UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Faculty of Electrical Engineering and Technology

Design and Development of a Spherical Robot for Mobile Applications

SITI NOR ATIRAH BINTI SAMUJI

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2023

Design and Development of a Spherical Robot for Mobile Applications

SITI NOR ATIRAH BINTI SAMUJI

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics)
with Honours

Faculty of Electrical Engineering and Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJUTERAAN ELEKTRIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Design and Development of A Spherical Robot For Mobile Application

Sesi Pengajian : 2023/2024

Saya …Siti Nor Atirah Binti Samuji……. mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)
Alamat Tetap: Kampung Batu 2 Tebuk Pulai, 45200, Sabak Bernam, Selangor.

(COP DAN TANDATANGAN PENYELIA)

Ts. Aminurrashid Bin Noordin
Pensyarah Kanan
Jabatan Teknologi Kejuruteraan Elektrik
Fakulti Teknologi dan Kejuruteraan Elektrik
Universiti Teknikal Malaysia Melaka

Tarikh: 12 /1 / 2024

Tarikh: 12.1.2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “Design And Development Of A Spherical Robot For Mobile Application” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : Siti Nor Atirah Binti Samuji

Date : 12 /1/ 2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature :

Supervisor Name : Ts. Aminurrashid Bin Noordin

Date : 12 /1/ 2024

DEDICATION

This project is dedicated to my parents and family, whose unwavering support and belief in my abilities have been the driving force behind my academic journey. Their sacrifices, encouragement, and love have been my constant source of inspiration.

I also dedicate this work to my supervisor as my mentor, who has shared his knowledge and expertise and has guided me through the challenges of this project. His patience and guidance have been invaluable in shaping my understanding and approach to problem-solving.

Lastly, to my friends and classmates, who have been my companions in this journey, offering their help, understanding, and camaraderie. Thank you for being a part of this journey.

ABSTRACT

This project focused on the design and development of a spherical robot, which can be wirelessly controlled and navigate various terrains through a wheel-driven mechanism. The robot, composed of two hemispheres, housed a small vehicle that facilitated locomotion. The movement of the robot was controlled via a smartphone through a wifi connection and was equipped with a Hibiscus sense board, which included gyroscopes and accelerometer sensors, two A4988 motor drivers, two stepper motors, and a graphical user interface (GUI). The project aimed to explore the challenges and limitations associated with the spherical robot's driving mechanics, control strategies, and sensing capabilities. In addition, the assessment of the robot's performance across different surfaces, including tar roads, grass, sand, cement, and carpet has been carried out. While the spherical robot could achieve omnidirectional movement on level ground, it encountered issues related to stability control on uneven or slippery surfaces such as grass, sand and tar toad. This project contributed to the development of a unique spherical robot that could execute various tasks in diverse environments using wireless communication. The findings from this project provided valuable insights for future research and development in the field of spherical robotics.

ABSTRAK

Projek ini memberi tumpuan kepada reka bentuk dan pembuatan robot berbentuk sfera, yang boleh dikawal secara tanpa wayar dan mampu dikawal atas pelbagai jenis permukaan melalui mekanisme roda. Robot ini, yang terdiri daripada dua hemisfera, menempatkan kenderaan kecil yang memudahkan pergerakan. Pergerakan robot dikawal melalui telefon pintar melalui sambungan wifi dan dilengkapi dengan papan sensor Hibiscus, yang merangkumi giroskop dan sensor accelerometer, dua pemacu motor A4988, dua motor langkah, dan antara muka pengguna grafik (GUI). Projek ini bertujuan untuk meneroka cabaran dan batasan yang berkaitan dengan mekanik pemanduan, strategi kawalan, dan keupayaan pengesanan robot sfera. Selain itu, penilaian prestasi robot di pelbagai permukaan, termasuk jalan tar, rumput, pasir, simen, dan karpet telah dijalankan. Walaupun robot sfera boleh mencapai pergerakan omnidireksional di tanah rata, ia menghadapi isu berkaitan dengan kawalan kestabilan di permukaan yang tidak rata atau licin seperti rumput, pasir dan jalan tar. Projek ini menyumbang kepada pembangunan robot sfera unik yang boleh melaksanakan pelbagai tugas dalam pelbagai persekitaran menggunakan komunikasi wayarles. Penemuan dari projek ini memberikan pandangan berharga untuk penyelidikan dan pembangunan masa depan dalam bidang robotik sfera.

TABLE OF CONTENTS

PAGE

DECLARATION

APPROVAL

DEDICATIONS

ABSTRACT i

ABSTRAK ii

TABLE OF CONTENTS 1

LIST OF TABLES 3

LIST OF FIGURES 4

CHAPTER 1 INTRODUCTION 6
1.1 Background 6
1.2 Problem Statement 8
1.3 Project Objective 9
1.4 Scope of Project 9

CHAPTER 2 11
2.1 Introduction 11
2.2 Disaster Risk Reduction 11
2.3 Driving Mechanisms 13
  2.3.1 Direct Driven Method 13
  2.3.2 Gravity Driven Method 16
  2.3.3 Angular Momentum Driven Method 20
  2.3.4 Shape Deformation and Hybrid 22
2.4 Sensor Fusion 25
  2.4.1 Kalman Filter 26
  2.4.2 Bayesian Network 26
  2.4.3 Dempster-Shafer Theory 27
  2.4.4 Complementary Filter 27
  2.4.5 Application 28
2.5 Controller Algorithms 28
  2.5.1 Proportional-Integral-Derivative Controller 29
  2.5.2 Baseline Controller 29
  2.5.3 Feedback Controller 30
2.6 Summary 30

CHAPTER 3 METHODOLOGY 31
3.1 Introduction 31
3.2 Selecting and Evaluating Tools for a Sustainable Development 33
3.3 Methodology 33
  3.3.1 First Milestone: Literature Review 34
  3.3.2 Second Milestone: Project Development 35

3.3.2.1 Robot Mechanical Design 36
A. Spherical Shell 36
B. Robot Chassis 37
C. Camera 38
D. Stepper Motor 39
E. Wheel Design and Comprehensive Robot Design 40
F. Adhesive weight 41
G. Castor wheels 41
i. N20 Castor Wheel 42
ii. W420 Castor Wheel 43
3.3.2.2 Electronic design 43
A. Hibiscus Sense Microcontroller 44
A. Inertia Measurement Unit 45
B. Esp 32 Series Module 46
C. Motor Driver 46
D. Battery 47
E. Voltage Regulator 48
F. Component Board Design 49
3.3.3 Third Milestone: Simulation 49
3.3.4 Designing GUI 50
3.3.5 Testing GUI interface 50
3.3.6 Fourth Milestones: Testing 52
3.3.6.1 Integration of The Mechanical and Electronic Hardware 52
A. Stepper Motor Testing 53
B. Stepper Motor Controlled using Smartphone 54
3.3.6.2 PID Testing 54
3.3.6.3 Testing and analysis of the robot locomotion 55
3.3.6.4 Camera Testing 57
3.4 Summary 57

CHAPTER 4 RESULTS AND DISCUSSIONS 59
4.1 Introduction 59
4.2 Spherical Robot Performance in Locomotion 59
4.2.1 The Locomotion Performance on Different Terrains 60
4.3 Preferable Surface of The Robot 62
4.4 Camera 64

CHAPTER 5 CONCLUSION 67
5.1 Conclusion 67
5.2 Potential and commercialization 68
5.3 Recommendation for future works 69

REFERENCES 70

APPENDIX 73

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1:	Comparison driving mechanism	25
Table 4.1:	The performance of the robot to reach target	61
Table 4.2:	The displacement of the robot from the target	61

LIST OF FIGURES

FIGURE **TITLE** **PAGE**

Figure 1.1: Overview of the project 7

Figure 2.1: SAR robot for navigation [1] 12

Figure 2.2: Disaster swarm robot [2] 12

Figure 2.3: The Spring-Loaded design [3] 14

Figure 2.4: BHQ-III [5] 15

Figure 2.5: Two wheel robot with gyroscope and magnetometer [6] 16

Figure 2.6: Two wheel robot internal driving unit (IDU) [7] 16

Figure 2.7: Single pendulum mode [9] 17

Figure 2.8: Spherical robot with two pendulum mode [10] 18

Figure 2.9: KisBot II design [11] 19

Figure 2.10: XK-I robot 20

Figure 2.11: Spherical robot with CMG 21

Figure 2.12: Spherical robot with two internal rotor 22

Figure 2.13: Spherical robot with omni-wheels 22

Figure 2.14: Spherical robot with bionic limb 23

Figure 2.15: Seformable shape spherical robot 23

Figure 2.16: deformable shape spherical robot 24

Figure 3.1: Flowchart for whole process 32

Figure 3.2: Methodology flowchart 34

Figure 3.3: Literature review flowchart 35

Figure 3.4: Dimension of spherical shell 36

Figure 3.5: Component Case 37

Figure 3.6: Component Case Cover 38

Figure 3.7: Motor Bracket 38

Figure 3.8: Spy Cam/mini camera 39

Figure 3.9: Bipolar stepper motor 39

Figure 3.10: Complete view of the complete robot 40

Figure 3.11: Adhesive weight 41

Figure 3.12: N20 castor wheel 42

Figure 3.13: W420 castor wheel 43

Figure 3.14: Hibiscus sense module 44

Figure 3.15: A4988 diagram 47

Figure 3.16: Lithium Ion battery 48

Figure 3.17: voltage regulator 48

Figure 3.18: Printed Circuit Board (PCB) 49

Figure 3.19: GUI design flowchart 50

Figure 3.20: The RGB light on the board changed color using a smartphone app. 51

Figure 3.21: Implementation of component flowchart 52

Figure 3.22: Illustration of robot's movement 53

Figure 3.23: Testing algorithm 56

Figure 3.24: Spherical robot testing 57

Figure 4.1: Robot performance testing procedure 60

Figure 4.2: Robot placed on different terrains 64

Figure 4.3: Graph of robot's performance 64

Figure 4.4: Robot's point of view during the day 65

Figure 4.5: Robot's point of view during the night 66

CHAPTER 1

INTRODUCTION

1.1 Background

Mobile robots are machines that move in physical environments. These robots use sensors and software to identify their surroundings and plan their motions. Mobile robot locomotion can be based on different principles such as predefined paths or navigate autonomously. There are different types of mobile robots based on the usage of the robot in different field. Mobile robot can employ various devices for locomotion, such as wheels, legs, tracks or propellers.

