

**Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Engineering
Electrical Engineering Dept.**



Design and Simulation of a Mobile Robot with a Moving Arm Controlled by Custimized Maps

A Thesis

*Submitted to the Department of Electrical Engineering /
College of Engineering / University of Babylon in Partial
Fulfillment
of the Requirements for the Degree of Master of Science
in Engineering / Electrical Engineering / Electronics.*

By

Abather Saad Kadhim

Supervised by

Prof. Dr. Ehab AbdulRazzaq Hussein

Assist. Prof. Hayder Mahdi Abdulridha

2023 A.D.

1445 A.H.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ
وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ)

صدق الله العظيم

سورة المجادلة : 11

Supervisors Certification

I certify that this thesis titled “**Design and Simulation of a Mobile Robot with a Moving Arm Controlled by Custimized Maps**” was prepared by under my supervision at the Electrical Engineering Department, College of Engineering at University of Babylon, in partial fulfillment Abather Saad Kadhim of the requirements for the degree of Master of Science in Electrical Engineering/Electronics.

Supervisors

Signature:

Name: Prof. Dr. Ehab AbdulRazzaq Hussein

Date: / / 2023

Supervisors

Signature:

Name: Assist. Prof. Hayder Mahdi Abdulridha

Date: / / 2023

In view of the available recommendation, we forward this thesis for debate by the examining committee.

Head of the Electrical Engineering Department

Signature:

Name: Prof. Dr. Qais Karim Omran

Date: / / 2023

Acknowledgements

Thanks be to God for the insight that inspired me, and praise be to Him for the blessings He has bestowed upon me, and to Him is the credit for achieving what I aspire to in this research. Praise be to God, who by His praise opens every book and by His remembrance every speech is issued, and by His praise the people of blessings enjoy the abode of reward.

*It gives me great pleasure to extend my thanks and gratitude to (**Prof. Dr. Ehab AbdulRazzaq Hussein**,*

Assist. Prof. Hayder Mahdi Abdulridha), who provided me with the information I needed in my research

And for the great efforts he made for me since he proposed the subject of the research and his continuous supervision, and for his advice that helped me a lot in overcoming the difficulties I faced, calling from God success and brilliance in the scientific career.

I would like to extend my thanks and gratitude to the Deanship of the College of Engineering and the Presidency of the Electrical Engineering Department for the facilities and administrative procedures they provided me, which effectively contributed to the completion of the requirements of this research.

Researcher

Dedication

To my kind father... who taught me how to stand firmly above the earth, my role model, and my ideal in life; He is the one who taught me how to live with dignity and honor.

To my tender mother... the source of love, altruism and generosity, I cannot find words that can give her her due, for she is the epic of love and the joy of a lifetime, and an example of dedication and giving.

To my brothers... my support and I share my joys and sorrows.

To all my friends and to all those from whom I received advice and support;

I dedicate to you the summary of my scientific effort.

Abstract

This project presents the design and implementation of an omnidirectional wheel mobile robot with an integrated robotic arm, focusing on the development of control algorithms using MATLAB. The primary objectives of this work include achieving precise control of both the robot's omnidirectional motion and the arm's end-effector position through inverse kinematics and Jacobian matrix calculations.

The omnidirectional wheel system is designed to provide exceptional maneuverability, enabling the robot to move in any direction with ease. By incorporating inverse kinematics, Developed algorithms that translate desired end-effector positions into joint angles for the robotic arm, allowing for precise control over the arm's movements. Additionally, the Jacobian matrix is used to establish a relationship between joint velocities and end-effector velocities, enabling coordinated motion of the arm.

It also designed and implemented an omnidirectional wheel mobile robot equipped with an integrated robotic arm, controlled through a smartphone application. The primary objective is to provide a user-friendly interface for the remote operation of both the omnidirectional motion of the robot and the precise positioning of the robotic arm using a smartphone app.

The implementation leverages the capabilities of modern smartphones, integrating Bluetooth or Wi-Fi connectivity to establish a seamless link between the user's commands and the robot's actions.

List of Contents

Subject	Page
Abstract	I
List of Content	II
List of Tables	V
List of Figures	VI
List of Abbreviation	IX
List of Symbols	X
Chapter One: Introduction	
1.1 Introduction	1
1.2 Literature Review	3
1.3 Problem Statement	7
1.4 Aims of the Thesis	7
1.5 Thesis Organization	8
Chapter Two: Background and Theoretical Concepts	
2.1 Introduction	9
2.2 Moving Robot Arm Structure	9
2.3 Kinematics and Inverse Kinematics Definition	12
2.3.1 Kinematics Definition	12
2.3.2 Inverse Kinematics Definition	15
2.4 Robot Object Movement Planning	17
2.4.1 Kinds of Robotic Movements	19
A) Linear Motion	19
B) Articulation Movement	20
C) Circular Motion	20
2.4.2 Robot Car	21
A) Two-Wheel Robot Car	22
B) Four-Wheel Robot Car	23
C) Omnidirectional Robot Car	25

List of Contents

2.4.3 Performance Criteria	26
2.5 Microcontroller with AI Techniques	30
2.5.1 AI Applications and Future Technology	32
2.5.2 Opportunities, limitations and Ethics of AI	33
2.6 Kinds of AI Strategies Applied in Track Planning	34
2.6.1 Problem Demonstration	34
2.6.2 Problem Analysis	35
2.6.3 Optimization Of The Artificial Intelligence Algorithm	36
2.6.3.1 Robot Reference System Design of Artificial Intelligence Algorithm	36
2.6.3.2 Robot Direction Scheme Structuring and Optimization	37
Chapter Three: Methodology	
3.1 Introduction	40
3.2 The Suggested System Details	41
3.2.1 The Car_Wheels_ Subsystem	42
3.2.2 The Robot Arm Section	45
3.2.3 The trajectory determination	53
A) The Trajectory Motion Error	54
B) Jacobean matrix	55
Chapter Four: Result and Discussion	
4.1 Implementation Results	58
A. Catch Something	59
B. Draw Sign (∞)	67
C. Sine Wave	70
D. Coordinate Error	74
Chapter Five :Prototype Implementation	
5.1 Introduction	78

List of Contents

5.2 Robot Construction	79
5.2.1 Design of the Robot	79
5.2.2 3D Printer	79
5.2.3 3D Printer Materials	80
5.2.4 Assembly	81
5.3 Robot Components	83
5.3.1 Microcontroller	83
5.3.2 Motors	84
5.3.3 Electronics Assembly	86
5.4 Programming	88
5.4.1 Arduino Programming	88
5.4.2 Phone application	89
Chapter Six :Conclusion and future research directions	
6.1 Conclusions	94
6.2 Future Research Directions	95
References	96
الخلاصة	أ

List of Tables

Table. No.	Table Titles	Page No.
Table(3.1)	The detailed parameters and design specifications of the Suggested system.	42
Table(3.2)	Signs for the Trigonometric Ratio in Each Quadrant	50
Table(4.1)	The detailed parameters and design specifications of the proposed system.	59
Table(5.1)	Characteristics of Motors	86

List of Figures

Fig. No.	Figure Titles	Page No.
Figure 1.1	Standard robot car with cloud controlled system [4].	3
Figure 2.1	A typical practical real view of the moving electromechanical robotic arm [25].	11
Figure 2.2	The basic construction units of the moving arm robot [25].	12
Figure 2.3	Examples of kinematics movement actions [32].	14
Figure 2.4	Examples of kinematics motion (distance, and displacement) [32].	15
Figure 2.5	Demonstrating of the forward versus the inverse kinematics problem[34].	17
Figure 2.6	Linear robotic motion example [38].	19
Figure 2.7	An example of 6 axis articulated robots motion [38].	20
Figure 2.8	Demonstration of typical robot circular motion [38].	21
Figure 2.9	Designing a useful schema which represents the "mobile robot"[39].	22
Figure 2.10	Expanded "obstacles" volume regarding the "mobile object" volume [44].	26
Figure 2.11	Current and next position of the i^{th} "object" [47].	27
Figure2.12	Summing Angles Errors: The Path Smoothness [48].	28
Figure 2.13	For multi-objective route planning, choosing midpoints is essential [48].	29
Figure 2.14	To stop colliding with an obstacle, choose (x_i', y_i') along (x_i, y_i) [43] .	30
Figure 3.1	Block diagram of the suggested robotic system system.	40
Figure 3.2	A general scheme of the proposed design system of a four-wheeled mobile robot with arm.	41

List of Figures

Fig. No.	Figure Titles	Page No.
Figure 3.3	Structure of the designed mobile robot 4-wheel car.	43
Figure 3.4	The first longitudinal movement of the robotic arm.	46
Figure 3.5	The second rotational movement of the robotic arm along the base center.	47
Figure 3.6	Joints of robotic arm.	49
Figure 3.7	Calculated the angle for joint 1.	50
Figure 3.8	Merged the X and Y-axis as a new axis and defend as the r-axis.	51
Figure 3.9	Flowchart of inverse kinematic method.	52
Figure 3.10	Mat lab simulation details of the map plan area specified to control the movement of the robot.	53
Figure 3.11	The robotic arm on the three-axis.	55
Figure 4.1	Mat Lab simulation diagram of the modified proposed system of a four wheeled mobile robot with arm.	58
Figure 4.2	The implementation of the proposed four-wheeled mobile robot with arm system with operating area with [3,3] x, y dimensions.	60
Figure 4.3	The implementation of the proposed four-wheeled mobile robot with arm system with rotation in three dimensional viewing.	61
Figure 4.4	Robot arm motion trajectory error signals of the proposed system.	62
Figure 4.5	The motion control speeds U_f, U_l, and ω, of the moving robot car.	63
Figure 4.6	The robot arm joint angles for separate action control q_1, q_2, and q_3.	64
Figure 4.7	The angular positioning angles of the robot arm q_1, q_2, and q_3.	65

List of Figures

Fig. No.	Figure Titles	Page No.
Figure 4.8	The torque signals values of the robot motion.	66
Figure 4.9	The motion of the robot during the draw of sign (∞).	67
Figure 4.10	The error signals motion of the robot during the draw of sign (∞).	68
Figure 4.11	The motion control speeds U_f , U_l , and ω of the moving robot car and the response of the robot arm joint angles for separate action control during the draw of sign (∞).	69
Figure 4.12	The motion of the robot during the draw of sin wave.	70
Figure 4.13	The error signals motion of the robot during the draw of sin wave.	71
Figure 4.14	The motion control speeds U_f , U_l , and ω of the moving robot car and the response of the robot arm joint angles for separate action control during the draw of sin wave.	72
Figure 4.15	The torque signals values of the robot motion during the draw of sin wave.	73
Figure 4.16	The implementation of the proposed four-wheeled mobile robot with arm system with rotation in three dimensional viewing (4,-4,2).	74
Figure 4.17	The error signals motion of the robot during the motion at (4,-4,2).	75
Figure 4.18	The motion control speeds U_f , U_l , and ω , of the moving robot car at (4,-4,2).	76
Figure 4.19	The robot arm joint angles for separate action control q_1 , q_2 , and q_3 at (4,-4,2).	77
Figure 5.1	Omni-directional mobile robot with a moving arm.	80
Figure 5.2	Robot parts during assembly.	

List of Figures

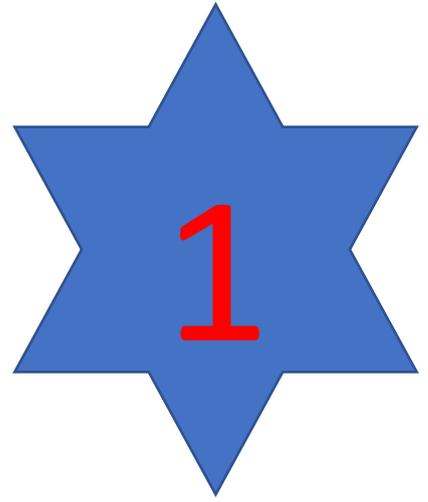
Fig. No.	Figure Titles	Page No.
Figure 5.3	Implemented project model.	
Figure 5.4	Arduino Mega 2560 Rev3	
Figure 5.5	The motors used in an Omni-directional mobile robot	
Figure 5.6	Assembling electronic components	
Figure 5.7	Electrical wiring diagram	
Figure 5.8	Flowchart of Control System	
Figure 5.9	user interface of the application	
Figure 5.10	MIT APP Inventor programming block (for forward and backward movement)	
Figure 5.11	MIT APP Inventor programming block (for side and rotate movement)	
Figure 5.12	MIT APP Inventor programming block (for Arm movement)	

List of Abbreviations

Abbreviation	Definition
Wi-Fi	Wireless Fi.
IOT	The Internet of Things.
FMT	Fast Marching Tree.
PDE	Partial Differential Equation.
PC	Personal Computer.
CATIA	Computer Aided Three-Dimensional Interactive Application.
3D	Three-dimensional.
AMDO	Articulated Mobility and Deformable Objects.
WMR	Wheel Mobile Robot.
DC	Direct Current.
DDM	Differential Drive Mode.
S	beginning location.
G	final location.
MR	Mobile Robot.
AI	Artificial Intelligence.
CI	Computational Insight.
ANN	Artificial Neural Network.
FL	Fuzzy Logic.
EC	Evolutionary Computation
SI	Swarm Intelligence.
Go	Game of.
AGI	Artificial General Intelligence.

List of Symbols

Symbol	Definition
\mathbf{r}	The Unit Vector
s	The arc-distance
$\Delta\mathbf{v}$	the velocity vector difference,
Δt	the time interval difference
v	the arc-speed
$d(.,.)$	the Euclidean Length.
$\theta_1, \theta_2, \text{ and } \theta_3$	the thetas among line segment



Chapter One

INTRODUCTION

Chapter one**INTRODUCTION****1.1 Introduction**

Moving arm robots cut down on the number of tasks and activities that an employee or worker needs to complete because, in addition to being effective and precise at work, they save time and effort. Each of these mechanical vehicles and robots is programmed to carry out a specific function. Depending on the kind of design, the programming mechanism, the nature of the task, and the scope of its implementation, it is also possible to develop them for additional tasks. The robot could unreservedly recognize the obstructions before it and pick the activity and get to know it intently through photography, and recognize it by video reconnaissance[1]. Due to the enormous industrial development implemented by factories and laboratories in a variety of medical, military, service, and civil fields, as well as a variety of device industries, the robotics industry has experienced tremendous growth over the past three decades. By looking at the most recent and important studies, references, and scientific literature on the topic of controlling mobile robots with moving arms, this study. Numerous studies have been presented, particularly those focusing on the mechanisms for controlling robots with moving arms and the classification of mobile phone control and self-programming methods[2]. By contrasting their various benefits and applications, the various types of mobile robotic driving technologies like WiFi, IOT, Arduino UNO, and microcontrollers have been examined. Despite actively avoiding obstacles through sensors, the robot may advance openly based on the drives that are connected to it via the phone. Consequently, a variety of sensors are positioned in the robot car's direction[3]. The robot may move toward the IOT platform when an obstacle is identified. This mission's objective is to develop a system of a cell-control robot car that uses more device landings than previously thought

to maintain a highly robust and adaptable robot[4]. The Arduino UNO is used as the center of the project, connecting a wide range of different components. A Wi-Fi module is used to remotely control the planned vehicle from a mobile phone.

A notification is sent to the phone when the robot detects an obstruction. Additionally, the robot uses an automated arm attached to the vehicle to perform pick-and-place tasks. A combination of temperature, gas, and fire sensors is used for blast detection[5]. and a bell indicates the location of a blast-prone area. In addition, a live reconnaissance is conducted to monitor each vehicle development[6]. The robot will save a lot of lives by spotting blast-prone areas and flagging something resembling a phone. It may be useful in identifying a variety of mishaps in which the environment poses a significant risk to individuals, such as working near radioactive or poisonous materials or an exploding propane tank. The robot could be used for reconnaissance because it can quickly cover areas that people can't reach and then automatically carry out its functions when there are obstacles. A typical robot car with a cloud-controlled system is depicted in figure(1.1).

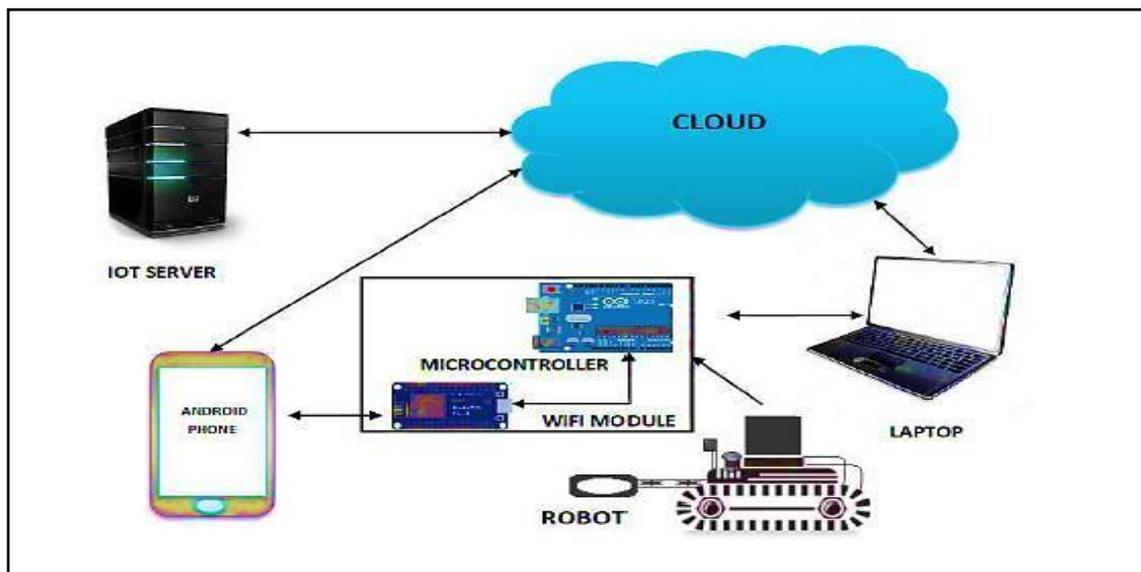


Fig 1.1: Standard robot car with cloud controlled system[4].

1.2 Literature Review

The important articles and papers which have discussed the work of robots that use motion sensors by using inverse kinetics have been collected and reviewed in the following paragraphs.

S. Gómez et al. 2015 [7] The paper describes the mechanical design of a 4-DOF robot manipulator with optimized algorithm for inverse kinematics. The robot has four degrees of freedom (DOF) and is designed to position points within a space in 3D. The first DOF has a rotational joint at the base, allowing movement in 360 degrees to increase the workspace, followed by three rotational joints. The mathematical equations that describe the inverse and direct kinematics are also obtained. The inverse kinematics algorithm used is a combination of numerical and geometric methods, which is used to make more efficient and accurate resolution of angles of each joint. The results show that the algorithm has appropriate performance with a relatively short processing time..

Panchanand jha. 2015 [8] This paper describes the methodology used for kinematic analysis of various configurations of robot manipulators. The conventional techniques like algebra, analytical method, iterative method, numerical method, geometric method, homogeneous matrix, DH algorithm, and quaternion algebra are presented. Additionally, different methodologies to solve the inverse kinematic problem are discussed in brief, including structural analysis of mechanisms, conventional methods for kinematics, intelligent or soft-computing approach, and optimization approach. The chapter also provides a table of adopted materials and methods for the research work, including conventional and intelligent approaches and various types of robots and structures..

Hong Zhao et al. 2017 [9] This paper discussing the kinematics of a four-degree-of-freedom manipulator. The paper explains the difference between forward and inverse kinematics and the importance of inverse kinematics in robot kinematics analysis, motion space, offline programming, and trajectory planning. The paper also discusses the two methods of solving inverse kinematics, closed-form solution, and numerical solution, and the three types of closed-form solutions, iterative, algebraic, and geometric. The paper then focuses on the algebraic and geometric methods and presents a simplified derivation process for solving inverse kinematics. The paper concludes with a simulation of the manipulator's kinematics and inverse kinematics using the Robotics Toolbox, which verifies the effectiveness of the modeling and simulation method and the correctness and feasibility of the inverse kinematics algorithm.

Květoslav Belda et al. 2017[10] This is a technical paper discussing the control objectives of mobile robotic systems with articulated robot arms. The paper outlines two main objectives for motion control: global and local. The global objective is the absolute motion of the mobile robotic system within its

workplace, while the local objective is the relative motion of the robot arm end-effector with respect to the mobile underframe of the system. The paper focuses on the local objective and proposes a model predictive control (MPC) design for local motion control of the robot arm. The paper also discusses suitable mathematical models for control design and provides a simulation of a 5 degrees of freedom robot arm. The proposed solution represents a suitable energy-optimal centralized control that takes into account the robot arm motion as one complex task.

Handry Khoswanto et al. 2019 [11] The article is a research paper that discusses the development of an application software for simulating the movement of robot manipulators. The software was designed using Unity engine and C# language and can simulate the movement of robot manipulators with 2 or 3 degrees of freedom. The aim of the research is to develop an application that can help students learn about robotics more effectively. The software was tested and compared to a similar application called RoboAnalyzer, and the results showed that the mathematical calculations were accurate. The article also provides information about robot manipulators and the components that make them up. The Unity engine is described as a useful tool for developing high-quality applications.

Tahseen F. Abaas et al. 2020 [12] The paper discusses the inverse kinematics analysis of a 5 DOF robotic arm using MATLAB and the Denavit-Hartenberg (D-H) parameters. The paper explains the importance of forward and inverse kinematics in determining the position and orientation of the end-effector of the robotic arm. The paper also presents a geometric approach to solve the inverse kinematics problem and divide the path of the robotic arm into successive lines to accomplish the required tasks. The paper includes a simulation of the robotic arm movement using MATLAB Graphical User Interface (GUI). The paper concludes that this method can be adopted for engineering applications.

Dhiya Uddin Rijalusalam et al. 2021[13] This is a research paper on the implementation of kinematic modeling and odometry in the trajectory planning of an omnidirectional robot. The paper explains the architectural control system consisting of track planning, motion control, and the overall system architecture. The results of testing the robot movement are also discussed. The paper concludes that the use of odometric sensors produces better movement results when compared to using only kinematic control. The paper also provides a graphical user interface of the application being made.

María José et al. 2023 [14] This is a research article that discusses the importance of studying robotics, analytics, artificial intelligence, nanotechnology, and IoT in the new industrial revolution. The article focuses on the mathematical modeling of an omnidirectional mobile manipulator robot and its implementation in simulation software. The study aims to develop control algorithms for the robot that are accurate, autonomous, and stable. The article discusses the direct and inverse kinematic problems that arise in the construction of robots and how a Jacobian matrix can help solve these problems. The article also discusses the importance of kinematic modeling, control, planning, and simulation in the design of robots. The article concludes by discussing the results of the simulation tests and the importance of developing mathematical models for mobile manipulator robots that meet the requirements of accuracy, autonomy, and stability

1.3 Problem Statement

This study identified the essential problems and effective obstacles by reviewing the latest research papers on the subject of mobile robots with a vehicles using a moving arm as follows:

1. Inconveniences of robot vehicles Network safety influences robot vehicles, the vehicle's PC may be hacked, and the correspondence framework between the vehicles could likewise be the primary deterrent, controlling their activity.
2. The need for precise guidance by increasing the programming effort and preparing the appropriate software to describe the movement of the robot accurately and efficiently and to control the directives accurately.
3. The high cost of design, in addition to the cost of control and continuous maintenance, to ensure the efficient performance of the mobile robots' work correctly and with the lowest error rate.

1.4 Aims of the Thesis

According to the previous studies available in literature reviewing, the objectives of this thesis might be summarize as follows:

- 1) Reviewing of the most updated available multipurpose robotic studies based on microcontroller and wireless technology.
- 2) Propose a robust strategies that early predict and trying to minimize the consequences of multipurpose robotic based on microcontroller and wireless technology.
- 3) Proposing an efficient controlling modeling approach for the multipurpose robotic to produce minimal error effect and mistakes elimination.

1.5 Thesis Organization

Chapter one: This chapter provided an introduction, problem statements, and objectives of to the thesis.

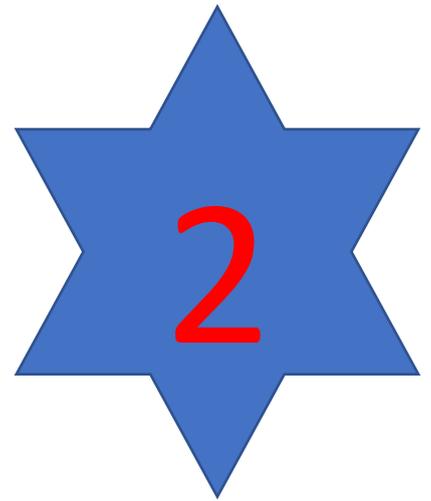
Chapter two: This chapter introduces the basic concepts of the theoretical backgrounds of multi-purpose robotics and will be presented with mathematical equations and architectural modeling of the most typically applied controllers.

Chapter three: This chapter provides a simulation for the designed system using the MATLAB program.

Chapter four: This chapter provides simulation results for the designed system using the MATLAB program.

Chapter five: This chapter presents the details of the proposed improved structure for multipurpose robotic based on microcontroller and wireless technology.

Chapter six: This chapter contains the conclusions and the future research direction.



Chapter Two

BACKGROUND AND THEORETICAL CONCEPTS

CHAPTER TWO

BACKGROUND AND THEORETICAL CONCEPTS

2.1 Introduction

In this chapter, the subject of a mobile robotic car with a movable arm will be discussed in detail. Theoretical aspects of a moving robotic car with a moving arm will be presented with a review of mathematical equations and theoretical principles in addition to illustrations and diagonal diagrams. In this chapter, aspects of control and transmission of control signals through IOT units and Wi-Fi controllers for the moving arm and for the moving vehicle are also explained in detail.

The principles of kinematics versus reverse kinematics principles for precise control of each part of the moving arm will also be studied, as well as on the units of connecting wheels and the transmission of movement to and from the moving parts in a balanced manner through sending and receiving control signals, which will also be discussed and demonstrated in a full discussion.

2.2 Moving Robot Arm Structure

A moving robot arm, often referred to as a robotic manipulator or simply a robot arm, is a mechanical structure designed to mimic the movement capabilities of a human arm. These robot arms are commonly used in various applications such as manufacturing, assembly, material handling, medical surgery, and more. The structure of a moving robot arm typically consists of several key components[15]:

- 1- **Base:** The base is the stationary part of the robot arm that is usually mounted to a fixed surface, such as a floor or a robotic work cell. It provides stability and serves as the anchor point for the rest of the arm. 2-

2- **Joints:** Robot arms typically consist of multiple joints, which are the movable connections between different segments of the arm. There are various types of joints used in robot arms, including:[16].

- **Revolute Joints** (also known as rotary joints): These joints allow rotation around an axis, much like the human elbow or wrist joint[17].
- **Prismatic Joints** (also known as linear joints): These joints allow linear motion along a single axis, similar to the motion of a sliding piston.
- **Spherical Joints:** These joints provide three degrees of rotational freedom and are often used at the wrist of a robot arm to provide orientation control.
- **Cylindrical Joints:** These joints combine both revolute and prismatic motions, offering a combination of rotation and linear movement.
- **Planar Joints:** These joints restrict motion to a two-dimensional plane, often used in simpler robotic arms.

3- **Links:** Links are the rigid segments that connect the joints. The number and length of links can vary, and they determine the reach and flexibility of the robot arm. Each link is usually connected to two joints.

4- **End-Effector:** The end-effector is the tool or device attached to the last link of the robot arm. It is responsible for performing the specific task the robot is designed for, such as gripping, welding, painting, or other operations. The end-effector can be interchangeable to accommodate different tasks.

5- **Actuators:** Actuators are the components responsible for driving the motion of the robot arm's joints. They can be electric motors, hydraulic

cylinders, pneumatic cylinders, or a combination of these, depending on the design and application requirements.

- 6- **Sensors:** Robot arms are often equipped with sensors to provide feedback on their position, orientation, and the objects they interact with. Common sensors include encoders, accelerometers, force/torque sensors, and vision systems.
- 7- **Control System:** The control system includes the hardware and software responsible for controlling the robot arm's movements and ensuring precise and accurate operation. It interprets input commands, processes sensor data, and calculates the necessary joint angles or positions to achieve the desired end-effector motion.
- 8- **Programming Interface:** Users can program robot arms through various interfaces, including teach pendants, computer software, or programming languages designed for robotics, such as Robot Operating System (ROS) or proprietary programming languages.

The specific design and configuration of a moving robot arm depend on the intended application, payload capacity, reach requirements, and other factors. Robot arms can range from small, lightweight units used in precision tasks as shown in figure (2.1) to large, heavy-duty industrial robots capable of lifting and manipulating heavy objects.



Fig 2.1: A typical practical real view of the moving electromechanical robotic arm[18].

2.3 Kinematics and Inverse Kinematics Definition

2.3.1 Kinematics Definition

Forward kinematics is a mathematical approach used to determine the position and orientation (pose) of the end-effector (tool or tip) of a robotic arm in three-dimensional space based on the known joint angles or joint positions of the arm's individual segments (links). The Denavit-Hartenberg (DH) parameters are commonly used in the context of forward kinematics to model the relationships between these joint angles, link lengths, and other geometric properties of the arm. Below, I'll explain how forward kinematics and DH parameters work together.

Forward kinematics is primarily concerned with calculating the transformation matrix that describes the pose of the end-effector relative to a fixed reference

frame, typically the base of the robot. This transformation matrix represents both the position and orientation of the end-effector in 3D space.

The transformation matrix is calculated by successively applying transformation matrices for each joint-link pair along the robotic arm. These matrices are typically based on the DH parameters[19].

The forward kinematics equations for each joint can be expressed using these transformation matrices. Here is a simplified representation for a serial manipulator (a robot with joints in series):

T_0 be the transformation matrix representing the base frame (usually the robot's reference frame).

T_1, T_2, \dots, T_n be the transformation matrices representing each joint's transformation with respect to the previous joint.

$T_{end-effector}$ be the transformation matrix representing the end-effector frame. The forward kinematics equations can be expressed as[20].

$$T_{end-effector} = T_0 * T_1 * T_2 * \dots * T_n \quad (2.1)$$

Each T_i matrix represents the transformation due to the i th joint's movement, and by multiplying them together, you obtain the transformation of the end-effector relative to the base frame[21].

The DH parameters are a standardized set of parameters used to describe the geometric properties and relationships between joints and links in a robotic arm. These parameters include link lengths (a), link twists (α), link offsets (d), and joint angles (θ) as shown in figure(2.3).

The DH parameters help establish a coordinate frame at each joint, defining the position and orientation of the coordinate axes. These coordinate frames are crucial for calculating the transformation matrices that relate adjacent joints.

Each joint-link pair is assigned a set of DH parameters, and these parameters are used to construct the transformation matrix that represents the relative pose between two consecutive joints[21].

In summary, forward kinematics uses the DH parameters to calculate the transformation matrices that describe the position and orientation of the end-effector based on the joint angles or positions. The DH parameters provide a systematic and standardized way to model the kinematics of robotic arms, making it easier to perform accurate forward kinematics calculations for various robot configurations kinematics actions are illustrated in figure (2.4).

