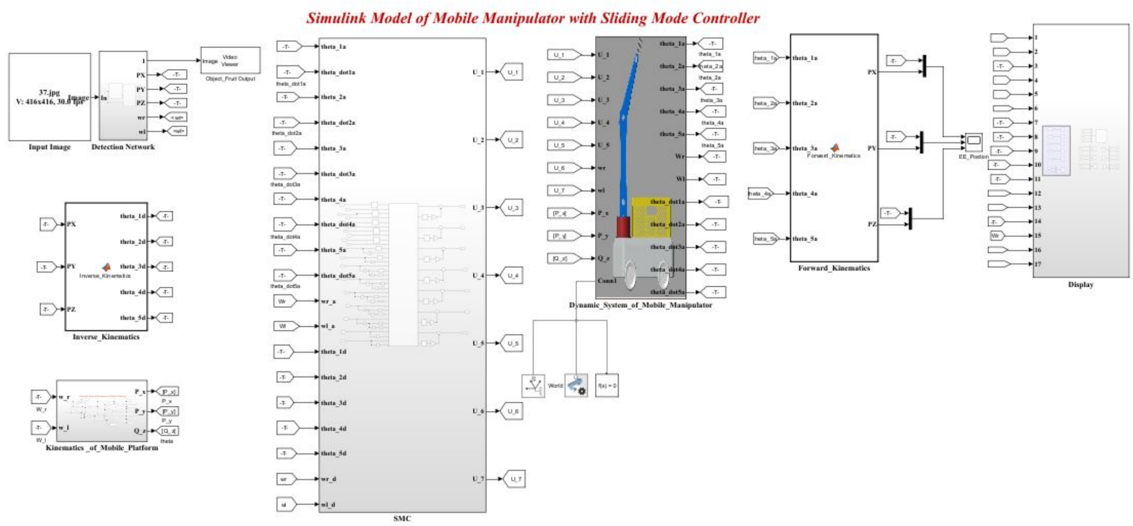
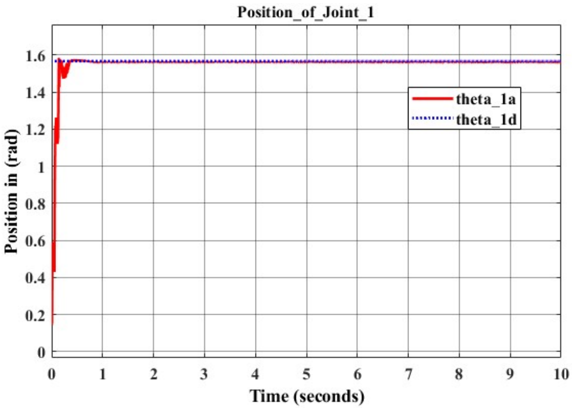


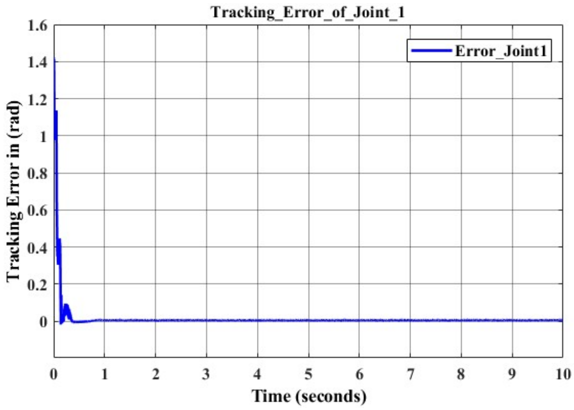
Figure 5.10: Simulink Model of Mobile Manipulator with SMC

5.5.1 Simulation Simulation Results of Mobile Manipulators with SMC

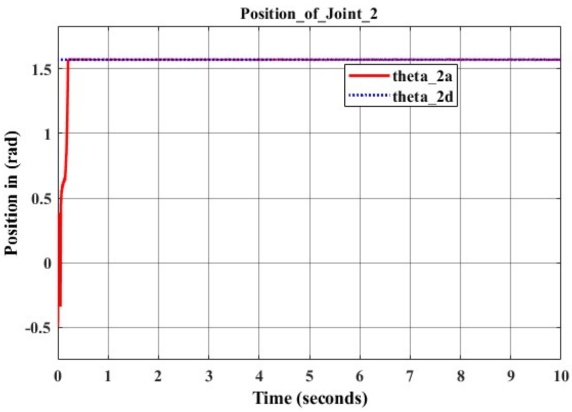
The result indicated in figures (5.11) and (5.12) is the simulation result obtained for a designed mobile manipulator by devoting a sliding mode controller.



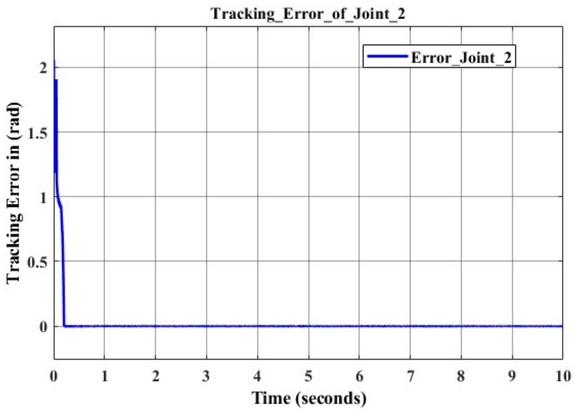
(a) Position of Joint 1



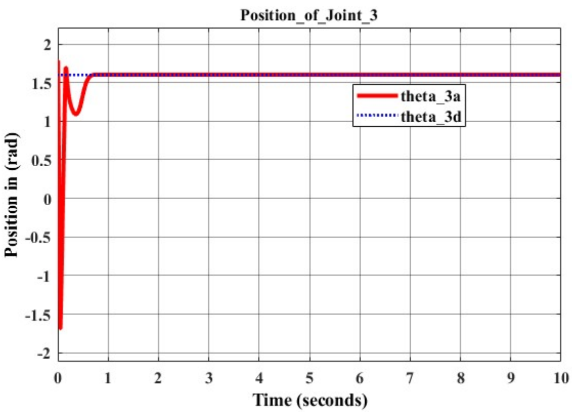
(b) Error Tracking of Joint 1



(c) Position of Joint 2



(d) Error Tracking of Joint 2

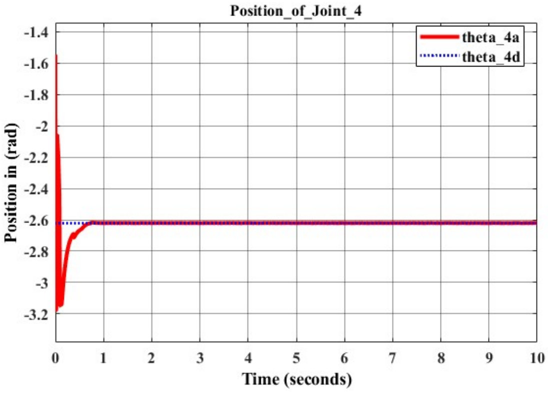


(e) Position of Joint 3

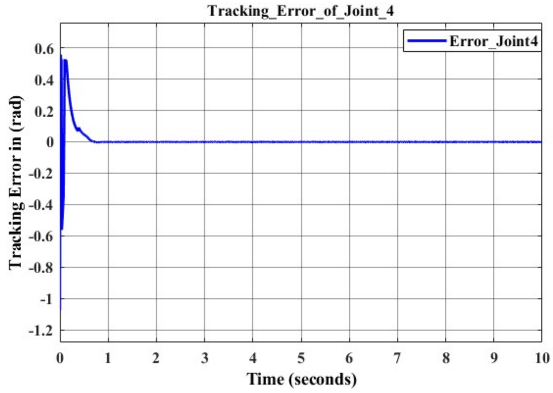
(f) Error Tracking of Joint 3

Figure 5.11: Position and Tracking Error of Joints of a Designed MM with SMC

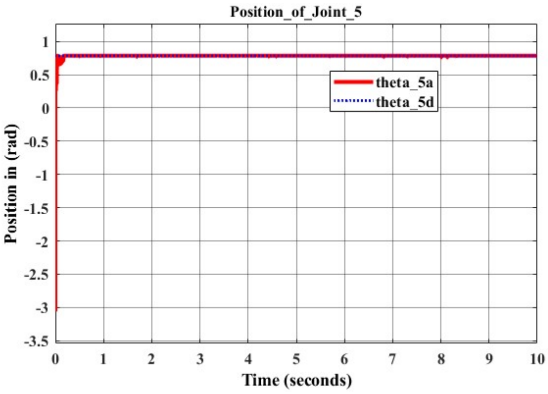
Figure (5.11) shows SMC's ability to stabilize the manipulator position, and reduce the error between desired and actual, ensuring good performance and consistency in the output.