Mobile robots have the potential to assist in disaster management by performing various tasks in disaster-affected areas, such as data acquisition and transmission, navigation and rescue planning and execution, situation awareness and adaptation, survivor identification and assistance, and infrastructure damage assessment. Additionally, mobile robots can enhance the safety and efficiency of search and rescue operations by reducing human exposure to hazards, reaching inaccessible area, and providing real-time information.

However, Debris field environments pose high risks of entanglement of mobile robots mechanism and may be inaccessible or hazardous for mobile robots to investigate. Search and rescue mission is a global problem that demands innovative solutions, such as spherical robots. Spherical robots are capable of moving in any direction without changing their orientation, protecting their internal mechanism with a shell, and equipping various driving systems and sensors. Nevertheless, spherical robots face many difficulties in designing and operating in complex environments, such as sensor fusion, control algorithms, obstacle avoidance, and wireless communication.

A sensor fusion algorithm will be used to integrate the data from the sensors and provide a robust and accurate perception of the robot’s surroundings. The project will consist of four phases: mechanical design, electronic design, testing and evaluation. The mechanical design will involve creating the spherical structure and the driving system of the robot. The electronic design will involve selecting and connecting the sensors, actuators, camera, and wireless module. The testing and evaluation will involve executing a simulation based on the primary results acquired from the component. The project will have implications of spherical robots in diverse environments such as surveillance, data collection and communication. Figure 1.1 shows the overview of the project.

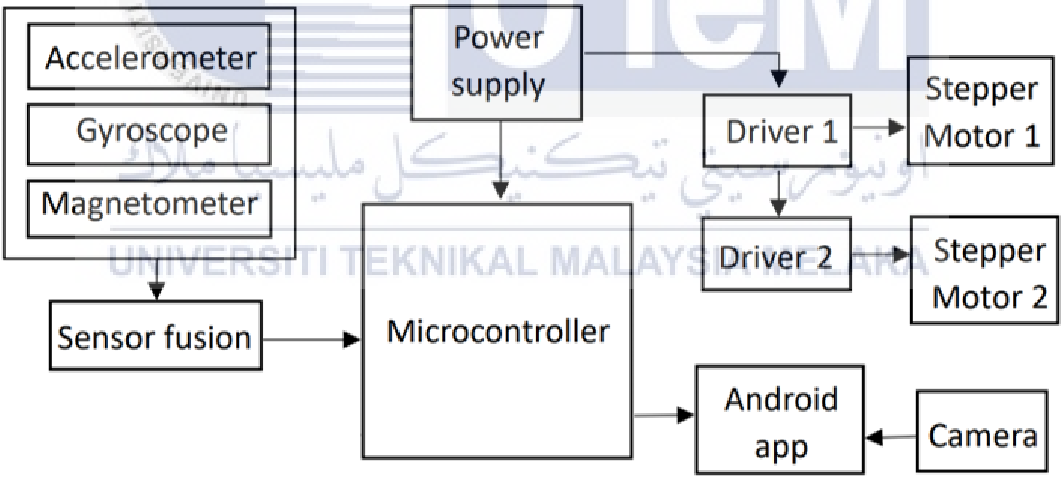


Figure 1.1: Overview of the project

1.2 Problem Statement

Working in dirty, dull, and dangerous environments, such as underwater, underground, and disaster areas, poses many risks and challenges for humans. Humans may face physical injuries, chemical hazards, biological threats, and psychological stress when performing these jobs. For example, rescue workers in road tunnel accidents may encounter hazardous substances, cramped spaces, electricity, high temperatures, and toxic materials. These hazards may cause trauma or long-term health effects for the rescue workers, such as injuries, illnesses, diseases, disabilities, or even death. Therefore, there is a need for an advance solution that can perform these tasks safely and efficiently.

One possible robotic solution is a wheel mobile robot, which is a type of robot that uses wheels to provide locomotion. However, wheel mobile robots have several limitations in these environments. The robot may have trouble balancing and stabilizing themselves on uneven or slippery surfaces. Mobile robots also have trouble avoiding obstacles or hazards that may block their path or damage their mechanism. Moreover, the robot may have trouble controlling their motion and orientation in confined spaces. These limitations may reduce their performance and capabilities in these tasks.

This project proposes a spherical robot as an alternative solution for these problems. A spherical robot is a type of robot that has a spherical shape and can roll over different terrains and obstacles. Spherical robots can withstand harsh environmental conditions and perform various tasks such as surveillance, data collection, and communication.

1.3 Project Objective

The objectives of the project are as follows:

1) To design and develop a spherical robot capable of wireless control via smartphone or computer apps, which can navigate in any direction and perform complex tasks such as environmental monitoring, and surveillance.
2) To integrate IMU sensor, and a wireless camera to provide the robot with a comprehensive understanding of its surroundings, enabling it to operate more accurately and safely.
3) To conduct extensive testing of the robot's performance and capabilities in various real-world scenarios, assessing its speed, stability, and safety, and identifying areas for further improvement.

1.4 Scope of Project

The scope of this project are as follows:

1) The spherical robot will be designed with a transparent or acrylic material shell, with a diameter of 14cm.
2) The robot will use a Hamster ball mechanism for movement.
3) The robot's inner system will include a microcontroller, two motors, IMU sensors, and a wireless camera for comprehensive understanding of its surroundings.
4) The robot will use Wi-Fi for wireless communication.

5) A sensor fusion approach will be employed to integrate the various sensors, including the gyroscope, accelerometer, and magnetic sensor in the IMU system.
6) A wireless camera will be integrated to provide visual feedback of the robot’s environment for monitoring purposes.
7) A smartphone or computer app will be developed to control the movement of the robot, monitor its states such as speed, velocity, and display a perspective view of the robot.
8) A comprehensive assessment will be conducted to evaluate the robot's speed, stability, safety, and vibrations during forward, backward, left, and right movement.
9) Specific tasks, such as navigating through different terrain, will be measured to evaluate the robot's performance.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This section provides literature review from primary information sources are reliable websites, articles, and journals regarding the previous development of spherical robots. It comprises four main subsections that include global issues regarding disaster risk reduction, driving mechanism of the robot, sensor fusion and control algorithms that have been used in past spherical robot development.

2.2 Disaster Risk Reduction

Disaster risk reduction (DRR) is the concept and practice of reducing disaster risks through systematic efforts to analyze and reduce the causal factors of disasters. DRR aims to prevent new and reduce existing disaster risk and manage residual risk, all of which contribute to strengthening resilience and therefore to the achievement of sustainable development. One of the applications of DRR is the use of robots for disaster scenarios, such as search and rescue (SAR).

Search and Rescue (SAR) Robot is a term that refers to a single or a group of small and portable mobile robots that can be deployed in disaster areas to locate and rescue victims, especially after a disaster. SAR Robot is distinct from military robots, as it has to satisfy three design constraints extreme operating conditions, ability to function in environments where Global Positioning System (GPS) and wireless signals are unavailable or limited, ability to operate autonomously as well as cooperatively with operators and victims [1]. SAR robot illustration is shown in Figure 2.1.

In addition, swarm robots are regarded as a suitable platform for disaster situations, as the robot can cope with cluttered and unstructured environments that are often very narrow. The swarm robot platform enables each robot to be small enough to access narrow areas, and also to cooperate with other robots to perform certain tasks, such as some limited rescue operations [2]. Disaster swarm robot illustration is shown in Figure 2.2.

Figure 2.1: SAR robot for navigation [1]

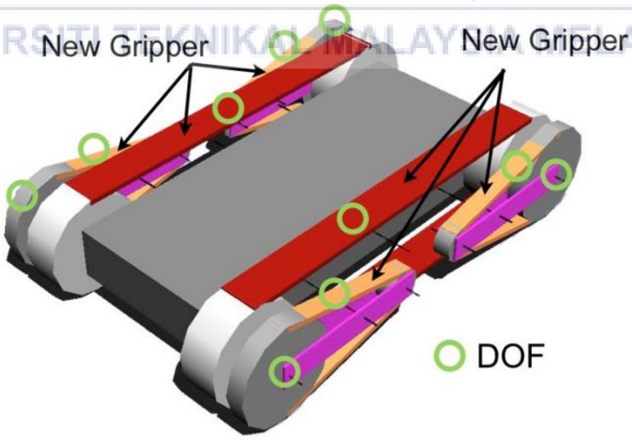


Figure 2.2: Disaster swarm robot [2]

2.3 Driving Mechanisms

The main term of the design and development of autonomous robot for spherical robots is the driving mechanisms. The technique or approach that allows a spherical robot to move or change direction is referred to as the driving mechanism. For spherical robots, several types of driving systems have been developed and put into practice, each having benefits and limitations. Spherical robots utilize four distinct types of driving mechanisms [3] as follows:

1) Direct Driven Method
2) Gravity Driven Method
3) Angular Momentum Driven Method
4) Shape Deformation and Hybrid Method

2.3.1 Direct Driven Method

The shell is navigated non-holonomically similar to a car. The heading of the internal robot must be changed in order to change the direction of travel. Single-wheeled or multi-wheeled vehicles can be utilized, and a four-wheeled differential-drive vehicle will create different motion curves as opposed to a single wheeled vehicle . The benefit of this approach is that it enables the robot to travel without any singular points in any direction, which is known as singularity-free omnidirectional locomotion [3] .

The Spring-Loaded type with IDU system is a common low-cost design, but it struggles to control the robot’s heading at high speeds. Slippage can happen between the wheels and the shell, and between the shell and the medium. The slippage can be minimized by adjusting the tension between the spring-loaded system and the internal robot, but this increases the friction forces in the robot. Also, an IDU system cannot use stored momentum: if the wheels stop, the robot will act erratically. An IDU system needs power to move down a small incline: it cannot roll down without assistance. On the other hand, rolling down steep inclines without power will cause unpredictable movement. The IDU system must also be very well balanced. An off-axis center of mass may make the robot travel in an unwanted pattern [4]. The Spring-Loaded design is shown in Figure 2.3.