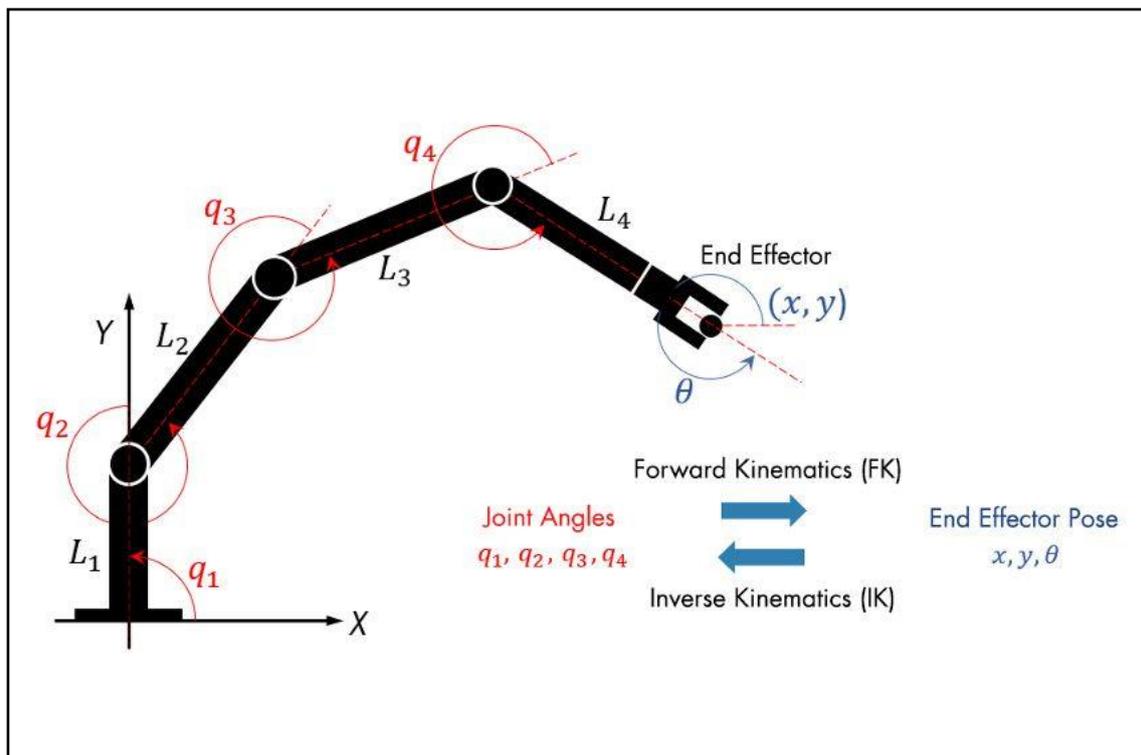


Fig 2.3: Examples of kinematics movement actions[22].

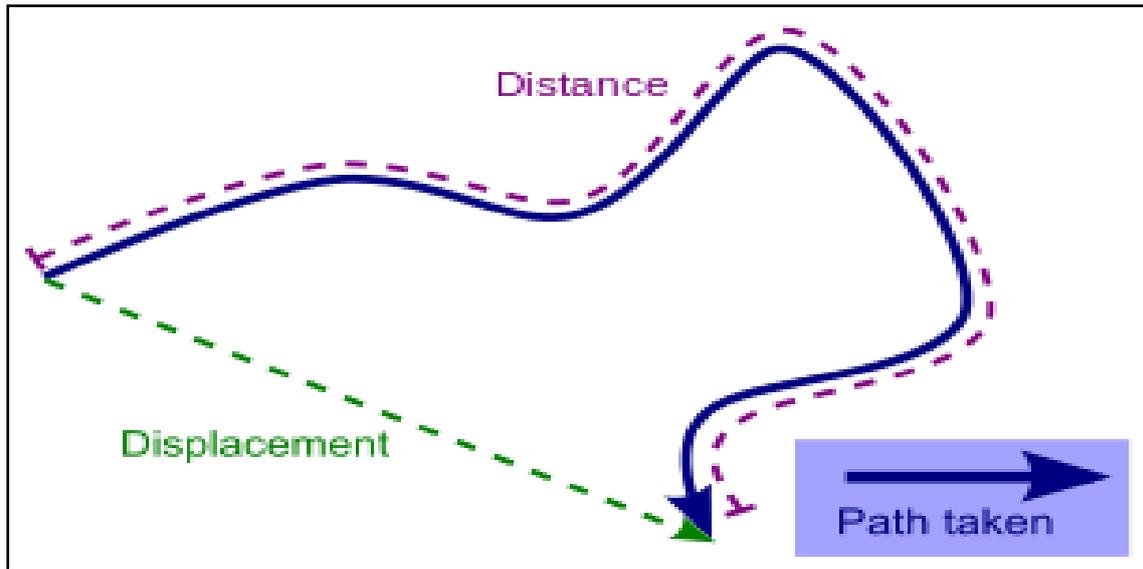


Fig 2.4: Examples of kinematics motion [23]

2.3.2 Inverse Kinematics Definition

is a fundamental concept in robotics and mathematics that deals with finding the joint angles or positions of a robotic system to achieve a desired end-effector position and orientation (pose) in three-dimensional space. It is essentially the reverse problem of Forward Kinematics, where you know the joint angles and want to determine the end-effector's position and orientation[24].

Here are the key points and components of Inverse Kinematics:

- **End-Effector Goal:** In Inverse Kinematics, you start with a specific desired position and orientation (pose) for the robot's end-effector. This could represent the desired position of a robot's hand, tool, or any other part of the robot.

- **Unknown Joint Values:** The goal is to find the joint angles or joint positions that will allow the robot to achieve the desired end-effector pose. These joint values are unknown initially.
- **Robot Geometry:** To solve Inverse Kinematics problems, you need a thorough understanding of the robot's geometry, including the lengths of its links, the locations of its joints, and any constraints on joint motion.
- **Inverse Kinematics Solver:** An Inverse Kinematics solver is a mathematical algorithm or process used to calculate the joint values that satisfy the desired end-effector pose. There are various methods to solve Inverse Kinematics problems, ranging from closed-form solutions for simple cases to numerical optimization techniques for more complex and high-dimensional robots.
- **Constraints:** Real-world robots often have physical limitations and constraints on their joint motion, such as joint limits, joint velocity limits, or obstacles to avoid. These constraints must be taken into account when solving Inverse Kinematics problems.
- **Applications:** Inverse Kinematics is used in robotics for tasks such as robot control, motion planning, trajectory generation, and programming. It is essential for applications where precise positioning and control of the robot's end-effector are required, such as pick-and-place tasks, manipulation, and industrial automation[25].

Solving Inverse Kinematics problems can be challenging, especially for robots with complex kinematic structures and multiple degrees of freedom. Researchers and engineers use a variety of mathematical techniques and computational methods to find solutions, making it possible for robots to

perform tasks with accuracy and dexterity in real-world scenarios. Figure (2.5) displays a diagram demonstrating the forward versus the inverse kinematics .

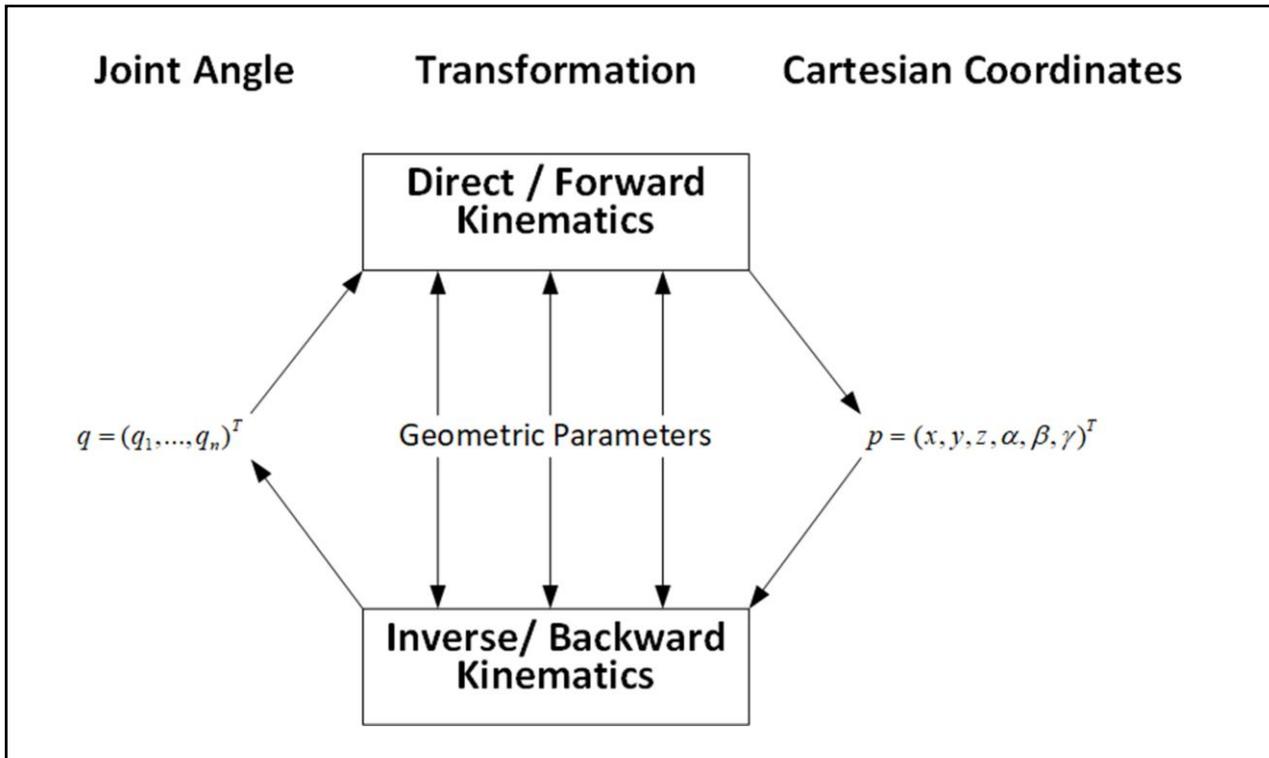


Fig 2.5: Demonstrating of the forward versus the inverse kinematics problem[26].

2.4 Robot Object Movement Planning

Robot object movement planning, often referred to as motion planning or path planning, is a critical component of robotics that involves determining how a robot should move to reach its desired goal while avoiding obstacles and adhering to various constraints. This process is essential for enabling robots to perform tasks autonomously and safely in dynamic environments. Here are the key aspects of robot object movement planning:[27].

- **Goal Definition:** The first step in movement planning is to define the robot's goal. This can be the desired position and orientation (pose) for

the robot's end-effector (e.g., a robotic arm's gripper) or any other specific objective, such as reaching a target location.

- **Environment Modeling:** The robot needs a representation of its environment, including the locations and shapes of obstacles, the robot's own kinematic and dynamic constraints, and any other relevant information. This representation can be in the form of a map, point cloud data, or other sensor inputs.
- **Path Planning:** Path planning is the process of finding a collision-free path from the robot's current position to its goal. This can involve algorithms that search through the environment's representation to determine the optimal or feasible path, considering factors like distance, clearance from obstacles, and kinematic constraints.
- **Trajectory Generation:** Once a path is planned, the next step is to generate a smooth trajectory that the robot can follow to move from its current position to the goal. Trajectory generation considers the robot's dynamics, velocity limits, and acceleration limits to ensure a feasible and safe motion.
- **Safety:** Ensuring the safety of both the robot and its surroundings is paramount. Safety measures, including emergency stop mechanisms and collision detection systems, are integrated into the planning and control systems.
- **Optimization:** In some cases, optimization techniques are used to find the most efficient or optimal path or trajectory while satisfying various constraints. These optimizations can minimize energy consumption, time, or other relevant criteria.

Effective planning algorithms and techniques are essential for enabling robots to navigate and manipulate their environments effectively and autonomously.[28].

2.4.1 Kinds of Robotic Movements

There are three essential kinds of robotic movements which a robot system utilizes to step around the physical world: linear, articulated, and circular motions. While the goal of all such movements is the same - to obtain along point A to point B - the direction which the robot catches along the path is the basic limiting factor for every kind of movement [29].

A) Linear Motion

Linear motion refers to the movement of a robot or a robotic component along a straight-line path from one point to another. This type of movement is fundamental for tasks such as precise positioning, pick-and-place operations, and material handling. Linear motion is crucial in applications where the robot needs to move in a straight line, either in one, two, or three dimensions. An example of the linear robotic motion is displayed in figure (2.6).

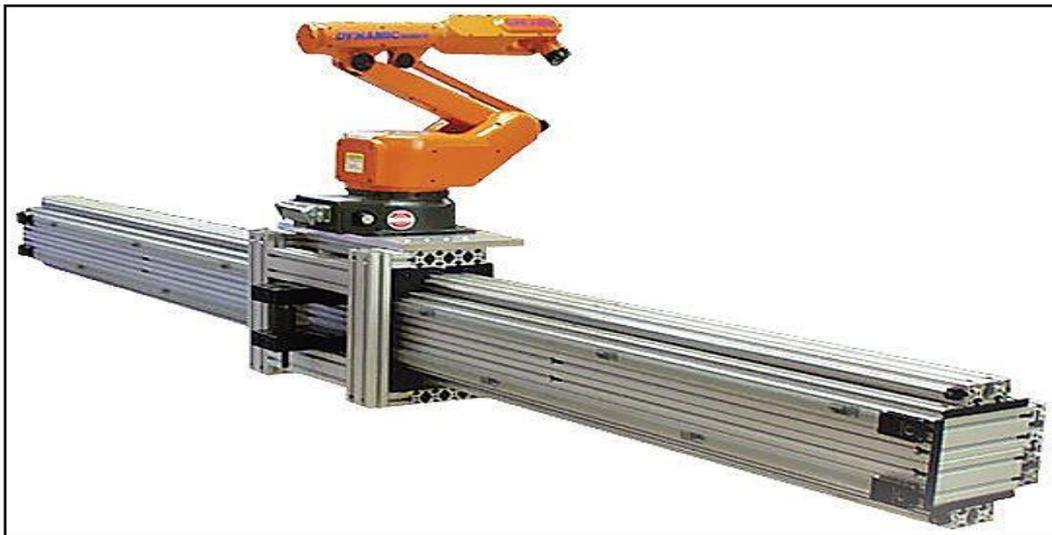


Fig 2.6: Linear robotic motion example[29].

B) Articulation Movement

The Articulated Mobility and Deformable Objects (AMDO) is a demanding domain of investigation which concentrates on the automatic planning of complicated objects, just like the human body, which present long variables in particulars of spatio-temporal paths. The arm is fixed to a base with a rotation joint. The arm itself might feature anywhere along two revolving joints up to ten rotating joints operating as axes, against every further joint or pivot permitting a greater movement degree. Only four or six axis in maximum account have been utilized by the articulated robots[30]. Figure (2.7) presents an example of 6 axis articulated robots motion.

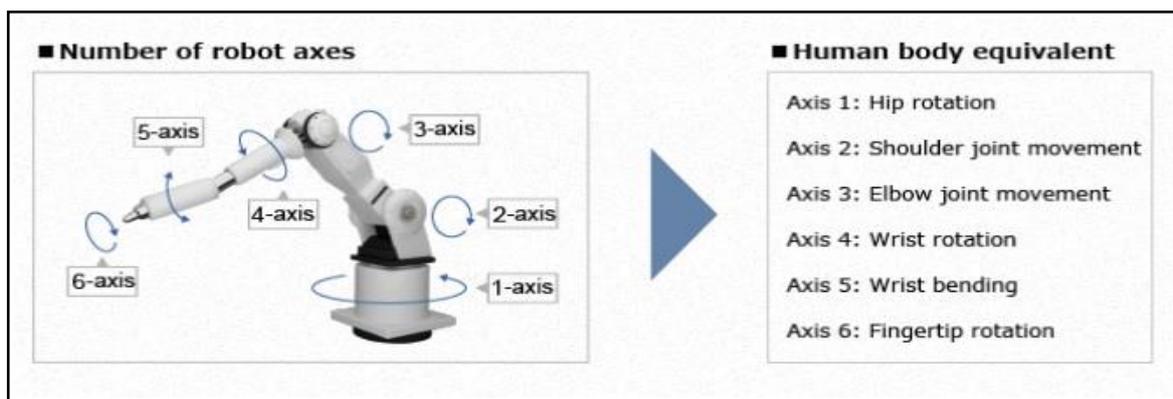


Fig 2.7: An example of 6 axis articulated robots motion[30].

C) Circular Motion

The rotational movement of the robot is compatible with the rest of the other movements, the linear movement and the axial movement to be complementary to them in coordinating the entire movement of the robot. The rotational movement runs parallel to the area of the movement plane and at a full 360-degree rotation angle.

The rotational motion must also act according to the same kinematic relations in accordance with the rest of the equations of other linear and axial

movements so that the final movement is integrated and consistent with each other, as well as with the general scheme of the robot's movement path . Typical robot circular motion is illustrated in figure (2.8).

In the displayed figure (2.8), it is obvious the various types of circular robotic motions employed by this example of typical robotic arm model. Actually, for each joint in the arm, the circular movement might be applied with different angle degree of freedom. In fact, The moving angle degree of freedom will depend mainly upon the task type of the specified joint action.



Fig 2.8: Demonstration of typical robot circular motion[31].

2.4.2 Robot Car

Robot cars, like other types of vehicles, can be classified based on the number of wheels they have. The number of wheels often corresponds to their design and intended use. Here are some classifications of robot cars based on the number of wheels:

A) Two-Wheel Robot Cars

The two "Wheel Mobile Robot" ("WMR") type is controlled by two DC (direct current) motors and every wheel is connected with one encoder for measuring the rotating velocity and direction which are based on differential drive mode (DDM). The two WMR type have left and right wheels which are independent of every other in motion. In the two WMR type, there is often an additional ineffective wheel at the front of the robot is attached to make the robot's motion smoother. Several autonomous navigation research projects have been extended to various forms of "mobile robots." [32]. The wheeled "mobile robot" (WMR) of The schematic scheme is presented in figure (2.9).

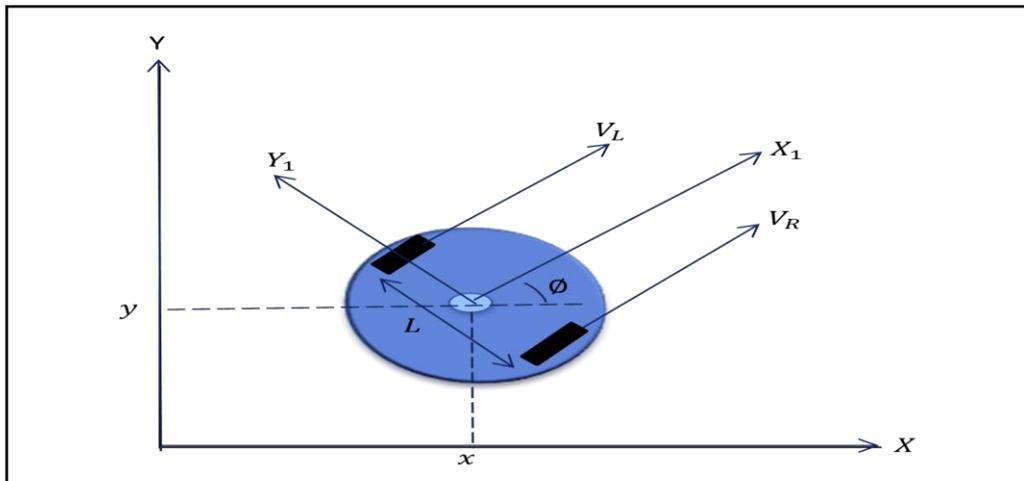


Fig 2.9: Designing a useful schema which represents the "mobile robot" [33].

The kinematic model of a non-homonymic "mobile robot" is provided as succeeds [34].

$$\frac{dx}{dt} = \frac{V_R + V_L}{2} \cos \phi, \quad (2.2)$$

$$\frac{dy}{dt} = \frac{V_R + V_L}{2} \sin \phi, \quad (2.3)$$

$$\frac{d\phi}{dt} = \frac{V_R - V_L}{L}, \quad (2.4)$$

Based on the computational angular velocity ω , the "robot" right and left wheel speeds could be calculated as succeeds[35].

$$V_L = V + \frac{1}{2} L\omega , \quad (2.5)$$

$$V_R = V - \frac{1}{2} L\omega , \quad (2.6)$$

Where ω angular velocity , V_L and V_R right and left wheel speeds and L The distance between wheels.

The assuming a moving "object" located at a beginning location (S) which is needed to catch a final location (G) in a certain counter. Many stable restrictions are suggested for occurring in the versatile "object" work area. The goal of a " path planning problem" is to search a perfect or close-maximal way (shortest, safest also smoothest) beyond smashing beside an issue, part of the postulates made in here enquire must be formed obvious[36]. Early: The restrictions are expressed as circular patterns. "Besides: The mobile object is a natural object; hence, to consider the true extent of the moving object, the obstructions are extended by the mobile robot (MR) diameter, hence the mobile robot might be represented as a mark. Thirdly there are no kinematic conditions which disturb the mobile robot motion[37] The alone active supply is the obstacles movement.

B) Four-Wheel Robot Cars

The journey of the four-wheel robot car began with a shared dream – a vision of a world where vehicles could navigate our bustling urban landscapes safely, efficiently, and autonomously. As the problems of traffic congestion, accidents, and pollution intensified, a need arose for a smarter and more sustainable approach to transportation.

The fundamental concept behind these four-wheeled marvels is autonomy. Equipped with an array of sensors, from cameras and LiDAR to radar and ultrasonic devices, these cars continually gather data about their surroundings. Coupled with advanced artificial intelligence and machine learning algorithms, they can analyze this data to make real-time decisions, navigate complex traffic situations, and respond to unexpected obstacles.

The development of four-wheel robot cars began with the pioneering work of companies like Google's Waymo, which introduced the world to the concept of fully autonomous vehicles. These cars were not just self-driving; they could do so with remarkable precision, safely transporting passengers from point A to point B.

As these four-wheel robot cars evolved, they became more than just autonomous vehicles; they represented a shift in the way we think about transportation. No longer was driving solely a manual task; it was something that could be delegated to a machine, allowing humans to reclaim valuable time for work, relaxation, or leisure during their commutes.

Four-wheel robot cars also promised safer roads. With the potential to reduce human error, a leading cause of accidents, these vehicles had the potential to save countless lives. Their strict adherence to traffic rules, impeccable reaction times, and lack of distractions made them safer than the average human driver.

The development and deployment of four-wheel robot cars have not just been about technology but a profound shift in our relationship with mobility. These vehicles are redefining urban landscapes, reducing traffic congestion, lowering emissions, and reshaping city planning. They are making the dream of a more efficient, sustainable, and accessible transportation future a reality.

C) Omnidirectional Robot Cars

Compact vehicle with a futuristic design as shown in figure. At its base, it's equipped with unique wheels that set it apart from traditional cars. These wheels are usually omni-wheels, Mecanum wheels, or a similar specialized wheel design that allows for omnidirectional movement.

- **Omni-Wheels:** Omni-wheels are wheels with multiple rollers mounted at angles around the wheel's circumference. These rollers allow the wheel to move not only forward and backward but also laterally (sideways) and rotate freely.
- **Mecanum Wheels:** Mecanum wheels are equipped with rollers set at specific angles to the wheel's axis, allowing the vehicle to move in any direction and even diagonally by spinning the wheels in different directions.

On top of this base, there's often a payload or cargo area, depending on the vehicle's purpose. It might be equipped with sensors such as cameras, lidar, and ultrasonic devices to perceive its environment and obstacles. Advanced control systems and artificial intelligence algorithms process this sensor data and calculate the precise wheel movements needed for the vehicle to move in the desired direction.

These vehicles can move forward, backward, sideways, and rotate effortlessly, offering exceptional agility and precision in confined spaces. They are often used in settings like factories, warehouses, hospitals, and other environments where precise navigation and maneuverability are essential.

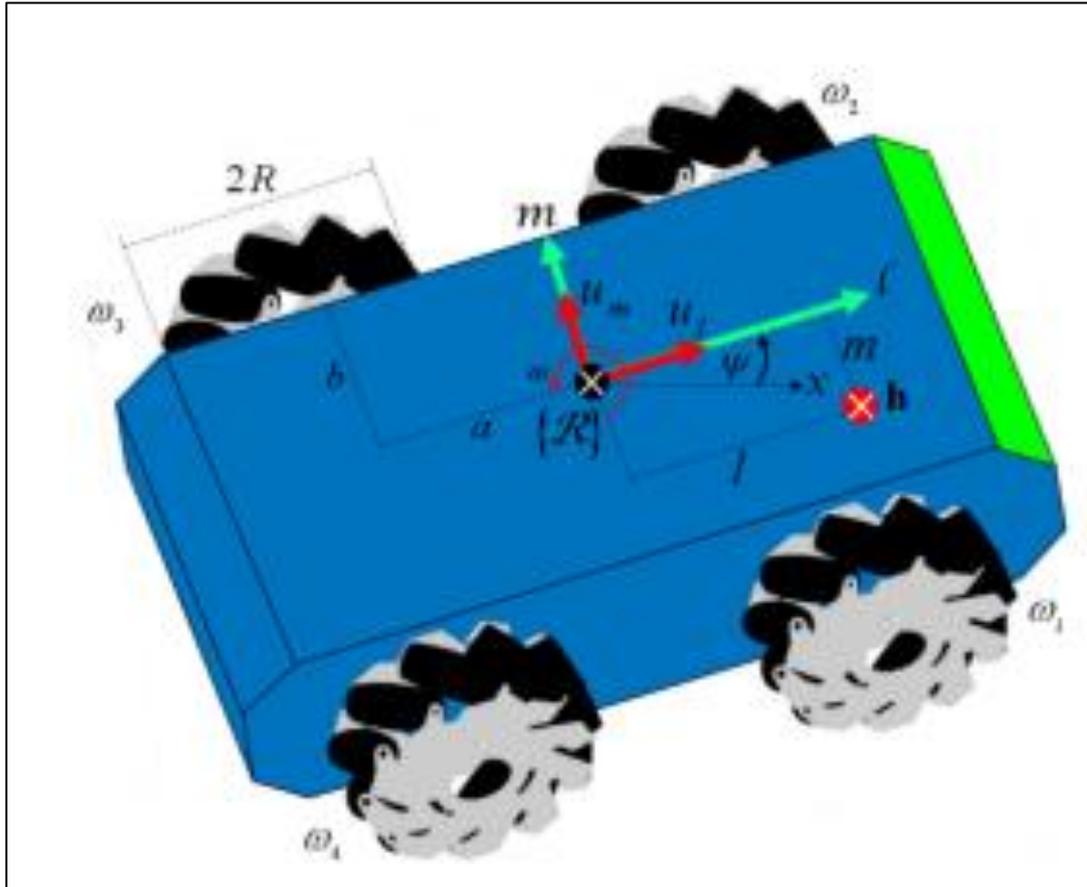


Fig 2.10: Omnidirectional Robot Cars.

2.4.3 Performance Criteria

Minimal Length: Of the path planning field, “Shortest Distance” defines reducing the way distance within the beginning as well destination points. For either repetition, the mark $p_j(i)$ is chosen as the favorite one when it has more concise length to the final mark $p(N)$ in which[38].

$$f_1(x, y) = d(p_j(i), p(N)) \quad (2.7)$$

is smallest, whereas $d(.,.)$ is the "Euclidean Length". The "Shortest Path Length" ("SPL") is the addition of every paths among "mid-points" ($p_j(2) \cdots p_j(N-1)$) created by the "path planning algorithm" among the beginning mark (S) $p(1)$ also the final Goal Point (G) $p(N)$, obtained by[39].

$$SPL = \sum_{i=1}^{N-1} d(p_j(i), p_j(i+1)) = \sum_{i=1}^{N-1} d_i \quad (2.8)$$

So that j is the symbol of the finest evaluation created by the "HGWO-PSO" situated "path planning algorithm"[40].

$$d_i = \sqrt{(x_{pj}(i+1) - x_{pj}(i))^2 + (y_{pj}(i+1) - y_{pj}(i))^2} \quad (2.9)$$

Figure (2.11) illustrate the shortest distance idea between the current and the next position.

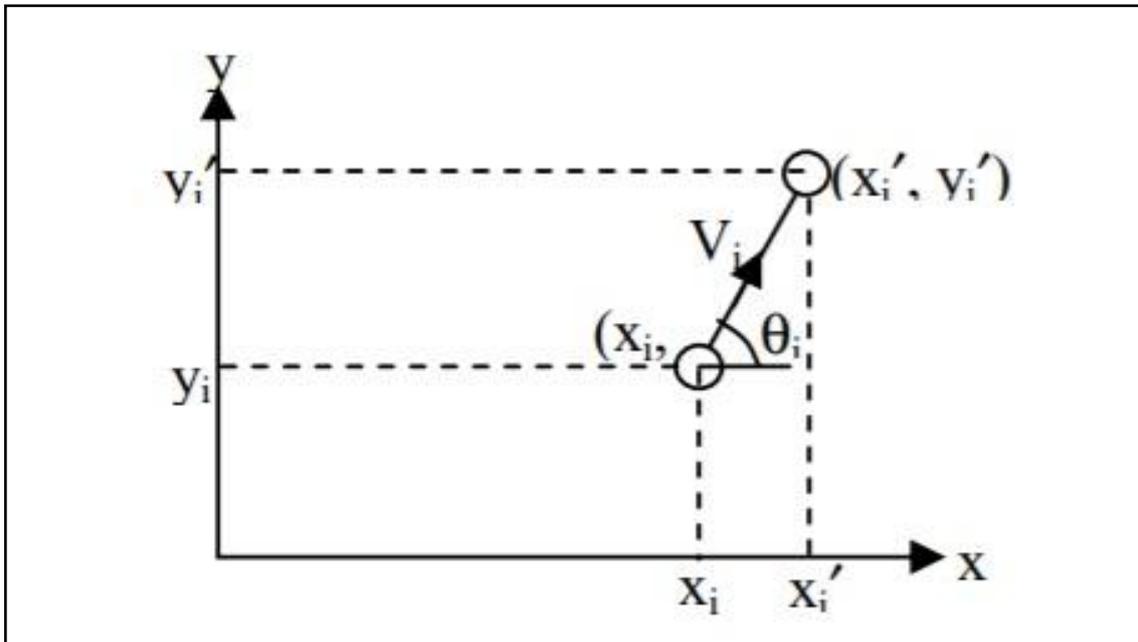


Fig 2.11: Current and next position of the i^{th} "object"[40].

Path Smoothness: As shown in figure (2.12), is involved reducing the inequality of the inclines among the solid traces ("objective-present marks with proposed-present marks"), and obtained by[41].

$$f_2(x, y) = \sum_{i=1}^{N-1} |\theta(p(i), p(i+1)) - \theta(p(i), p(N))| \quad (2.10)$$

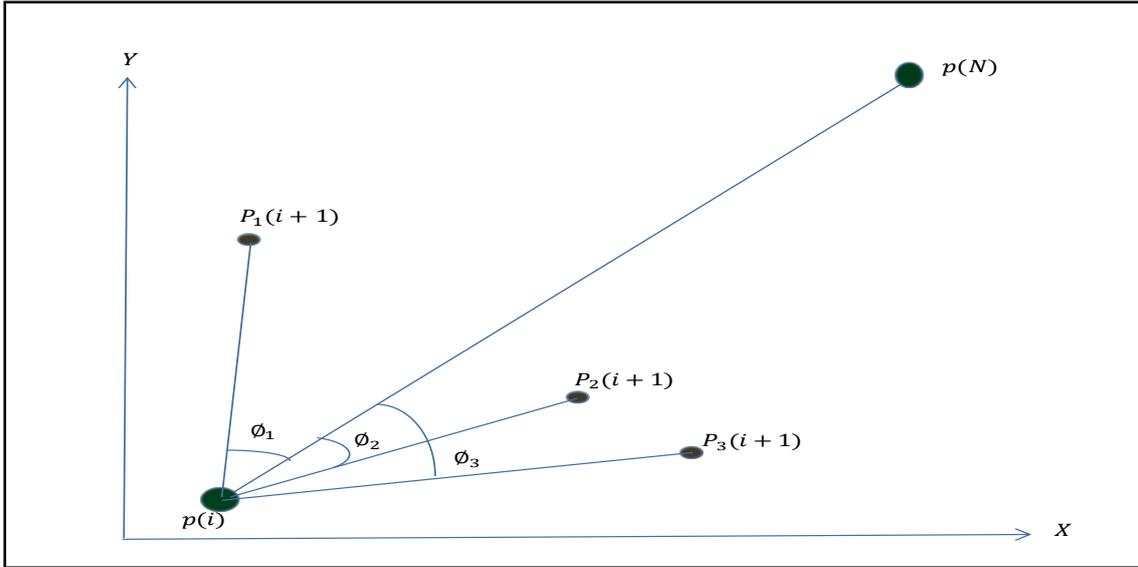


Fig 2.12: Summing Angles Errors: The Path Smoothness[41].