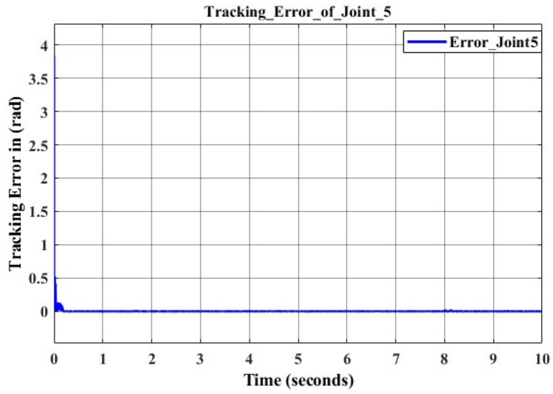
(a) Position of Joint 4



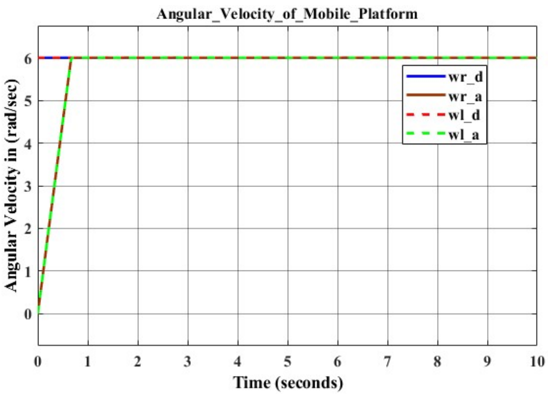
(b) Tracking Error of Joint 4



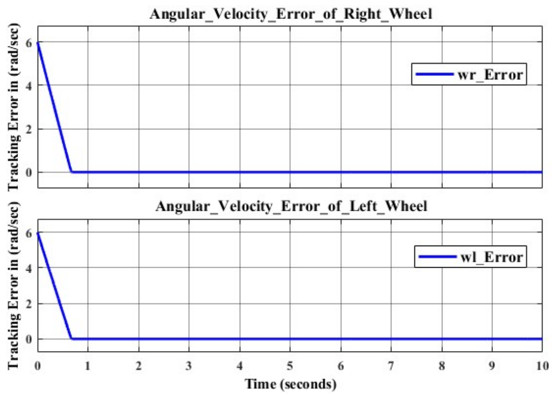
(c) Position of Joint 5



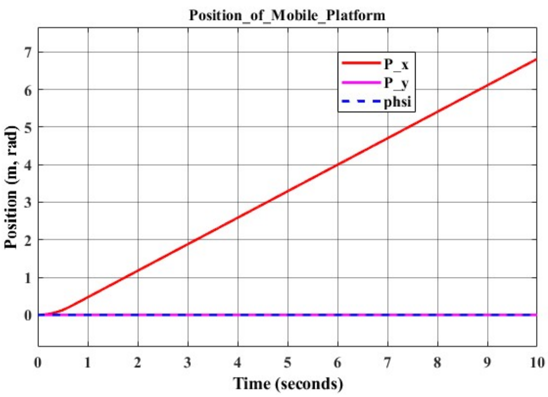
(d) Tracking Error of Joint 5



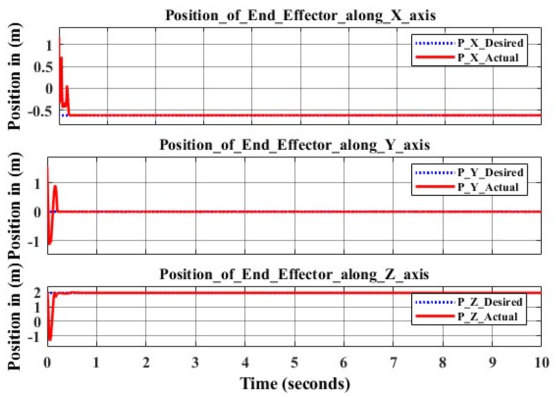
(e) Angular Velocity of Wheels



(f) Tracking Error of Wheels



(g) Position of Mobile Platform



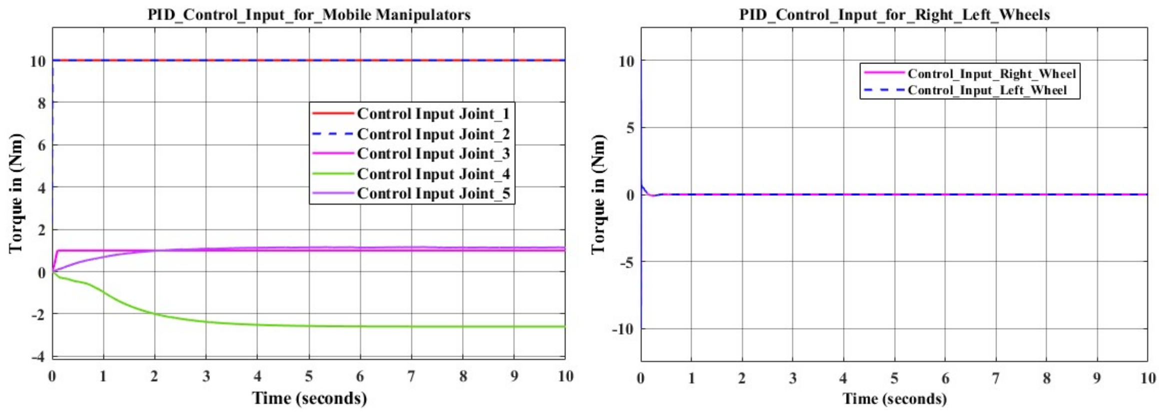
(h) Position of EE

Figure 5.12: Position and Velocity of a Designed Mobile Platform with SMC

The figures (5.11a), (5.11c), (5.11e), (5.12a), and (5.12c) show that all joints of the manipulator follow the desired trajectory position derived from the camera or image. The tracking error for the manipulator's joints is depicted in figures (5.11b), (5.11d), (5.11f), (5.12b), and (5.12d), indicate that the joints accurately track the desired target. The plot displayed in figure (5.12h) illustrates the position of a manipulator's end effector. It represents the attained position, which was obtained through image processing and tracked or reached by manipulating the joints of the manipulator. Figures (5.12e) and (5.12g) depict the angular velocity of the wheels and the position of the mobile platform, respectively. The simulation result in figure (5.11) and (5.12) exhibited that the mobile manipulator became stable and controlled by using SMC. Eventually, the sliding surface is minimized to zero, and the manipulator joint position reaches the desired position to perform a steady function of harvesting fruit in agricultural applications.

5.5.2 Control Input of Mobile Manipulator

To achieve effective control of a mobile manipulator, the primary objective of the PID controller is to generate control efforts that bring it closer to the desired position or enable smooth tracking of a given trajectory. Therefore, it is crucial to appropriately tune the PID controller's gains to ensure stable and accurate control of the mobile manipulator.

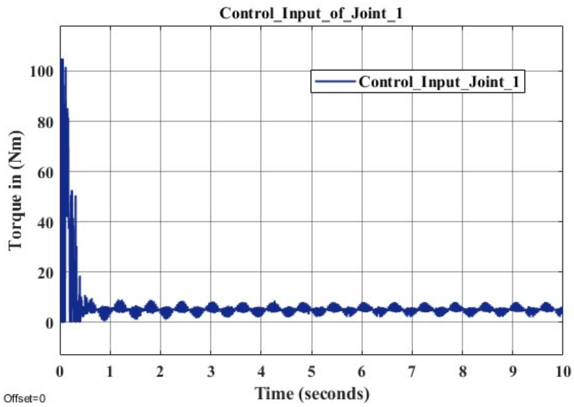


(a) PID Control Effort for Manipulators Joints

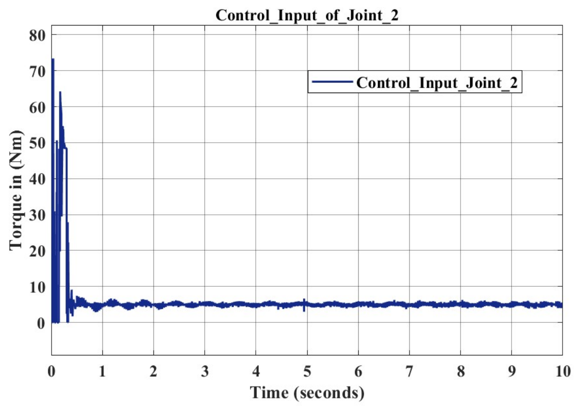
(b) PID Control Effort for Wheels

Figure 5.13: PID Control Input used for Mobile Manipulator

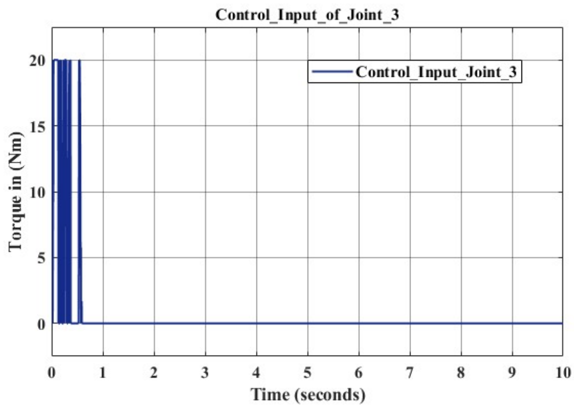
In SMC, the control input is designed to drive the system state trajectories to slide along the sliding surface toward a desired equilibrium point. In this study, a conventional sliding mode controller was employed, and the level of control input is depicted in the figure (5.14).