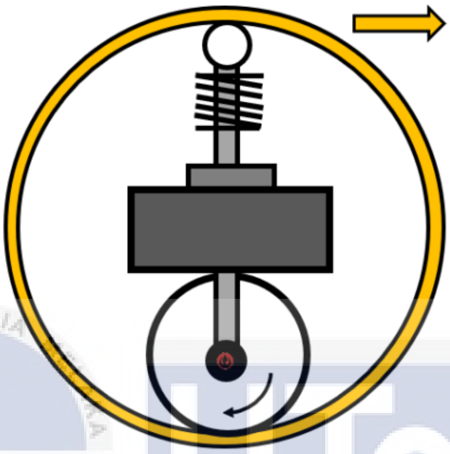


Figure 2.3: The Spring-Loaded design [3]

In [5], this research created a robot called BHQ-III, its name derived from the abbreviating Boltzmann-Hamel equation. One drive motor, which is directly coupled to the drive wheel, makes up this system. As in the first for example, the drive wheel makes direct contact with the spherical robot inner shell. Additionally, the IDU, which directly controls the rotation of the robot, is managed by the second motor. This approach unquestionably makes it possible for the spherical robot to travel with nearly no turning radius, accounting for a high degree of holonomy. The platform hits the inside shell of the robot at four spots, with the drive wheel providing one touch and the sponge wheels providing the other three. This results in more friction and allows the robot to go over uneven ground. The BHQ-III robot is shown in Figure 2.4.

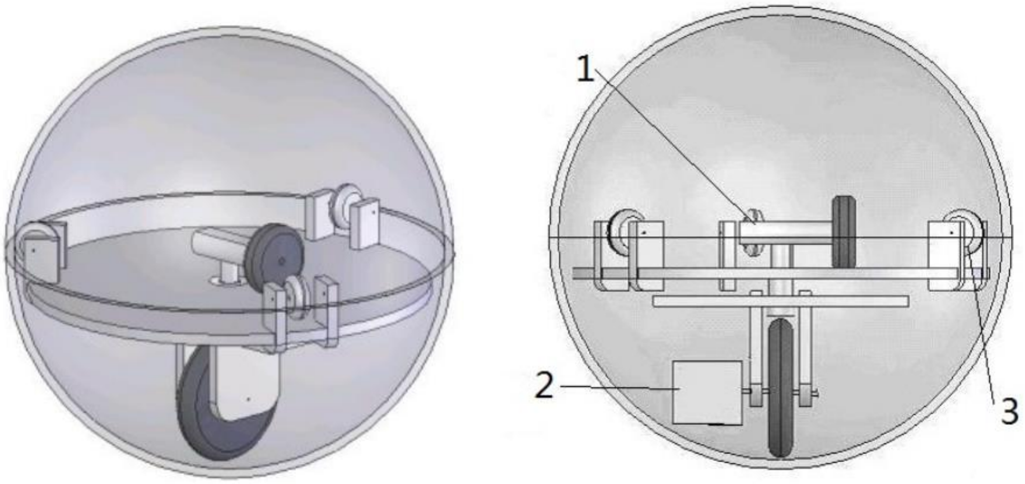


Figure 2.4: BHQ-III [5]

Two wheel driving mechanism for spherical robot is a method of propulsion that uses two wheels inside the sphere to drive the spherical shell. The robot has a mechanical design that enables it to have multiple degrees of freedom and to change its direction of movement instantly. Steering can be achieved by varying the velocity or the direction of the wheels. This design is simple, but it necessitates a uniform sphere and the wheels may detach from the sphere's inner surface because of motion disturbances [6]. The advantage of this method is that it can achieve high maneuverability and stability. The disadvantage is that it requires a complex control system and a high power consumption.

The spherical robot moves in response to the remote-control car's motion. To steer the sphere, the car can go left, right, forward, and backward . To gauge its orientation and direction, the robot also incorporates a gyroscope and a magnetometer shown in Figure 2.5. On level terrain, the robot is capable of omnidirectional mobility . In addition, some with two wheel mechanism robot has been designed for children with development disorder. The spherical robot that can jump and roll toward children actively. The robot used soft shell to prevent injuries to the children [7]. Figure 2.6 shows the illustration of jumping robot using two wheel robot internal driving unit (IDU).

Figure 2.5: Two wheel robot with gyroscope and magnetometer [6]

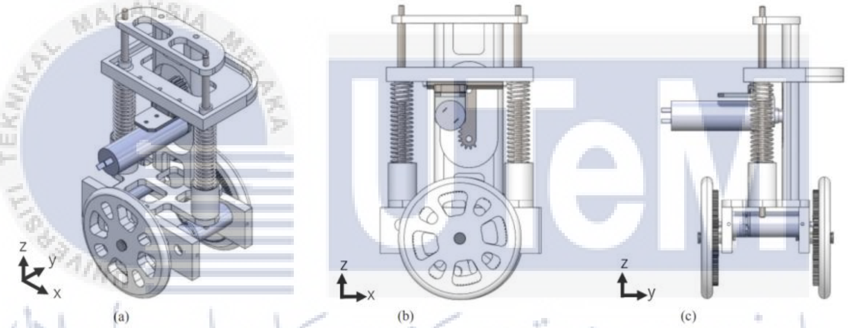


Figure 2.6: Two wheel robot internal driving unit (IDU) [7]

2.3.2 Gravity Driven Method

Spherical robot employs a original locomotion system by shifting its center of gravity to produce torque and rotate itself. However, spherical robot can only generate small torque for rotation. Therefore, this limits its locomotion capability for uphill climbing and obstacle overcoming. Pendulum method can be used to displace mass for locomotion. Pendulum are usually connected to the geomatric center of the spherical shell, without touching the exoskeleton. Therefore, any type otspherrical body can be employed such as inflatable, deployable or wired body [8].

Single pendulum method used in spherical robot is a method of propulsion that uses a pendulum inside the sphere to shift the center of mass (CoM) and generate torque. However, it can only generate small torque and has low stability. On the other hand, it can achieve omnidirectional movement and low power consumption. The pendulum can be controlled by a motor to swing in different directions. Despite its durability, simplicity, and low speed, this approach has numerous drawbacks, including difficulty stabilizing and low speed, and low precision. The pendulums are designed to rotate without any interference with the main shaft of the base frame to generate the movement paths This method can be employ to spy robot for surveilance ue to its stability to monitor the environment [9]. Figure 2.7 shows the illustration for single pendulum mode.

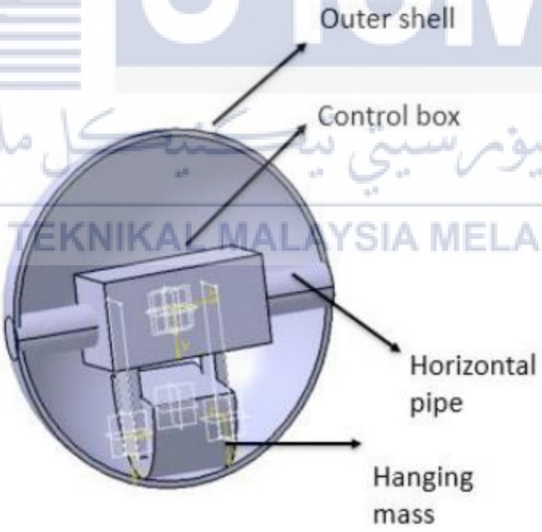


Figure 2.7: Single pendulum mode [9]

Spherical robots that uses two pendulums inside the robot to generate motion and steering as shown in Figure 2.8. The pendulums are mounted on a cross-shape frame that is fixed to the outer shell of the robot. The pendulums produce two types of moments that affect the robot's movement eccentric moment and inertial moment [10]. The eccentric moment is caused by the shift of the center of mass of the robot away from the center of the sphere due to the position of the pendulums. The inertial moment is caused by the change of angular momentum of the robot due to the rotation of the pendulums. The robot can perform turning in place motion by rotating both pendulums in opposite directions with equal magnitude. The robot can perform omni directional locomotion by combining linear motion and turning in place motion in a smooth way.

KisBot II is an active spherical robot that has a curved two-pendulum driving system and enhanced driving capability. It has advantages over the conventional one-pendulum or two-pendulum spherical robot, such as improved movement direction capability from a stationary state, omnidirectional driving capability using a base frame operation for two-pendulum driving, and the capability to change the driving direction by adding a rotational degree of freedom to the pendulum along the horizontal axis [11]. Figure 2.9 shows the illustration of KisBot II design.

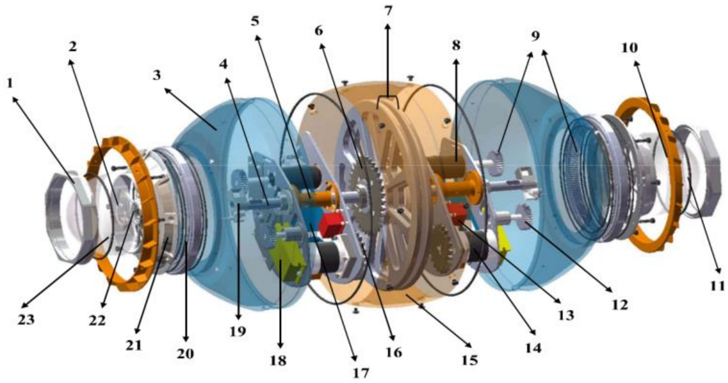


Figure 2.8: Spherical robot with two pendulum mode [10]

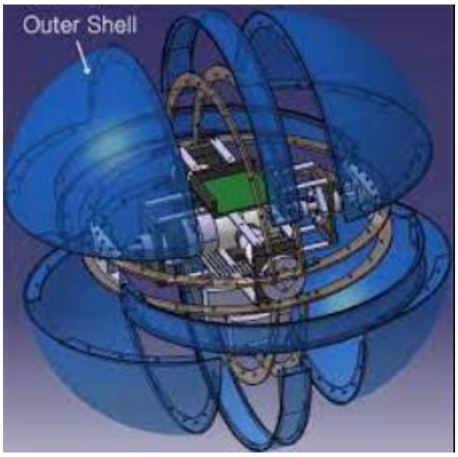


Figure 2.9: KisBot II design [11]

The shifting mass method in spherical robot is a type of driving mechanism that operates by changing the robot’s center of mass (the barycenter) in order to produce a desired motion [12]. the direction of motion can be changed orthogonally while moving. It has the possibility, in conjunction with its design, to constantly accelerate, decelerate, and maintain a constant speed. The rotor is mounted to the robot's spherical shell from the inside. A two-degree-of-freedom spherical robot XK-I with a swing around the X and Y axes [13]. The dynamic model of linear motion is established by Lagrangian equation. The XK-I robot steering is implemented by controlling the number of steps of the stepping motor by the pulse signal, so that the XK-I robot can move in any direction. The XK-I is shown in Figure 2.10. A single rotor rotates along the Z axis, while two rotors revolve in tandem along the X axis to form a single rigid linked body. The conservation of angular momentum causes the spherical robot to roll in the opposite direction when the rotors are turned.