Where,

$$\theta(p(i), p(N)) = \arctan \frac{yp(N) - yp(i)}{xp(N) - xp(i)} \quad (2.11)$$

Where the θ_1 , θ_2 , and θ_3 are the thetas among line segment $(p(i), p(N))$ and $(p(i), p_1(i+1))$, $(p(i), p_2(i+1))$, $(p(i), p_3(i+1))$ respectively. The θ_2 has a minimum angle among the competing points (p_1, p_2, p_3) .

The weighted sum of the two goals above is the final "multi-objective optimization":

$$f(x, y) = w_1 f_1(x, y) + w_2 f_2(x, y) \quad (2.12)$$

Where w_1 and w_2 the two goals have varying degrees of significance. Their ideals shall meet the following requirements:

$$w_1 + w_2 = 1 \quad (2.13)$$

The total shape formula is computed as follows:

$$fitness = \frac{1}{f(x, y) + \gamma} \quad (2.14)$$

Where γ is a very small value ($\gamma=0.001$) used to avert zero division in some situations. The mechanism of choosing the finest clarification along challenging suitable privileges in every repetition trusts in upon the adjust among the double actions purposes stated in (2.12) as well (2.14) for every possible clarifications. In figure (2.13), the finest mark along six challenging marks is mark p_3 for the i repetition, also mark p_5 in the $(i+1)$ emphasis, otherwise in the $(i+2)$ repetition, marks p_2 as well p_3 provide less separation yet longer variation inclines, in variance, to mark p_1 that gives a symmetry among the two standards; therefore, mark p_1 is chosen. Such action is remained up to meeting G.

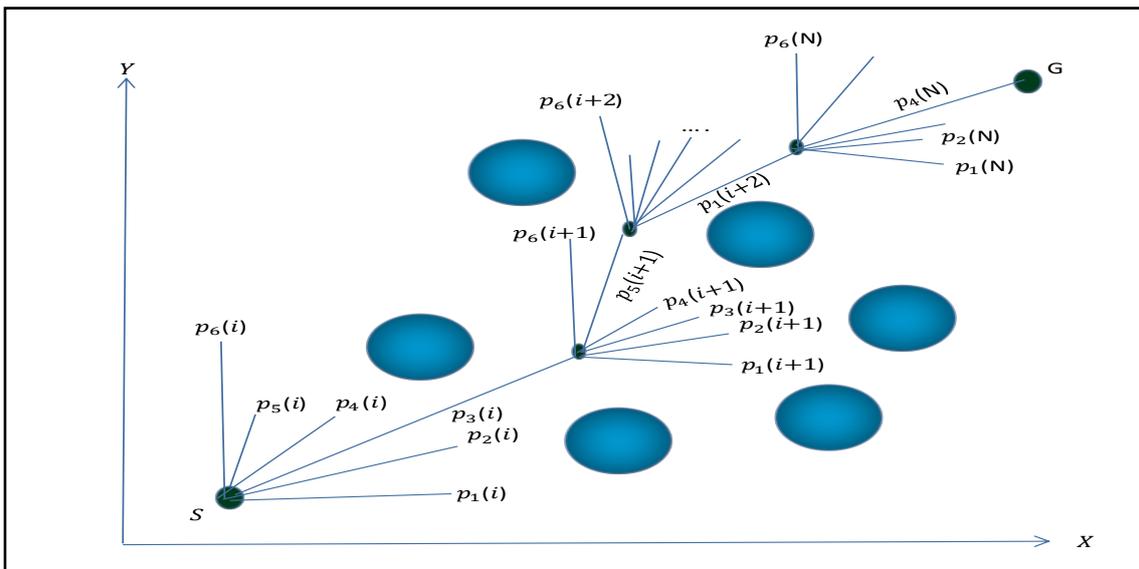


Fig 2.13: For multi-objective route planning, choosing midpoints is essential[42].

Also, figure (2.14) illustrates the Selection of (x'_i, y'_i) along (x_i, y_i) avoiding a collision with another obstacle.

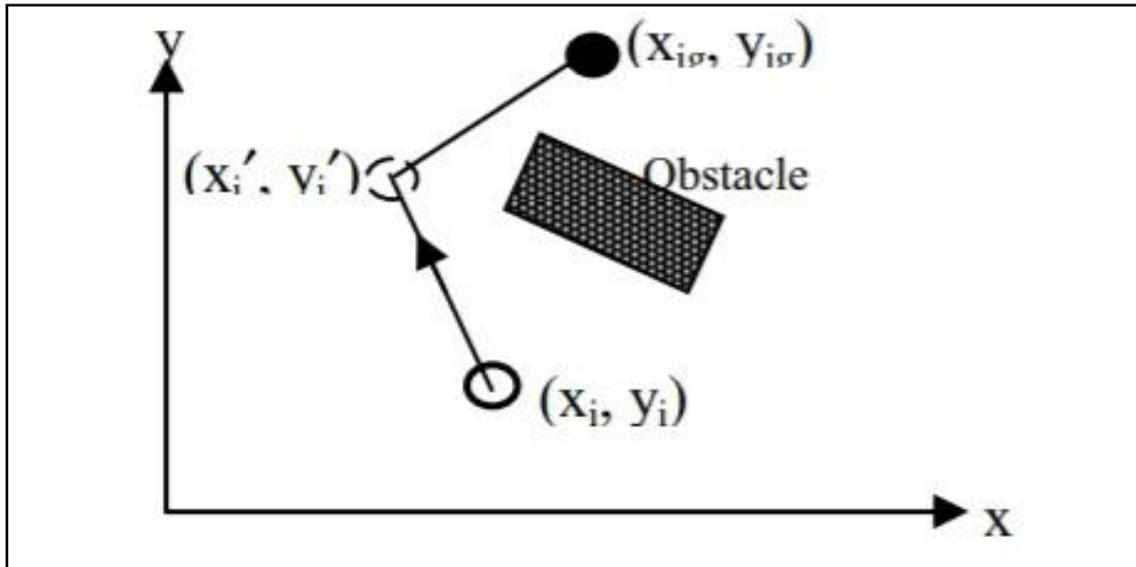


Fig 2.14: To stop colliding with an obstacle, choose (x_i', y_i') along (x_i, y_i) [42].

2.5 Microcontroller with AI Techniques

Artificial Intelligence (AI) refers to the simulation of human intelligence processes by machines, particularly computer systems. It involves the development of algorithms, models, and systems that enable machines to perform tasks that typically require human intelligence. AI encompasses a broad range of techniques and technologies, including machine learning, natural language processing, computer vision, robotics, and more. Here are some key aspects of artificial intelligence[43].

- **Machine Learning:** Machine learning is a subset of AI that focuses on the development of algorithms and models that allow computers to learn and make predictions or decisions based on data. Supervised learning, unsupervised learning, and reinforcement learning are common types of machine learning.

- **Deep Learning:** Deep learning is a subset of machine learning that uses artificial neural networks with many layers (deep neural networks) to process and learn from data. It has been particularly successful in tasks like image and speech recognition.[44].
- **Computer Vision:** Computer vision involves teaching machines to interpret and understand visual information from the world, including images and videos. It is used in facial recognition, object detection, autonomous vehicles, and medical image analysis.
- **Robotics:** AI plays a critical role in robotics, enabling robots to perceive their environment, make decisions, and perform tasks autonomously. Robotic applications range from manufacturing and healthcare to exploration and entertainment[45].
- **AI in Healthcare:** AI is used in healthcare for tasks like disease diagnosis, drug discovery, personalized medicine, and patient monitoring. It can analyze medical images, electronic health records, and genomic data.
- **AI in Finance:** In the financial industry, AI is used for fraud detection, algorithmic trading, credit risk assessment, and customer service chatbots.
- **AI in Autonomous Vehicles:** Autonomous vehicles (self-driving cars and drones) rely on AI technologies such as computer vision, sensor fusion, and machine learning to navigate and make decisions.
- **Ethical and Social Considerations:** As AI technology advances, there are ethical and social considerations, including issues related to bias in AI algorithms, privacy concerns, and the impact of AI on employment and society[46].

AI is a rapidly evolving field with the potential to revolutionize various industries and domains. It continues to advance and find applications in new areas, making it a topic of significant interest and research. Natural Language Processing (NLP): NLP is a branch of AI that focuses on the interaction between computers and human language. It enables machines to understand, interpret, and generate human language. NLP is used in applications like chatbots, language translation, and sentiment analysis[47].

2.5.1 AI Applications and Future Technology

Artificial intelligence is pervasive and isn't simply restricted to software engineering be that as it may, has advanced to incorporate different regions like wellbeing, security, instruction, music, workmanship, also, business implementation[48].

Various AI implementations are completely inserted in the bases of every plants. Man-made brilliance is expected, in a couple of years, to contact essentially all the enterprises furthermore, there are a lot of approaches AI is also might alter certain ventures. Computer based intelligence is at present being utilized for a wide scope of exercises containing clinical finding, electronic exchanging stages, robot control, and distant detecting. It has been utilized to create and advance various fields and ventures, containing money, medical care, training, transportation, and mechanical technology[49].

Man-made intelligence analysts have made numerous devices to explain the most troublesome issues in software engineering and different fields. The current AI execution ranges between sub-human, ideal, what's more, super-human presentation. A wide scope of assignments might be understood by AI implementations containing facial acknowledgment, discourse acknowledgment , object acknowledgment, pictures characterization and outperforming human-level knowledge in The Game of Go, Chess. Another focal point of AI

advancements lies between the regions of medical services and protection with the headway of unified learning and security seeing AI[50]

Artificial intelligence is utilized as a clinical choice emotionally supportive networks for clinical findings, PC-assisted translation of clinical pictures, and partner robots. It can even create music usable in a clinical setting PC-created music for stress and relief along with discomfort. Besides, activities like Google Magenta , directed by the Google Brain group, need to search out if AI can be fit for creating convincing craftsmanship and music. One further AI research dynamic domain is the implementation of AI to compose another AI. Such an approach will merge Google's AutoML task to advance other neural organization geographies. along advanced structures with Override presentation of all recently distributed regions ImageNet execution[51].

2.5.2 Opportunities, limitations and Ethics of AI

Assuming the sensational climb of interest in AI, huge assessments have started the impact of AI on society, not simply in that frame of mind furthermore in authentic, and moral areas. This also consolidates the speculation that autonomous super AI may ultimately displace the mental abilities of individuals. This future circumstance is known as the "man-made intelligence SINGULARITY", portrayed as the limit of machines to gather better machines without help along any other person. Force AI experts are more revolved around making structures that are perfect at tasks in a tight reach of purposes. This middle is at chances with the chance of the interest of fake general understanding (AGI) that could reflect all uncommon mental limits related to human knowledge for instance, care and excited data. Current AI progression and the situation with our power as the most cunning species on the planet, further social worries are raised. Regardless, AI propels really confined to very specific executions. One hindrance of AI is the shortfall of "ordinary sense" the

capacity to condemn information past its acquired data. Artificial intelligence is moreover limited similar to enthusiastic information[52].

Artificial intelligence can simply perceive fundamental human energetic states like shock, satisfaction, feel sorry for, stress, anguish, fear, and unprejudiced nature. Excited information is one of the accompanying edges of additional huge degrees of personalization. The software designing guidelines driving AI forward, are rapidly advancing and assessing its influence, along a creative point as well as along a social, moral and real viewpoint is basic[53].

2.6 Kinds of AI Strategies Applied in Track Planning

2.6.1 Problem Demonstration

In the approach, arranging of present day robots, the direction orientation arranging of the robots is a significant part of the cycle. In robot technique arranging, car method issue is likewise evolved with the improvement of robot method arranging. Toward the start of the advancement of robot method arranging, the issue of vehicle planning around then centered around fixed data vehicle planning for the workshop[54]. The entire cycle of booking control is exceptionally basic, simply thinking about planning along the viewpoint of operational exploration. With the advancement of robot approach arranging, the focal point of robot approach arranging is continuously confronted with more noteworthy transportation limit interest. Hence, with the change of robot approach arranging strategy, a huge territorial place is set up in numerous metropolitan transportation center points to meet the interest of vehicle planning for various areas[55]. The fundamental administration aim is to understand the greatest robot bearing orientation prerequisites by utilizing more limited mileage and a smaller amount vehicle assets. In the practical robot approach arranging, the relating augmentation of various issues is slowly framed[56].

To finish the model conversation of the above real circumstance, the accompanying circumstances are set up:

- (1) the robots direction orientation focus arranging data is the equivalent.
- (2) every sending point in city is known.
- (3) the quantity of data of every beneficiary is known.
- (4) the real distance between every accepting site and the hub is recognized.
- (5) all interest focuses in city must be covered.
- (6) the check and stacking of the robot bearing reference car must be carefully controlled.
- (7) there is a foreordained maximum cutoff on the complete season of sending the vehicle
- (8) must arrangement on schedule

To summarize, the difficult model is the means by which to apportion restricted arranging vehicle assets under the above requirements, how to design a sensible course for the arranging vehicle, and complete the comparing conveyance interest with the base time and transportation cost[57].

2.6.2 Problem Analysis

In the depiction portrayed in the past area, the approach factor of the issue can be characterized appropriately, beginning along the arranging hub. The succession of sending focuses came to by finishing other sending necessities is known as the real sub method, while the relating crossing method does exclude the arrangement of a few real sub methods of continued arranging focuses[58]. In this streamlining model, we add some virtual arranging hubs and relate with arranging vehicles. On this premise, all courses are coordinated into a few significant courses, and just one arranging network is set for the entire course. A sending vehicle can't pass at least two courses. For instance, a arranging hub

(number 0) necessities to convey data to the excess number 1,2... N sending focuses, M vehicles are accessible, and m-1 virtual arranging hub is set (really a similar arranging hub, number is N+1,N+2,... ,N+ m-1), and set the distance network as indicated by the genuine importance. In this method, we turn the CMTSP issue into the (N+M) point TSP issue[59].

2.6.3 Optimization Of The Artificial Intelligence Algorithm

2.6.3.1 Robot Reference System Design of Artificial Intelligence Algorithm

In the day by day project of the arranging framework, the primary examination project is the key exploration substance of the street arranging. In this cycle, it is important to locate the most brief course, which is additionally the most brief method. All in all, the method arranging and the most brief approach configuration can be mutually investigated. The issue of arranging approach most brief truly is that there are numerous practical constraints to be investigated, containing time , different components, domain, expense and estimation surface. The most brief approach issue is employed to the bear signal connection of the robot. There is no rejecting that this is an impression of the exhibition of explicit investigation, yet the activity of the calculation in the particular examination measure is a typical highlight, and its differential substance is primarily reflected in various bends with various loads. In the event that the most brief length is to be discovered, the weight might be planned as the genuine length in the way. In the event that it is to be planned as the vehicle time, it is important to plan it as the quickest distance in the way. Now is the ideal time furthermore, distance are relative to one another, so significant activity impact can be accomplished through the count of the calculation, which is additionally to guarantee that it can accomplish most extreme monetary expense execution and construct the ideal approach at one time[60].

2.6.3.2 Robot Direction Scheme Structuring and Optimization

To start with, investigating the relating future transmitting locations as per the main realized transmitting marks, decide if the obtained length information is the most brief thought, and afterward sum the important limitation of time limit. At that point if isn't the best approach is resolved, the likelihood estimation, in the following approaches as per the transmission, the directions of focuses provided locations, discover the transmitting mark nearest to the street, also as indicated by the distance and speed estimation inside a recommended period all might being arrived at the following point, get these sent probably the most brief approach, here for the most limited approach in street distance for weight, Dijkstra calculation has been end up being one of the most ideal calculation might get the most brief approach.

By the calculation portrayed over, the cycle of the calculation is created of a huge cycle, the external circle run $n - 1$, internal circle for 2, are running $n - 1$, so the time multifaceted nature of the calculation above to amount to is $O(n) + O(1) + O(n) + O(n^2) = O(n^2)$. Man-made consciousness calculation is applied to the neighborhood the guide. In light of the calculation, the likelihood of the most brief method that can be reached at a getting point is determined as follows[61].

$$Pr_r = \prod_{i=1}^{Nr} P_{n(S_{i+1}, S_i)}^m (S_{i+1}/S_i) \quad (2.15)$$

Such that, m addresses the MTH vehicle of the provincial web-based store's vehicle source, n signifies the comparing convergence point as point n , and way r indicates $\{S_1, S_2 \dots\}$. For the blend of SN_r series of areas, N_r alludes to the quantity of crossing focuses for the way r , S_i and S_{i+1} separately address the segments ceaselessly passing on the way r . The average probability of all paths is[62].

$$P = \frac{1}{N} \sum_{r=1}^{N_P} Pr_r \quad (2.16)$$

Where N_p speaks to all sending areas inside 3 minutes, PR speaks to the likelihood of relating methods inside a specific timeframe. At long last, the model reconciliation is done for the information of all vehicles of the vehicle source in the network, in which the weight is instated, which is set as the most elevated on the initial not many days of the week, followed by working days or ends of the week, and the rest is the base. Since this article plan execution measure, comprehending the disconnected likelihood and without utilizing Hadoop group to acknowledge, so the cycle to accomplish the necessary time is too since quite a while ago, since it's getting late requirements, for the accompanying equation P at the hour of the test highlight illuminate the likelihood, set to 0, which contains just the direction arranging vehicle proportion, if the genuine determined estimations of P , will undoubtedly additionally upgrade the determining model to tackle the exactness of the likelihood[63].

$$Pro = \frac{1}{M} \sum_{m=1}^M (w_m \cdot (P + P_{real}) \cdot (1 + \alpha)) \quad (2.17)$$

M as the examination on the quantity of vehicles, W_m is comparing to each vehicle arranging data measure of weight (comparative with the far off regions, metropolitan region inside the extent of purpose of the vehicle's weight is set higher), α is a weighted boundaries, if how much data in the full reach, is set to 0 to 1, generally set to 0, P (2) that the above equation in computing the likelihood; P_{real} corresponds to the proportion of the conveyance load pace of a

specific vehicle in verifiable information and the general conveyance amount, and the computation recipe is as per the succeeding:

$$P_{\text{real}} = \frac{C_{\text{pass}}}{C_{\text{all}}} \quad (2.18)$$

Such that, the C_{all} is the aggregate sum of all the data that may be conveyed inside the predetermined time, and the C_{pass} is the quantity of vehicles that have really finished the conveyance inside the predefined time. As per the likelihood worth of the latest portion determination, each briefest way might be obtained.



Chapter Three

Methodology

CHAPTER THREE

THE PROPOSED METHODOLOGY

3.1 Introduction

In this chapter, an efficient AI multipurpose robotic system will be proposed based on a microcontroller and wireless technology as a suggested structure. The suggested robotic system has been divided into several sections which are necessary for operating, monitoring, and controlling the proposed block diagram system. as illustrated in figure (3.1).

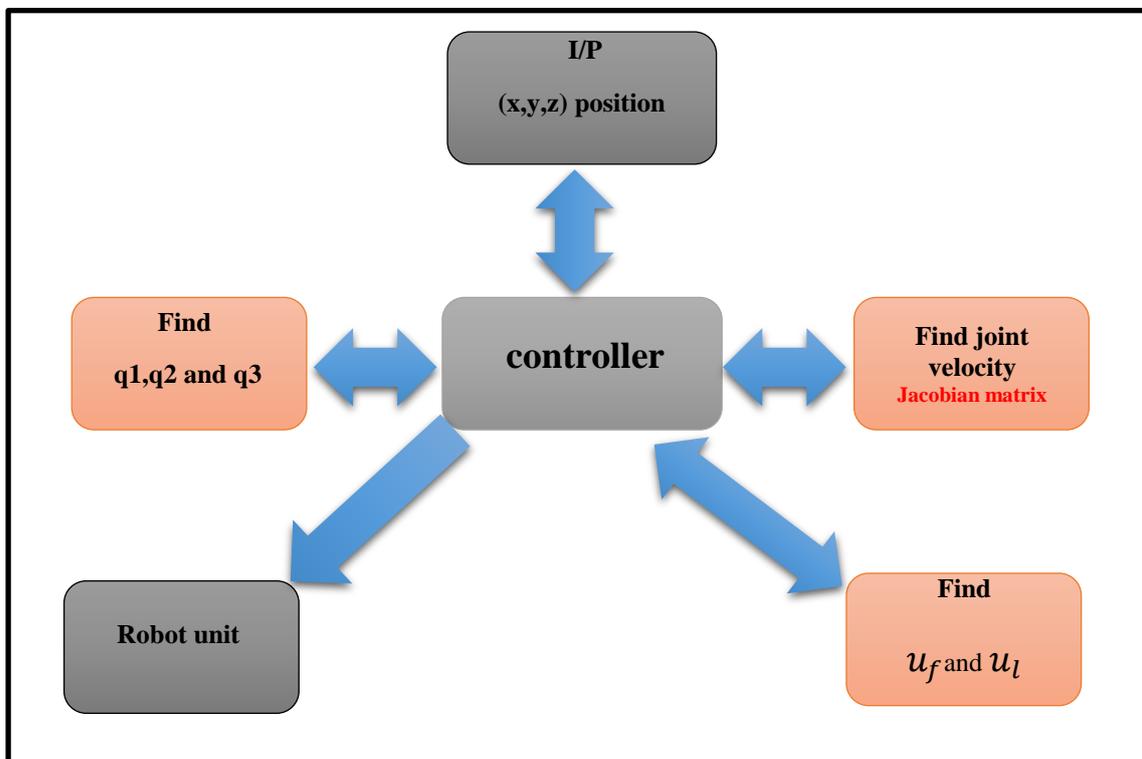


Fig. 3.1: Block diagram of the suggested robotic system.

3.2 The Suggested System Details

The proposed system for this proposition was proposed to execute a versatile robot with four wheels, with the chance of controlling the development of the arm and wheels through movement facilitates planned by the mission relegated to it. Figure (3.2) shows the general plan of the proposed plan system of a four-wheeled portable robot.

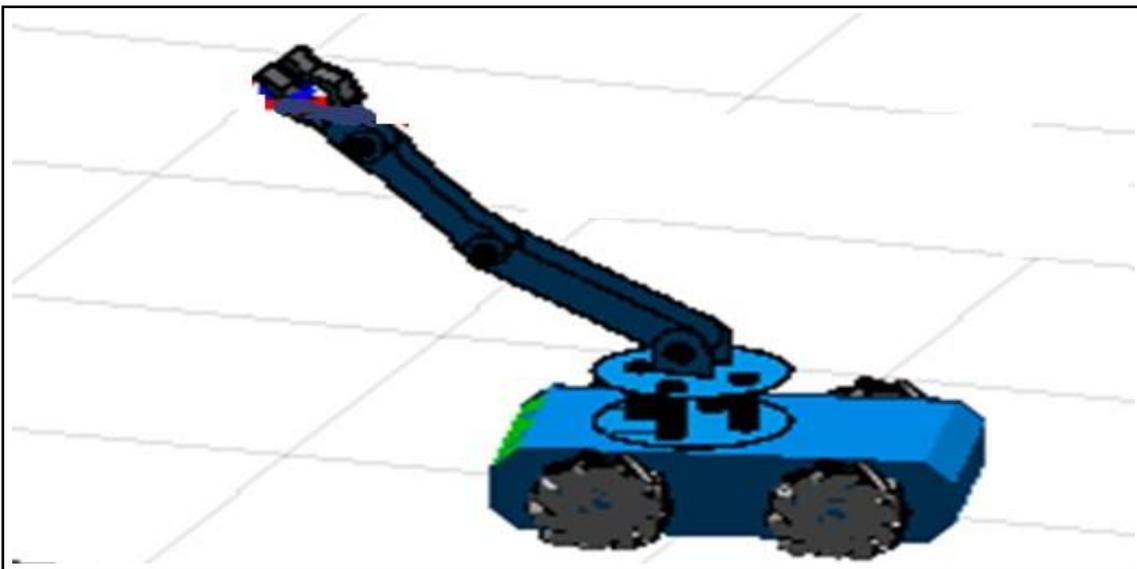


Fig 3.2: A general scheme of the proposed design system of a four-wheeled mobile robot with arm.

The detailed parameters and design specifications of the system displayed in figure (3.2) have been shown in Table (3.1).

Table 3.1: The detailed parameters and design specifications of the Suggested system.

Robot Parameters	Distance to interest point (m)	Resultant Robot Car Distance c (m)	Height of Car from Land b (m)	1 st Shaft Arm Length l_1 (m)	2 nd Shaft Arm Length l_2 (m)	Distance between wheels d (m)	Wheel Radius R (m)
Values	0.1	0.25	0.25	0.5	0.55	0.5	0.1

Actually, the suggested mobile robot car with arm consist of the following main sections:

- 1) The Car_Body and Car_Weels_ Section.
- 2) The Arm Section, and
- 3) The Control Section.

3.2.1 The Car_Weels_ Subsystem

Preparation of the design controls for the mobile robot cart takes place in this section. where the fundamental components and parameters required to create the ideal image of the robot cart carrying the movable arm are provided. As a result, in accordance with the project's design, this cart has a rectangular parallelepiped base whose dimensions are set by using the built-in functions for ready-made MATLAB2022b m. files, as depicted in figure (3.3).

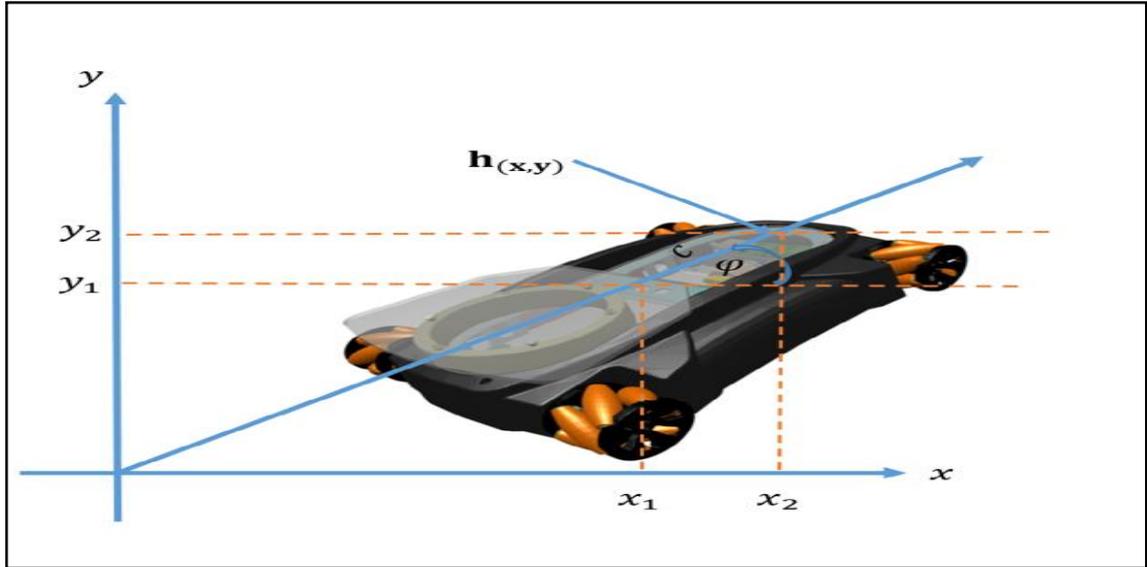


Fig 3.3: Structure of the designed mobile robot 4-wheeled car.

The wagon unit consists of four wheels with two axles, each axle connected to two wheels opposite the axle in a cylindrical shape. It is formed in a circular motion, and all shapes of these parts are chosen from parts of the built-in m.files utilities provided by MATLAB. The inverse kinematics equations of the mobile 4-wheeled car motion are listed as below:

$$h_x = x_1 + x_2, \quad (3.1)$$

$$h_y = y_1 + y_2, \quad (3.2)$$

$$h_x = x_1 + c \cos\varphi, \quad (3.3)$$

$$h_y = y_1 + c \sin\varphi, \quad (3.4)$$

Where x_1 , and y_1 , are the coordinates of the start point of the mobile robot car, while, x_2 , and y_2 , are the coordinates of the point of the destination that the robot car will travel. Also, h_x , represents the x-axis component of the resulting displacement of the mobile robot car direction, whereas, h_y , denotes the y-axis component of this resulting displacement that is indicated by $h(x, y)$.

Furthermore, φ , denotes the phase angle of the mobile robote car resulting displacement from the x-axis, and c , is the length of this resulting displacement. All units in meters of that parameters in meter.

Now applying the kinematic equations, which figure (3.3) refers to it. Firstly, it must be known that the car moves at two velocities. The first velocity is the front velocity, and the second velocity is the lateral velocity.

after that, it must be found the derivation of equations (3.3) and (3.4). the derivation of the first term in each equation becomes from dividing each velocity into two components, which are given as follows:

$$x_{u_f h} = u_f \cos \varphi \tag{3.5}$$

$$y_{u_f v} = u_f \sin \varphi \tag{3.6}$$

$$x_{u_l h} = u_l \sin \varphi \tag{3.7}$$

$$y_{u_l v} = u_l \cos \varphi \tag{3.8}$$

Where $x_{u_f h}$ is the horizontal component of the front velocity, $y_{u_f v}$ is the vertical component of the front velocity, $x_{u_l h}$ is the horizontal component of the lateral velocity, $y_{u_l v}$ is the vertical component of the lateral velocity, u_f is the front velocity, and u_l represent the lateral velocity in m/s. Also it will be defined $\dot{\varphi}$ as the angular velocity of the mobile robote car in rad/s. Based on this, the derivation of equations 3.3 and 3.4 are given as follows:

$$\dot{h}_x = u_f \cos \varphi - u_l \sin \varphi - c \omega \sin \varphi, \tag{3.9}$$

$$\dot{h}_y = u_f \sin \varphi + u_l \cos \varphi + c \omega \cos \varphi, \tag{3.10}$$

$$\dot{\varphi} = \omega, \tag{3.11}$$

where \dot{h}_x , and \dot{h}_z , are the the velocities of the horizontal and vertical displacements of the mobile robot car direction. Next, the previous kinematics equations of the mobile robot car might be expressed in matrix:

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & -c\sin\varphi \\ \sin\varphi & \cos\varphi & c\cos\varphi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_f \\ u_l \\ \omega \end{bmatrix}, \quad (3.12)$$

3.2.2 The Robot Arm Section

In this section, the details of the forward kinematic equations and inverse kinematic equations concerning the motion of the robotic arm will be presented and discussed in details. The robotic arm will be constructed of two sub arms and thus divided in to two sub sections called shafts, the first moving shaft, and the second moving shaft. These two shafts will construct the complete robotic arm in which, each shaft will have length of l_1 , and l_2 respectively. The robotic arm will have two types of motions, the first movement will be up and down in longitudinal action for the two arm shafts, and the second movement will be in rotational way in which the base carrying the robotic arm will rotate 360° around its center. All these motions will be discussed and demonstrated with comprehensive inverse kinematic equations and forward kinematic equations.

Firstly, it will be write the necessary kinematic equations of the first movement of the robotic arm. The method that depends here to find these equations is the same it in the car section. Therefore, and basing the figure (3.4) the necessary kinematic equations of the first movement of the robotic arm can be given as follow:

$$h_x = x_1 + x_2, \quad (3.13)$$

$$h_z = b + z_1 + z_2, \quad (3.14)$$

$$h_x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2), \quad (3.15)$$

$$h_z = b + l_1 \sin q_1 + l_2 \sin(q_1 + q_2), \quad (3.16)$$

Where, h_x , and h_z , are the horizontal and vertical displacements of the first robotic arm motion respectively. Also, x_1, z_1 , are the first joint coordinates, and x_2, z_2 , are the second joint coordinates of the first robotic arm shaft. Further, b , is the altitude of the robotic arm base from the ground. All these values are measured in meters. At last, q_1 , and q_2 , will denote the angles of the robotic arm first and second shafts respectively.