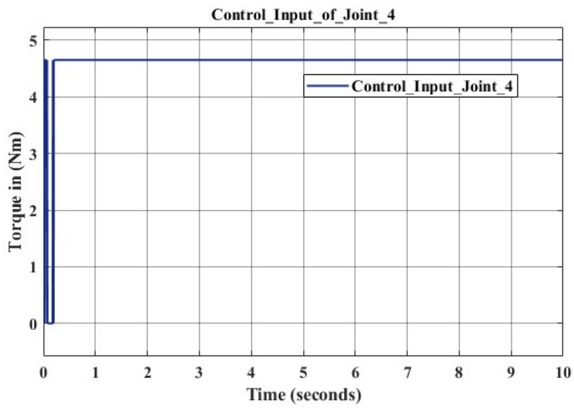
(a) Control Effort of Joint 1



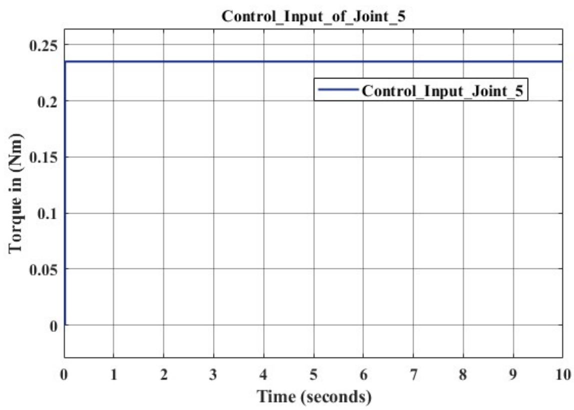
(b) Control Effort of Joint 2



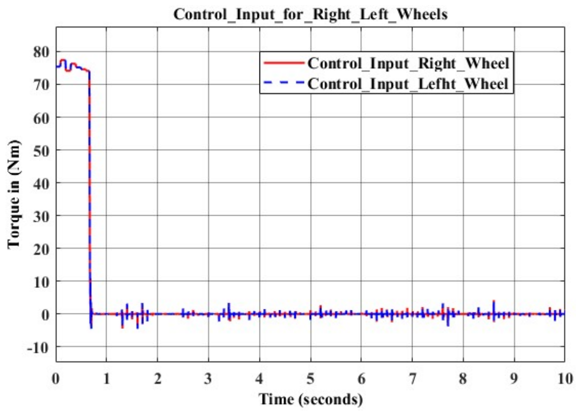
(c) Control Effort of Joint 3



(d) Control Effort of Joint 4



(e) Control Effort of Joint 5



(f) Control Effort of Wheels

Figure 5.14: SMC Control Input used for Mobile Manipulator

The control effort generated by the control law, which is designed based on the sliding surface information, to control the modeled mobile manipulator is shown in figure (5.14a), (5.14b), (5.14c), (5.14d), (5.14e), and (5.14f). The purpose of this control effort is to guide and stabilize the model by ensuring that it follows the sliding surface and behaves as desired. Moreover, this control effort is equivalent to the actuator size designed for both the mobile platform wheels and the manipulator's joints.

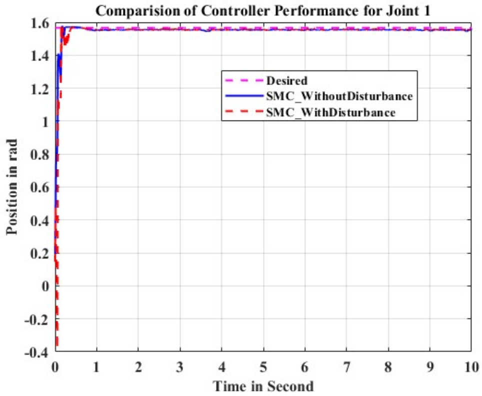
In essence, the control law determines the necessary adjustments or commands required to align a model with a desired trajectory or behavior represented by the sliding surface. To implement this control law on a mobile manipulator, the control effort must be applied to both the mobile base wheels' actuators and the manipulator's joint actuators. The actuators for the mobile base enable the model to navigate and reach different locations along rows of planted trees, controlling its movement and positioning. Conversely, the joint actuators on the manipulator's arm enable articulation and the execution of fruit harvesting or picking tasks. By simultaneously applying the control effort to both sets of actuators, the system coordinates the motion of the mobile base and manipulator's arm, facilitating smooth and precise control of the entire mobile manipulator system. This integration guarantees the effective and accurate achievement of the desired behavior represented by the sliding surface. However, there might be slight chattering observed on specific joints (5.14a), and (5.14b), which is a known drawback of the sliding mode controller.

5.6 Simulink Model of MM with Perturbation

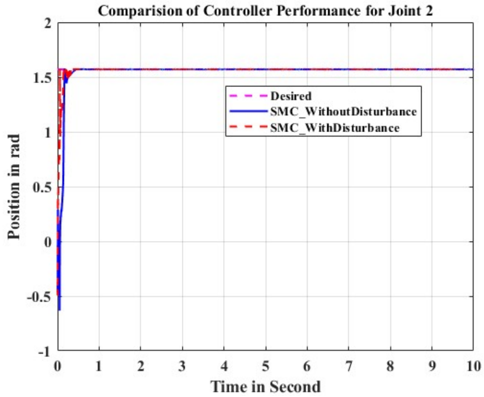
In scenarios where the exact model of a nonlinear system is not available in real time, the system's performance and stability may be compromised due to the uncertainty in a dynamic model. This uncertainty arises from the varying loads imposed by manipulators and the platform's carrying capacities, which influence model parameters. In order to ensure the resilience of the model against uncertainty in parameters and external disturbances, the manipulator was assigned weights of [$0.5kg$, $1.0kg$, and $1.5kg$]. Additionally, a disturbance torque equivalent to 20% of the maximum torque was introduced. To guarantee the stability of the platform's carrying capacity across different conditions, extra weights of [$5kg$, $15kg$, and $25kg$] were added to the crate holding the fruit. The simulation results for modeled mobile manipulator under different perturbations are presented in subsequent subsections.

5.6.1 Simulation Result of Mobile Manipulator with Disturbance

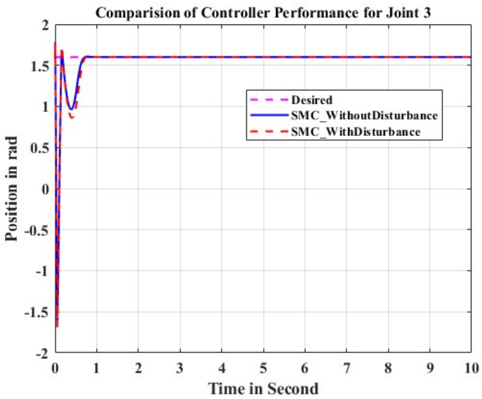
SMC is a robust control technique that can mitigate the effect of white noise disturbance on control systems. Sliding mode control uses a sliding surface to force the system output toward a desired reference trajectory. By doing so, it can ensure that the system output is not affected by the white noise disturbance and remains close to the desired trajectory.



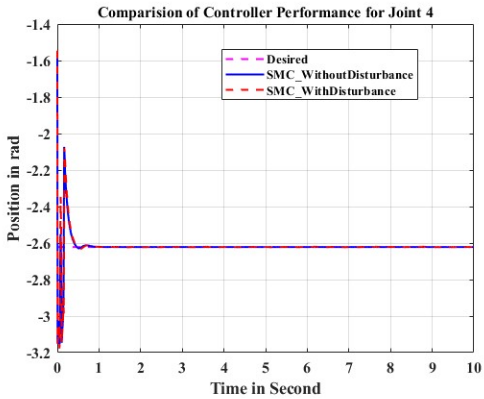
(a) Position Tracking of Joint 1



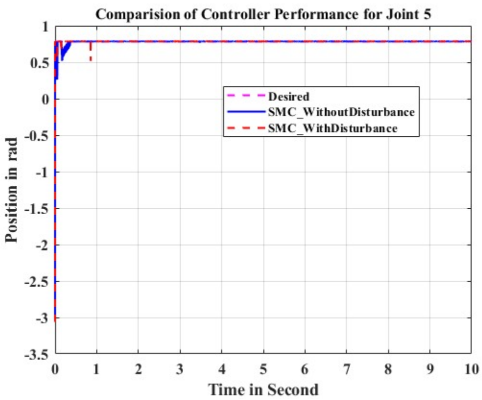
(b) Position Tracking of Joint 2



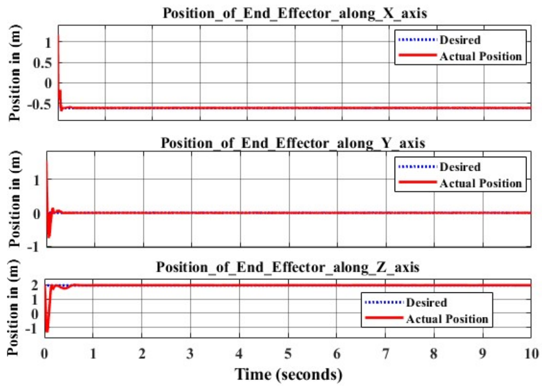
(c) Position Tracking of Joint 3



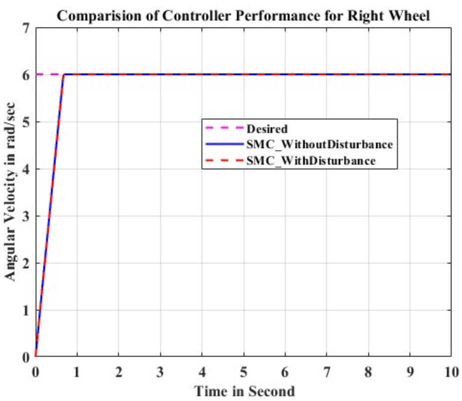
(d) Position Tracking of Joint 4



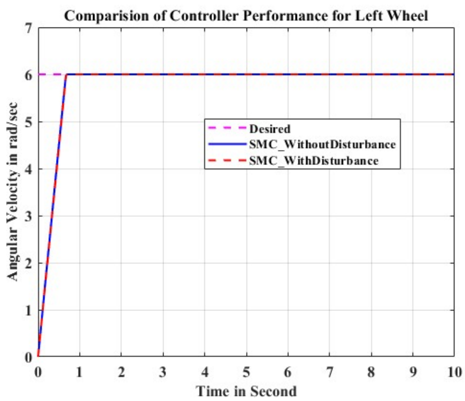
(e) Position Tracking of Joint 5



(f) Position Tracking of End Effector



(g) Angular Velocity of Right Wheel

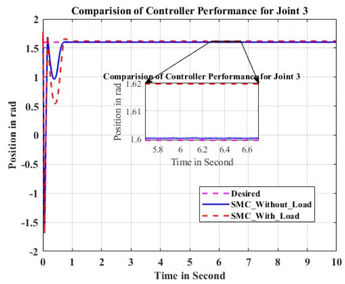


(h) Angular Velocity of Left Wheel

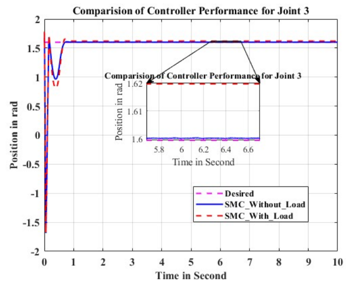
Figure 5.15: A Controller's Performance with White Noise Disturbances

The figures (5.15) showcase the effectiveness of the designed systems in handling the introduction of white noise. They demonstrate the system's resilience as depicted in figures (5.15a), (5.15b), (5.15c), (5.15d), and (5.15e), which indicate that the addition of white noise disturbance to the manipulator joint position model does not result in significant changes. Similarly, the introduction of disruptions in the wheels of a mobile platform does not impact the overall performance of the system as indicated in the figures (5.15g) and (5.15h. This indicates that the system is robust against external disturbances, particularly in the context of sliding mode control.