.

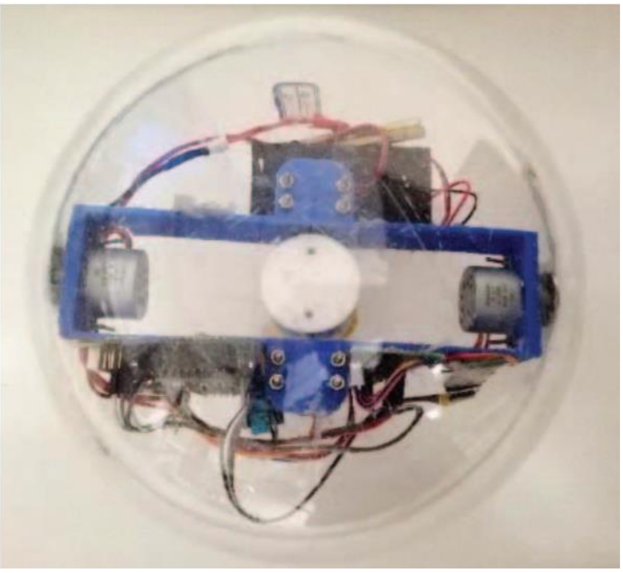


Figure 2.10: XK-I robot

2.3.3 Angular Momentum Driven Method

A spherical robot with angular momentum uses spinning internal weights to produce torque and move in the desired direction. A separate motor is utilised to move the wheel forward and backward. An internal gyroscope stabilises the wheel's rolling axis. The spherical robot has a Control Moment Gyroscope (CMG) inside it [14]. It can roll forward or backward by using a straight motor that creates torques. It can also steer by using a steering weight that shifts the center of gravity. The steering mechanism has a steering motor, a steering weight, a lead screw and some member bars. The robot rolls forward or backward by the friction between the driving wheel and the inner shell. The spherical shell is attached to the hollow axles that are connected to the frame by hinges shown in Figure 2.11.

The spherical robot rolls on a plane by using two internal rotors that work on the principle of conservation of angular momentum shown in Figure 2.12. The driving unit has a pendulum and an arched body. The arched body and pendulum can control the pitch angle, which is the angle of tilting up or down. The arched body below the pendulum can control the roll angle, which is the angle of tilting left or right. The roll angle is controlled at the same time as the pitch angle. The IDU has a wheel that rolls inside the spherical shell and is driven by a motor. The robot moves because the inside construction makes the system unbalanced [15].

Another types is the spherical robot can move in any direction by using three sets of omni-wheels. The omni-wheels are part of an internal driving unit that is inside the hollow spherical shell [16]. This protects the driving unit and also makes the motion more precise. The omni wheels are attached to the motor. The omni wheels can move forward and sideways. The IDU has 3 omni wheels that are attached to the motors on the edge of the links. The robot can move in any direction by using three omni-directional wheels that are 120 degrees apart. Each wheel is connected to a DC motor. A PID controller controls the robot's motion and Bluetooth communication sends and receives data from the robot [17]. The main frame of the IDU has three links that are at right angles to each other and joined at a common point. This makes the frame look like a cone and makes it stable and easy to fit inside the spherical shell shown in Figure 2.13.

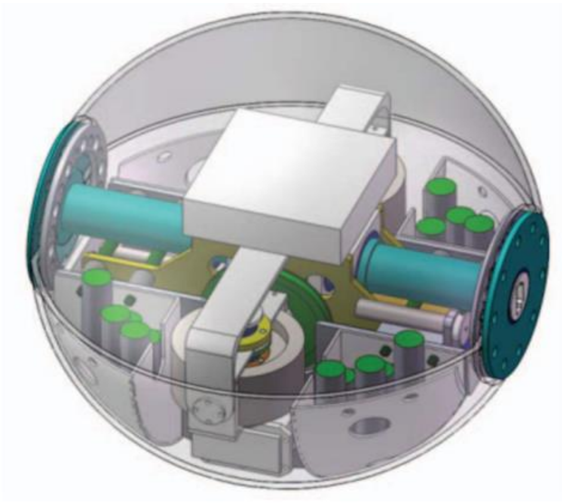


Figure 2.11: Spherical robot with CMG

Figure 2.12: Spherical robot with two internal rotor

Figure 2.13: Spherical robot with omni-wheels

2.3.4 Shape Deformation and Hybrid

The robot is a new deformable robot that can roll like a sphere or walk like a quadruped. The robot mechanism can change to different modes depending on the terrain. The deformable mechanism makes the robot move with high motion efficiency and energy saving on flat terrain, and also makes the robot adapt to unstructured terrain [18]. The robot uses the advantages of energy-saving and stability of spherical rolling, and also tries to improve the robot’s ability to move on uneven terrain. The main idea is to combine the spherical structures with the bionic limb structures . The robot will change to different motion states, such as spherical rolling, quadruped walking, and spherical-legged rolling, according to the terrain changes as shown is Figure 2.14.

In [19], The robot can change its shape from a sphere to two connected halves: white and red. Each half has three legs with outer shell segments that have omnidirectional wheels. The robot can move in two ways rolling by changing its shape and wheel movement by using the omnidirectional wheels. Both ways need to change the center of gravity of the robot to make the internal forces that move the robot in the desired direction. The outer shell of the sphere shape can protect the internal parts of the robot as shown in Figure 2.15.

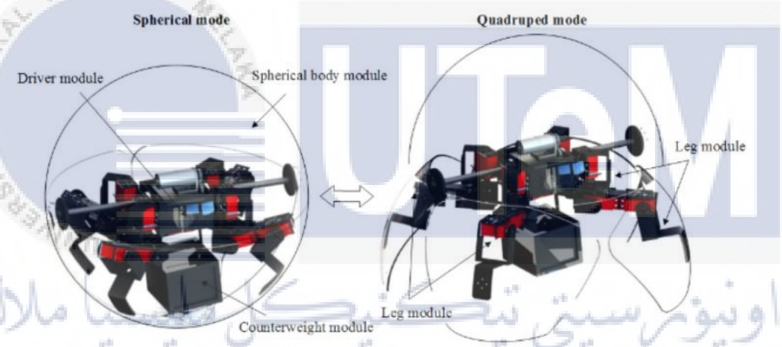


Figure 2.14: Spherical robot with bionic limb

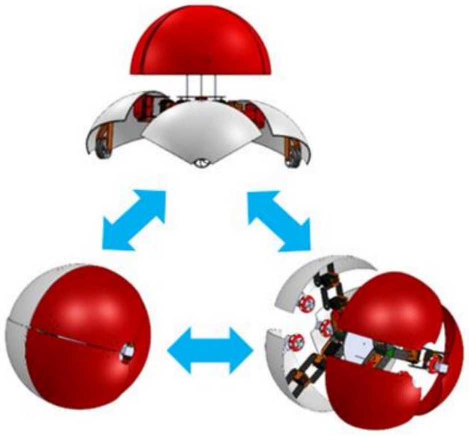


Figure 2.15: Seformable shape spherical robot

A hybrid type spherical robot can move in different ways by using different drive mechanisms. The spherical robot has a symmetrical structure that helps it explore underwater. The new designs of spherical robot can move freely under water. They have two propellers outside that make them move forward, backward, and turn on the same plane. The propellers are driven by the motor as shown in Figure 2.16. The robot can also float and sink by changing the size of the ball with buoyancy [20].

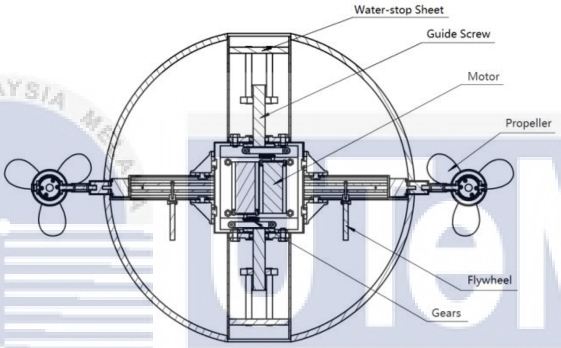


Figure 2.16: deformable shape spherical robot

Table 2.1: Comparison driving mechanism

Comparison	Driving method	Holonomy	Stability	Mobility	Durability	References
Single wheel	D	-	High	Medium	High	[4][5]
Hamster ball	D	R	Medium	Medium	High	[6][7]
Pendulum	G	R*	High	Medium	Medium	[8][9][10][11]
Shifting mass	G	R	High	High	Medium	[12][13]
Novel spherical	M	-	Medium	High	Low	[14][15]
Omni-directional	M	R*	Medium	High	Low	[16][17]
Deformation	S	R	High	Medium	Low	[18][19]
Hybrid	H	R*	High	Medium	Low	[20]

D – Direct driven method
G – Gravitational driven method
M – Angular momentum driven method
S – Shape deformation method
H – Hybrid method
R – Ability to perform specific movement
R* - Ability to perform specific movement in specific condition

2.4 Sensor Fusion

Sensor fusion is applied in autonomous robotics to enhance the perception, localization, mapping, and navigation of the robots. Sensor fusion integrates data from multiple sensors on and off the robot to reduce uncertainty and balance the positive and negative of different sensors. Sensor fusion can assist the robots in making decisions, estimations, and actions in dynamic and complex environments [21]. The sensors employed in autonomous robotics are cameras, radars, lidars, ultrasonic sensors, odometric sensors, and inertial measurement units (IMUs). Some of the sensor fusion methods employed in autonomous robotics are Kalman filter, Bayesian networks, Dempster-Shafer theory and Complementary filter.