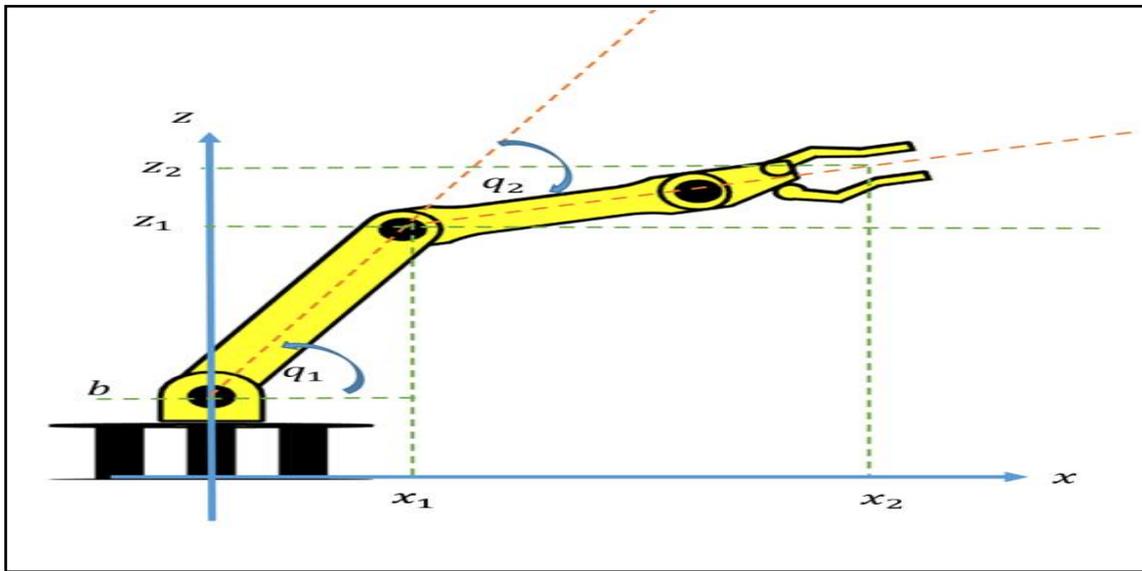


Fig 3.4: The first longitudinal movement of the robotic arm.

Completing the derivations of the kinematics equations of the first longitudinal robotic arm motion, and depending on Eqs. (3.13) to (3.16), the velocity of the horizontal and vertical displacements of the first robotic arm motion will be as follows:

$$\dot{h}_x = (-l_1 \sin q_1 - l_2 \sin(q_1 + q_2))\dot{q}_1 + (-l_2 \sin(q_1 + q_2))\dot{q}_2, \quad (3.17)$$

$$\dot{h}_z = (l_1 \cos q_1 + l_2 \cos(q_1 + q_2))\dot{q}_1 + (l_2 \cos(q_1 + q_2))\dot{q}_2, \quad (3.18)$$

In which, \dot{h}_x , and \dot{h}_z , are the the velocities of the horizontal and vertical displacements of the first robotic arm motion respectively. Next, it can also represent the above equations in matrix form as succeeded:

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_z \end{bmatrix} = \begin{bmatrix} (-l_1 \sin q_1 - l_2 \sin(q_1 + q_2)) - l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) + (l_2 \cos(q_1 + q_2)) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \quad (3.19)$$

Now, concerning with the second rotational movement of the robotic arm along the base center which has been displayed in figure (3.5). Firstly, it will be write the necessary kinematic equations of the second movement of the robotic arm.:

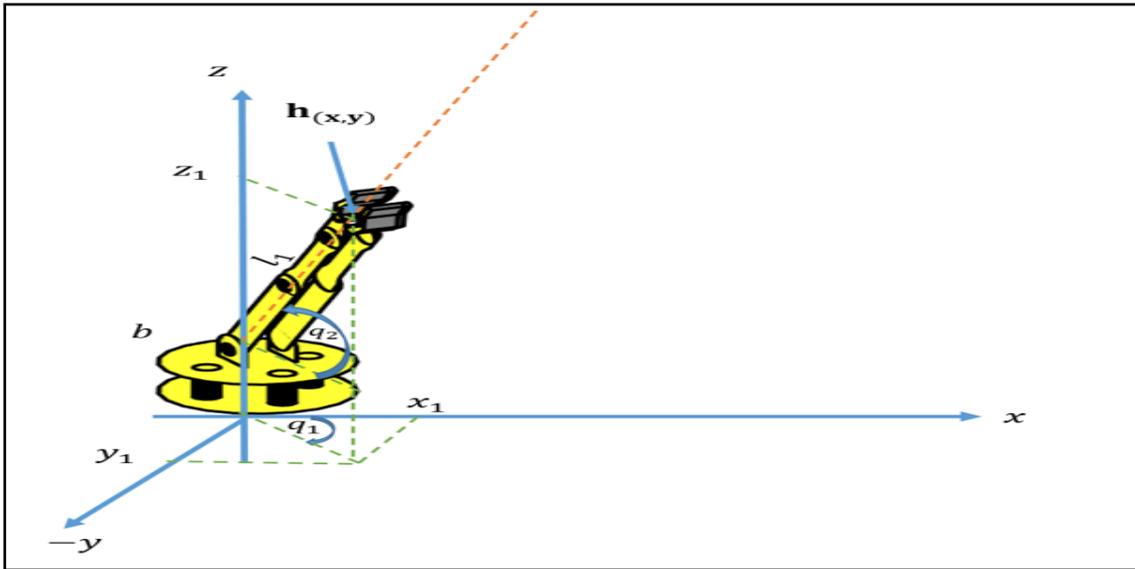


Fig 3.5: The second rotational movement of the robotic arm along the base center.

$$h_x = x_1, \quad (3.20)$$

$$h_y = y_1, \quad (3.21)$$

$$h_z = b + z_1, \quad (3.22)$$

$$h_x = l_1 \cos q_2 \cos q_1, \quad (3.23)$$

$$h_y = l_1 \cos q_2 \sin q_1, \quad (3.24)$$

$$h_z = b + l_1 \sin q_2, \quad (3.25)$$

Where now, h_x , h_y , and h_z , are the x,y,z coordinates of the robotic arm terminal catcher handle location all in meters. Also, q_1 , is the angle of horizontal robotic arm rotation , and q_2 , is the angle of the vertical robotic arm elivation from ground all in degree. Furthermore, l_1 , is the length of the two shafis sections for the total robote arm in meters.

Hence, completing the derivations of the kinematics equations of the second rotational robotic arm motion, and depending on Eqs. (3.20) to (3.25), the velocities the x , y , and z displacements of the second robotic arm motion will be as follows:

$$\dot{h}_x = -(l_1 \cos q_2 \sin q_1) \dot{q}_1 + (-l_1 \cos q_1 \sin q_2) \dot{q}_2, \quad (3.22)$$

$$\dot{h}_y = (l_1 \cos q_2 \cos q_1) \dot{q}_1 + (-l_1 \sin q_1 \sin q_2) \dot{q}_2, \quad (3.23)$$

$$\dot{h}_z = l_1 \cos q_2 \dot{q}_1, \quad (3.24)$$

Where \dot{h}_x , \dot{h}_y , and \dot{h}_z , are x,y, and z, rotaitonal velocities, of the second robotic arm motion in m/s. Similarly, the last kinematics equations of the mobile robot arm secon motion might be expressed in matrix form as follows:

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{h}_z \end{bmatrix} = \begin{bmatrix} -l_1 \cos q_2 \sin q_1 & -l_1 \cos q_1 \sin q_2 \\ l_1 \cos q_2 \cos q_1 & -l_1 \sin q_1 \sin q_2 \\ 0 & l_1 \cos q_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \quad (3.25)$$

Finally, it will be write the necessary inverse kinematic equations of the movements of the robotic arm. The shut structure arrangement technique including mathematical and arithmetical arrangements is utilized to display the robot arm's opposite kinematic. In light of figure (3.6), the arrangement is separated into two cycles. The first step is to calculate the angle of joint 1 as the first step. The development of the robot arm on the $x - y$ surface relies just upon the point of joint 1. Hence, and from figure (3.7) , the plot for joint 1 can be determined as follows:

$$q1 = \tan^{-1}\left(\frac{Py}{Px}\right) \quad (3.26)$$

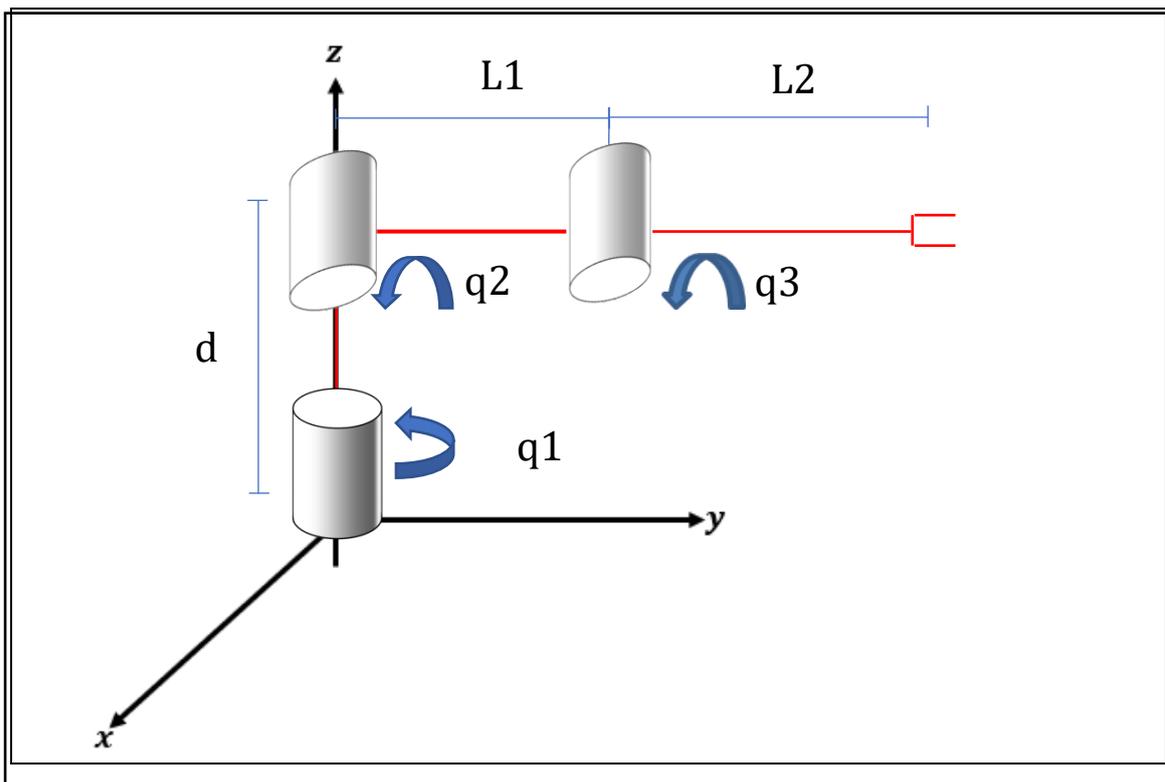


Fig 3.6: Joints of robotic arm.

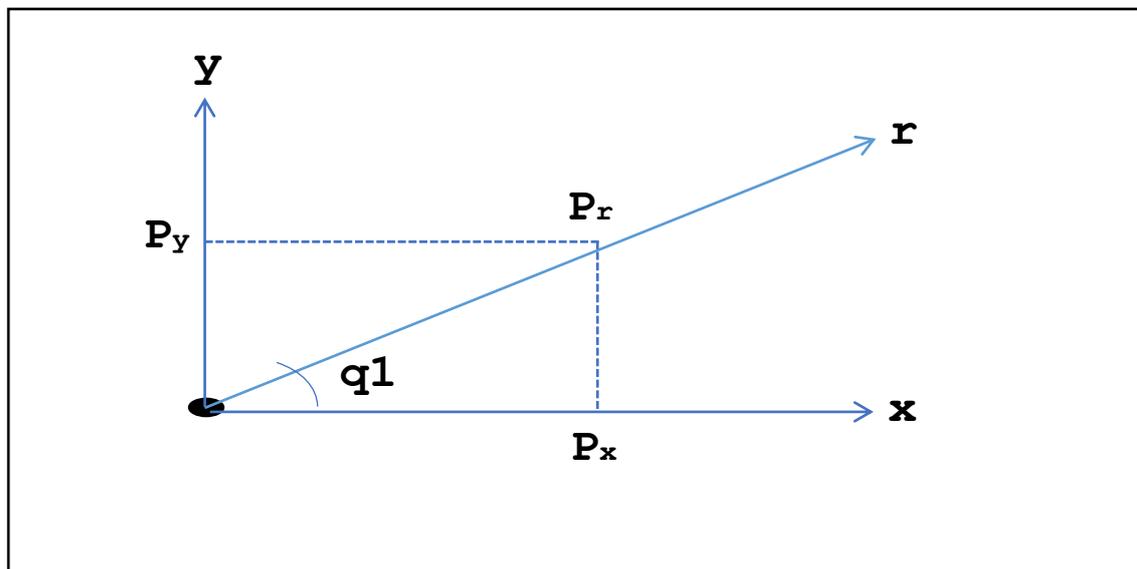


Fig 3.7: Calculated the angle for joint 1.

Since (q_1) is the rotational plot for the robot arm's base, the point range is somewhere in the range of -180° and 180° . P_x and P_y are the situations on the x and y hub separately. The quadrant for (q_1) is distinguished to decide the sign for each geometrical proportion in a given quadrant. In each quadrant, the signs for the trigonometric ratio are shown in the table.

Table 3.2: Signs for the Trigonometric Ratio in Each Quadrant

P_x	P_y	Quarter	angle (q_1)
+	+	1	q_1
-	+	2	q_1+180
-	-	3	q_1-180
+	-	4	q_1

In addition, Calculating the angles for joints 2 and 3 (q_2 and q_3) is the second step. For getting the answer for q_1 and q_2 the 3-layered (3D) space which is comprising of the x, y, and z coordinate tomahawks, is streamlined to a 2-layered (2D) surface. The x and y-hub are converged as another pivot and

known as the r-hub, as displayed in figure (3.8) . Using the Pythagorean equation, the end-effector's x- and y-coordinates are combined to form the r-coordinate

$$Pr = \sqrt{Px^2 + Py^2} \tag{3.27}$$

Where Pr is the resultant from the axes X and Y.

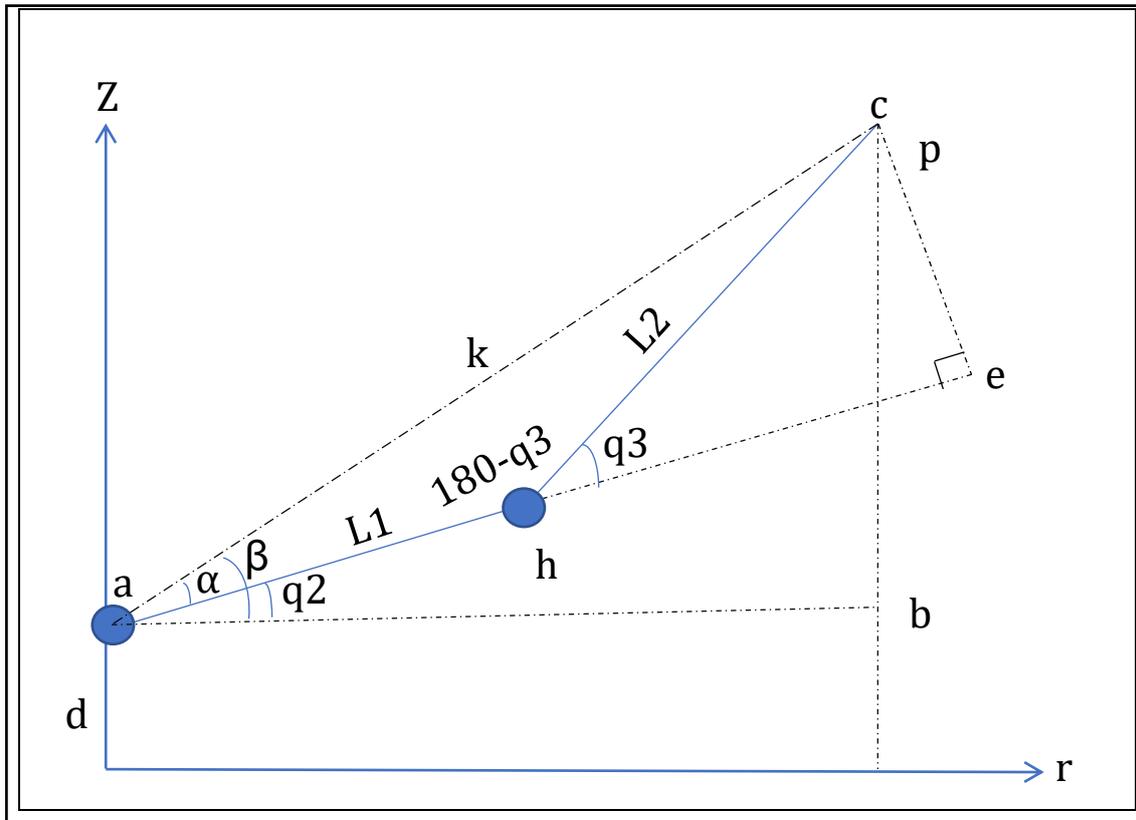


Fig 3.8: Merged the X and Y-axis as a new axis and defend as the r-axis.

From the figure formed a group of triangles such as (abc, ahc, and hec)

$\Delta(abc)$

$$k = \sqrt{(p_z - d)^2 + r^2} \tag{3.28}$$

$$\beta = \tan^{-1}\left(\frac{p_z - d}{r}\right) \tag{3.29}$$

$\Delta(ahc)$, And from cos law

$$k^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos(180 - q_3) \quad (3.30)$$

$$k^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos(q_3) \quad (3.31)$$

$$\cos(q_3) = \frac{k^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (3.32)$$

$$q_3 = \cos^{-1}\left(\frac{k^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \quad (3.33)$$

$\Delta(aec)$ and $\Delta(hec)$

$$\alpha = \tan^{-1}\left(\frac{L_2 \sin q_3}{L_1 + \cos q_3 L_2}\right) \quad (3.34)$$

$$q_2 = \beta - \alpha \quad (3.35)$$

Moreover, the inverse kinematic method it can be modeled as a flowchart, which is given in the following figure:

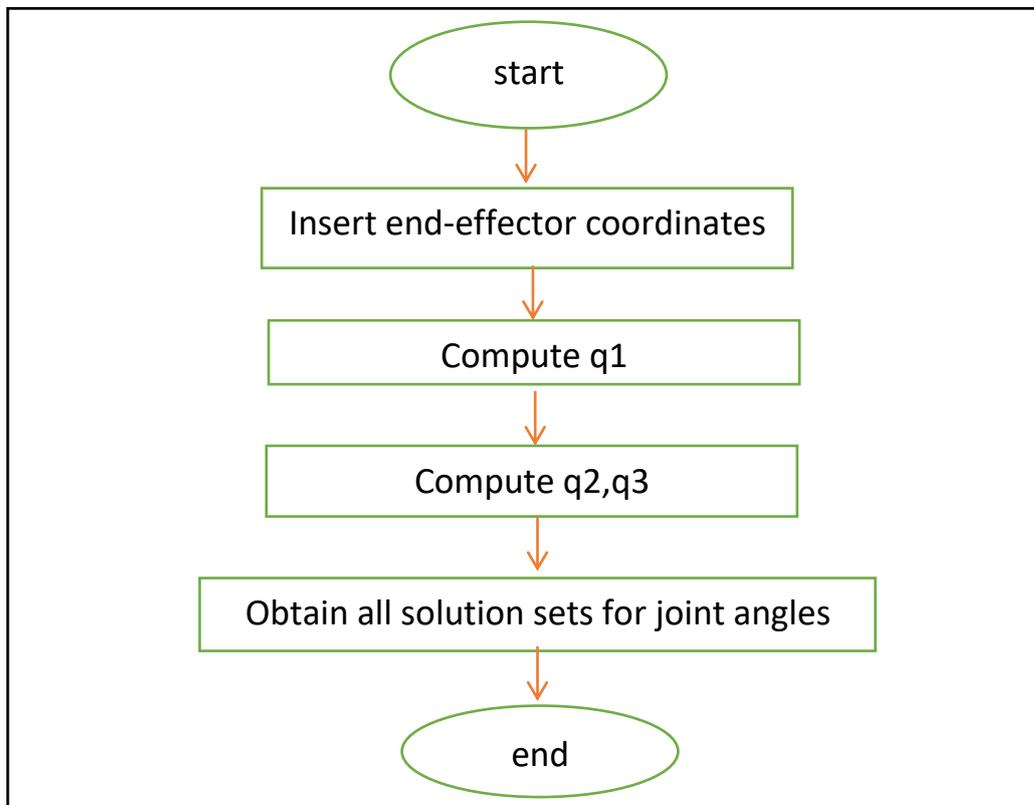


Fig 3.9: flowchart of inverse kinematic method.

3.2.3 The trajectory determination

This part is considered one of the most important sections of making and representing the movement of the robot, as it can be considered as the controlling mind of the robot, through which it is possible to control all the movements of its parts and to determine the directions, speed, acceleration, forces and moments necessary for the smooth functioning of the robot. This is done by describing an action plan or a robot's movement map and specifying the values and amounts of all variables according to the movement path specified in the work plan or map. In this study, a plan or a map to move the robot according to the task assigned to it is drawn and translated into numbers and values as inputs to the control unit, which works to distribute tasks to all parts of the robot. To facilitate and organize this task, the control and control unit was divided into two parts, so that the first part is responsible for regulating the movement of the wheels, while the second part controls the transmission of commands to the robot's limbs. Figure (3.10) presents the details of the map plan area specified to control the movement of the robot.

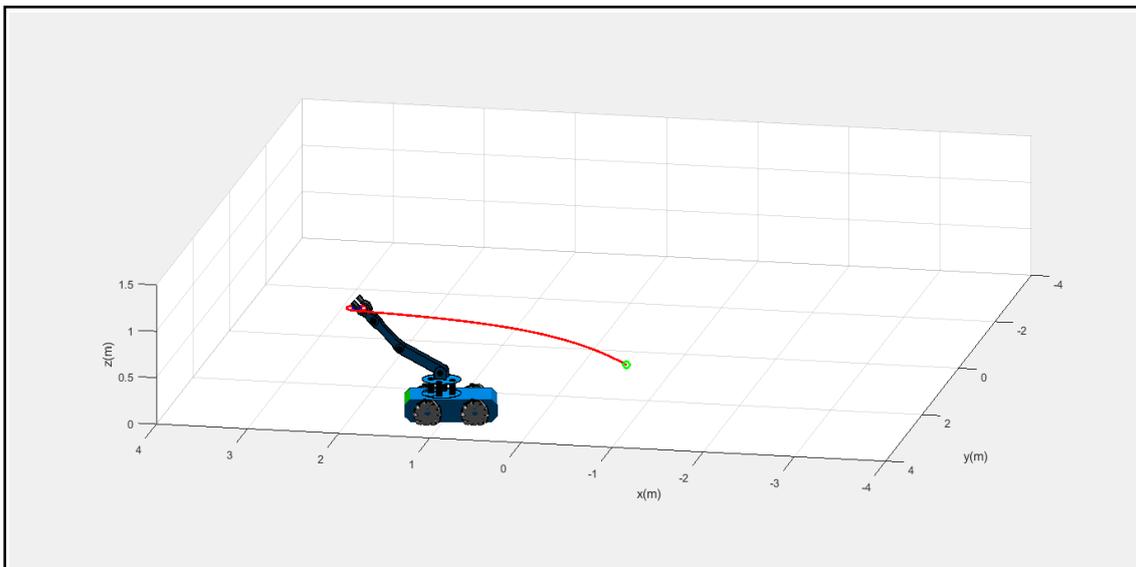


Fig 3.10: MATLAB simulation details of the map plan area specified to control the movement of the robot.

As it is clear from the previous figure 3.6, that the plan area map specified for the robot motion is designed to have x-axis and y-axis ranges from (-3 to 3) meters, while the z-axis range has designed with range from (0 to 1.5) meters. Thus, these dimensions will determine the area of motion that the designed robot will travel within and might be called the robot motion area limitations. Hence, and in order to complete the design requirements, it must consider the robot as a trajectory and calculate the desired trajectory equations as below:

$$h_{xd} = 0.05 \cos(0.1t) - (-x_2), \quad (3.36)$$

$$h_{yd} = 0.05 \sin(0.1t) - (-y_2), \quad (3.37)$$

$$h_{zd} = 0.6 \text{ ones}(1, \text{size}(t, 2)) - (-z_2) - 0.5, \quad (3.38)$$

Such that, h_{xd} , is the desired trajectory of the (x-axis) in meters, h_{yd} , is the desired trajectory in the (y-axis) in meters, and h_{zd} , is the desired trajectory (z-axis) also in meters. Therefore, in order to find the controller relations, it will be evaluated two necessary measures, first the motion error, and second, the Jacobean matrix.

A) The Trajectory Motion Error

The mobile robot trajectory motion error equations might be expressed as follows:

$$h_{xe}(t) = h_{xd}(t) - h_x(t), \quad (3.39)$$

$$h_{ye}(t) = h_{yd}(t) - h_y(t), \quad (3.40)$$

$$h_{ze}(t) = h_{zd}(t) - h_z(t), \quad (3.41)$$

Where, $h_{xe}(t)$, $h_{ye}(t)$, and $h_{ze}(t)$ are the x , y , and z coordinates of the robot trajectory motion error components. Also, $h_{xd}(t)$, $h_{yd}(t)$, and $h_{zd}(t)$ are the x , y , and z coordinates of the desired robot trajectory displacement.

Furthermore, $h_x(t)$, $h_y(t)$, and $h_z(t)$ are the the x , y , and z coordinates of the robot trajectory instantaneous motion displacement. Moreover, the final resulting error vector might be represented by the next expression:

$$h_e = [h_{xe}(t) \ h_{ye}(t) \ h_{ze}(t)] \quad (3.42)$$

B) Jacobean matrix

Firstly, it must be write the equations of $h_x(t)$, $h_y(t)$, and $h_z(t)$. These equations are written based on the figurer (3.11) as follows:

$$h_x = h_x(car) + l1 * \cos(q_2) * \cos(\varphi + q_1) + l2 * \cos(q_2 + q_3) * \cos(\varphi + q_1) \quad (3.43)$$

$$h_y = h_y(car) + l1 * \cos(q_2) * \sin(\varphi + q_1) + l2 * \cos(q_2 + q_3) * \sin(\varphi + q_1) \quad (3.44)$$

$$h_z = b + l1 * \sin(q_2) + l2 * \sin(q_2 + q_3) \quad (3.45)$$

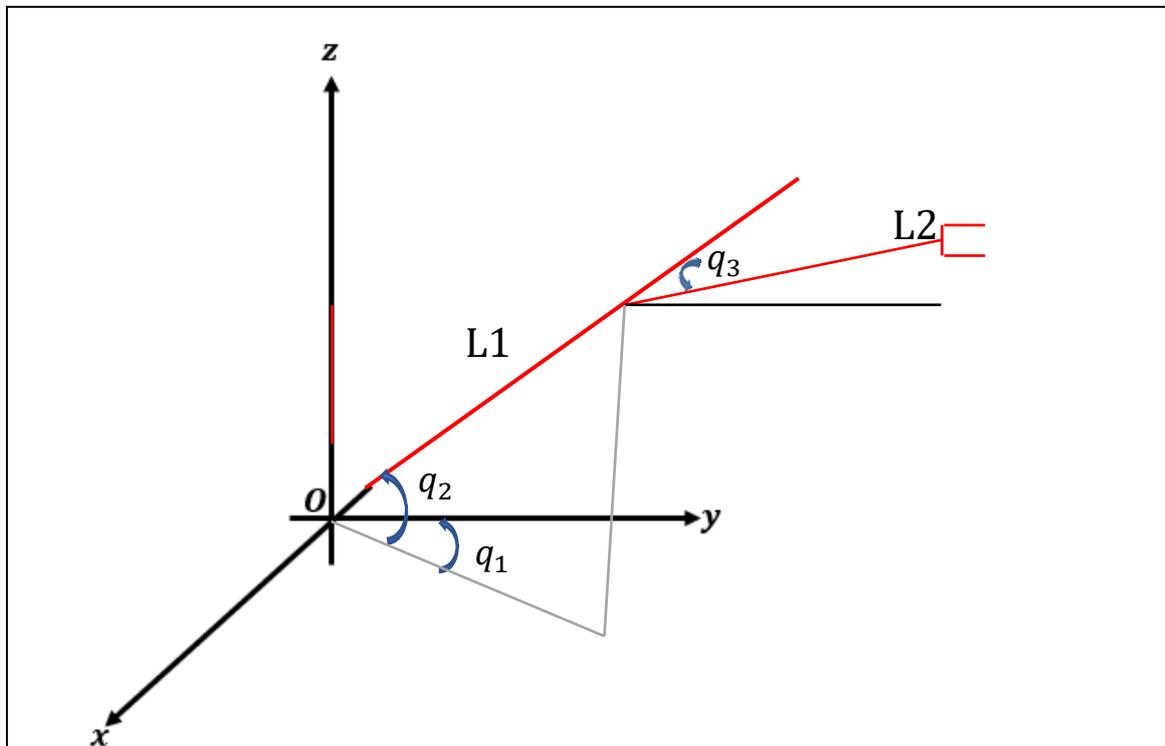


Fig 3.11: The robotic arm on the three-axis.

Now regarding the Jacobean matrix that is necessary to complete the control calculations and might be written as follows:

$$J_{11} = \cos\varphi \quad (3.46)$$

$$J_{12} = -\sin\varphi \quad (3.47)$$

$$J_{13} = -c\sin\varphi - l_1\cos q_2 \sin(\varphi + q_1) - l_2\cos(q_2 + q_3)\sin(\varphi + q_1) \quad (3.48)$$

$$J_{14} = -l_1\cos q_2 \sin(\varphi + q_1) - l_2\cos(q_2 + q_3)\sin(\varphi + q_1) \quad (3.49)$$

$$J_{15} = -l_1\sin q_2 * \cos(\varphi + q_1) - l_2\sin(q_2 + q_3)\cos(\varphi + q_1) \quad (3.50)$$

$$J_{16} = -l_2\sin(q_2 + q_3) * \cos(\varphi + q_1) \quad (3.51)$$

$$J_{21} = \sin\varphi \quad (3.52)$$

$$J_{22} = \cos\varphi \quad (3.53)$$

$$J_{23} = c\cos\varphi + l_1\cos q_2 \cos(\varphi + q_1) + l_2\cos(q_2 + q_3)\cos(\varphi + q_1) \quad (3.54)$$

$$J_{24} = l_1\cos q_2 \cos(\varphi + q_1) + l_2\cos(q_2 + q_3)\cos(\varphi + q_1) \quad (3.55)$$

$$J_{25} = -l_1\sin q_2 \sin(\varphi + q_1) - l_2\sin(q_2 + q_3)\sin(\varphi + q_1) \quad (3.56)$$

$$J_{26} = -l_2\sin(q_2 + q_3) \sin(\varphi + q_1) \quad (3.57)$$

$$J_{31} = 0 \quad (3.58)$$

$$J_{32} = 0 \quad (3.59)$$

$$J_{33} = 0 \quad (3.60)$$

$$J_{34} = 0 \quad (3.61)$$

$$J_{35} = l_1\cos q_2 + l_2\cos(q_2 + q_3) \quad (3.62)$$

$$J_{36} = l_2\cos(q_2 + q_3) \quad (3.63)$$

$$\mathbf{Jacobian\ matrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \end{bmatrix}$$

Where, l_1 , l_2 , are the lengths of the robot arm first and second shafts respectively measured in meters. Also, here q_1 , q_2 , and q_3 are the angular positions angles of the robotic arm two motions measured in radian. Also, It should be noted that the angle φ was combined with the angle q_1 in order to make the cart and the arm in the same direction.

Finally finding the torque relations of the robot arm motion as follows:

$$T = M.v \quad (3.64)$$

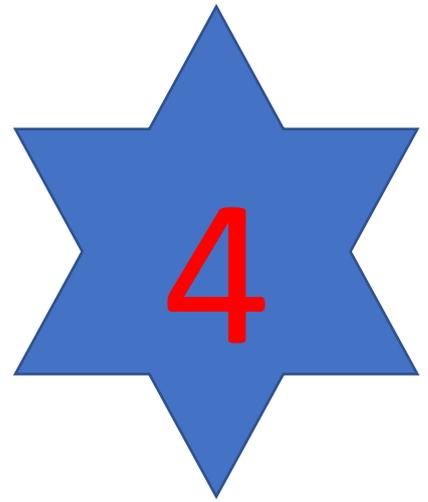
Such that, T , is the estimated torque in N.m units, M , is the suggested robotic mass in Kg, and v , is the velocity of the robot motion in m/sec. now, assuming standard robot body mass of $M=50$ Kg, the three robot equations of torque will be calculated for the three obtained robotic motion speeds, u_f , u_l , and ω as below:

$$T_f = M.u_f \quad (3.65)$$

$$T_l = M.u_l \quad (3.66)$$

$$T_a = M.\omega \quad (3.67)$$

Where, T_f , is the front torque, T_l , is the lateral torque, and T_a , is the angular torque in N.m. units.