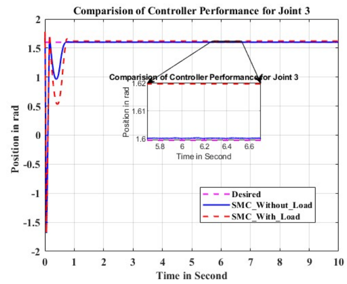
5.6.2 Performance Evaluation of Designed Controller with Load Variation on Manipulator Joint



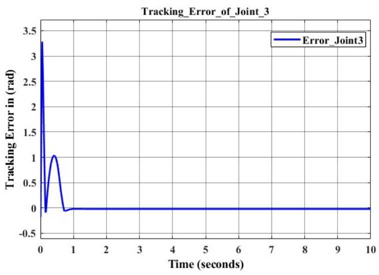
(a) Position Tracking of Joint 3 with 0.5 Kg



(b) Position Tracking of Joint 3 with 1.0 Kg



(c) Position Tracking of Joint 3 with 1.5 Kg



(d) Tracking Error of Joint 3 with 0.5 Kg

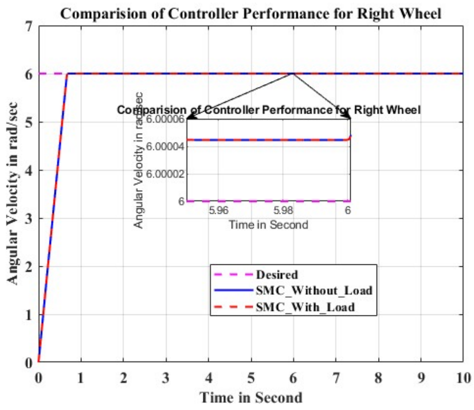
(e) Tracking Error of Joint 3 with 1.0 Kg

(f) Tracking Error of Joint 3 with 1.5 Kg

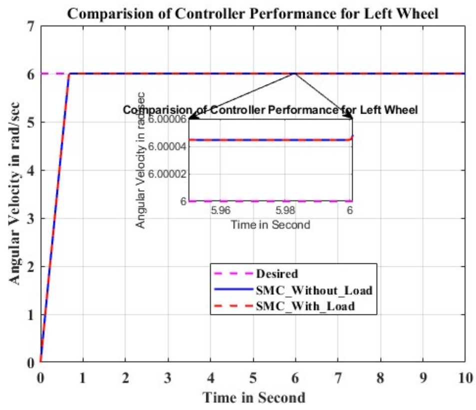
Figure 5.16: A controller's performance with Load Variation on Joint 3

To assess how much weight the controller can handle and evaluate its durability when faced with varying loads, modify the load on joint three of the manipulator. The results depicted in figures (5.16a), (5.16b), and (5.16c) demonstrate that the controller remains robust and stable even with additional masses of up to 1.5 kg on the manipulator's joint. When compared to figure (5.11f), the position tracking error for joint three is consistent with figures (5.16d), (5.16e), and (5.16f), despite the introduction of different types of masses.

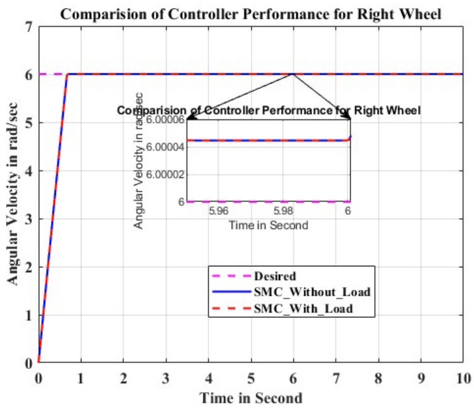
5.6.3 Performance Evaluation of Designed Controller with Load Variation on Mobile Platform



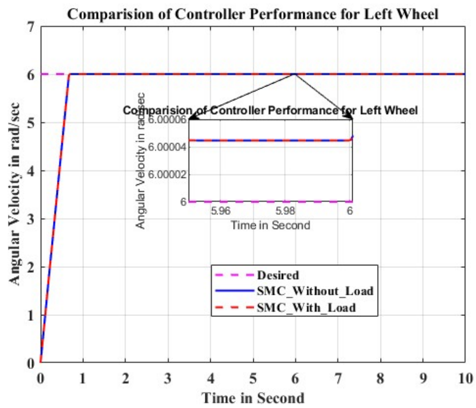
(a) Right Wheel Angular Velocity with 5 Kg



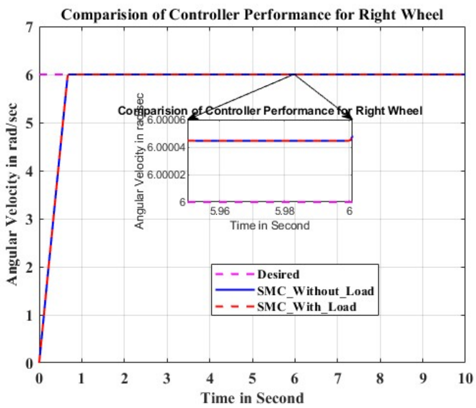
(b) Left Wheel Angular Velocity with 5 Kg



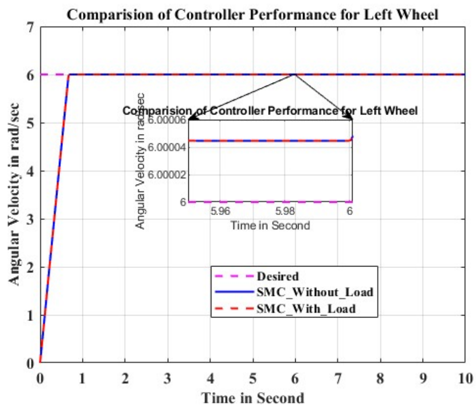
(c) Right Wheel Angular Velocity with 15 Kg



(d) Left Wheel Angular Velocity with 15 Kg



(e) Right Wheel Angular Velocity with 25 Kg



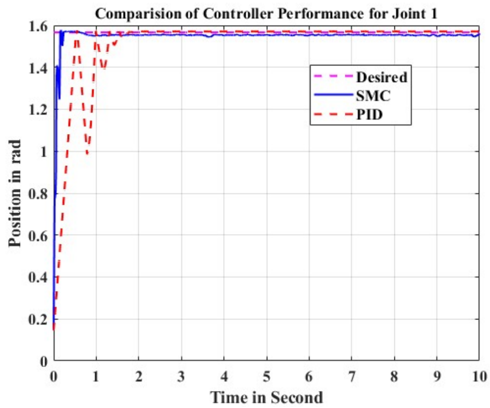
(f) Left Wheel Angular Velocity with 25 Kg

Figure 5.17: Performance of a Controller under Different Types of Weight on a Mobile Platform

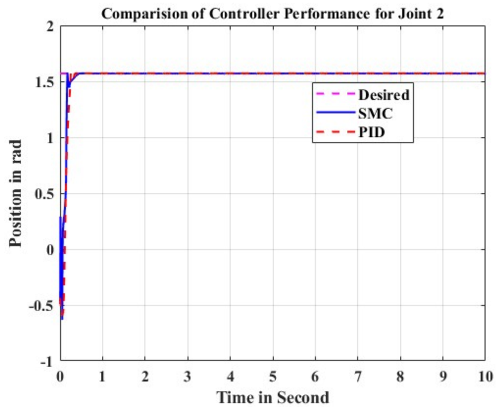
The strength and load-bearing capability of a platform is assessed by changing the weight of a fruit crate at various levels. The findings presented in figures (5.17) demonstrate that the controller maintains stability and robustness even when confronted with different masses.

5.6.4 Performance Evaluation and Comparison of Designed Controller

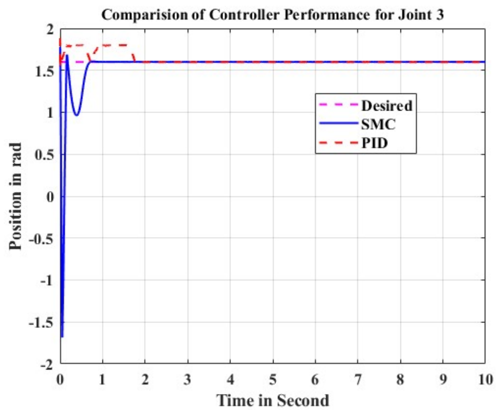
Figure (5.18) depicts the assessment of the designed controller's performance in relation to the PID controller, specifically regarding the tracking of manipulator joint positions and the angular velocity of the mobile platform.