2.4.1 Kalman Filter

A Kalman filter is a way of using math to find out the state of a system, such as its position, speed, or temperature. It does this by using a series of calculations that are fast and accurate. The filter can estimate the state of the system in the past, present, or future, even if some details of the system are unknown or uncertain. In addition, a discrete Kalman filter is a type of Kalman filter that works for systems that change and are measured at regular time intervals. A discrete Kalman filter uses simpler and faster math than a continuous Kalman filter, which works for systems that change and are measured continuously. A discrete Kalman filter can be made from a continuous Kalman filter by finding a special matrix that connects the system state at one time to the system state at the next time [21].

However, there is another version of Kalman filter that have been develop. An extended Kalman filter is a kind of Kalman filter that works for systems that are not linear, meaning that the state and observations are not connected by straight-line functions. An extended Kalman filter uses a method called linearization, which means making the non-straight-line functions look like straight-line functions near the current guess of the state. An extended Kalman filter uses the same math as a Kalman filter, but with the changed functions instead of the original ones [22].

2.4.2 Bayesian Network

Bayesian networks utilize probability theory and graphical models to represent and reason with uncertain knowledge. Such networks have various applications, such as analysing sensor data, supporting decisions, understanding situations, and learning from data. A Bayesian network can demonstrate the causal relationships among variables and update the beliefs based on new evidence. Moreover, a Bayesian network can cope with missing data, noisy data, and different data sources . Additionally, a Bayesian network can be learned from data or constructed from expert knowledge, or both [23].

2.4.3 Dempster-Shafer Theory

Dempster-Shafer theory is a mathematical framework that uses probability theory and evidence theory to represent and deal with uncertainty in sensor fusion. It is based on the work of Arthur Dempster and Glenn Shafer, who suggested a method of combining pieces of evidence from different sensors and finding a degree of belief for different hypotheses. Moreover, Dempster-Shafer theory can cope with various types of uncertainty such as vagueness, ambiguity, conflict, and ignorance. Additionally, Dempster-Shafer theory can be applied for various sensor fusion applications such as target recognition, fault diagnosis, complex event processing, and situation awareness [24].

2.4.4 Complementary Filter

A low-pass and a high-pass filter are used in a sensor fusion technique called a complementary filter. It is used to integrate data obtained from the inertial measurement unit's (IMU's) accelerometer and gyroscope sensors. The sensor fusion combine the orientation angles obtained from these sensors to get accurate estimates. The complementary filter is a widely used technique that depends on the proper selection of its gain parameters. In [25], a novel cascaded architecture of the complementary filter that consists of a nonlinear and a linear version within one framework has been proposed. The nonlinear version corrects the gyroscope bias, while the linear version estimates the attitude angle. The main advantage of the proposed architecture is that it does not depend on the filter parameters, which eliminates the need for tuning the filter’s gain parameters. Moreover, the design does not require any mathematical modeling of the system and is computationally inexpensive.of the orientation. The filtered data are then combined to produce a smoother and more precise estimate of the orientation.

2.4.5 Application

A low-cost inertial measurement unit composed of a 3-axis accelerometer and 3-axis gyroscope was used to estimate the attitude of a quadrotor UAV [26]. A microprocessor software environment was used to make the sensors work together to simulate 6 degrees of freedom orientation sensing using sensor fusion. A simple complimentary filter and a more complex Kalman filter were used to combine the data from each sensor and get the best of both sensors [27]. The Kalman filter was also used to remove the gyroscopic drift in the pitch and roll axes for both still and moving situations [26][27]. The results show that smooth roll, pitch and yaw attitude angles were obtained from the cheap IMU by using the sensor fusion algorithm.

2.5 Controller Algorithms

A spherical robot typically uses a controller algorithm to maintain its balance while navigating. This algorithm takes input from various sensors on the robot, such as gyroscopes and accelerometers, to determine the robot’s current state, including its tilt angle and rate of change. The controller then adjusts the speed and direction of the robot’s wheels to counteract any tilt and keep the robot upright.

2.5.1 Proportional-Integral-Derivative Controller

Proportional-Integral-Derivative controllers are commonly utilized in control systems because of their straightforwardness and high efficiency. They operate based on an algorithm that adjusts three parameters - Proportional, Integral, and Derivative - to achieve the best control [28]. The Proportional component generates a control output that is proportional to the present error, the Integral component is proportional to the total of past errors, and the Derivative component is proportional to the forecast of future errors. Despite their effectiveness, the task of tuning the PID parameters is challenging as it demands a comprehensive understanding of the system dynamics and can be a lengthy process [28]. To address these challenges, several techniques have been suggested in scholarly works. These techniques encompass methods such as heuristic tuning, optimization algorithms, and machine learning techniques for the automatic tuning of PID parameters.

2.5.2 Baseline Controller

The Linear Quadratic Regulator (LQR) controller has been widely studied and implemented in the field of two-wheeled robots. Research has explored the use of particle swarm optimization algorithms to enhance the parameter matrix of an LQR controller, resulting in improved system stability [29]. The studies have successfully applied LQR controllers for balance control in two-wheeled legged robots, demonstrating their efficiency through both simulations and experiments. Additionally, research has been conducted on designing an embedded controller using velocity feedback alone to stabilize a custom-designed self-balancing two-wheeled robot. These studies highlight the potential of LQR controllers in improving the performance and capabilities of two-wheeled robots [29].

2.5.3 Feedback Controller

The Radial Basis Function Neural Network (RBFNN) is adept at providing high-quality approximations for curve-fitting problems [30]. It can be trained to understand various types of nonlinear functions and system dynamics in a simple and swift manner. One of the key benefits of RBFNN is the ability to handle linear weights associated with the output layers independently from the neurons in the hidden layer . The weights in the hidden layer are adjusted using a nonlinear optimization process, while the weights in the output layer are adjusted using linear optimization. The precision of the approximations and the rate of convergence of the RBFNN can be enhanced by employing a strategy that involves the selection of suitable centers and widths for the receptive fields [30].

2.6 Summary

In this section, the initial results of the data collection are shown and analysed from previous studies. The mixed-methods approach is used to investigate the effects of robot outcomes. The effectiveness of different approaches for developing effective and capabilities of spherical robot was established via analysis of the literature on development of spherical robot. The technique that have been choose is employs an Hibiscus sense microcontroller to perform driving process, which collect data from gyroscope and accelerometer to the robot for faster balancing and control the motor. The development of spherical robot across a range of technology can be aided by the implementation of the appropriate sensor fusion.

CHAPTER 3

METHODOLOGY

3.1 Introduction

A mobile robot having a spherical shape that can travel in any direction is called a spherical robot. But creating a spherical robot is difficult since it requires dealing with balance and accuracy issues. The robot's ability to retain stability and direction while rolling across various surfaces and navigating obstacles is referred to as balance. Accuracy refers to the robot's ability to precisely regulate its motion and speed in accordance with its intended direction and objective. A decent spherical robot should be accurate and able to balance simultaneously, requiring advanced sensors, actuators, and algorithms. A more accurate model usually requires more resources, while a more effective model sacrifices some accuracy for less resources. Figure 3.1 shows the flowcart of the whole process.