Chapter Four

Result and Discussion

CHAPTER FOUR RESULT AND DISCUSSION

4.1 Implementation Results

In this study, the proposed four-wheeled mobile robot with arm system displayed in Chapter 3 has been simulated and implemented using MATLAB2022b m. files codes with the help of the built in functions and the design structure presented previously in figure (3.2) is repeated as shown in figure (4.1).

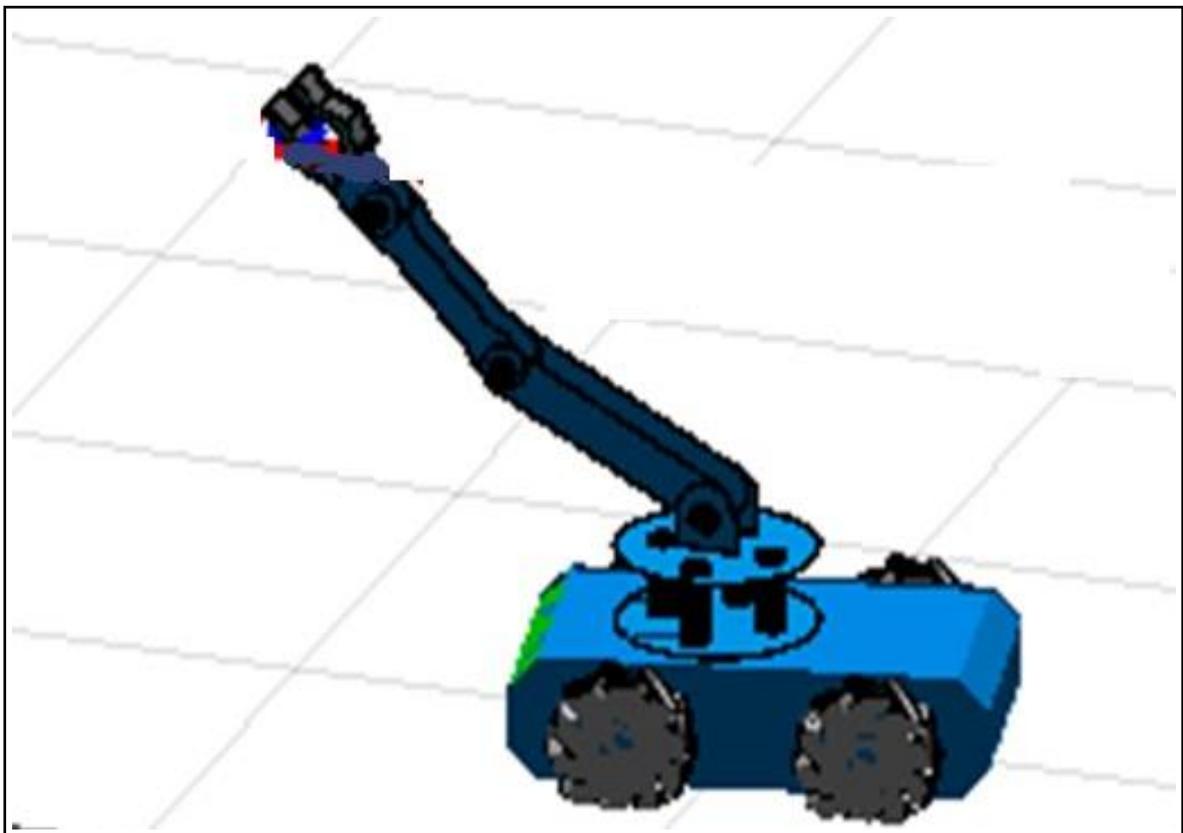


Fig 4.1: MATLAB simulation diagram of the modified proposed system of a four-wheeled mobile robot with arm.

The proposed system illustrated in figure (4.1) has been implemented with the design parameters and specifications as displayed in Table 4.1.

Table 4.1: The detailed parameters and design specifications of the proposed system.

Robot Parameters	Distance to interest point (m)	Resultant Robot Car Distance c (m)	Height of Car from Land b (m)	1 st Shaft Arm Length l_1 (m)	2 nd Shaft Arm Length l_2 (m)	Distance between wheels d (m)	Wheel Radius R (m)
Values	0.1	0.25	0.25	0.5	0.55	0.5	0.1

Firstly, it will be showed how the robot is able to catch something. Then it will be offered how the robot is able to draw a sign (∞). Finally, it will be observed how the robot is able to draw the sin wave. In all these parts it will be shown the results of motion, errors, car velocities, joints velocity, and torque.

A. Catch Something

The implementation of the proposed four-wheeled mobile robot with arm system has been accomplished with the planned area of motion having [-3,3] x,y dimentions displayed in figure (4.2).

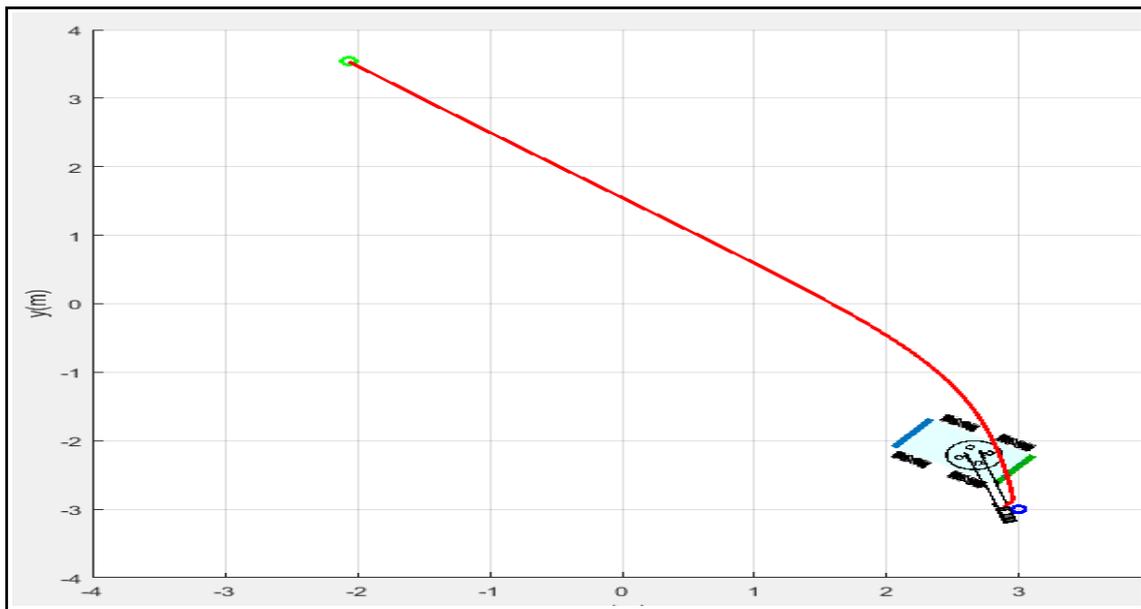


Fig 4.2: The implementation of the proposed four-wheeled mobile robot with arm system with operating area with $[-3,3]$ x, y dimensions.

It can be observed from figure (4.2), the implementation of the m. file program displayed the system in two dimensional configuration with possibility of rotation in three dimensional viewing. The robot motion has been successfully operated in smooth and soft action which indicates the high stability and smoothness of the movement provided by the efficient design controlling equations and simulators.

Similarly, the implementation of the proposed system is again run with three dimensional viewing as illustrated in figure (4.3).

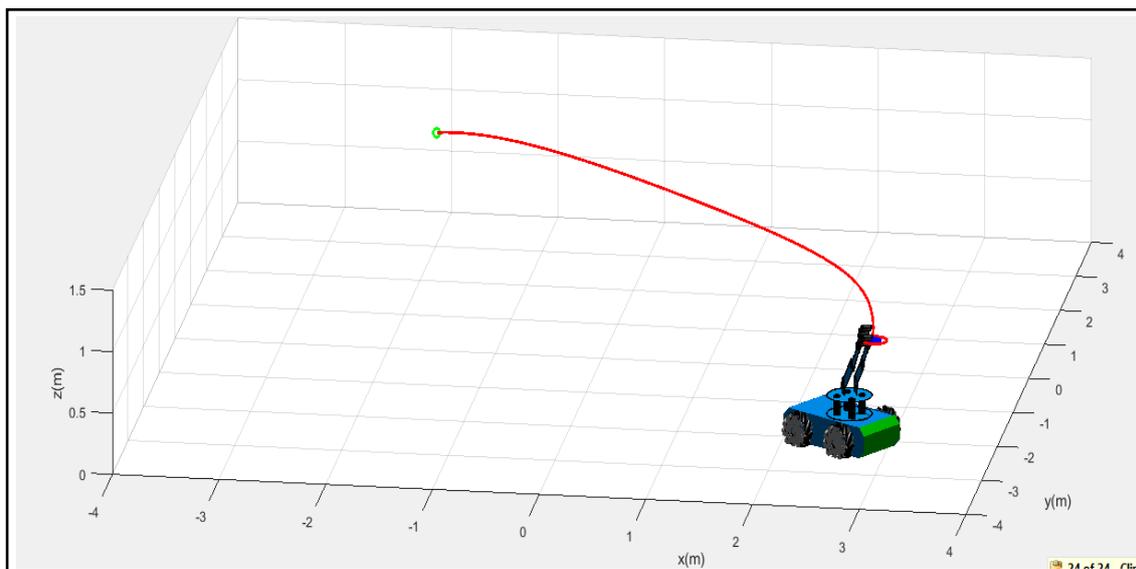


Fig 4.3: The implementation of the proposed four-wheeled mobile robot with arm system with rotation in three dimensional viewing.

In fact, it can be seen from Figure (4.3), the implementation of the inverse kinematic equations of the four-wheels mobile robot car section with the moving arm section have been successfully implemented according to the specified user defined motion coordinates.

Furthermore, the robot trajectory error signals have been computed using error equations demonstrated previously in Chapter 3 and the results presented in figure (4.4).

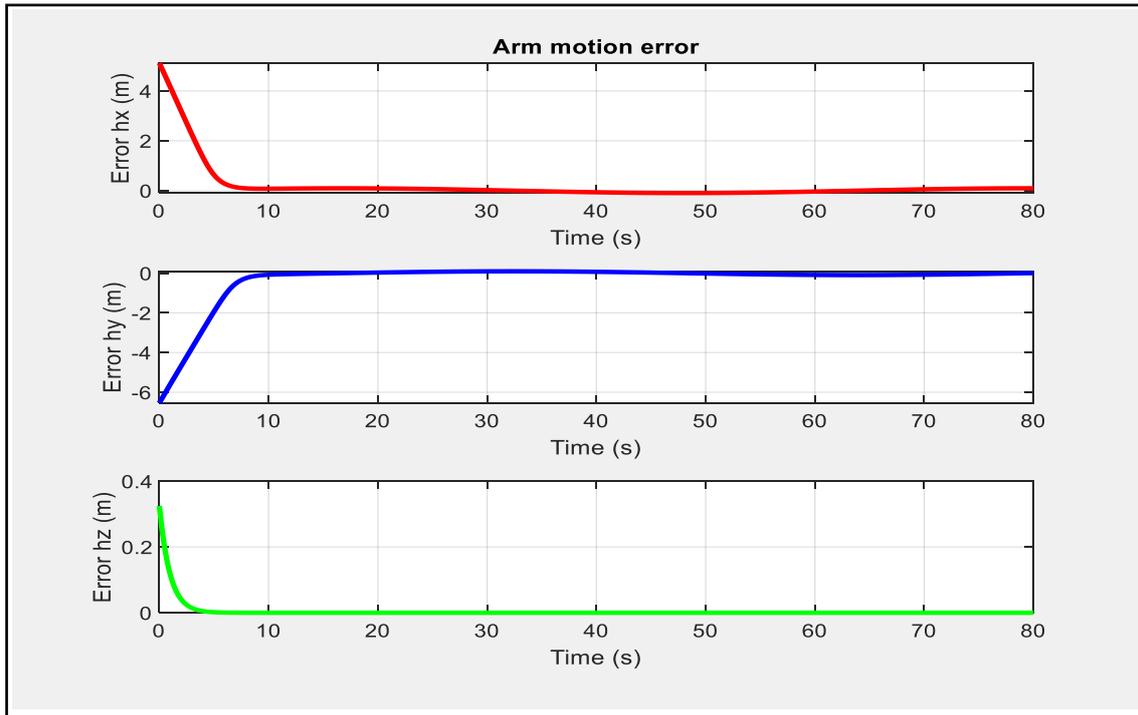


Fig 4.4: Robot arm motion trajectory error signals of the proposed system.

Figure (4.4), displays the proposed robot arm motion error signals in x,y, and z coordinates. It is clear from the past figure (4.4), that the error signals started with small values at the beginning time of robot action, and fastly reduced to zero values within very small amount of time. Actually, throughout the total 80 sec of operation period, the error signals are minimized to zero values within less than 5 seconds.

Next, the motion control speeds U_f , U_l , and ω , of the moving robot car have been computed from implementing their equations presented and discussed in Chapter 3, and displayed in figure (4.5).

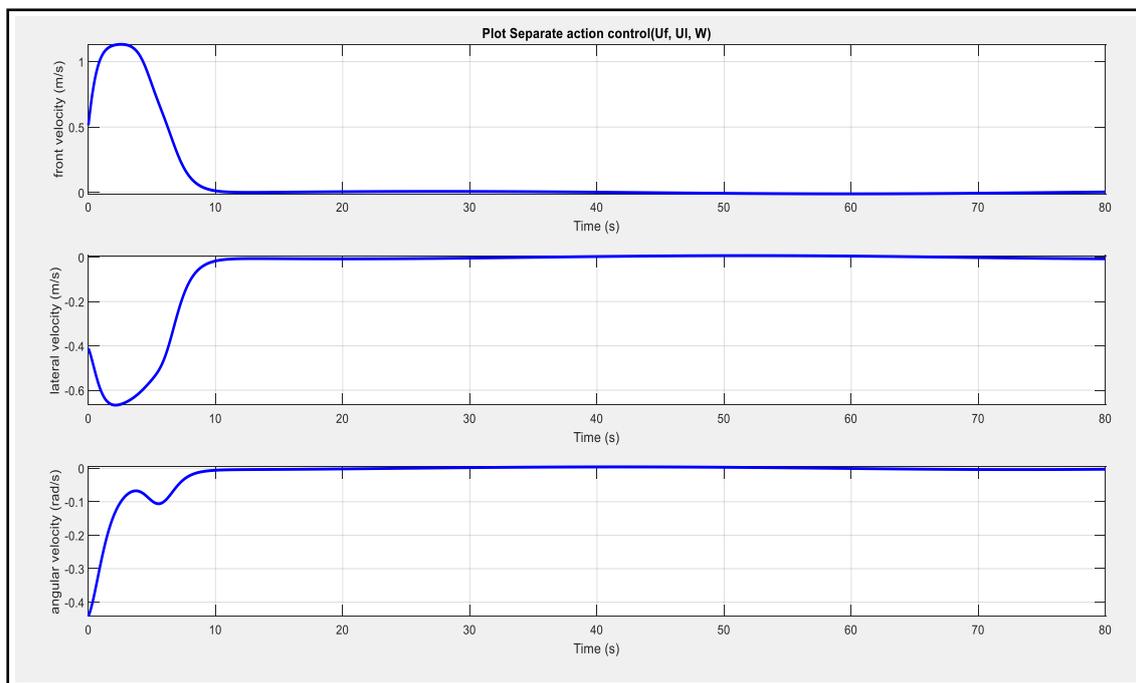


Fig 4.5: The motion control speeds U_f , U_l , and ω , of the moving robot car.

By observing the signals obtained in figure (4.5), it can be seen that the motion control speeds are starting at different values at time $t=0$, then decreased with various shapes for each one until reached to zero values at time instant $t=10$ sec along the total motion time $t=80$ sec. This is normal and maps the action of the mobile robot car with arm from user defined starting point to the end point. Moreover, the response of the robot arm joint angles for separate action control q_1 , q_2 , and q_3 have been extracted and demonstrated in figure (4.6).

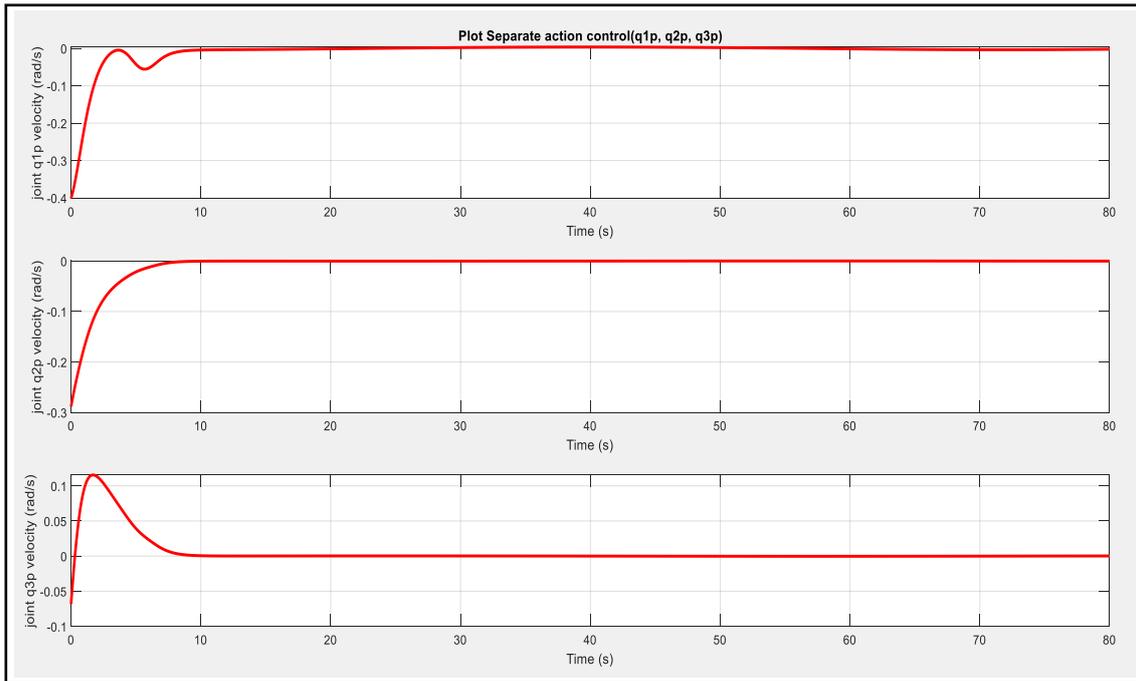


Fig 4.6: The robot arm joint angles for separate action control q_1 , q_2 , and q_3 .

Similarly, by referring to the signals achieved in figure (4.6), it can be also observed that the robot arm joint angles for separate action control q_1 , q_2 , and q_3 are starting at various amounts at time $t=0$, then reduced with different features for every one until approached to zero at a time instant $t=10$ sec along the total motion time $t=80$ sec. This is also natural and maps the motion of the mobile robot car with arm from user defined starting point to the terminated point.

Now, by continuing the analysis and program simulation of our project, the angular positioning angles of the robot arm q_1 , q_2 , and q_3 have been implemented and obtained from the design equations presented previously in Chapter 3 and displayed in figure (4.7).

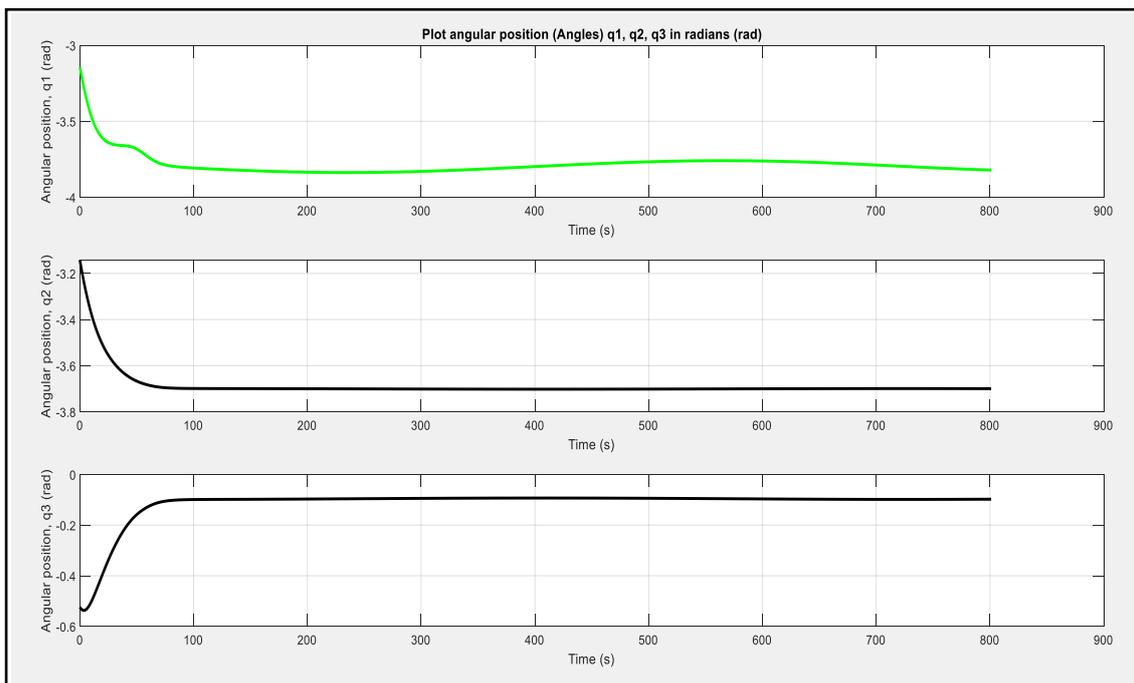


Fig 4.7: The angular positioning angles of the robot arm q_1 , q_2 , and q_3 .

In fact, and from noting the signals of the robot arm angular positions q_1 , q_2 , and q_3 , it is visible that each angle will vary according to the specified motion of the arm joints. These variations will determine the motion of the robot arm to reach and catch the target placed at the end point of the robot motion. Actually, such parameters illustrated from Figures (4.5, 4.6, and 4.7) have been obtained by solving the Jacobian matrix of robot position equations discussed previously in Chapter 3.

Finally, the torque values of the robot motion have been computed and shown in figure (4.8).

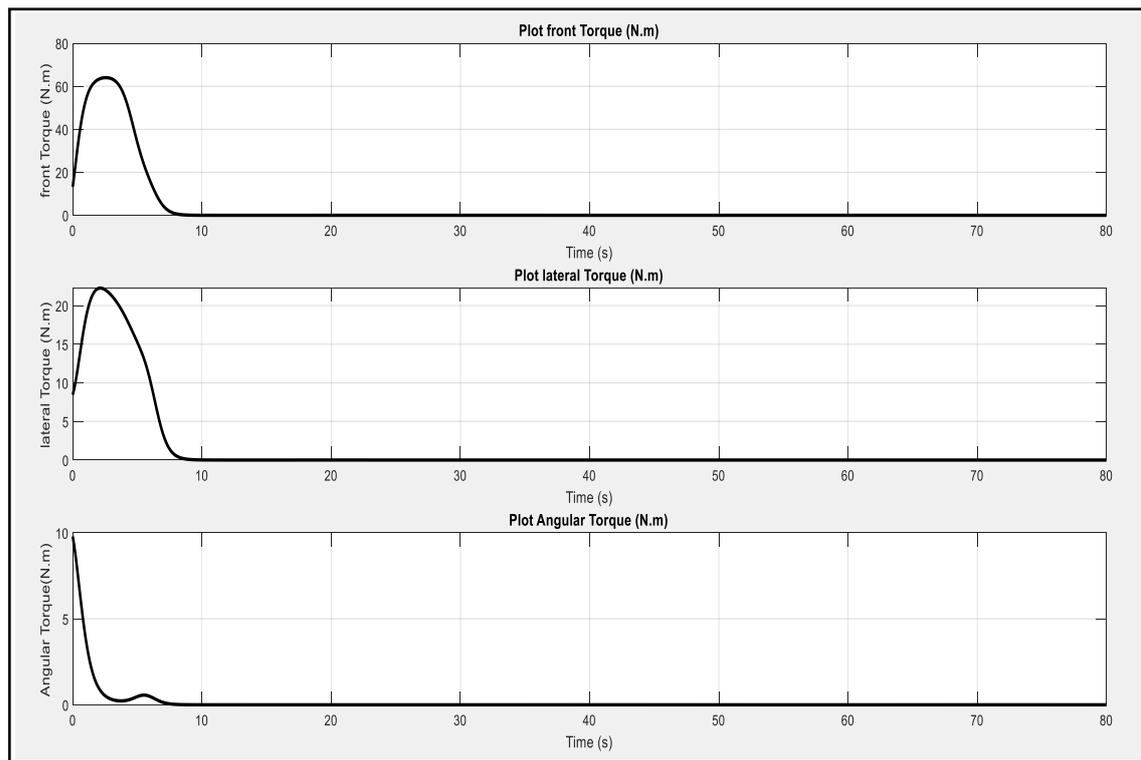


Fig 4.8: The torque signals values of the robot motion.

In fact, and as it is obvious from the torque signals illustrated in figure (4.8), the effective torque values are existing from the robot motion starting time $t=0$ sec and remain to the final robot car stopped at time $t=10$ sec, which are very regular results since represented the actual robot action. Such values have been evaluated according to the torque equations and assuming overall robot mass of $m=50$ Kg.

B. Draw Sign (∞)

In this subsection the results of the implementation of the proposed four-wheeled when it draws a sign (∞) will be offered, figure (4.9) showed the motion of the robot during the drawing.

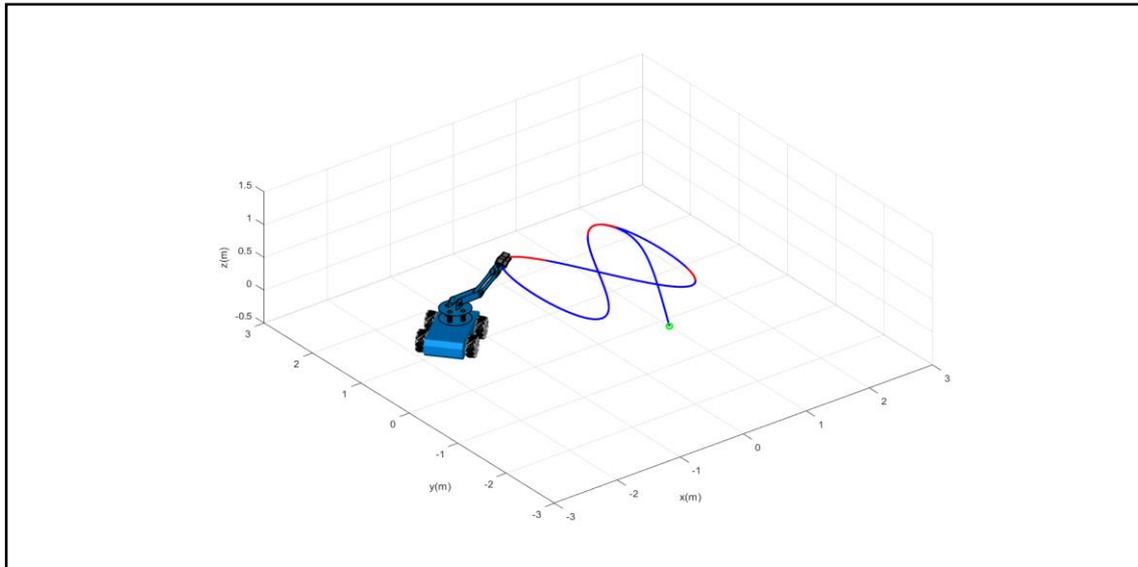


Fig 4.9: The motion of the robot during the draw of sign (∞).

The above figure shows that the robot draw the sign in a simple form, where the red track shows the chosen sign and the blue track shows the sign made by the robot. Also, it can be seen that there is notice a mismatch between the lines in the beginning because of the initial place of the robot.

Furthermore, the robot trajectory error signals results of the mobile robot during the draw of sign (∞) presented in figure (4.10).

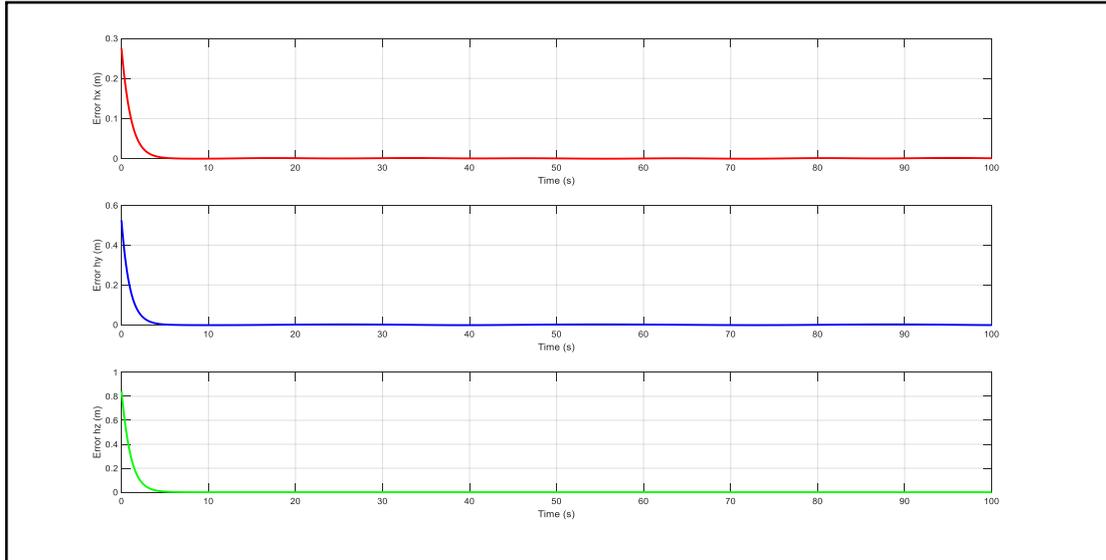


Fig4.10: The error signals motion of the robot during the draw of sign (∞).

The suggested robot arm motion error signals are shown in x, y, and z coordinates in figure (4.10) The previous Figure makes it evident that the error signals initially had low values at the commencement of the robot activity and quickly decreased to zero values in a very short period of time. Actually, within less than 5 seconds of beginning the 100-second operation period, the error signals are minimized to zero values. The reason of not arrive the signals to zero in starting is the initial position of the mobile robot.

Now, the motion control speeds U_f , U_l , and ω of the moving robot car and the response of the robot arm joint angles for separate action control q_1 , q_2 , and q_3 have been extracted and demonstrated in figure (4.11).