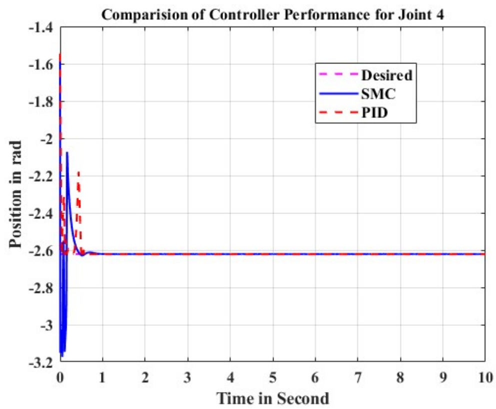
(a) Position Tracking of Joint 1



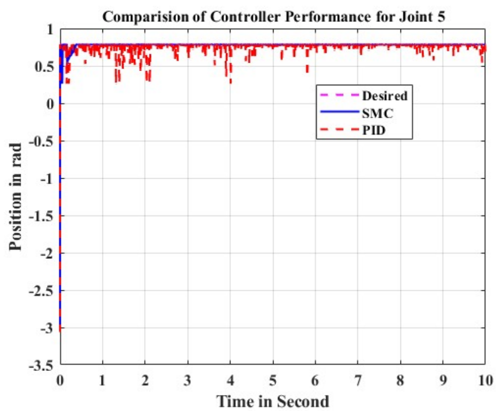
(b) Position Tracking of Joint 2



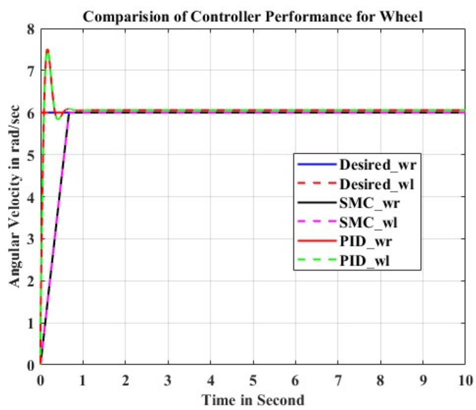
(c) Position Tracking of Joint 3



(d) Position Tracking of Joint 4



(e) Position Tracking of Joint 5



(f) Angular Velocity of Wheels

Figure 5.18: Comparative Evaluation of Designed Controller with PID for Position Tracking and Angular Velocity

The figures presented in (5.18a), (5.18b), (5.18c), (5.18d), and (5.18e) were utilized to compare the performance of the manipulator's joint controller (joint one, two, three, four, and five) to a conventional PID controller. The simulation results demonstrated that the SMC surpassed PID in effectively managing highly nonlinear systems that are affected by uncertainties and disturbances. This can be seen in the figure (5.18), where SMC exhibited a rapid and accurate response with minimal overshoot and chattering. Conversely, PID controllers perform well in situations where the system dynamics are relatively linear and predictable. Furthermore, the performance of the designed controller for the angular velocity of the mobile platform wheels was compared to that of PID, and as depicted in figure (5.18f), SMC showcased even greater superiority over PID. This suggests that SMC is a more robust controller for highly nonlinear systems that encounter uncertainties and disturbances.

In general, Sliding mode controllers exhibit remarkable capability in handling external disturbances and uncertainties, as demonstrated in the depicted figures (5.15), (5.16), (5.17) and (5.18). These controllers are specifically designed to cope with system uncertainties, disturbances, and nonlinearities, resulting in enhanced robustness. They deliver exceptional performance even when confronted with a model mismatch or external disturbances. In short, while PID controllers are extensively employed and comparatively easier to implement, sliding mode controllers offer robustness against uncertainties and disturbances. Hence, sliding mode controllers are often the preferred choice for applications involving nonlinear system dynamics or unpredictable disturbances.

The table (5.1) provides metrics for each joint's integral time absolute error, used to compare controller performances, and indicates that the controller with the lower ITAE value has better performance in minimizing the integral error over time.

Table 5.1: Comparison of Controller Performance in Terms of Integral Time Absolute Error

Joints	Controller		
	SMC	SMC_Uncertainty	PID
Joint_1	0.2952	0.5325	0.2641
Joint_2	0.04438	0.0448	0.07182
Joint_3	0.1275	0.1533	3.196
Joint_4	0.02466	0.04565	0.3752
Joint_5	0.1468	0.1516	1.142
Right_Wheel	0.4505	0.4542	2.764
Left_Wheel	0.4505	0.4542	2.764

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

In this study, a mobile manipulator system based on artificial intelligence has been designed specifically for agricultural purposes. The system incorporates a SMC for precise control of manipulator and mobile platform positions. The entire system was meticulously modeled, designed, and simulated to ensure its feasibility and effectiveness. Advanced CAD software was utilized for the mechanical design, resulting in the creation of a $3D$ physical model. Extensive verification and testing were conducted to determine the acceptability and functionality of the modeled system, without the need for additional actuation. The mathematical analysis was briefly discussed and applied to design a controller that uses image inputs to regulate the positioning of both the manipulator and the mobile platform.

One of the crucial aspects of the developed system is the ability to detect and ascertain the position of fruits suitable for harvesting. This was achieved through the integration of well-established AI components including image processing, computer vision, and deep learning algorithms. The YOLOv2 object detection algorithm, known for its real-time analysis capabilities, was employed for fruit detection. Employing an eye-in-hand camera configuration, positioned on top of the manipulator's end-effector, fruit positions were determined using a perspective projection approach and a position-based servoing control mechanism. To validate the effectiveness of the system, the detected fruit positions were utilized as inputs for the inverse kinematics of the designed model. This dynamic control scheme allowed for precise control, stabilization, and tracking of real-world physical plants within the MATLAB Simulink environment. Though training was conducted with a small dataset, it is noted that incorporating a larger dataset could potentially enhance precision and robustness at the cost of additional training time. To address uncertainties and variations in parameters, SMC was employed. The SMC scheme resulted in a stable and resilient system capable of withstanding various uncertainties. This controller design approach is advantageous due to its simplicity and ease of implementation. Performance analysis was conducted, specifically evaluating the robustness of the designed controller against uncertainties and parameter variations by introducing a payload representative of the manipulator and mobile platform's maximum carrying capacity.

Through extensive simulations, it was demonstrated that the proposed SMC effectively stabilized the system, even in the presence of parameter variations and external disturbances. Furthermore, the deployment of SMC successfully countered perturbations resulting in a quick transient response. Comparative analysis with a conventional PID controller revealed the superiority of the SMC strategy in terms of its ability to effectively mitigate uncertainties. The simulation results indicated that the modeled mobile manipulator achieved its intended objectives in obstacle-free scenarios. The model proved to be robust under external disturbances with a maximum weight of $25kg$, giving confidence in its real-world application within agricultural environments. Overall, the findings of this research demonstrate the success of modeled AI-based mobile manipulators in ensuring stability, accuracy, and adaptability in agricultural applications for fruit harvesting.

6.2 Recommendation

As the design, modeling, and simulation of the mobile manipulator with AI-based algorithms cover a huge discipline and subsections of different fields, this work was specific to some of them stated in the specific objective, and showed results. But in the future, the following work will be included and worked on further by interested researchers:

- This work addresses obstacle-free movement; therefore, it should be considered navigation for mobile manipulators by taking into account complicated environments with obstacles.
- Despite the existence of a more advantageous SMC, chattering remains a significant problem. However, by integrating a system like a neuro-fuzzy with other variations of SMC, the overall performance of the controller can be greatly improved.

Bibliography

[1] Thomas R Kurfess et al. *Robotics and automation handbook*, volume 414. CRC press Boca Raton, FL, 2005.

[2] Ben-Zion Sandler. *Robotics: designing the mechanisms for automated machinery*. Academic Press, 1999.

[3] Burra Hymavathi, J Hariharan, K Manideep, and DV Srikanth. Design and structure of multi-purpose agricultural robot. *International Journal of Engineering Research & Technology (IJERT)*, 9(5):475–478, 2020.

[4] Farzin Piltan, Sara Emamzadeh, Zahra Hivand, Fatemeh Shahriyari, and Mina Mirazaei. Puma-560 robot manipulator position sliding mode control methods using matlab/simulink and their integration into graduate/undergraduate nonlinear control, robotics and matlab courses. *International Journal of Robotics and Automation*, 3(3):106–150, 2012.

[5] Hassan K Khalil. *Nonlinear control*, volume 406. Pearson New York, 2015.

[6] J-X Xu. Sliding mode control in engineering-wilfrid perruquetti and jean pierre barbot (eds.); marcel dekker, new york, 2002, isbn 0-419-19280-8. *Automatica*, 5(39):951–954, 2003.

[7] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

[8] Wolfgang Ertel. *Introduction to artificial intelligence*. Springer, 2018.

[9] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[10] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[11] Spyros G Tzafestas. Mobile robots: General concepts. *Introduction to Mobile Robot Control, 1st ed. Athens, Greece: Elsevier Inc*, pages 1–29, 2014.

[12] Peilin Li, Sang-heon Lee, and Hung-Yao Hsu. Review on fruit harvesting method for potential use of automatic fruit harvesting systems. *Procedia Engineering*, 23:351–366, 2011.

[13] Björn Åstrand and Albert-Jan Baerveldt. An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous robots*, 13:21–35, 2002.

[14] Rubens Andre Tabile, Rafael Vieira de Sousa, Ricardo Yassushi Inamasu, and Arthur José Vieira Porto. Agribot: development of a mobile robotic platform to support agricultural data collection. In *Proceedings*, 2014.

[15] R Chandana, M Nisha, B Pavithra, S Suresh, and RN Nagashree. A multipurpose agricultural robot for automatic ploughing, seeding and plant health monitoring. *International Journal of Engineering Research & Technology (IJERT) IETE*, 8(11), 2020.

[16] Ivan Beloev, Diyana Kinaneva, Georgi Georgiev, Georgi Hristov, and Plamen Zahariev. Artificial intelligence-driven autonomous robot for precision agriculture. *Acta Technologica Agriculturae*, 24(1):48–54, 2021.

[17] Eleni Vrochidou, Konstantinos Tziridis, Alexandros Nikolaou, Theofanis Kalampokas, George A Papakostas, Theodore P Pachidis, Spyridon Mamalis, Stefanos Koundouras, and Vassilis G Kaburlasos. An autonomous grape-harvester robot: integrated system architecture. *Electronics*, 10(9):1056, 2021.