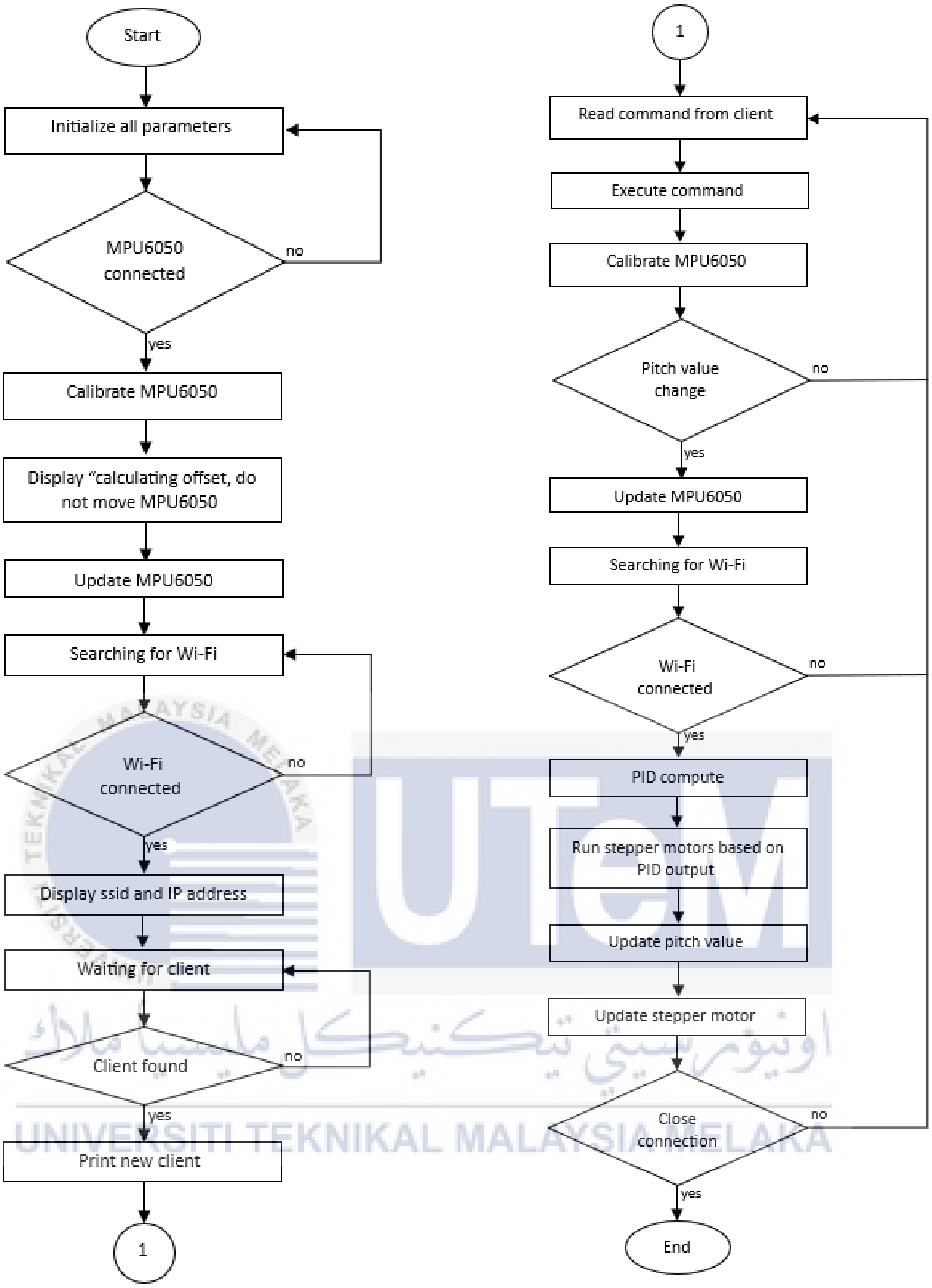


Figure 3.1: Flowchart for whole process

3.2 Selecting and Evaluating Tools for a Sustainable Development

A tough challenge that requires considering into account a number of characteristics of the robot's design, performance, and application are choosing and assessing tools for the sustainable development of spherical robots. A mobile robot having a spherical exterior shape that commonly rolls across surfaces is referred to as a "spherical robot". Some benefits of spherical robots include their ability to function on land and water, to shield internal components from the external environment, and to change their shape for various jobs. The complexity of their dynamics and kinematics, the trade-off between stability and maneuverability, and the challenge of managing their orientation and location are just a few of the difficulties that spherical robots encounter.

3.3 Methodology

This project's performance process has been divided into a four stages to ensure that it is completed with success includes milestone 1, milestone 2, milestone 3 and milestone 4. Each milestone consist activity that needs to be complete. The methodology flowchart for this project is shown in Figure 3.1.

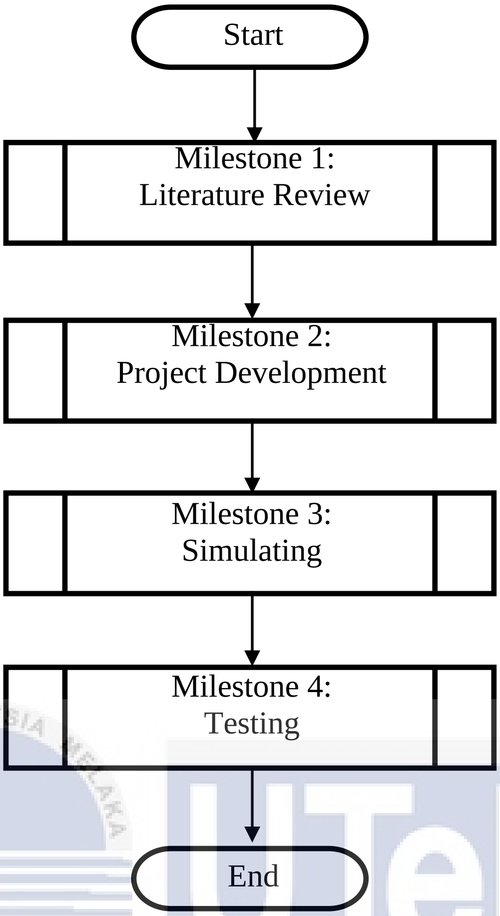


Figure 3.2: Methodology flowchart

3.3.1 First Milestone: Literature Review

To make sure the project meets with the requirements of the research project, the objectives of the project were discussed with the supervisor. The aim of this project is to design and develop a spherical robot that can be wirelessly controlled using smartphone or computer apps. To gather concepts and information regarding the development of a spherical robot from previous projects as well as research carried out by other researchers in various places or academic institutions. To develop the scope for this project and to focus on the research medium, the data from the published papers had been examined. The structure of the literature review conducted for this project is shown in Figure 3.2.

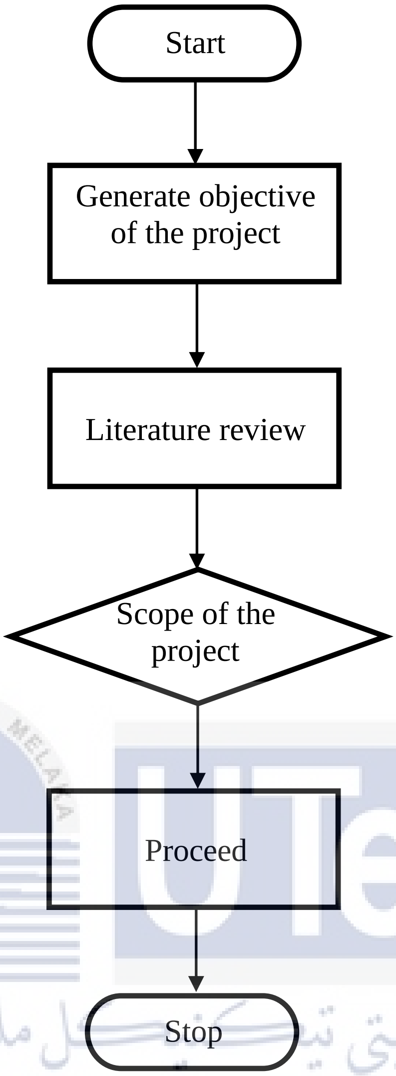


Figure 3.3: Literature review flowchart

3.3.2 Second Milestone: Project Development

This section describes the activities that are involved in the hardware design of the spherical robot. The hardware design consists of three main components: the spherical shell, the type of wheels and motors. The spherical shell is the outer layer of the robot that protects the internal component. The type of wheels and motors determines the robot’s performance on locomotion. All the components will be assembled to become a complete robot.

3.3.2.1 Robot Mechanical Design

In this section, the mechanical design of a robot encompasses the development of its physical architecture. This process involves the meticulous design of the robot’s structural chassis, the mechanisms of its body, and other essential components. These elements collectively contribute to the robot’s overall functionality and performance.

A. Spherical Shell

The size of the spherical shell and the type of material utilized are two specifications that must be taken into consideration when developing the robot's spherical shells. The primary consideration when designing the spherical robot's shell is the material. Based on its ability to be shaped into a sphere and high durability, the acrylic with a translucent appearance type material is chosen as the material for the shell of the spherical robot. The spherical shell consists of two hemispheres with same dimension each. The spherical robot's dimensions are then 180 mm in diameter for the outer shell and 177 mm for the inner shell. The spherical shell's dimensions are shown in Figure 3.3.

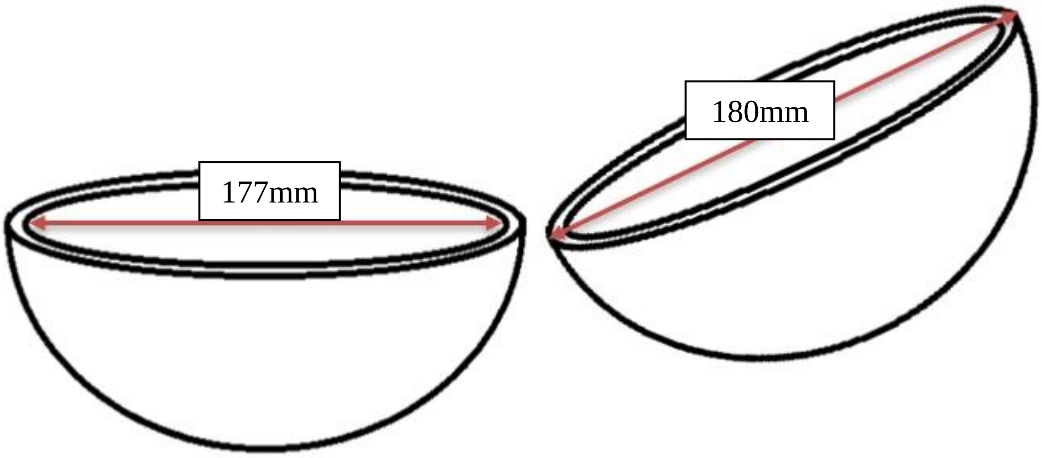


Figure 3.4: Dimension of spherical shell

B. Robot Chassis

A robot chassis is meticulously designed to meet specific requirements. This design process takes into account various factors such as the robot’s intended function, environment, and operational demands. The chassis serves as the robot’s fundamental structure, housing components like the power source, control system, and locomotion mechanisms. Figure shows the chassis design.

i. Component case

This case can fit a Hibiscus sense board and two motor drivers (A4988). The case has several holes for the jumper wire between the motor and the motor driver. Figure 3.5 shows the design and dimension of the case (mm).

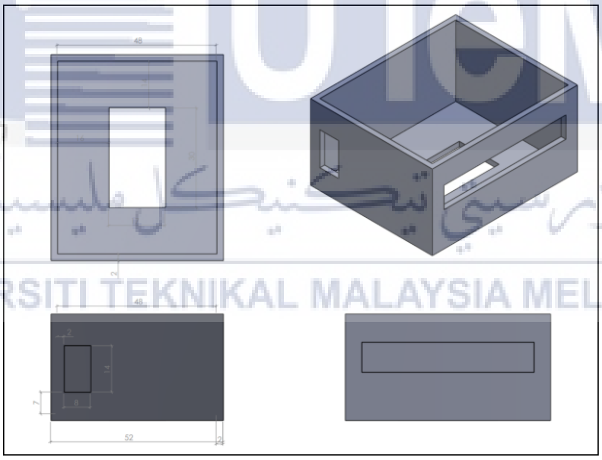


Figure 3.5: Component Case

ii. Component case cover

This cover is attached to the camera bracket which will hold the camera of the robot. Figure 3.6 shows the design and dimensions of the cover.