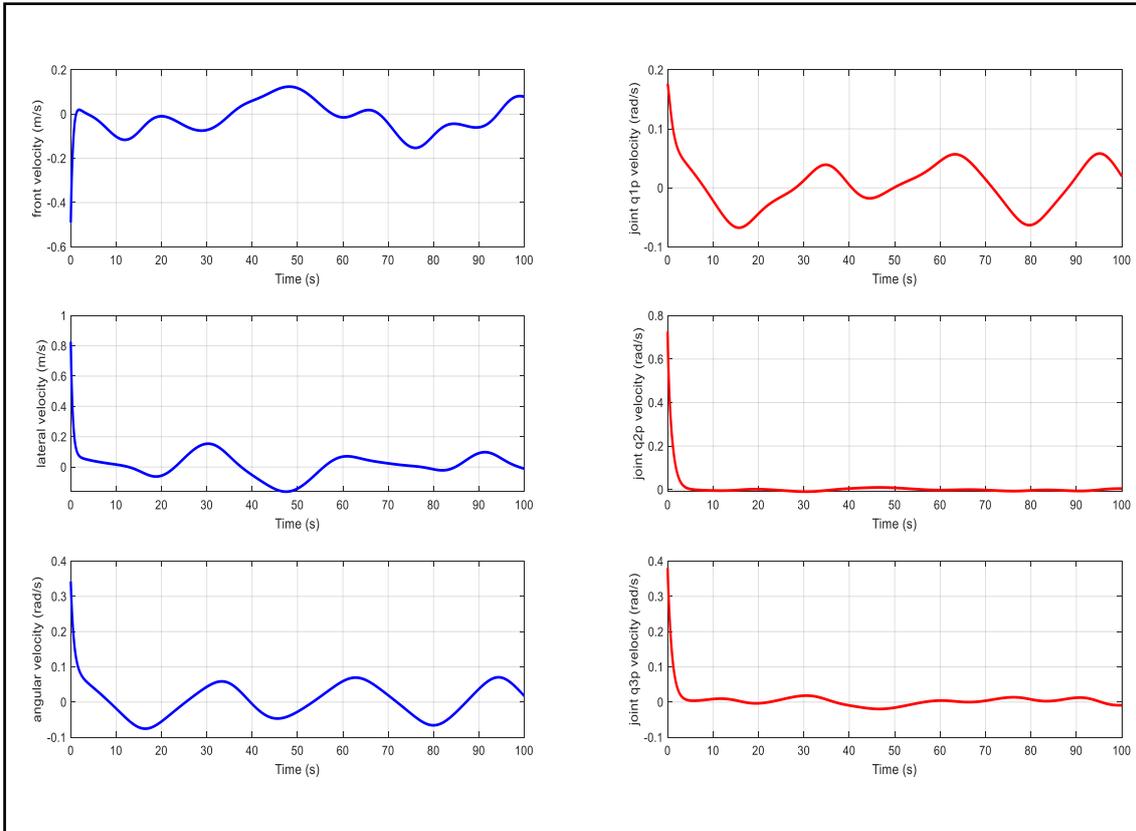


Fig 4.11: The motion control speeds U_f , U_l , and ω of the moving robot car and the response of the robot arm joint angles for separate action control during the draw of sign (∞).

It can be seen from the signals given in figure (4.11) that the motion control velocities start at different values at time $t = 0$. These signals are then repeated at $t = 65$ s due to the continuity of motion of the car. Also, the above figure proves that the robot upon more on the joint q1 during the draw of sign (∞). While the joints q2 and q3 are starting at various amounts at time $t=0$, then reduced with different features for every one until approached to zero at time instant $t=5$ sec along the total motion time $t=100$ sec

C. Sine Wave

In this subsection the outcomes of the operation of the proposed four-wheeled when it draws a sin wave will be offered, figure (4.12) showed the motion of the robot during the drawing.

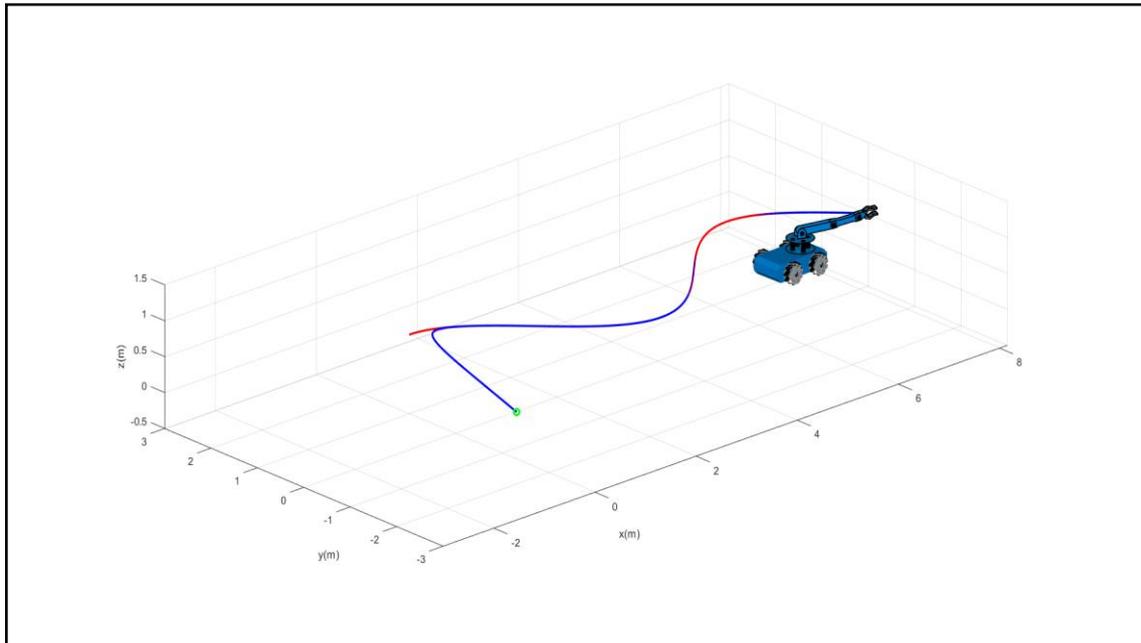


Fig 4.12: The motion of the robot during the draw of sin wave.

The above figure shows that the robot draw the sin wave in a simple form, where the red track shows the chosen sign and the blue track shows the sign made by the robot. Also, it can be seen that there is notice a mismatch between the lines in the beginning because of the initial place of the robot.

Moreover, the robot trajectory error signals results of the mobile robot during the draw of sin wave presented in figure (4.13).

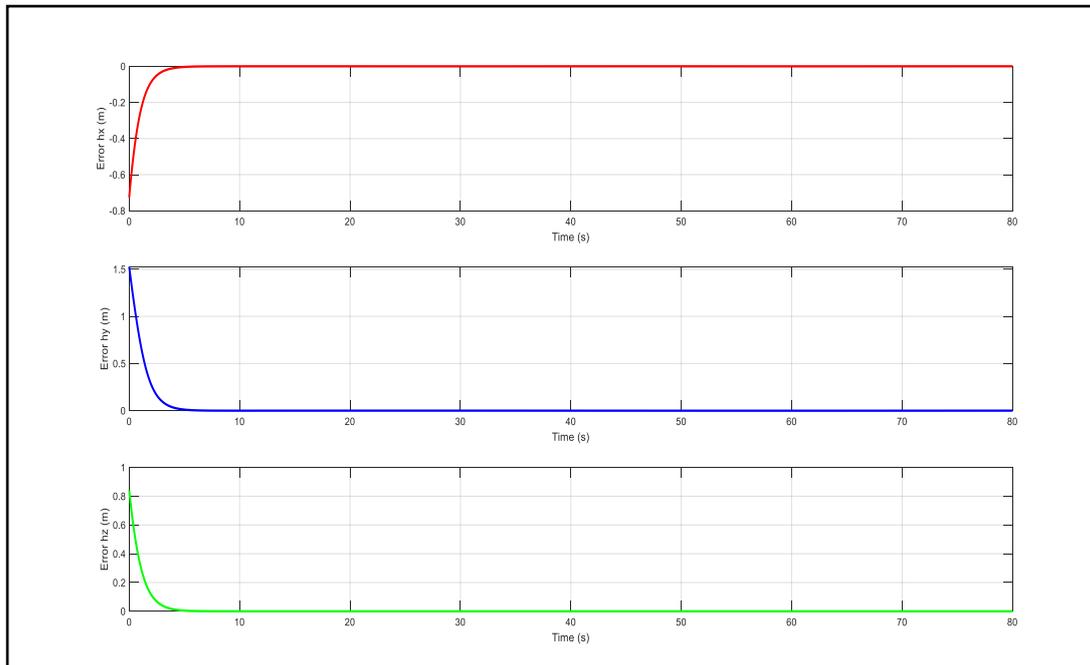


Fig 4.13: The error signals motion of the robot during the draw of sin wave.

Figure (4.13) displays the suggested robot arm motion error signals in x, y, and z axes. According to the previous Figure, the error signals started off with low values at the start of the robot activity and swiftly dropped to zero values in a very short amount of time. Actually, the error signals are minimized to zero values fewer than 5 seconds after the 100-second operation period starts. The initial position of the mobile robot is what causes the signals to not arrive at zero in starting.

Figure (4.14) now shows the motion control speeds U_f , U_l , and of the driving robot car as well as the reaction of the robot arm joint angles for each individual action control q_1 , q_2 , and q_3 .

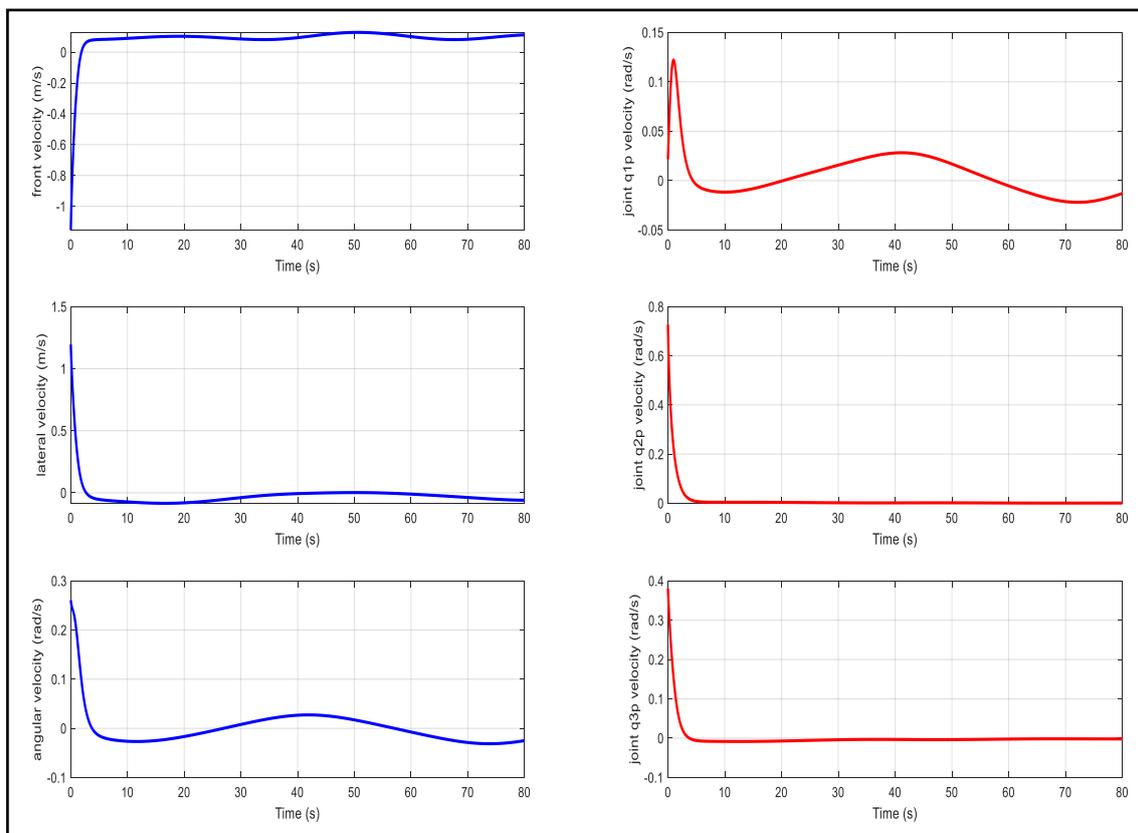


Fig 4.14: The motion control speeds U_f , U_l , and ω of the moving robot car and the response of the robot arm joint angles for separate action control during the draw of sin wave.

The signals shown in figure (4.14) demonstrate that the motion control velocities start at various values at time $t = 0$. Then, because of the car's continuous travel, these signals are repeated at $t = 65$ s. Additionally, the aforementioned image demonstrates that the robot relies more on joint q1 when drawing the symbol (). While the joints q2 and q3 are first reduced with separate characteristics for each other till they reach zero at time instant $t=5$ sec along the whole motion time $t=100$ sec.

Finally, in the case of drawing the sin wave, it will be added an additional figure, which is the torques of the car. these signals are displayed in figure (4.15).

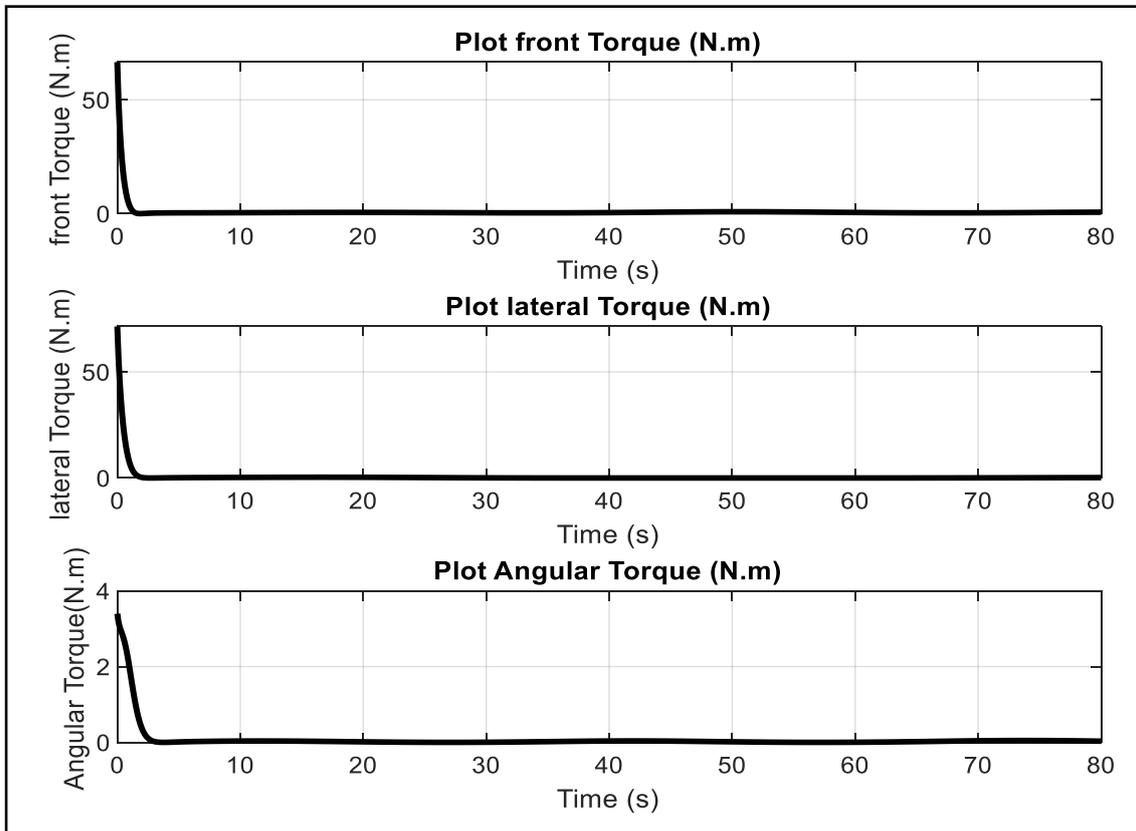


Fig 4.15: The torque signals values of the robot motion during the draw of sin wave.

The above figure (4.15) proves that the car does not exert strong torque in drawing functions. Where the drawing process requires accuracy more than strength, and the greatest effort falls on the joints of the arm, especially the first joint.

D. Coordinate Error

In this section, it will be offered the motion, errors, car velocities, and joints velocity when the final Coordinates are outside the limitations (3,-3,1.5). Figure (4.16) pictures the motion of the robot at (4,-4,2).

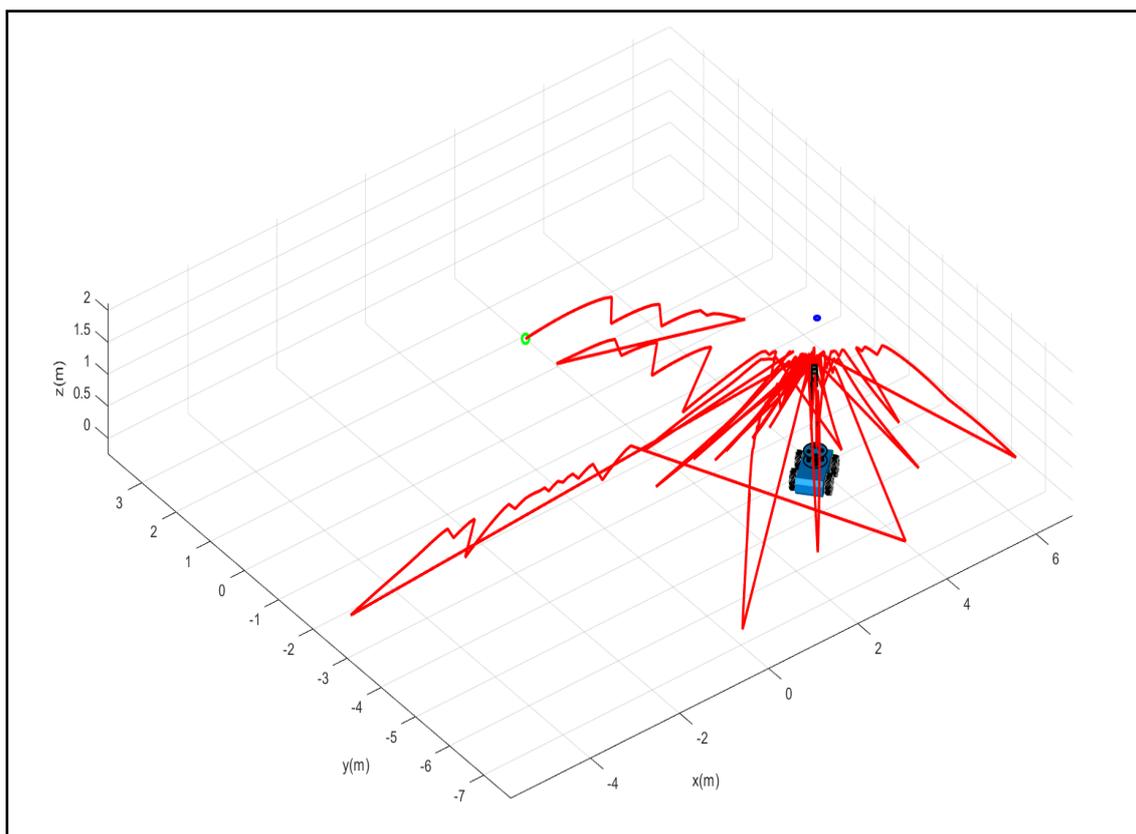


Fig 4.16: The implementation of the proposed four-wheeled mobile robot with arm system with rotation in three dimensional viewing (4,-4,2).

The above figure shows that the robot motion on the three dimensional in a randomly form. In other words, the robot cannot set the final destination to be reached. The reason for this is that the coordinates are outside the range programmed to the robot.

Now, , the robot trajectory error signals results of the mobile robot during the motion at (4,-4,2) presented in figure (4.17).

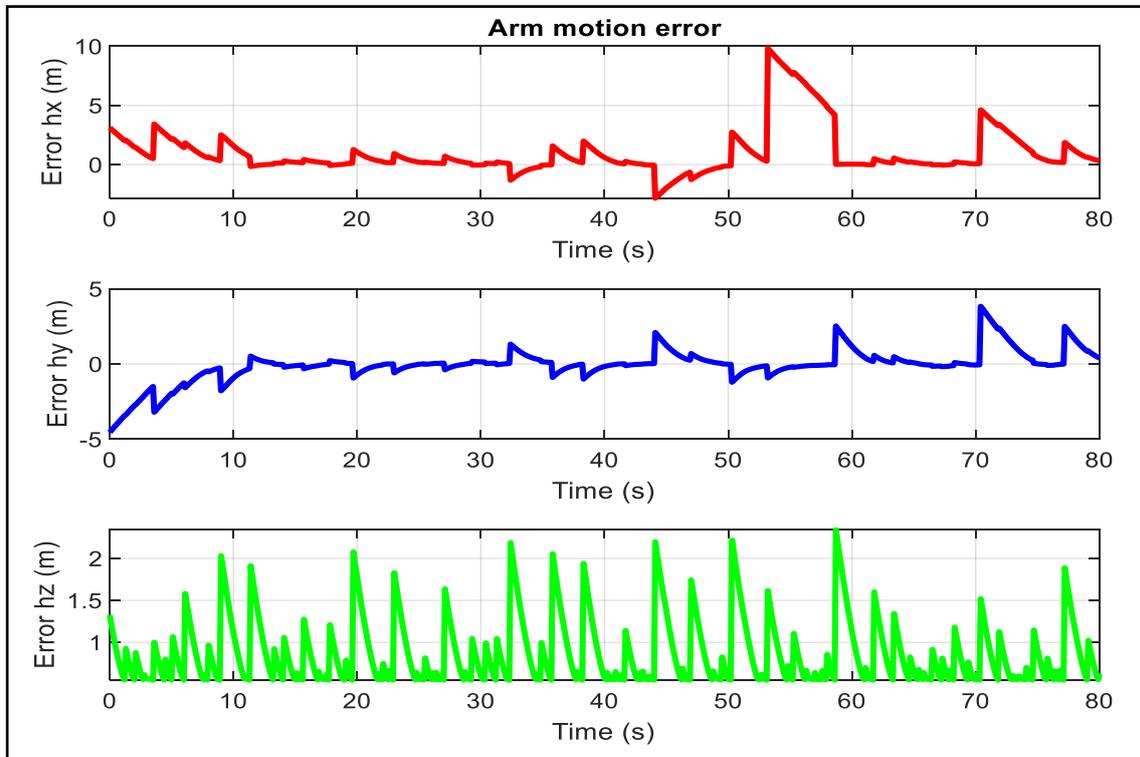


Fig 4.17: The error signals motion of the robot during the motion at (4,-4,2).

The above figure is self-explanatory, as it can be seen that the error rate on each axis is high. In other words, the error rate in each axis touches zero at times, then returns and rises at other times, and this indicates the randomness that the robot suffers from during its movement.

Moreover, the motion control speeds U_f , U_l , and ω , of the moving robot car at (4,-4,2) and displayed in figure (4.18).

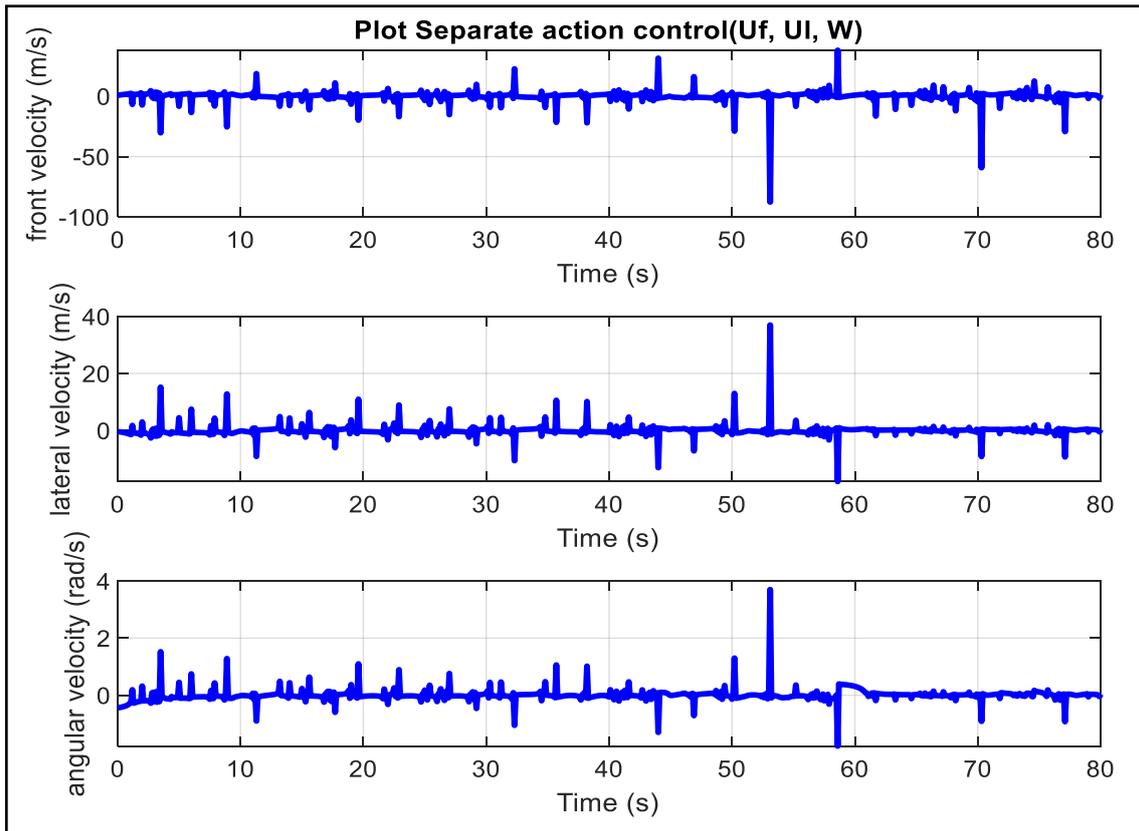


Fig 4.18: The motion control speeds U_f , U_l , and ω , of the moving robot car at $(4,-4,2)$.

The simplest thing that can be seen from the above figure is that the car makes a random movement at a random speed because it is unable to determine the final destination. Also, it can display the velocities of joints that make the arm, which is shown in figure (4.19). this figure also proves that the joints suffer from randomness because of the error coordinates.

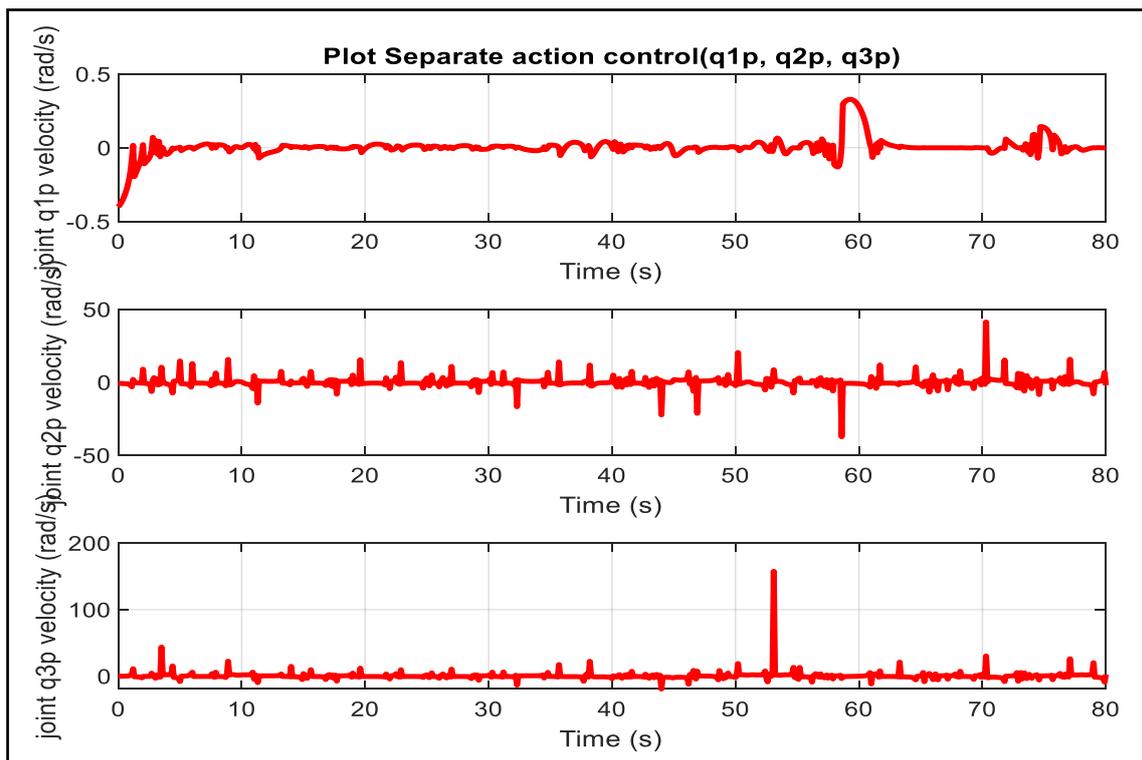


Fig 4.19: The robot arm joint angles for separate action control q_1 , q_2 , and q_3 at (4,-4,2)



Chapter Five

Prototype

Implementation

CHAPTER FIVE

PROTOTYPE IMPLEMENTATION

5.1 Introduction

The Omnidirectional Mobile Manipulator Robot with an arm is a remarkable robotic system that combines the unique capabilities of omnidirectional mobility and robotic manipulation into a single, versatile platform. This advanced robot represents a groundbreaking fusion of mobility and dexterity, providing solutions to a wide array of complex tasks and applications.

At its core, this robot boasts omnidirectional mobility, a remarkable feature that sets it apart from conventional robots. Equipped with specialized wheels or holonomic drive systems, it has the extraordinary ability to move in any direction effortlessly. This includes gliding forward, backward, sideways, and performing precise rotations without cumbersome turns. This level of agility and maneuverability is a game-changer, allowing the robot to operate effectively in confined and intricate spaces.

What makes this robot exceptional is its robotic arm. This arm is equipped end-effector and 6degree of freedom, granting the robot the capacity to interact with its environment and execute a multitude of tasks with remarkable precision and adaptability. Whether it's object manipulation, or other intricate operations, the robotic arm empowers the robot to tackle a wide variety of challenges.

5.2 Robot Construction

Creating a fabrication robot using a 3D printer is a fascinating project that combines the advantages of additive manufacturing with robotics. Such a robot can be designed to build, assemble, or maintain structures autonomously, making it a valuable asset in various industries. Here's an overview of how to fabricate a robot using a 3D printer:

5.2.1 Design of the Robot

Start by designing the robot's structure and components. This involves creating a 3D model of the robot using computer-aided design software.

Use a 3D printer to fabricate the individual components of the robot. Depending on the complexity of the design, this may involve printing multiple parts, such as the chassis, arms, joints, and specialized components .

Ensure that the design is optimized for 3D printing, taking into account structural integrity and assembly. as shown in figure (5.1).

5.2.2 3D Printer

Selecting the "best" 3D printer for robot parts can depend on your specific requirements, such as budget, materials, and precision. However, a 3D printer that is widely recognized for its quality and versatility is the CREALITY K1 (or its newer versions like the K1 MAX).

Reasons for choosing this model(Print Quality,Autocalibration etc)

Remember that the "best" 3D printer can vary depending on your specific needs and budget. It's essential to consider factors like the size of your prints, the type

of materials you intend to use, the level of precision required, and any specific features important for your robot project.

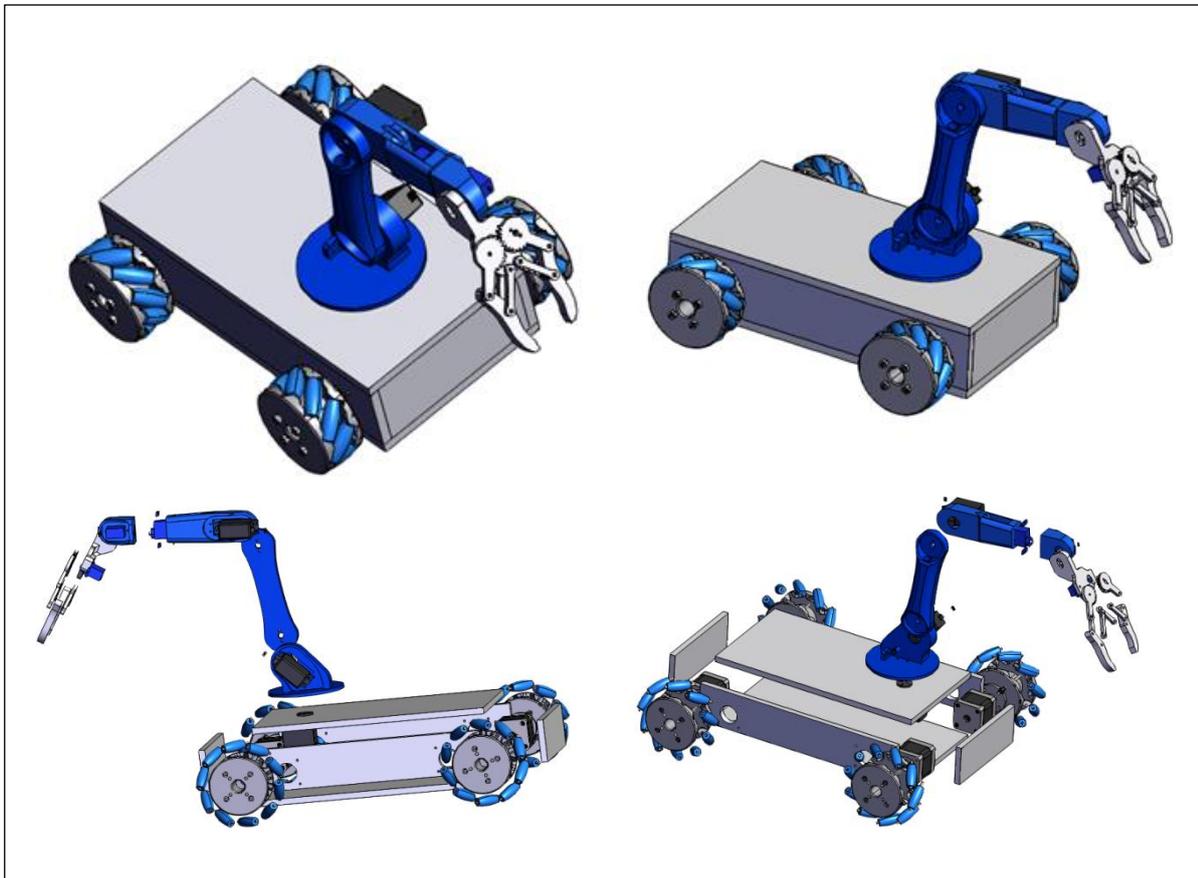


Fig 5.1: Omni-directional mobile robot with a moving arm.