[18] Bei He, Gang Liu, Ying Ji, Yongsheng Si, and Rui Gao. Auto recognition of navigation path for harvest robot based on machine vision. In *Computer and Computing Technologies in Agriculture IV: 4th IFIP TC 12 Conference, CCTA 2010, Nanchang, China, October 22-25, 2010, Selected Papers, Part I 4*, pages 138–148. Springer, 2011.

[19] Ya Xiong, Cheng Peng, Lars Grimstad, Pål Johan From, and Volkan Isler. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Computers and electronics in agriculture*, 157:392–402, 2019.

[20] Abhijeet Ravankar, Ankit A. Ravankar, Michiko Watanabe, Yohei Hoshino, and Arpit Rawankar. Development of a low-cost semantic monitoring system for vineyards using autonomous robots. *Agriculture*, 10(5):182, 2020.

[21] Eduardo Navas, Roemi Fernandez, Delia Sepúlveda, Manuel Armada, and Pablo Gonzalez-de Santos. Soft grippers for automatic crop harvesting: A review. *Sensors*, 21(8):2689, 2021.

[22] Qingchun Feng, Wei Zou, Pengfei Fan, Chunfeng Zhang, and Xiu Wang. Design and test of robotic harvesting system for cherry tomato. *International Journal of Agricultural and Biological Engineering*, 11(1):96–100, 2018.

[23] Alistair J Scarfe, Rory C Flemmer, HH Bakker, and Claire L Flemmer. Development of an autonomous kiwifruit picking robot. In *2009 4th International Conference on Autonomous Robots and Agents*, pages 380–384. IEEE, 2009.

[24] Divyansh Khare, Sandra Cherussery, and Santhakumar Mohan. Investigation on design and control aspects of a new autonomous mobile agricultural fruit harvesting robot. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236(18):9966–9977, 2022.

[25] Faiza Gul, Wan Rahiman, and Syed Sahal Nazli Alhady. A comprehensive study for robot navigation techniques. *Cogent Engineering*, 6(1):1632046, 2019.

[26] Ahmad Taher Azar, Azher M Abed, Farah Ayad Abdul-Majeed, Ibrahim A Hameed, Anwar Ja'afar Mohamad Jawad, Wameedh Riyadh Abdul-Adheem, Ibraheem Kasim Ibraheem, and Nashwa Ahmad Kamal. Design and stability analysis of sliding mode controller for non-holonomic differential drive mobile robots. *Machines*, 11(4):470, 2023.

[27] Lixiong Lin, Peixin Wu, Bingwei He, Yanjie Chen, Jiachun Zheng, and Xiafu Peng. The sliding mode control approach design for nonholonomic mobile robots based on non-negative piecewise predefined-time control law. *IET Control Theory & Applications*, 15(9):1286–1296, 2021.

[28] Kevin M Lynch and Frank C Park. Modern robotics-mechanics, planning, and control: Video supplements and software. 2017.

[29] BBVL Deepak, Dayal R Parhi, and Ravi Praksh. Kinematic control of a mobile manipulator. In *Proceedings of the International Conference on Signal, Networks, Computing, and Systems: ICSNCS 2016, Volume 2*, pages 339–346. Springer, 2016.

[30] Guy Campion, Georges Bastin, and Brigitte D'Andréa-Novel. Structural properties and classification on kinematic and dynamic models of wheeled mobile robots. *Russian Journal of Nonlinear Dynamics*, 7(4):733–769, 2011.

[31] Shimon Y Nof. *Handbook of industrial robotics*. John Wiley & Sons, 1999.

[32] Peter C Müller. Robot dynamics and control: Mark w. spong and m. vidyasagar, 1992.

[33] Tahseen F Abaas, Ali A Khleif, and Mohanad Q Abbood. Inverse kinematics analysis and simulation of a 5 dof robotic arm using matlab. *Al-Khwarizmi Engineering Journal*, 16(1):1–10, 2020.

[34] Saeed B Niku. *Introduction to robotics: analysis, control, applications*. John Wiley & Sons, 2020.

[35] Alonzo Kelly. *Mobile robotics: mathematics, models, and methods*. Cambridge University Press, 2013.

[36] Vladimir Prada-Jiménez, Paola A Niño-Suárez, Mauricio F Mauledoux-Monroy, and J Aponte-Rodríguez. Coupled dynamic model for a mobile manipulator. *International Journal of Applied Engineering Research*, 14(11):2622–2629, 2019.

[37] Ignacy Duleba. Modeling and control of mobile manipulators. *IFAC Proceedings Volumes*, 33(27):447–452, 2000.

[38] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[39] Gerald Cook and Feitian Zhang. *Mobile robots: Navigation, control and sensing, surface robots and AUVs*. John Wiley & Sons, 2020.

[40] Abhishek Chandak, Ketki Gosavi, Shalaka Giri, Sumeet Agrawal, and P Kulkarni. Path planning for mobile robot navigation using image processing. *International Journal of Scientific & Engineering Research*, 4(6):1490–1495, 2013.

[41] Shahed Shojaeipour, Sallehuddin Mohamed Haris, and Muhammad Ihsan Khairir. Vision-based mobile robot navigation using image processing and cell decomposition. *IVIC*, 9:90–96, 2009.

[42] Mahmut DİRİK and A Fatih KOCAMAZ. Global vision based path planning for avgs using a* algorithm. *Journal of Soft Computing and Artificial Intelligence*, 1(1):18–27, 2020.

[43] Héctor M Becerra, Carlos Sagüés, Héctor M Becerra, and Carlos Sagüés. Robust visual control based on the epipolar geometry. *Visual Control of Wheeled Mobile Robots: Unifying Vision and Control in Generic Approaches*, pages 21–44, 2014.

[44] Tom Taulli. *Artificial Intelligence basics: A non-technical introduction*. Apress, 2019.

[45] Deep Learning Algorithms javatpoint. `https://www.javatpoint.com/deep-learning-algorithms`. Accessed: Dec. 29, 2022.

[46] Sakshi Gupta and Dr T Uma Devi. Yolov2 based real-time object detection. *Int. J. Comput. Sci. Trends Technol. IJCST*, 8:26–30, 2020.

[47] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[48] Peter I Corke and Oussama Khatib. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer, 2011.

[49] Chaojun Hou, Xiaodi Zhang, Yu Tang, Jiajun Zhuang, Zhiping Tan, Huasheng Huang, Weilin Chen, Sheng Wei, Yong He, and Shaoming Luo. Detection and localization of citrus fruit based on improved you only look once v5s and binocular vision in the orchard. *Frontiers in Plant Science*, 13, 2022.

[50] Yuri Shtessel, Christopher Edwards, Leonid Fridman, Arie Levant, Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. Disturbance observer based control: aerospace applications. *Sliding mode control and observation*, pages 291–320, 2014.

[51] Jinkun Liu. *Sliding mode control using MATLAB*. Academic Press, 2017.

Appendix A

Design Specification of Mobile Manipulator

This section provides a description of the design specifications and model parameters utilized in plant design and control. Additionally, the physical parameter values used in the simulation are derived from the design created in SolidWorks software.

Table A.1: Design Specification of Mobile Manipulator

List of Physical Parameter used in Design and Simulation of MM					
Symbols	Values	Units	Symbols	Values	Units
L	0.6	m	m_5	0.16	kg
r	0.1175	m	L_{cmx1}	0.01	kg.m^2
d	0.39	m	L_{cmx2}	0.01	kg.m^2
L_{cmxp}	11.27	m	L_{cmx3}	0.01	kg.m^2
L_0	0	m	I_p	19.35	kg.m^2
L_1	0.33	m	I_{xx1}	0.01	kg.m^2
L_2	0.79	m	I_{yy1}	0.01	kg.m^2
L_3	0.65	m	I_{zz1}	0.01	kg.m^2
L_4	0	m	I_{xx2}	0.02	kg.m^2
L_5	0.3	m	I_{yy2}	0.33	kg.m^2
m_p	50.99	kg	I_{zz2}	0.34	kg.m^2
m_w	1.08	kg	I_{xx3}	0.01	kg.m^2
m_1	1.34	kg	I_{yy3}	0.15	kg.m^2
m_2	4.57	kg	I_{zz3}	0.15	kg.m^2
m_3	3.15	kg	I_{yyw}	0.01	kg.m^2
m_4	0	kg	I_{zzw}	0.01	kg.m^2

Appendix B

Dynamic Model of Mobile Manipulator

In this section, the focus is on explaining the design of the mathematical expression of a mobile manipulator by discussing the transformation and inertia matrix. The ultimate goal is to derive the general dynamic equation of the robot.