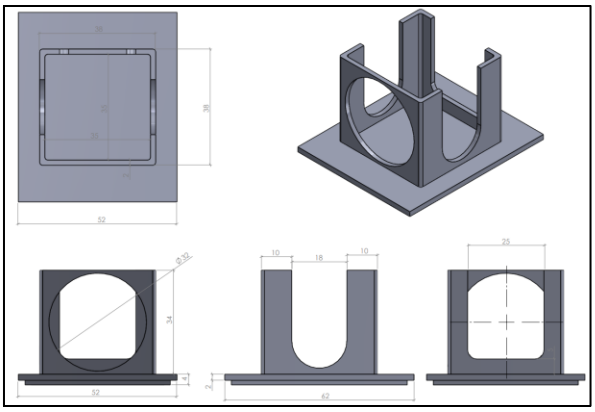


Figure 3.6: Component Case Cover

iii. Motor bracket

This bracket can fit a stepper motor (NEMA 17-42mmx23mm) and includes fourscrew holes that can attach to each of the motors. Figure 3.7 shows the design and the dimensions of the motor bracket.

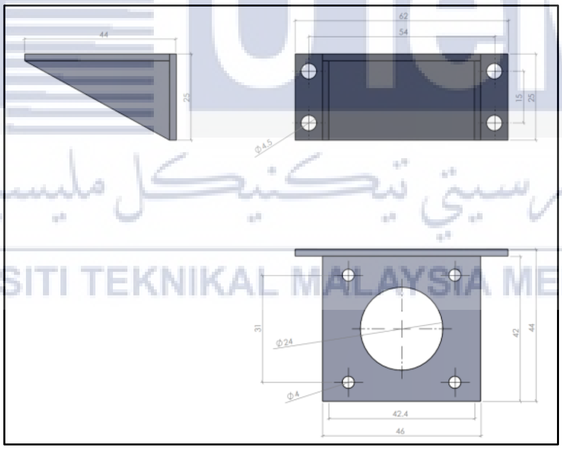


Figure 3.7: Motor Bracket

C. Camera

Due to its compact size (3.4cmx3.4cm x4.5cm), spycams or hidden cameras are used. This camera is a compact, portable gadget that can record high-definition photos and videos. In-built batteries, memory card slots, and USB ports for charging and data transfer are all features of the cameras. Additionally, the cameras offer functions including loop recording, motion detection, and infrared vision. The cameras can be wirelessly connected to PCs or smartphones via Wi-Fi. Figure 3.8 shows mini camera that will be use in this project.

Figure 3.8: Spy Cam/mini camera

D. Stepper Motor

Two bipolar stepper motor will be use in this spherical. It is a synchronous motor that converts digital pulses into mechanical shaft rotation. It has a rotor and a stator, where the rotor is a permanent magnet or an iron core and the stator has multiple coils organized in phases. By switching the coils on and off in sequence, the motor can move one step at a time. This allows for precise positioning and speed control. Figure 3.9 shows illustration of a bipolar stepper motor that will be used in this project.

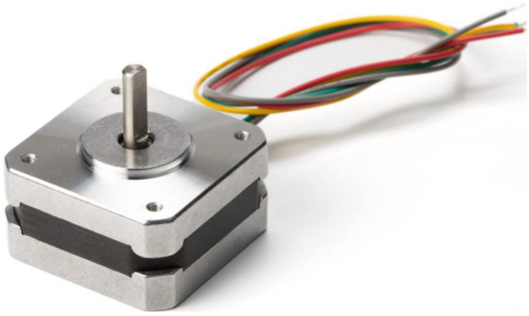


Figure 3.9: Bipolar stepper motor

E. Wheel Design and Comprehensive Robot Design

The wheel resembles a bicycle wheel, with a slender rim and spokes to reduce the contact area with the inner shell. The material used for the wheel is rubber type to give more grip between wheel and inner shell surface. The wheel is compact enough to fit snugly inside the spherical shell. The wheel measures 6 cm in diameter. Since there will be two motors implemented in this project, the motor bracket is used to hold the motor. The size of the components, such as the motor, battery, and microcontroller sizes, had an impact on the robot chassis development. The component case on which all of the electrical components will be mounted on motor bracket. The complete robot structure is shown in Figure 3.10.

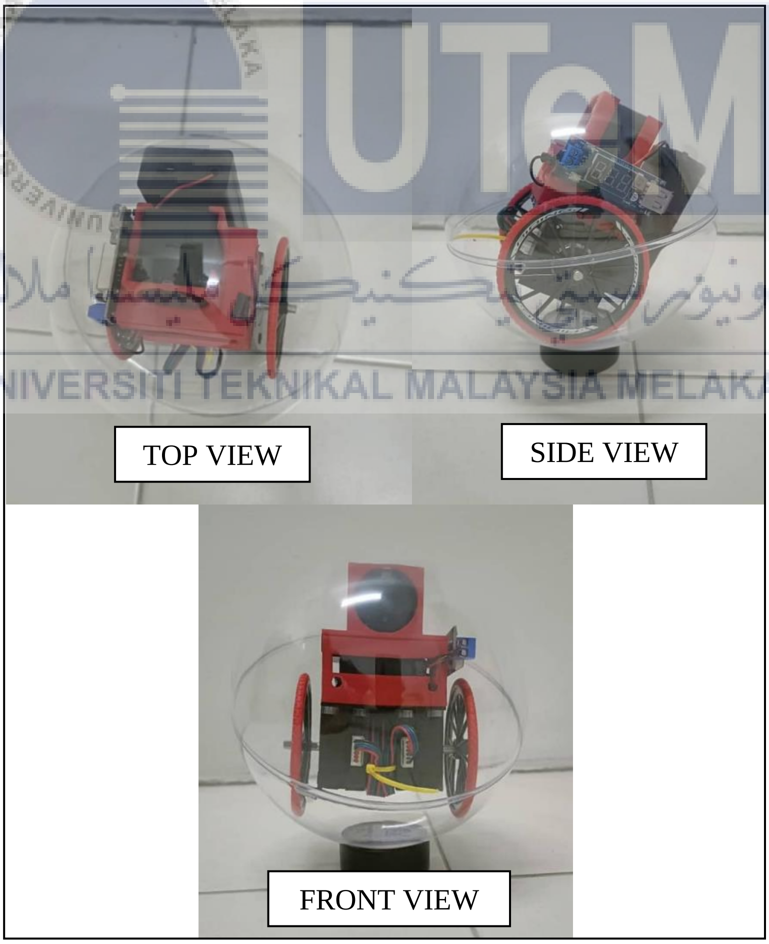


Figure 3.10: Complete view of the complete robot

F. Adhesive weight

Addressing the issue of weight imbalance in the robot's mass center, a series of adjustments have been implemented to ensure the robot's balance and maintain its upright position. A weight of approximately 80g has been strategically attached to the front of the robot chassis. This modification effectively shifts the center of mass, counterbalancing the previously existing weight imbalance. As a result, the robot's stability is significantly enhanced, enabling it to maintain an upright position even under varying operational conditions. Figure 3.11 shows the adhesive weight that is placed on the robot chassis.

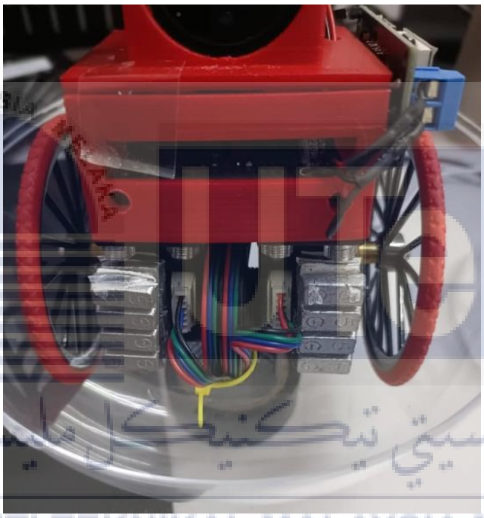


Figure 3.11: Adhesive weight

G. Castor Wheels

In response to recurring collisions between the robot's top and the shell during balancing operations, a strategic modification has been implemented: the addition of a castor wheel. This preventive measure serves as a protective buffer, effectively eliminating any direct contact between the robot chassis and the shell. The castor wheel absorbs the impact of these collisions, thereby preserving the integrity of the robot's structure and preventing potential damage. This innovative solution not only enhances the robot's durability but also optimizes its balancing capabilities, ensuring smoother and more efficient operation.

i. N20 Castor Wheel

The N20 Castor Wheel is a type of castor wheel that is often used in robotics, particularly for mobile robots. Material: The N20 Castor Wheel is made of metal and plastic. It consists of four small balls and a large steel ball with a diameter of 12mm. All the steel balls are made of 304 stainless steels to ensure long-term use without rust. The overall size of the N20 Castor Wheel is approximately 23.7mm x 15.7mm x 16mm. Mounting: The height from the mounting surface to the ball is 16mm, and the mounting hole is 4.0 mm. The distance between two screw holes is 1.5m35. The N20 Castor Wheel weighs about 10g. This type of castor wheel is perfect for small mobile robots, especially two-wheeled robots, as it provides better navigation and prevents the base from falling forward. Figure 3.12 shows the shape of N20 castor wheel.