5.2.3 3D Printer Materials

The choice of appropriate 3D printing materials is based on your robot's requirements as durability, weight, and temperature resistance . Common options include PLA, ABS, PETG, and nylon.

The best one for print the robot part is PLA as is easy to work with, environmentally friendly, and has low printing temperatures

5.2.4 Assembly

After all the 3D-printed parts are ready as shown in figure (5.2), the assembly process begins. need additional hardware, such as screws, nuts, bolts, motors, and a control unit.

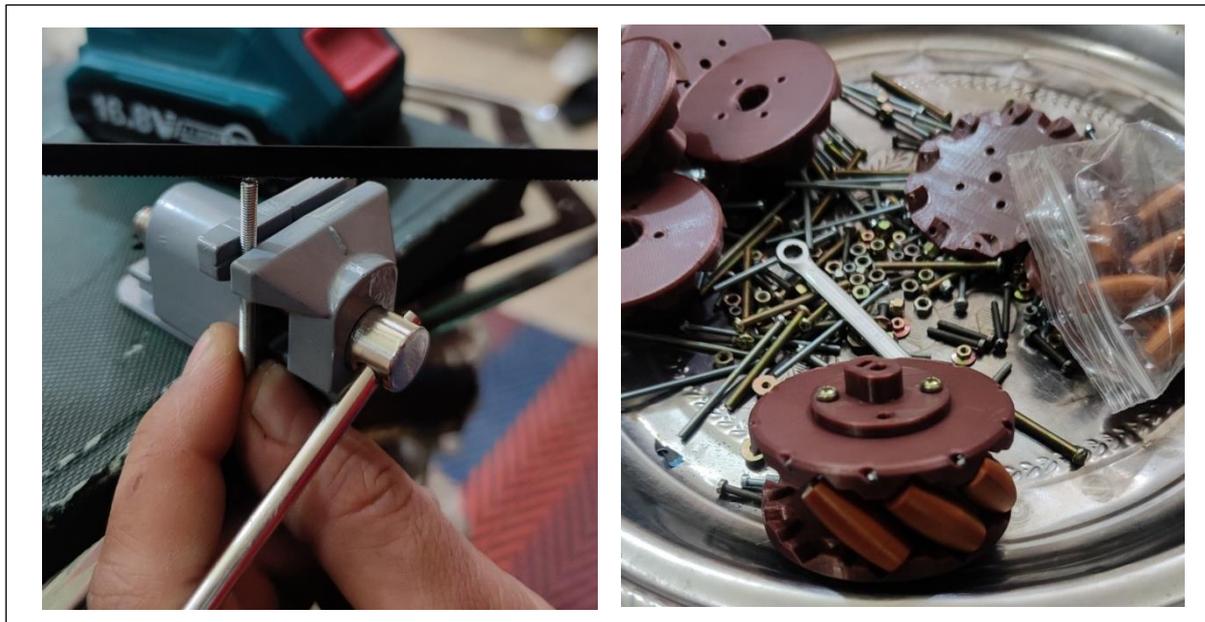


Fig 5.2 : Robot parts during assembly.

The figure (5.3), represents the robotic arm mounted on the chassis. The chassis is supported for the displacement of the robotic arm by four Omni wheels. The robotic arm implemented has six degrees of freedom.

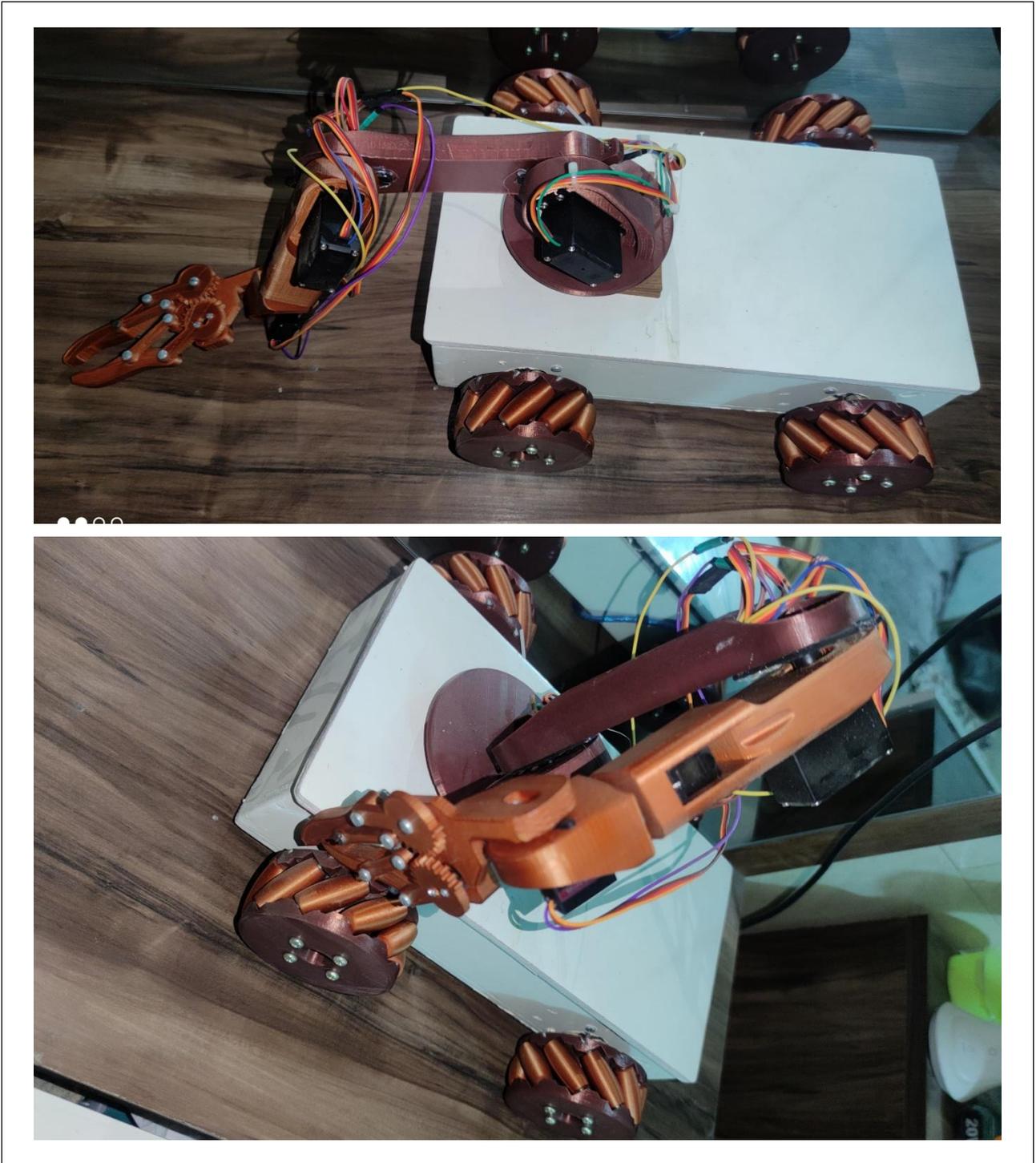


Fig 5.3 : Implemented project model.

5.3 Robot Components

Selecting components for a robot depends on the robot's purpose and design. Different types of robots may require different components. Here's a list of components used :

5.3.1 Microcontroller

Using an Arduino Mega for a robot is a common choice, especially since the robot project requires a lot of I/O pins, memory, and processing power as shown in the figure (5.4). The Arduino Mega is one of the more powerful and versatile boards in the Arduino family due to its numerous digital and analog I/O pins, larger flash memory, and greater computational capability. Here are some considerations when using an Arduino Mega for a robot:

- **I/O Pins:** The Arduino Mega has a significant number of digital and analog pins (54 digital and 16 analog). This is especially useful if your robot project involves multiple sensors, motors, or other peripherals.
- **Processing Power:** The Arduino Mega typically requires a voltage supply between 7V and 12V, and you can power it via USB, an external power supply, or batteries, following the specified voltage and current requirements in the board's documentation.
- **Memory:** The Mega offers 256KB of flash memory and 8KB of SRAM, which is considerably more than the smaller Arduino boards. This extra memory is helpful when your robot project requires storing and processing a lot of data.

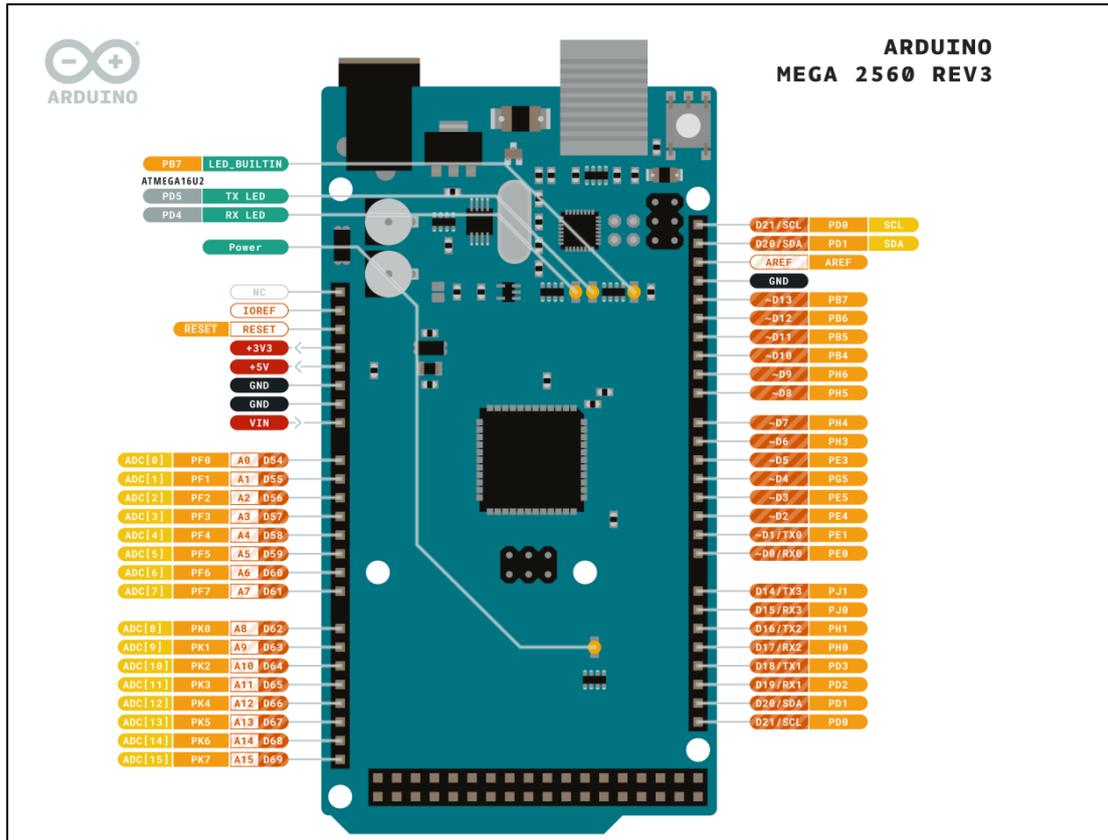


Fig 5.4 : Arduino Mega 2560 Rev3.

5.3.2 Motors

Robot motors are the key components responsible for controlling motion and movement in robots. These motors provide the mechanical power needed for the robot to perform tasks and navigate its environment. There are various types of motors used in robotics, including:

- **Servo Motors:** Servo motors are known for their precision and feedback control. They are often used in robotic arms and mechanisms where accurate positioning and control are essential.
- **Stepper Motors:** Stepper motors move in discrete steps, making them suitable for applications that require precise positioning without the need for feedback systems.

The choice of motor for a robot depends on factors like the robot's intended application, the required precision, speed, torque, and budget. Different types of motors are suitable for different tasks within a robot, such as driving wheels, moving arms, or controlling various mechanisms.

Both types of motors were chosen depending on the type of movement, as shown in the figure(5.5).

The most important characteristics of motors are shown in the table (5.1).

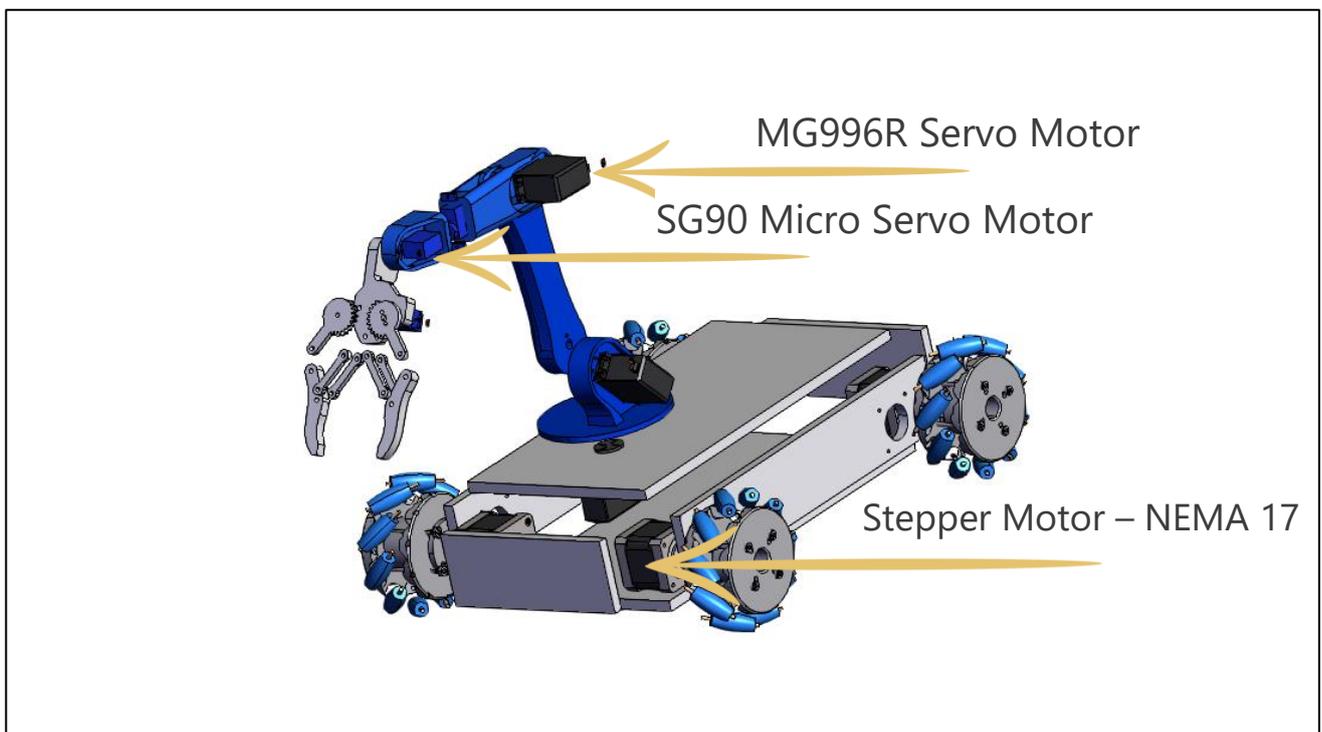


Fig 5.5 : The motors used in an Omni-directional mobile robot.

Table 5.1 : Characteristics of Motors.

Property	NEMA 17	MG996R	SG90
Type	Stepper Motor	Analog Servo Motor	Analog Micro Servo
Size	NEMA 17 (42mm x 42mm)	Standard-sized servo	Mini-sized servo
Step Angle	Typically 1.8 degrees	—————	—————
Torque	Variable, 0.4-0.6 Nm	High, 10-15 kg/cm	Approx. 2.5 kg/cm
Speed	Variable	Moderate to high	Moderate
Voltage	2.8V to 4.2V	4.8V to 6.0V	4.8V to 6.0V
Gears	—————	Metal gears	Plastic gears
Control	Stepper Driver	PWM Control	PWM Control
Feedback	No feedback	Typically analog, may require external feedback	Typically analog, may require external feedback
Use Cases	3D printers, CNC, robotics	Various robotics, automation	Micro-robotics, RC models

5.3.3 Electronics Assembly

Electronic assembly is a critical phase in the construction of robots. It involves carefully integrating various electronic components into the robot's structure to ensure it functions as intended as the shown in figure (5.6).

At the heart of the assembly is the microcontroller, an Arduino or Raspberry Pi, which serves as the robot's "brain."

Motor drivers or control boards are introduced to manage the robot's movements, connecting to motors that drive wheels or joints. Power is supplied through batteries or power sources, with voltage regulation and distribution considered to keep everything running smoothly.

Wiring, connectors, and cables are laid out with precision, ensuring that communication is seamless between components As shown in the figure(5.7) .

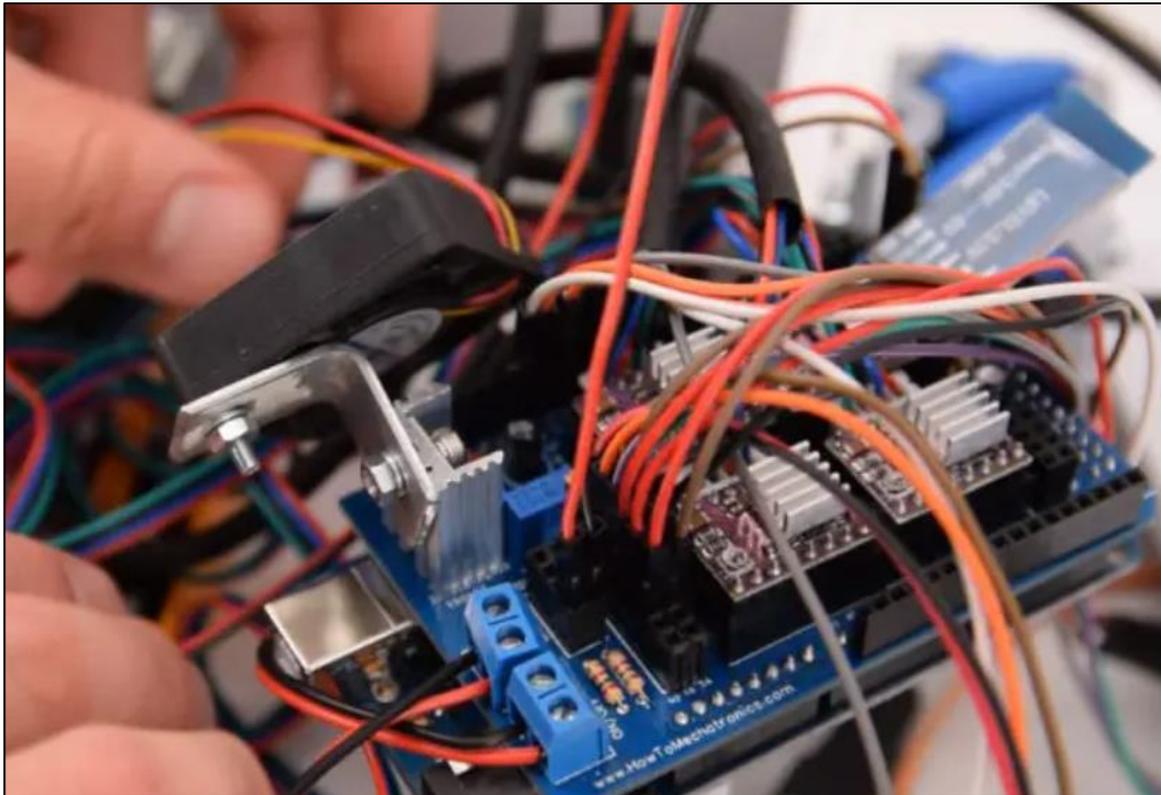


Fig 5.6 : Assembling electronic components.

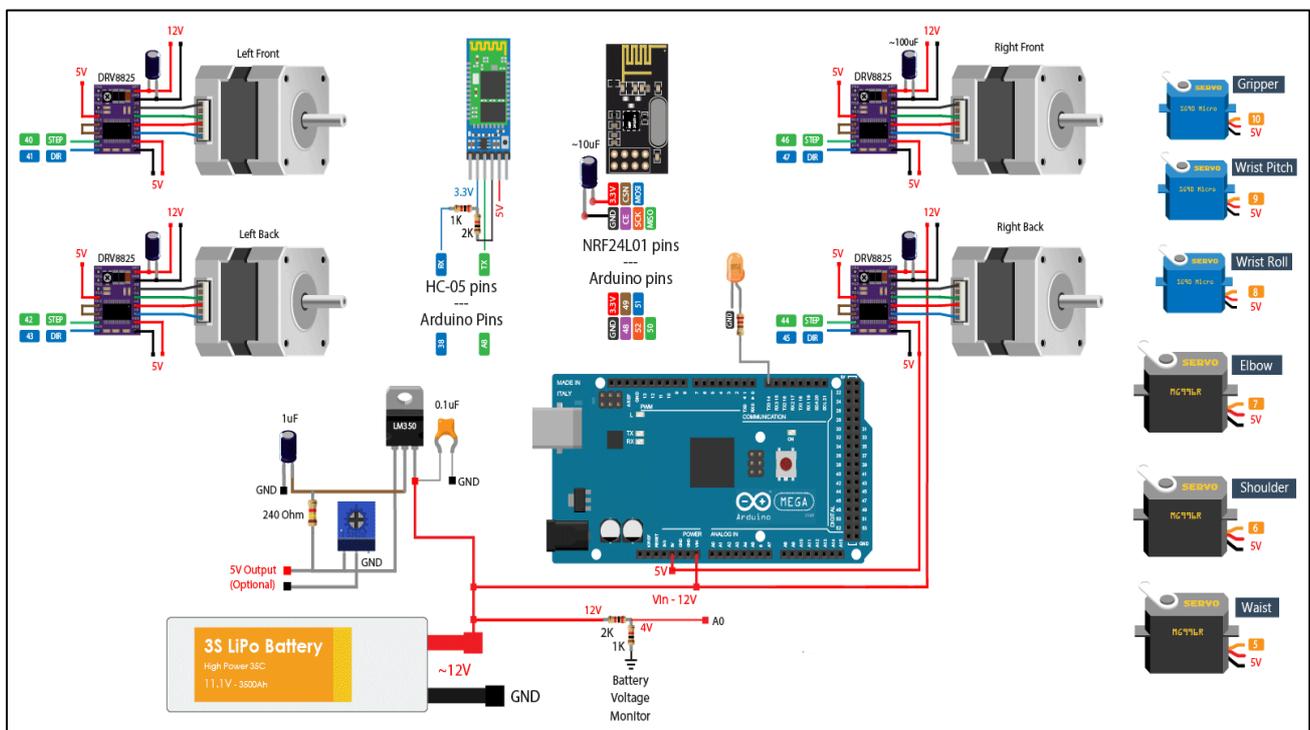


Fig 5.7 : Electrical wiring diagram.

5.4 Programming

Write and upload the software that controls the robot's behavior. This may involve coding for tasks like navigation .

5.4.1 Arduino programming

Programming an Arduino to control a robot via a mobile app is a captivating fusion of hardware and software. This endeavor involves the creation of code for both the Arduino and the mobile app to enable seamless communication and remote operation. On the Arduino side, you craft code to receive commands from the mobile app, interpreting them as specific actions like forward motion, turning, or stopping. Safety features, such as emergency stops, are integrated to ensure the robot's well-being.

Simultaneously, delve into mobile app development, designing an interface with buttons or controls that convey your intentions to the robot. The app's code establishes a connection with the Arduino via Bluetooth, transmitting commands swiftly and efficiently. This opens a world of possibilities, from guiding your robot through explorations to executing tasks from a distance.

The code can be expressed as a diagram in the figure(5.8).

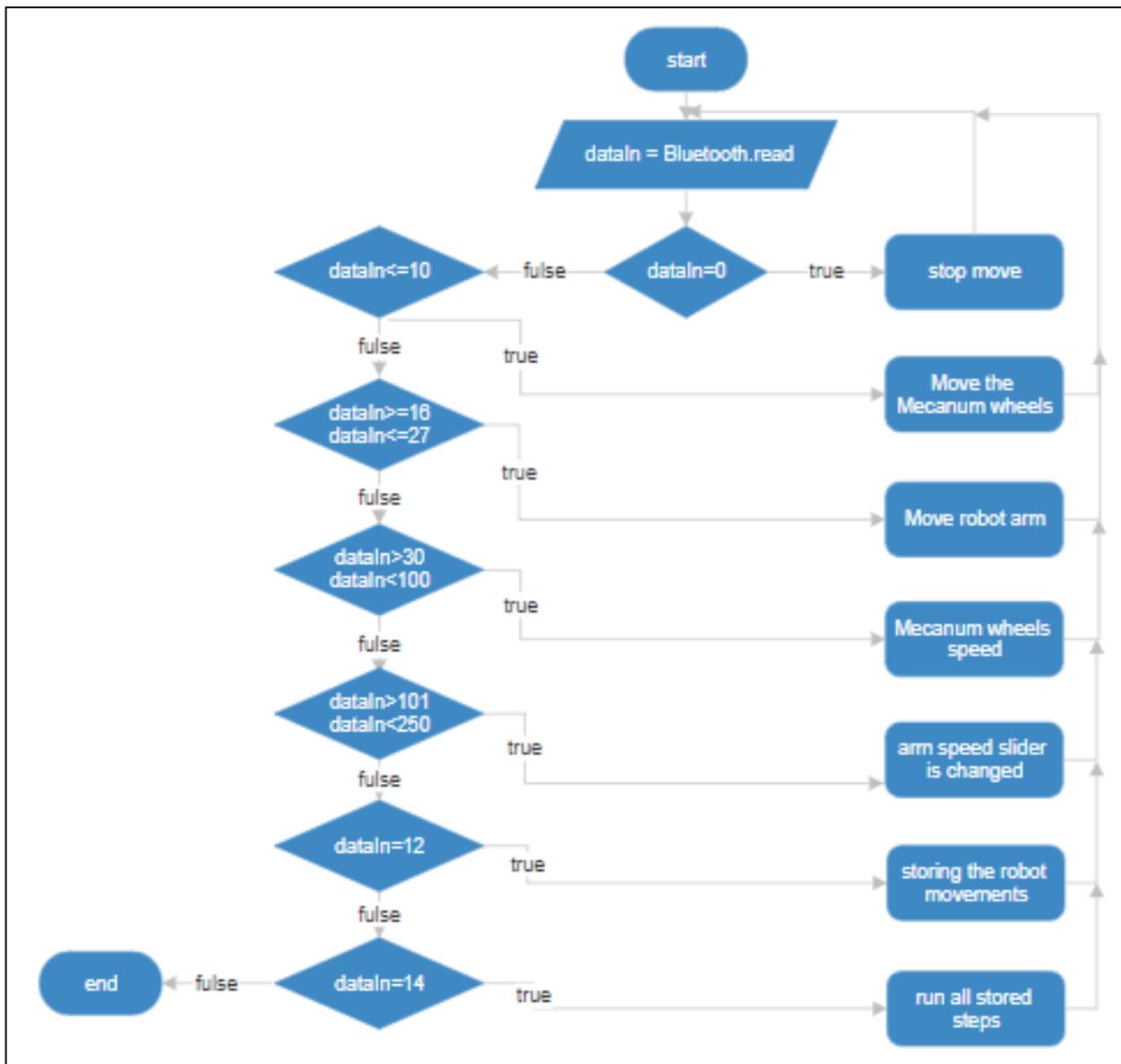


Fig 5.8 : Flowchart of Control System .

5.4.2 Phone application

Creating a complete app for controlling a robot with omnidirectional wheels and a robot arm in MIT App Inventor is a complex project that involves both hardware and software components. requires a significant amount of time and effort. maybe provide with a basic outline of the steps involved.

Here are the general steps:

- **Design the User Interface:** Use MIT App Inventor to design the user interface of the app. can create buttons, sliders, or other UI elements to control the robot's movement and arm.

as shown in the figure (5.9).

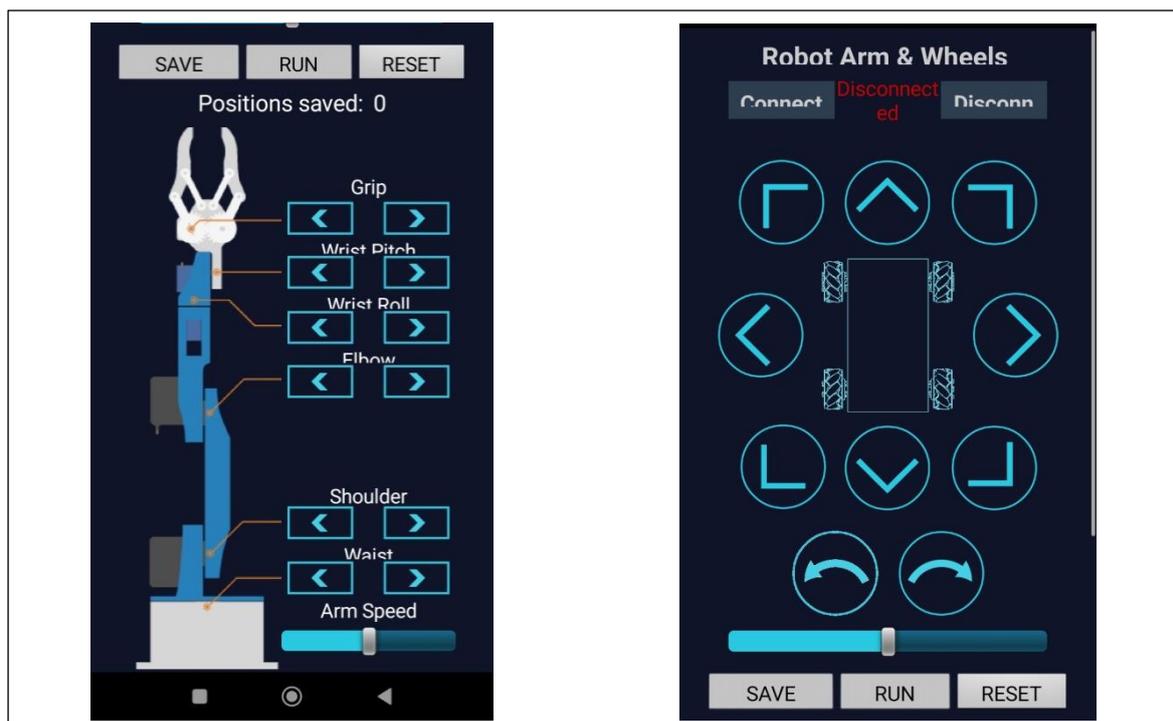


Fig 5.9 : user interface of the application.