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

The inertial matrix of a mobile manipulator. Inertia Matrix of row1

$$M_{11} = p_0 - p_1 * cos(\theta_{1a}) * sin(\theta_{2a}) + p_2 * (sin(\theta_{2a})^2) + p_3 * sin(\theta_{1a}) * sin(\theta_{2a})$$

$$M_{12} = q_0 + p_1 * cos(\theta_{1a}) * sin(\theta_{2a}) - p_2 * (sin(\theta_{2a})^2)$$

$$M_{13} = q_1 * sin(\theta_{1a}) * sin(\theta_{2a}) - q2 * cos(\theta_{1a}) * sin(\theta_{2a}) + q_3 * (sin(t_2a)^2) + q_4$$

$$M_{14} = -q_1 * cos(\theta_{1a}) * cos(\theta_{2a}) - q_2 * sin(\theta_{1a}) * cos(\theta_{2a})$$

$$M_{15} = -q_5 * cos(\theta_{1a}) * cos(\theta_{2a}) - q_{6*} sin(\theta_{1a}) * cos(\theta_{2a})$$

Inertia Matrix of row2

$$M_{21} = M_{12}$$

$$M_{22} = p_0 - p_1 * cos(\theta_{1a}) * sin(\theta_{2a}) + p_2 * (sin(\theta_{2a})^2) - p_3 * sin(\theta_{1a}) * sin(\theta_{2a})$$

$$M_{23} = q_1 * sin(\theta_{1a}) * sin(\theta_{2a}) + q_2 * cos(\theta_{1a}) * sin(\theta_{2a}) - q_3 * (sin(\theta_{2a})^2) - q_4$$

$$M_{24} = -q_1 * cos(\theta_{1a}) * cos(\theta_{2a}) + q_2 * sin(\theta_{1a}) * cos(\theta_{2a})$$

$$M_{25} = -q_5 * cos(\theta_{1a}) * cos(\theta_{2a}) + q_6 * sin(\theta_{1a}) * cos(\theta_{2a})$$

Inertia Matrix of row3

$$M_{31} = -q_1 * sin(\theta_{1a}) * sin(\theta_{2a}) - q_2 * cos(\theta_{1a}) * sin(\theta_{2a}) + q_3 * (sin(\theta_{2a})^2) + q_4$$

$$M_{32} = q_1 * sin(\theta_{1a}) * sin(\theta_{2a}) + q_2 * cos(\theta_{1a}) * sin(\theta_{2a}) - q_3 * (sin(\theta_{2a})^2) - q_4$$

$$M_{33} = I_{21} + I_{22} + m_2 * a_2^2 * (sin(\theta_{2a})^2)$$

$$M_{34} = M_{43} = M_{35} == 0$$

Inertia Matrix of row4

$$M_{41} = -q_1 * cos(\theta_{1a}) * cos(\theta_{2a}) - q_2 * sin(\theta_{1a}) * cos(\theta_{2a})$$

$$M_{42} = -q_1 * cos(\theta_{1a}) * cos(\theta_{2a}) + q_2 * sin(\theta_{1a}) * cos(\theta_{2a})$$

$$M_{44} = q_4; \qquad M_{45} = 0$$

Inertia Matrix of row5

$$M_{51} = M_{15}; \qquad M_{52} = M_{25}; \qquad M_{55} = q_7; \qquad M_{53} = M_{35} = M_{54} = M_{45} == 0$$

The Corololis matrix of a mobile manipulator

Corololis Matrix of row1

$$C_{11} = (p_1 * sin(\theta_{1a}) * sin(\theta_{2a}) + p_3 * cos(\theta_{1a}) * sin(\theta_{2a})) * \frac{\dot{\theta}_{1a}}{2} + (2 * p_2 * sin(\theta_{2a}) * cos(\theta_{2a})$$

$$+p_3 * sin(\theta_{1a}) * cos(\theta_{2a}) - p_1 * cos(\theta_{1a}) * cos(\theta_{2a})) * \frac{\dot{\theta}_{2a}}{2})$$

$$C_{12} = -p_1 * sin(\theta_{1a}) * sin(\theta_{2a}) * \dot{\theta}_{1a} + (p_1 * cos(\theta_{1a}) * cos(\theta_{2a}) - 2 * p_2 * sin(\theta_{2a}) * cos(\theta_{2a})) * \dot{\theta}_{2a}$$

$$C_{13} = (p_1 * sin(\theta_{1a}) * sin(\theta_{2a}) + p_3 * cos(\theta_{1a}) * sin(\theta_{2a})) * \frac{w_r}{2} - p_3 * sin(\theta_{1a}) * sin(\theta_{2a}) * w_l$$

$$+q_1 * cos(\theta_{1a}) * sin(\theta_{2a}) - q_2 * sin(\theta_{1a}) * sin(\theta_{2a})) * \dot{\theta}_{1a} + (q_1 * sin(\theta_{1a}) * cos(\theta_{2a})$$

$$-q_2 * cos(\theta_{1a}) * cos(\theta_{2a}) + q_3 * sin(\theta_{2a}) * cos(\theta_{2a})) * \dot{\theta}_{2a}$$

$$C_{14} = (-p_1 * cos(\theta_{1a}) * cos(\theta_{2a}) + 2 * p_2 * sin(\theta_{2a}) * cos(\theta_{2a}) + p_3 * sin(\theta_{1a}) * cos(\theta_{2a})) * \frac{w_r}{2}$$

$$+(p_1*cos(\theta_{1a})*cos(\theta_{2a})+2*p_2*sin(\theta_{2a})*cos(\theta_{2a}))*w_l+(q_1*sin(\theta_{1a})*cos(\theta_{2a})$$

$$-q_2*cos(\theta_{1a})*cos(\theta_{2a})+q_3*sin(\theta_{2a})*cos(\theta_{2a}))*\dot{\theta}_{1a}+(q_1*cos(\theta_{1a})*sin(\theta_{2a})$$

$$+q_2*sin(\theta_{1a})*sin(\theta_{2a}))*\dot{\theta}_{2a}$$

$$C_{15}=(-p_1*cos(\theta_{1a})*cos(\theta_{2a})+2*p_2*sin(\theta_{2a})*cos(\theta_{2a})+p_3*sin(\theta_{1a})*cos(\theta_{2a}))*\frac{w_r}{2}$$

$$+(p*cos(\theta_{1a})*cos(\theta_{2a})+2*p_2*sin(\theta_{2a})*cos(\theta_{2a}))*w_l+(q_1*sin(\theta_{1a})*cos(\theta_{2a})-q_2*cos(\theta_{1a})*cos(\theta_{2a})$$

$$+q_3*sin(\theta_{2a})*cos(\theta_{2a}))*\dot{\theta}_{1a}+(q_1*cos(\theta_{1a})*sin(\theta_{2a})+q_2*sin(\theta_{1a})*sin(\theta_{2a}))*\dot{\theta}_{2a}$$

Corololis Matrix of row2

$$C_{21}=(-p_1*sin(\theta_{1a})*sin(\theta_{2a}))*\dot{\theta}_{1a}/2+(p_1*cos(\theta_{1a})*cos(\theta_{2a})$$

$$-2*p_2*sin(\theta_{2a})*cos(\theta_{2a}))*\dot{\theta}_{2a}$$

$$C_{22}=(p_1*sin(\theta_{1a})*sin(\theta_{2a})-p_2*cos(\theta_{1a})*sin(\theta_{2a})-p_3*cos(\theta_{1a})*sin(\theta_{2a}))*\dot{\theta}_{1a}/2$$

$$+(2*p_2*sin(\theta)*cos(\theta_{2a})-p_3*sin(\theta_{1a})*cos(\theta_{2a})-p_1*cos(\theta_{1a})*cos(\theta_{2a}))*\dot{\theta}_{2a}/2$$

$$C_{23}=(-p_1*sin(\theta_{1a})*sin(\theta_{2a}))*\frac{w_r}{2}+(p_1*sin(\theta_{1a})*sin(\theta_{2a})-p_3*cos(\theta_{1a})*sin(\theta_{2a}))*\frac{w_l}{2}$$

$$+(q_1*cos(\theta_{1a})*sin(\theta_{2a})-q_2*sin(\theta_{1a})*sin(\theta_{2a}))*\dot{\theta}_{1a}+(q_1*sin(\theta_{1a})*cos(\theta_{2a})$$

$$+q_2*cos(\theta_{1a})*cos(\theta_{2a})-q_3*sin(\theta_{2a})*cos(\theta_{2a})*t\dot{\theta}_{2a})$$

$$C_{24}=(p_1*cos(\theta_{1a})*cos(\theta_{2a})-2*p_2*sin(\theta_{2a})*cos(\theta_{2a}))*\frac{w_r}{2}$$

$$+(2*p_2*sin(\theta_{2a})*cos(\theta_{2a})-p_1*cos(\theta_{1a})*cos(\theta_{2a})-p_3*sin(\theta_{1a})*cos(\theta_{2a}))*\frac{w_l}{2}$$

$$+(q_1*sin(\theta_{1a})*cos(\theta_{2a})+q_2*cos(\theta_{1a})*cos(\theta_{2a})-q_3*sin(\theta_{2a})*cos(\theta_{2a})*\dot{\theta}_{1a})$$

$$C_{25}=(p_1*cos(\theta_{1a})*cos(\theta_{2a})-2*p_2*sin(\theta_{2a})*cos(\theta_{2a}))*\frac{w_r}{2}+(2*p_2*sin(\theta_{2a})*cos(\theta_{2a})$$

$$-p_1*cos(\theta_{1a})*cos(\theta_{2a})-p_3*sin(\theta_{1a})*cos(\theta_{2a}))*\frac{w_l}{2}+(q_1*sin(\theta_{1a})*cos(\theta_{2a})+q_2*cos(\theta_{1a})*cos(\theta_{2a})$$

$$-q_3*sin(\theta_{2a})*cos(\theta_{2a}))*\dot{\theta}_{1a})$$

Corololis Matrix of row3

$$C_{31}=-(p_1*sin(\theta_{1a})*sin(\theta_{2a})+p_3*cos(\theta_{1a})*sin(\theta_{2a}))*\frac{w_r}{2}+(p_1*sin(\theta_{1a})*sin(\theta_{2a})-$$

$$p_3 * cos(\theta_{1a}) * sin(\theta_{2a})) * \frac{w_l}{2} + q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * \dot{\theta}_{2a}$$

$$C_{32} = (p_1 * sin(\theta_{1a}) * sin(\theta_{2a}) - p_3 * cos(\theta_{1a}) * sin(\theta_{2a})) * \frac{w_r}{2} + (p_3 * cos(\theta_{1a}) * sin(\theta_{2a}) -$$

$$p_1 * sin(\theta_{1a}) * sin(\theta_{2a})) * \frac{w_l}{2} - q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * \dot{\theta}_{2a}$$

$$C33 = 0$$

$$C_{34} = q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * w_r - q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * w_l + m_2 * a_2^2 * sin(\theta_{2a}) * cos(\theta_{2a}) * w_l$$

$$C_{35} = q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * w_r - q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * w_l + m_2 * a_2^2 * sin(\theta_{2a}) * cos(\theta_{2a}) * w_l$$