- **Connect to Hardware:** Depending on the robot's hardware, need to establish a connection between the app and the robot. This might involve Bluetooth, Wi-Fi, or other communication methods. MIT App Inventor has components for Bluetooth and Wi-Fi communication.
- **Control the Omnidirectional Wheels:** Implement the logic to control the robot's omnidirectional wheels. send commands from mobile (forward, backward, left, right, etc.) to the robot bas on the user's input through the app.

Figure (5.10),(5.11) indicates the operating process of the Omnidirectional Wheels.

```

when LeftForward . Click
do
  if LeftForward . Image = LeftForward_0.png
  then
    set LeftForward . Image to LeftForward_1.png
    set Left . Image to Left_0.png
    set Right . Image to Right_0.png
    set Backward . Image to Backward_0.png
    set Forward . Image to Forward_0.png
    set RightForward . Image to RightForward_0.png
    set LeftBackward . Image to LeftBackward_0.png
    set RightBackward . Image to RightBackward_0.png
    set RotateLeft . Image to RotateLeft_0.png
    set RotateRight . Image to RotateRight_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 1
  else
    set LeftForward . Image to LeftForward_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 0
  
```

f) Left Forward

```

when RightForward . Click
do
  if RightForward . Image = RightForward_0.png
  then
    set RightForward . Image to RightForward_1.png
    set Left . Image to Left_0.png
    set Right . Image to Right_0.png
    set Backward . Image to Backward_0.png
    set LeftForward . Image to LeftForward_0.png
    set Forward . Image to Forward_0.png
    set LeftBackward . Image to LeftBackward_0.png
    set RightBackward . Image to RightBackward_0.png
    set RotateLeft . Image to RotateLeft_0.png
    set RotateRight . Image to RotateRight_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 3
  else
    set RightForward . Image to RightForward_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 0
  
```

e) Right Forward

```

when Forward . Click
do
  if Forward . Image = Forward_0.png
  then
    set Forward . Image to Forward_1.png
    set Left . Image to Left_0.png
    set Right . Image to Right_0.png
    set Backward . Image to Backward_0.png
    set LeftForward . Image to LeftForward_0.png
    set RightForward . Image to RightForward_0.png
    set LeftBackward . Image to LeftBackward_0.png
    set RightBackward . Image to RightBackward_0.png
    set RotateLeft . Image to RotateLeft_0.png
    set RotateRight . Image to RotateRight_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 2
  else
    set Forward . Image to Forward_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 0
  
```

d) Forward

```

when Backward . Click
do
  if Backward . Image = Backward_0.png
  then
    set Backward . Image to Backward_1.png
    set LeftForward . Image to LeftForward_0.png
    set Right . Image to Right_0.png
    set LeftBackward . Image to LeftBackward_0.png
    set Forward . Image to Forward_0.png
    set RightForward . Image to RightForward_0.png
    set Left . Image to Left_0.png
    set RightBackward . Image to RightBackward_0.png
    set RotateLeft . Image to RotateLeft_0.png
    set RotateRight . Image to RotateRight_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 7
  else
    set Backward . Image to Backward_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 0
  
```

c) Backward

```

when LeftBackward . Click
do
  if LeftBackward . Image = LeftBackward_0.png
  then
    set LeftBackward . Image to LeftBackward_1.png
    set LeftForward . Image to LeftForward_0.png
    set Right . Image to Right_0.png
    set Backward . Image to Backward_0.png
    set Forward . Image to Forward_0.png
    set RightForward . Image to RightForward_0.png
    set Left . Image to Left_0.png
    set RightBackward . Image to RightBackward_0.png
    set RotateLeft . Image to RotateLeft_0.png
    set RotateRight . Image to RotateRight_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 6
  else
    set LeftBackward . Image to LeftBackward_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 0
  
```

b) Left Backward

```

when RightBackward . Click
do
  if RightBackward . Image = RightBackward_0.png
  then
    set RightBackward . Image to RightBackward_1.png
    set LeftForward . Image to LeftForward_0.png
    set Right . Image to Right_0.png
    set LeftBackward . Image to LeftBackward_0.png
    set Forward . Image to Forward_0.png
    set RightForward . Image to RightForward_0.png
    set Left . Image to Left_0.png
    set Backward . Image to Backward_0.png
    set RotateLeft . Image to RotateLeft_0.png
    set RotateRight . Image to RotateRight_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 8
  else
    set RightBackward . Image to RightBackward_0.png
    call BluetoothClient1 . Send1ByteNumber
    number 0
  
```

a) Right Backward

Fig5.10 : MIT APP Inventor programming block (for forward and backward movement) .

```

when Right Click
do
  if Right Image = Right_0.png
  then
    set Right Image to Right_1.png
    set LeftForward Image to LeftForward_0.png
    set Left Image to Left_0.png
    set Backward Image to Backward_0.png
    set Forward Image to Forward_0.png
    set RightForward Image to RightForward_0.png
    set LeftBackward Image to LeftBackward_0.png
    set RightBackward Image to RightBackward_0.png
    set RotateLeft Image to RotateLeft_0.png
    set RotateRight Image to RotateRight_0.png
    call BluetoothClient1 Send1ByteNumber
    number 5
  else
    set Right Image to Right_0.png
    call BluetoothClient1 Send1ByteNumber
    number 0
  
```

a) Right

```

when Left Click
do
  if Left Image = Left_0.png
  then
    set Left Image to Left_1.png
    set LeftForward Image to LeftForward_0.png
    set Right Image to Right_0.png
    set Backward Image to Backward_0.png
    set Forward Image to Forward_0.png
    set RightForward Image to RightForward_0.png
    set LeftBackward Image to LeftBackward_0.png
    set RightBackward Image to RightBackward_0.png
    set RotateLeft Image to RotateLeft_0.png
    set RotateRight Image to RotateRight_0.png
    call BluetoothClient1 Send1ByteNumber
    number 4
  else
    set Left Image to Left_0.png
    call BluetoothClient1 Send1ByteNumber
    number 0
  
```

b)Left

```

when RotateLeft Click
do
  if RotateLeft Image = RotateLeft_0.png
  then
    set RotateLeft Image to RotateLeft_1.png
    set Left Image to Left_0.png
    set Right Image to Right_0.png
    set Backward Image to Backward_0.png
    set LeftForward Image to LeftForward_0.png
    set Forward Image to Forward_0.png
    set LeftBackward Image to LeftBackward_0.png
    set RightBackward Image to RightBackward_0.png
    set RotateRight Image to RotateRight_0.png
    set RightForward Image to RightForward_0.png
    call BluetoothClient1 Send1ByteNumber
    number 9
  else
    set RotateLeft Image to RotateLeft_0.png
    call BluetoothClient1 Send1ByteNumber
    number 0
  
```

c)Rotate Left

```

when RotateRight Click
do
  if RotateRight Image = RotateRight_0.png
  then
    set RotateRight Image to RotateRight_1.png
    set Left Image to Left_0.png
    set Right Image to Right_0.png
    set Backward Image to Backward_0.png
    set LeftForward Image to LeftForward_0.png
    set Forward Image to Forward_0.png
    set LeftBackward Image to LeftBackward_0.png
    set RightBackward Image to RightBackward_0.png
    set RotateLeft Image to RotateLeft_0.png
    set RightForward Image to RightForward_0.png
    call BluetoothClient1 Send1ByteNumber
    number 10
  else
    set RotateRight Image to RotateRight_0.png
    call BluetoothClient1 Send1ByteNumber
    number 0
  
```

d)Rotate Right

```

when Speed_Platform PositionChanged
thumbPosition
do
  call BluetoothClient1 Send1ByteNumber
  number round get thumbPosition
  
```

f) Speed platform

Fig 5.11 : MIT APP Inventor programming block (for side and rotate movement).

- **Control the Robot Arm:** Create UI elements to control the robot arm movement. Depending on the robot arm design must send specific commands, angles, or positions to control it

Figure (5.12) indicates the operating process of the robot arm.

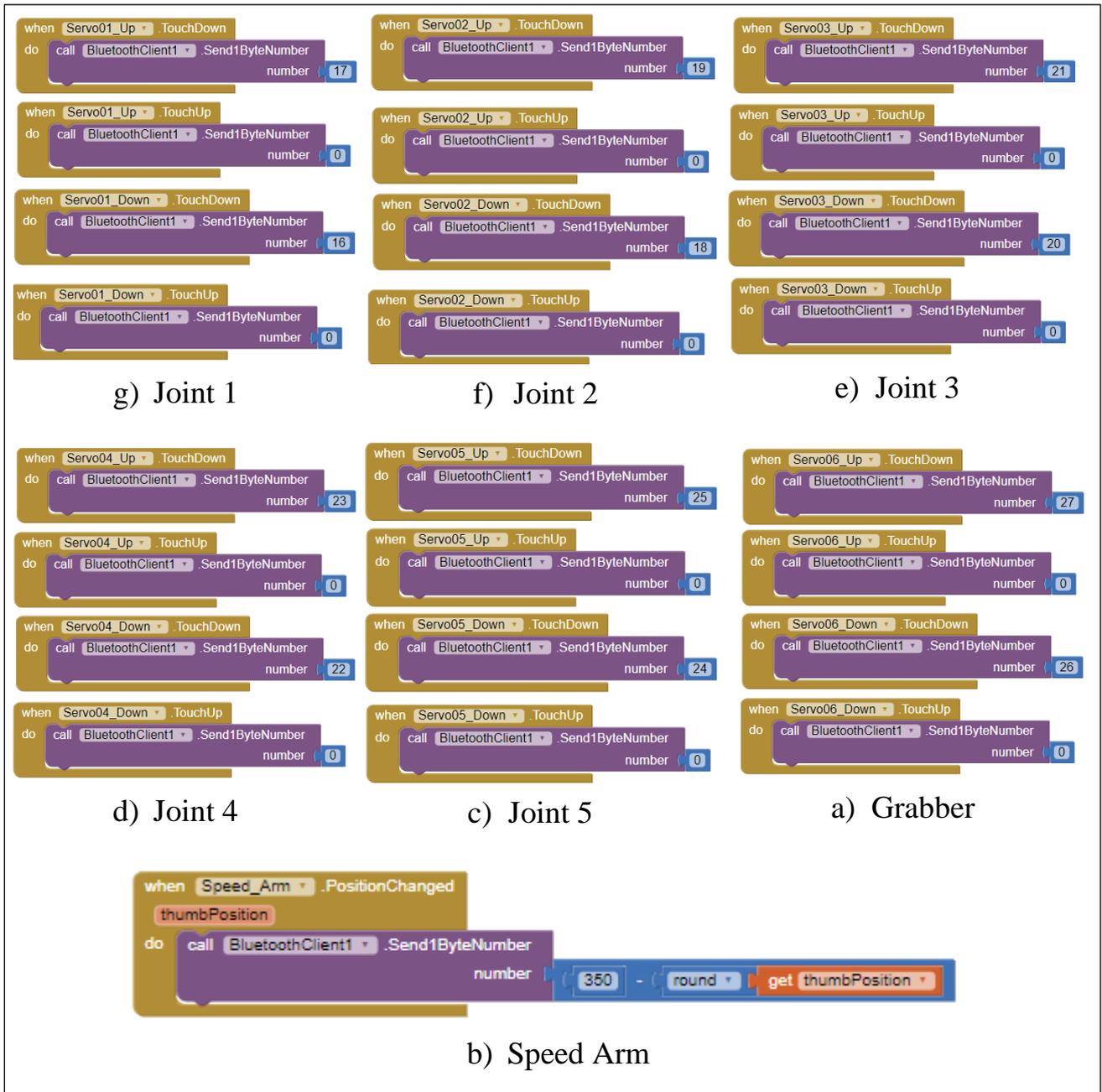


Fig 5.12 : MIT APP Inventor programming block (for Arm movement).



Chapter Six
Conclusion and
future research
directions

CHAPTER SIX

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

6.1 Conclusions

By observing the results of the robot motion control signals provided by the inverse kinematics mathematical equations of motion and the Jacobi matrix, we may conclude that the motion of the wheel is approximately constant for all four wheels as it must move at the same speed, direction and angle to be able to transmit torque to a fixed base and the body of the robot to move in a fully coordinated and synchronized manner. The movement of the robot cart is a uniform movement of all wheels, also, by looking at the movement of the robot arm shown in the results, we may conclude that the movements of the upper arm and forearm are identical and take the same specific path for the movement of the cart to reach the target point through the arm control signals. Referring to the obtained results, we can observe the control waves, the error, and the amount of movement angles for all the moving axes that receive transmission signals through the design equations in the control structures completely for all parts of the proposed system of the robotic arm vehicle. Through these readings, researchers can know the actual areas of influence in the proposed system and identify transmission axes with better accuracy and clarity.

By reviewing the simulation results obtained from implementing the proposed moving robot programming with arm structure, we may make important observations to discuss these results. The robot movement was successfully operated with a smooth and smooth movement, which indicates the high stability and smoothness of the movements provided by the effective counter motion equations that control the execution of the movement of the cart

and the arm towards the target point specified by the inputs. Further, by observing the consequences of the robot motion control signals that, we can assume that the motion of the wheel is practically constant for each of the four tires as they must move at similar speed, direction and angle uniformly to have the option to deliver force to the appropriate base and the body of the robot to move in an orderly manner And completely synchronized with the movement of the arm.

6.2 Future Research Directions

As suggestions for further research and study in this topic, it might be proposed to implement the suggested robotic system in cooperation with a supportive company for its practical implementation. It might also be proposed to include smart sensors in the robotic arm and body to estimate distances and displacement and translate them into speeds, moments and angles calculations and signals which are fed through control devices to the various transmissions and axes in the proposed system of the mobile robot.



References



-
- [1] T. L. Chien, H. Guo, K. L. Su, and S. V. Shiau, "Develop a Multiple Interface Based Fire Fighting Robot," in *2007 IEEE International Conference on Mechatronics*, May 2007, pp. 1–6. doi: 10.1109/ICMECH.2007.4280040.
- [2] S. Mandal, S. K. Saw, S. Maji, V. Das, S. K. Ramakuri, and S. Kumar, "Low cost arduino wifi bluetooth integrated path following robotic vehicle with wireless GUI remote control," in *2016 International Conference on Information Communication and Embedded Systems (ICICES)*, Feb. 2016, pp. 1–5. doi: 10.1109/ICICES.2016.7518916.
- [3] H. Rissanen, J. Mahonen, K. Haataja, M. Johansson, J. Mielikainen, and P. Toivanen, "Designing and implementing an intelligent Bluetooth-enabled robot car," in *2009 IFIP International Conference on Wireless and Optical Communications Networks*, Apr. 2009, pp. 1–6. doi: 10.1109/WOCN.2009.5010563.
- [4] Zhao Wang, Eng Gee Lim, Weiwei Wang, M. Leach, and Ka Lok Man, "Design of an arduino-based smart car," in *2014 International SoC Design Conference (ISOCC)*, Nov. 2014, pp. 175–176. doi: 10.1109/ISOCC.2014.7087683.
- [5] Zhenjun He, Jiang Zhang, Peng Xu, Jiaheng Qin, and Yunkai Zhu, "Mine detecting robot based on wireless communication with multi-sensor," in *2013 IEEE 4th International Conference on Electronics Information and Emergency Communication*, Nov. 2013, pp. 117–120. doi: 10.1109/ICEIEC.2013.6835467.
- [6] Yong Zhang, B. K. Chen, Xinyu Liu, and Yu Sun, "Autonomous Robotic Pick-and-Place of Microobjects," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 200–207, Feb. 2010, doi: 10.1109/TRO.2009.2034831.
- [7] S. Gómez *et al.*, "Design of a 4-DOF robot manipulator with optimized
-

-
- algorithm for inverse kinematics,” *Int. J. Mech. Mechatronics Eng.*, vol. 9, no. 6, pp. 929–934, 2015.
- [8] Panc. JhA, “Inverse kinematic analysis of robot manipulators.” 2015.
- [9] G. Bao, S. Liu, and H. Zhao, “Kinematics Simulation of 4 DOF Manipulator,” in *Proceedings of the 2017 2nd International Conference on Materials Science, Machinery and Energy Engineering (MSMEE 2017)*, 2017, pp. 400–408. doi: 10.2991/msmee-17.2017.81.
- [10] K. Belda and O. Rovny, “Predictive control of 5 DOF robot arm of autonomous mobile robotic system motion control employing mathematical model of the robot arm dynamics,” in *2017 21st International Conference on Process Control (PC)*, Jun. 2017, pp. 339–344. doi: 10.1109/PC.2017.7976237.
- [11] H. Khoswanto, K. N. Sugiharto, and I. N. Sandjaja, “Forward and Inverse Kinematic of a Manipulator Simulator Software Using Unity Engine.” Petra Christian University, 2018.
- [12] T. F. Abaas, A. A. Khleif, and M. Q. Abbood, “Inverse kinematics analysis and simulation of a 5 DOF robotic arm using MATLAB,” *Al-Khwarizmi Eng. J.*, vol. 16, no. 1, pp. 1–10, 2020.
- [13] D. U. Rijalusalam and I. Iswanto, “Implementation Kinematics Modeling and Odometry of Four Omni Wheel Mobile Robot on The Trajectory Planning and Motion Control Based Microcontroller,” *J. Robot. Control*, vol. 2, no. 5, pp. 448–455, 2021, doi: 10.18196/jrc.25121.
- [14] M. J. M. Salazar, M. X. D. Villegas, and M. de L. P. Robalino, “Modeling, Simulation and Control of An Omnidirectional Mobile Manipulator Robot,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 5s, pp. 229–237, 2023.
-

-
- [15] C. Petres, Y. Pailhas, Y. Petillot, and D. Lane, “Underwater path planning using fast marching algorithms,” in *Europe Oceans 2005*, 2005, vol. 2, pp. 814-819 Vol. 2. doi: 10.1109/OCEANSE.2005.1513161.
- [16] N. Patel, R. Slade, and J. Clemmet, “The ExoMars rover locomotion subsystem,” *J. Terramechanics*, vol. 47, no. 4, pp. 227–242, Aug. 2010, doi: 10.1016/j.jterra.2010.02.004.
- [17] M. A.-M. Salem, “Multi-stage localization given topological map for autonomous robots,” in *2012 Seventh International Conference on Computer Engineering & Systems (ICCES)*, Nov. 2012, pp. 55–60. doi: 10.1109/ICCES.2012.6408483.
- [18] A. Nash and S. Koenig, “Any-Angle Path Planning,” *AI Mag.*, vol. 34, no. 4, pp. 85–107, Sep. 2013, doi: 10.1609/aimag.v34i4.2512.
- [19] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, “Path planning: A 2013 survey,” in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, 2013, pp. 1–8.
- [20] M. Brunner, T. Fiolka, D. Schulz, and C. M. Schlick, “Design and comparative evaluation of an iterative contact point estimation method for static stability estimation of mobile actively reconfigurable robots,” *Rob. Auton. Syst.*, vol. 63, pp. 89–107, Jan. 2015, doi: 10.1016/j.robot.2014.09.003.
- [21] M. Norouzi, J. V. Miro, and G. Dissanayake, “Planning Stable and Efficient Paths for Reconfigurable Robots On Uneven Terrain,” *J. Intell. Robot. Syst.*, vol. 87, no. 2, pp. 291–312, Aug. 2017, doi: 10.1007/s10846-017-0495-8.
- [22] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, M. Azkarate, L. Gerdes, and
-

-
- A. García-Cerezo, “Dynamic path planning for reconfigurable rovers using a multi-layered grid,” *Eng. Appl. Artif. Intell.*, vol. 86, pp. 32–42, Nov. 2019, doi: 10.1016/j.engappai.2019.08.011.
- [23] M. Effati, J.-S. Fiset, and K. Skonieczny, “Considering Slip-Track for Energy-Efficient Paths of Skid-Steer Rovers,” *J. Intell. Robot. Syst.*, vol. 100, no. 1, pp. 335–348, Oct. 2020, doi: 10.1007/s10846-020-01173-5.
- [24] K. Bergman, O. Ljungqvist, and D. Axehill, “Improved Path Planning by Tightly Combining Lattice-Based Path Planning and Optimal Control,” *IEEE Trans. Intell. Veh.*, vol. 6, no. 1, pp. 57–66, Mar. 2021, doi: 10.1109/TIV.2020.2991951.
- [25] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, “Path planning and collision avoidance for autonomous surface vehicles I: a review,” *J. Mar. Sci. Technol.*, vol. 26, no. 4, pp. 1292–1306, Dec. 2021, doi: 10.1007/s00773-020-00787-6.
- [26] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, “Path Planning for Autonomous Mobile Robots: A Review,” *Sensors*, vol. 21, no. 23, p. 7898, Nov. 2021, doi: 10.3390/s21237898.
- [27] P. Papadakis, “Terrain traversability analysis methods for unmanned ground vehicles: A survey,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, Apr. 2013, doi: 10.1016/j.engappai.2013.01.006.
- [28] H. Sun, W. Zhang, R. Yu, and Y. Zhang, “Motion Planning for Mobile Robots—Focusing on Deep Reinforcement Learning: A Systematic Review,” *IEEE Access*, vol. 9, pp. 69061–69081, 2021, doi: 10.1109/ACCESS.2021.3076530.
- [29] V. S. Raghavan, D. Kanoulas, A. Laurenzi, D. G. Caldwell, and N. G. Tsagarakis, “Variable Configuration Planner for Legged-Rolling Obstacle
-

-
- Negotiation Locomotion: Application on the CENTAURO Robot,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 4738–4745. doi: 10.1109/IROS40897.2019.8968014.
- [30] M. Sutoh, M. Otsuki, S. Wakabayashi, T. Hoshino, and T. Hashimoto, “The Right Path: Comprehensive Path Planning for Lunar Exploration Rovers,” *IEEE Robot. Autom. Mag.*, vol. 22, no. 1, pp. 22–33, Mar. 2015, doi: 10.1109/MRA.2014.2381359.
- [31] T. Hines *et al.*, “Virtual Surfaces and Attitude Aware Planning and Behaviours for Negative Obstacle Navigation,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 4048–4055, Apr. 2021, doi: 10.1109/LRA.2021.3065302.
- [32] H. Taghavifar, S. Rakheja, and G. Reina, “A novel optimal path-planning and following algorithm for wheeled robots on deformable terrains,” *J. Terramechanics*, vol. 96, pp. 147–157, Aug. 2021, doi: 10.1016/j.jterra.2020.12.001.
- [33] R. E. Arvidson *et al.*, “Spirit Mars Rover Mission: Overview and selected results from the northern Home Plate Winter Haven to the side of Scamander crater,” *J. Geophys. Res.*, vol. 115, no. E7, p. E00F03, Sep. 2010, doi: 10.1029/2010JE003633.
- [34] G. Ishigami, K. Nagatani, and K. Yoshida, “Path Planning and Evaluation for Planetary Rovers Based on Dynamic Mobility Index,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 601–606. doi: 10.1109/IROS.2011.6094768.
- [35] H. Inotsume, T. Kubota, and D. Wettergreen, “Robust Path Planning for Slope Traversing Under Uncertainty in Slip Prediction,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3390–3397, Apr. 2020, doi: 10.1109/LRA.2020.2975756.
-

-
- [36] H. Inotsume, C. Creager, D. Wettergreen, and W. R. L. Whittaker, “Finding routes for efficient and successful slope ascent for exploration rovers,” 2016.
- [37] P. Niksirat, A. Daca, and K. Skonieczny, “The effects of reduced-gravity on planetary rover mobility,” *Int. J. Rob. Res.*, vol. 39, no. 7, pp. 797–811, 2020, doi: 10.1177/0278364920913.
- [38] A. Kaplan, N. Kingry, P. Uhing, and R. Dai, “Time-Optimal Path Planning With Power Schedules for a Solar-Powered Ground Robot,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1235–1244, Apr. 2017, doi: 10.1109/TASE.2016.2533418.
- [39] S. S. Pujari, M. S. Patil, and S. S. Ingleshwar, “Remotely controlled autonomous robot using Android application,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Feb. 2017, pp. 588–593. doi: 10.1109/I-SMAC.2017.8058248.
- [40] K. Groves, E. Hernandez, A. West, T. Wright, and B. Lennox, “Robotic Exploration of an Unknown Nuclear Environment Using Radiation Informed Autonomous Navigation,” *Robotics*, vol. 10, no. 2, p. 78, May 2021, doi: 10.3390/robotics10020078.
- [41] M. R. Alenezi and A. M. Almeshal, “Optimal Path Planning for a Remote Sensing Unmanned Ground Vehicle in a Hazardous Indoor Environment,” *Intell. Control Autom.*, vol. 09, no. 04, pp. 147–157, 2018, doi: 10.4236/ica.2018.94011.
- [42] P. Raja, “Optimal path planning of mobile robots: A review,” *Int. J. Phys. Sci.*, vol. 7, no. 9, pp. 1314–1320, Feb. 2012, doi: 10.5897/IJPS11.1745.
- [43] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “Heuristic approaches in robot path planning: A survey,” *Rob. Auton. Syst.*, vol. 86, pp. 13–28,
-

-
- Dec. 2016, doi: 10.1016/j.robot.2016.08.001.
- [44] M. N. Zafar and J. C. Mohanta, “Methodology for Path Planning and Optimization of Mobile Robots: A Review,” *Procedia Comput. Sci.*, vol. 133, pp. 141–152, 2018, doi: 10.1016/j.procs.2018.07.018.
- [45] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, “A review: On path planning strategies for navigation of mobile robot,” *Def. Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019, doi: 10.1016/j.dt.2019.04.011.
- [46] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, “Experimental kinematics for wheeled skid-steer mobile robots,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007, pp. 1222–1227. doi: 10.1109/IROS.2007.4399139.
- [47] M. Elbanhawi and M. Simic, “Sampling-Based Robot Motion Planning: A Review,” *IEEE Access*, vol. 2, pp. 56–77, 2014, doi: 10.1109/ACCESS.2014.2302442.
- [48] I. Noreen, A. Khan, and Z. Habib, “Optimal path planning using RRT* based approaches: a survey and future directions,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, p. 12, 2016.
- [49] A. S. H. H. V Injarapu and S. K. Gawre, “A survey of autonomous mobile robot path planning approaches,” in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, Oct. 2017, pp. 624–628. doi: 10.1109/RISE.2017.8378228.
- [50] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, “Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges,” *Sensors*, vol. 18, no. 9, p. 3170, Sep.
-

-
- 2018, doi: 10.3390/s18093170.
- [51] M. M. Costa and M. F. Silva, “A Survey on Path Planning Algorithms for Mobile Robots,” in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Apr. 2019, pp. 1–7. doi: 10.1109/ICARSC.2019.8733623.
- [52] S. Garrido, D. Alvarez, and L. E. Moreno, “Marine Applications of the Fast Marching Method,” *Front. Robot. AI*, vol. 7, p. 2, Jan. 2020, doi: 10.3389/frobt.2020.00002.
- [53] R. S. Nair and P. Supriya, “Robotic Path Planning Using Recurrent Neural Networks,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020, pp. 1–5. doi: 10.1109/ICCCNT49239.2020.9225479.
- [54] W. Houtman, C. A. L. Martinez, S. Wang, A. Ketels, H. P. J. Bruyninckx, and M. J. G. van de Molengraft, “Dynamic control of steerable wheeled mobile platforms applied to an eight-wheeled RoboCup Middle Size League soccer robot,” *Mechatronics*, vol. 80, p. 102693, 2021.
- [55] S. Choi, Jaehyun Park, Eulgyoon Lim, and W. Yu, “Global path planning on uneven elevation maps,” in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Nov. 2012, pp. 49–54. doi: 10.1109/URAI.2012.6462928.
- [56] S. Baek, G. Y. Bae, J. Kwon, K. Cho, and S. Jung, “Flexible Pressure-Sensitive Contact Transistors Operating in the Subthreshold Regime,” *ACS Appl. Mater. Interfaces*, vol. 11, no. 34, pp. 31111–31118, Aug. 2019, doi: 10.1021/acsami.9b09636.
- [57] Z. Abd Algfoor, M. S. Sunar, and H. Kolivand, “A Comprehensive Study on Pathfinding Techniques for Robotics and Video Games,” *Int. J.*
-

-
- Comput. Games Technol.*, vol. 2015, pp. 1–11, 2015, doi: 10.1155/2015/736138.
- [58] M. Likhachev and D. Ferguson, “Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles,” *Int. J. Rob. Res.*, vol. 28, no. 8, pp. 933–945, Aug. 2009, doi: 10.1177/0278364909340445.
- [59] A. Shum, K. Morris, and A. Khajepour, “Direction-dependent optimal path planning for autonomous vehicles,” *Rob. Auton. Syst.*, vol. 70, pp. 202–214, Aug. 2015, doi: 10.1016/j.robot.2015.02.003.
- [60] M. Laîné, C. Tamakoshi, M. Touboulic, J. Walker, and K. Yoshida, “Initial Design Characteristics, Testing and Performance Optimisation for a Lunar Exploration Micro-Rover Prototype,” *Adv. Astronaut. Sci. Technol.*, vol. 1, no. 1, pp. 111–117, Sep. 2018, doi: 10.1007/s42423-018-0007-3.
- [61] L. Marin, M. Valles, A. Soriano, A. Valera, and P. Albertos, “Event-Based Localization in Ackermann Steering Limited Resource Mobile Robots,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1171–1182, Aug. 2014, doi: 10.1109/TMECH.2013.2277271.
- [62] M. Wu, S.-L. Dai, and C. Yang, “Mixed Reality Enhanced User Interactive Path Planning for Omnidirectional Mobile Robot,” *Appl. Sci.*, vol. 10, no. 3, p. 1135, Feb. 2020, doi: 10.3390/app10031135.
- [63] R. Takei and R. Tsai, “Optimal Trajectories of Curvature Constrained Motion in the Hamilton–Jacobi Formulation,” *J. Sci. Comput.*, vol. 54, no. 2–3, pp. 622–644, Feb. 2013, doi: 10.1007/s10915-012-9671-y.

الخلاصة

يقدم هذا المشروع تصميم وتنفيذ روبوت متحرك متعدد الاتجاهات مزود بذراع آلية متكاملة، مع التركيز على تطوير خوارزميات التحكم باستخدام MATLAB. تشمل الأهداف الأساسية لهذا العمل تحقيق التحكم الدقيق في كل من حركة الروبوت متعددة الاتجاهات وموضع المؤثر النهائي للذراع من خلال الحركية العكسية وحسابات المصفوفة اليعقوبية.

تم تصميم نظام العجلات متعدد الاتجاهات لتوفير قدرة استثنائية على المناورة، مما يمكّن الروبوت من التحرك في أي اتجاه بسهولة. من خلال دمج الحركية العكسية، تم تطوير خوارزميات تترجم مواقع المؤثر النهائي المطلوبة إلى زوايا مشتركة للذراع الآلي، مما يسمح بالتحكم الدقيق في حركات الذراع. بالإضافة إلى ذلك، يتم استخدام مصفوفة جاكوبي لإنشاء علاقة بين سرعات المفاصل وسرعات المؤثر النهائي، مما يتيح الحركة المنسقة للذراع.

كما قامت بتصميم وتنفيذ روبوت متنقل متعدد الاتجاهات مزود بذراع آلية متكاملة، يتم التحكم فيه من خلال تطبيق على الهاتف الذكي. الهدف الأساسي هو توفير واجهة سهلة الاستخدام للتشغيل عن بعد لكل من الحركة الشاملة للروبوت وتحديد الموقع الدقيق للذراع الآلي باستخدام تطبيق الهاتف الذكي.

ويستفيد التنفيذ من قدرات الهواتف الذكية الحديثة، ويدمج اتصال Bluetooth أو Wi-Fi لإنشاء رابط سلس بين أوامر المستخدم وإجراءات الروبوت.



وزارة التعليم العالي والبحث العلمي
جامعة بابل / كلية الهندسة
قسم الهندسة الكهربائية

تصميم ومحاكاة روبوت مسير مسيطر عليه بخارطة ذات تصميم شخصي

رسالة
مقدمة الى كلية الهندسة في جامعة بابل
كجزء من متطلبات نيل درجة الماجستير في علوم الهندسة الكهربائية/الهندسة
الكهربائية/الإلكترونيك

من قبل

اباذر سعد كاظم

اشراف

الاستاذ الدكتور ايهاب عبد الرزاق حسين

الاستاذ الدكتور حيدر مهدي عبد الرضا