Corololis Matrix of row4

$$C_{41} = (p_1 * cos(\theta_{1a}) * cos(\theta_{2a}) - I_{wp2} * sin(\theta_{2a}) * cos(\theta_{2a}) - p_3 * sin(\theta_{1a}) * cos(\theta_{2a})) * \frac{w_r}{2} -$$

$$(p_1 * cos(\theta_{1a}) * cos(\theta_{2a}) - 2 * p_2 * sin(\theta_{2a}) * cos(\theta_{2a})) * \frac{w_l}{2} + q_3 * sin(\theta_{2a}) * cos(\theta_2 a) * w_l$$

$$C_{42} = -(p_1 * cos(\theta_{1a}) * cos(\theta_{2a}) - 2 * p_2 * sin(\theta_{2a}) * cos(\theta_{2a})) * \frac{w_r}{2} + (p_1 * cos(\theta_{1a}) * cos(\theta_{2a}) -$$

$$I_{wp2} * sin(\theta_{2a}) * cos(\theta_{2a}) + p_3 * sin(\theta_{1a}) * cos(\theta_{2a})) * \frac{w_l}{2} + q_3 * sin(\theta_{1a}) * cos(\theta_{2a}) * w_l$$

$$C_{43} = q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * \dot{\theta}_r + q_3 * sin(\theta_{2a}) * cos(\theta_{2a}) * \dot{\theta}_l - m_2 * a_2^2 * sin(\theta_{2a}) * cos(\theta_{2a}) * \dot{\theta}_l$$

$$C_{44} = 0; \qquad C45 = 0;$$

Corololis Matrix of row5

$$C_{51} = C_{15}; \qquad C_{52} = C_{25}; \qquad C53 = C_{35}; \qquad C_{54} = C_{45}; \qquad C_{55} = 0;$$

$$G(\theta) = \begin{bmatrix} G_1 & G_2 & G_3 & G_4 & G_5 \end{bmatrix}^T$$

Gravitational Matrix

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ -m_2 * g * (d_1 + a_2 * sin(\theta_{2a})) \\ -m_3 * g * (d_1 + 2 * a_2 * sin(\theta_{2a}) + 2 * a_3 * sin(\theta_{2a} + \theta_{3a})) \end{bmatrix}$$

Appendix C

MATLAB Script and Simulink Block

C.1 Matlab Code used in Design Process

The MATLAB script used in the design for Object detection and Position determination is described below.

```
MATLAB code ...
%% ==============================================================
% Designing object detection network using YOLOv2 object detection
imageResizingNeeded = false;
if imageResizingNeeded
    folderName = 'FruitImage';
    srcDir = fullfile(folderName)
    srcFiles = dir([srcDir,'*.jpeg']);
    for i=1:length(srcFiles)
        filename =[srcDir ,srcFiles(i),name];
        im =imread(filename);
        k=imresisize(im, [416,416]);
        newfilename=[srcDir, srcFiles(i).name];
        imwrite(k,newfiles,'jpeg')
    end
end
splitFolder = false;
if splitFolder
    valFolderName = 'FruitImage'
```

```
    files = dir(fullfile(valFolderName));
    N = length(files);
    tf=randperm(N)>(0.50*N);
    mkdir TestResized;
    mkdir ValResized;
    for i = 3:length(tf)
        files_re = files(i).name;
        if tf(i)==1
            copyfile (fullfile(valFolderName,files_re), 'test')
        else
            copyfile (fullfile(valFolderName,files_re), 'val')
        end
    end
end
isGroundTruthAvailable = false;
if~isGroundTruthAvailable
        groundTruthLabeler;
else
        labelingSessionName ='Zedo\groundTruthLabelingSession.mat';
        groundTruthLabeler(labelingSessionName)
end
if ~exist('gTruthResizedTrain','var')
        load('TrainModel\gTruth.mat');
end
if ~isfolder(fullfile('TraingData'))
        mkdir TrainingData
end
trainingData = objectDetectorTrainingData(gTruth,'SamplingFactor'
,1,...'WriteLocation','TrainingData');
trainingData(1:4,:)
if ~exist('gTruthResizedVal','var')
        load('Zedo\gTruthVal.mat');
```

```
end
if ~isfolder(fullfile('ValData'))
        mkdir ValData
end
ValData = objectDetectorTrainingData(gTruthVal,'SamplingFactor',
1,...'WriteLocation','ValData');
if ~exist('gTruthResizedTest','var')
        load('Zedo\gTruthTest.mat');
end
if ~isfolder(fullfile('TestData'))
        mkdir TestData
end
TestData = objectDetectorTrainingData(gTruthTest,'SamplingFactor',
1,...'WriteLocation','TestData');
inputLayer = imageInputLayer([128 128 3],'Name','input',
'Normalization','none');
 filterSize = [3 3];
featureExtractionNetwork = resnet50;
numClasses = size(trainingData,2)-1;
open(fullfile("codeFiles","AnchorBoxes.m"));
Anchors = [43 59
    18 22
    23 29
    84 109];
lgraph = yolov2Layers([128 128 3], numClasses,
Anchors,featureExtractionNetwork,'activation_40_relu');
analyzeNetwork(lgraph);
doTraining = false;
if doTraining
    rng(0)
     options = trainingOptions('sgdm', ...
        'InitialLearnRate',0.01, ...
```

```
        'Verbose',true,'MiniBatchSize',16,'MaxEpochs',160,..
        'Shuffle','every-epoch','VerboseFrequency',50, ...
        'DispatchInBackground',false,...
        'ExecutionEnvironment','auto');
    [detectorYolo2, info] = trainYOLOv2ObjectDetector(trainingData,
    lgraph,options);
else
    load(fullfile("Zedo","detectorYolo2.mat"));
end
detectionResults = table('Size',[height(TestData) 3],...
    'VariableTypes',{'cell','cell','cell'},...
    'VariableNames',{'Boxes','Scores', 'Labels'});
depVideoPlayer = vision.DeployableVideoPlayer;
for i = 1:height(TestData)
    [bboxes,Scores,Labels] = detect(detectorYolo2,I);
    if ~isempty(bboxes)
        annotations = string(Labels) + ": " + string(Scores);
        I = insertObjectAnnotation(I,'rectangle',
        bboxes,cellstr(annotations));
        depVideoPlayer(I);
        pause(0.1);
    end
end
    detectionResults.Boxes{i} = floor(bboxes);
    detectionResults.Scores{i} = Scores;
    detectionResults.Labels{i} = Labels;
threshold = 0.5;
detectionResults = detect(detectorYolo2, TestData);
```

C.2 Simulink Block of Mobile Manipulator

The below Simulink block shows the Simulink model of a mobile manipulator designed in SolidWorks and imported to MATLAB using Simscape.

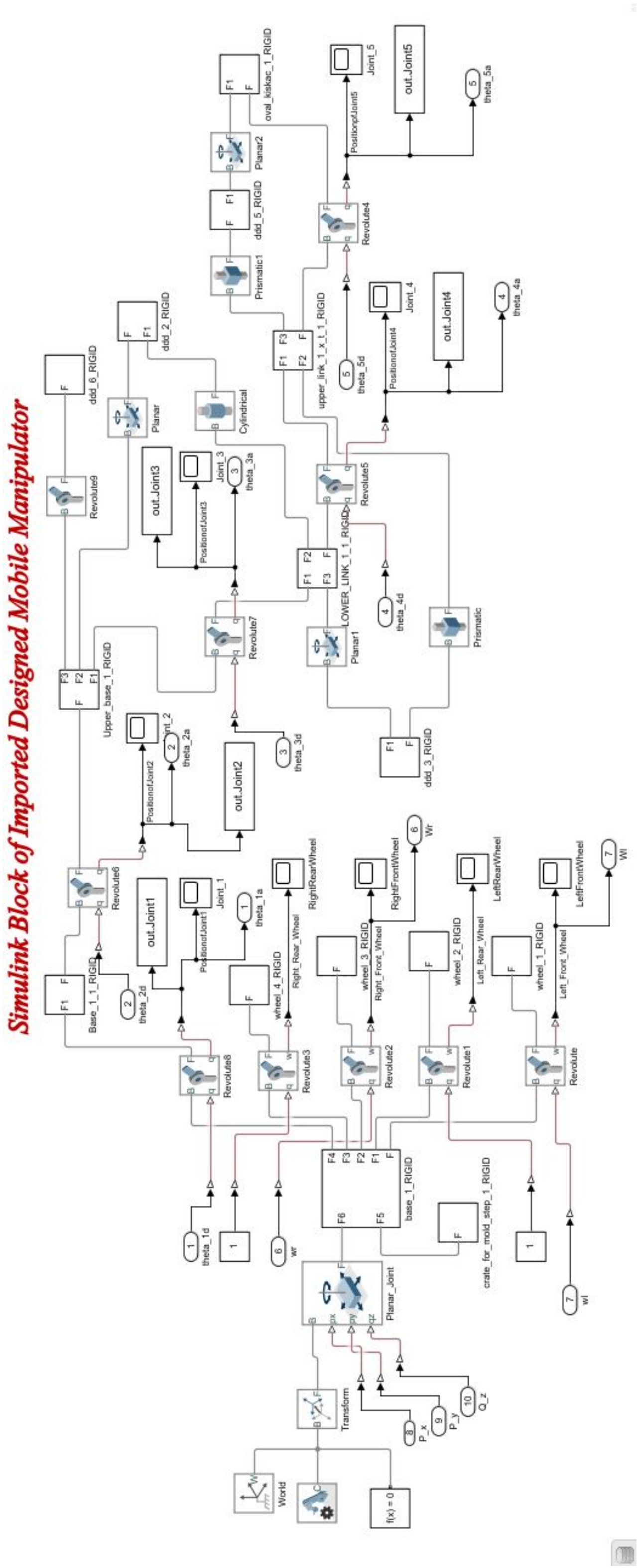


Figure C.1: Simulink block of Mobile Manipulator