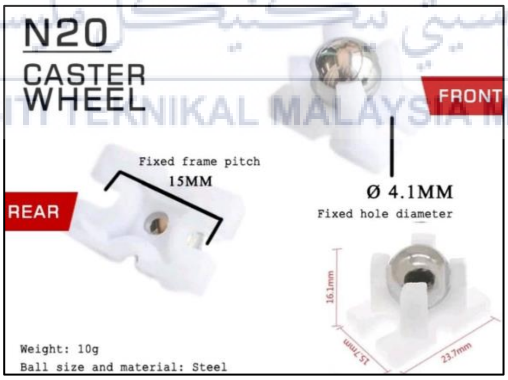


Figure 3.12: N20 castor wheel

ii. W420 Castor Wheel

The W420 Castor Wheel is a type of castor wheel that is often used in robotics, particularly for mobile robots. It is a spherical metal ball that mounts in a restraining fixture. The wheel is made of steel and nylon, ensuring durability and smooth movement. The overall height of the W420 Castor Wheel is 20mm, measured from the mounting surface to the ball. The mounting hole is 3.0 mm in diameter, and the distance between mounting holes is 40mm. This type of castor wheel is perfect for small mobile robots, especially two-wheeled robots, as it provides better navigation and prevents the base from falling forward. The W420 Castor Wheel is a light-duty caster wheel ball and is ideal for small scale projects. Figure 3.13 shows the shape of W420 castor wheel.

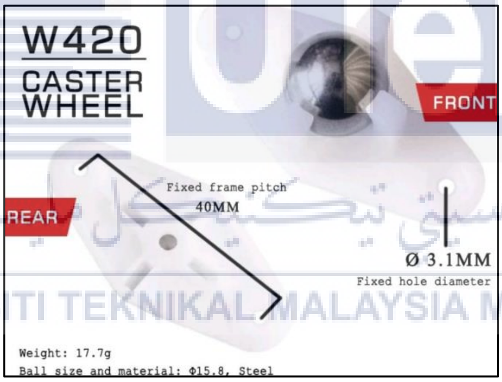


Figure 3.13: W420 castor wheel

3.3.2.2 Electronic Design

In this section, the electronic design of a robot involves the development of its control and power systems. This includes designing the robot’s power systems, sensors, actuators and microcontrollers, these elements collectively contribute to the robot’s overall functionality and performance.

A. Hibiscus Sense Microcontroller

The selected microcontroller for this project is Hibiscus Sense microcontroller. The information of the electronic component can refer to datasheet. Hibiscus sense microcontroller will be used in this project to control the stepper motors. Figure 3.14 shows the Hibiscus sense microcontroller includes with two actuators which are buzzer and LED light. The microcontroller also includes with three sensor which are PDS9960, BME280, and MPU6050 however in this project only used one sensor which is MPU6050 that embedded in the microcontroller board.

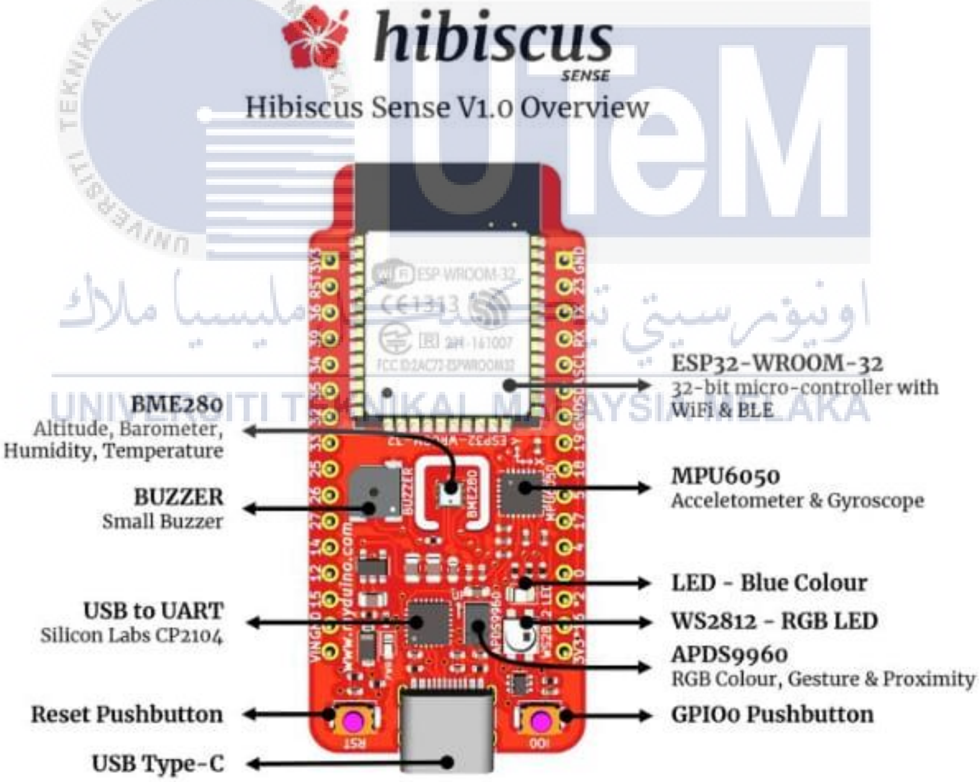


Figure 3.14: Hibiscus sense module

The Hibiscus Sense board uses the ESP WROOM-32 that operates at a voltage of 5V. The microcontroller comes with 3 sensors, providing a range of possibilities for various applications. It has 21 PWM pins and 6 input pins, offering flexibility for connecting peripheral devices. Each I/O pin can handle a DC current of up to 40 mA. With 520KB of SRAM, it provides ample memory for your programs. The clock speed of the microcontroller is 240 MHz, ensuring fast processing of tasks.

A. Inertia Measurement Unit

One of the three sensors incorporated on the Hibiscus board is the MPU6050. The acceleration and gyroscope of an object are measured by an IMU (Inertial Measurement Unit) with six degrees of freedom (DoF). The Hibiscus board's MPU6050 will be used to detect the board's physical movement and orientation. It can be programmed using programs like the ESP-IDF, Thonny IDE, or the Arduino IDE.

A gyroscope will be used as a sensor to measure the orientation and angular velocity of the spherical robot. This can help the robot to stabilize itself, navigate, and perform tasks that require precise motion control. A gyroscope can enhance the performance and functionality of a spherical robot by providing accurate sensing, flexible actuation, and versatile propulsion. An accelerometer can be used as a sensor to measure the linear acceleration of the spherical robot. This will help the robot to estimate its speed, position, and orientation by integrating the acceleration data.

However, this method is prone to errors due to noise and drift. Sensor fusion can prevent or reduce errors due to noise and drift by combining the measurements from different sensors. an accelerometer can measure linear acceleration and gravity, but it is affected by noise and drift. A gyroscope can measure angular velocity, but it is affected by bias and drift. By fusing the measurements from these sensors, the result obtain will be more accurate estimates of the orientation and motion of the robot.

B. Esp 32 Series Module

A Wi-Fi + Bluetooth + Bluetooth LE module called ESP 32-WROOM-32 is based on the ESP32-D0WDQ6 chip which includes in Hibiscus sense board. It contains a dual-core, 32-bit LX6 processor, 320 KB RAM, 4/8/16 MB of flash memory, and a number of peripherals, including GPIOs, ADCs, DACs, SPI, I2C, and UARTs. It operates between 3.0 and 3.6 V DC and between -40 and 85 °C ambient temperature. . Applications for it include low-power sensor networks, voice encoding, music streaming and MP3 decoding. A spherical robot will use ESP 32-WROOM-32 to provide wireless connectivity via Wi-Fi and interface with a graphical user interface (GUI) on a web browser or an application that shows the robot’s status, and control buttons. Additionally, the robot can stream video from a camera mounted on the robot to the GUI display, allowing the user to monitor the robot’s surroundings

C. Motor Driver

A4988 is a type of stepper motor driver that can control the speed and direction of a bipolar stepper motor by sending pulses to its coils. It has a built-in translator that simplifies the interface with a microcontroller or a controller logic. It can operate in full-step, half-step, quarter-step, eighth-step and sixteenth-step modes, with a maximum output of 35 V and ±2 A. It also has several protection features such as overcurrent, short circuit, under-voltage lockout, and over-temperature protection. The illustration of A4988 is shown in Figure 3.15.

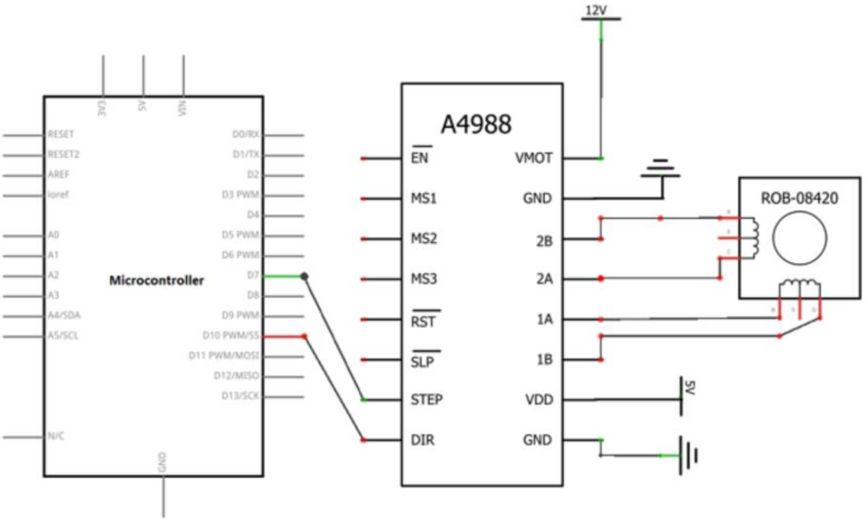


Figure 3.15: A4988 diagram

D. Battery

Lithium-ion (Li-ion) is used to operate the robot. The battery is a rechargeable battery that uses lithium ions to store energy. They are known for their high energy density, efficiency, and long life cycle. Constructed from positive and negative electrodes separated by a liquid chemical electrolyte, they are used in a variety of applications, including portable electronics, electric cars, and grid-scale energy storage. Safety measures are in place to prevent overheating and potential fires. Figure 3.16 shows the lithium Ion battery used in this project.

Figure 3.16: Lithium Ion battery

E. Voltage Regulator

A voltage regulator is utilized to step down the voltage from 7.4V to 5V. This adjustment is crucial because the microcontroller and the camera, which are integral components of the robot, operate on a 5V supply. By reducing the voltage to this level, the voltage regulator ensures the optimal functioning of these components. The motors are an exception to this as they operate at a higher voltage of 7.4V. Figure 3.17 shows the type of voltage regulator used.

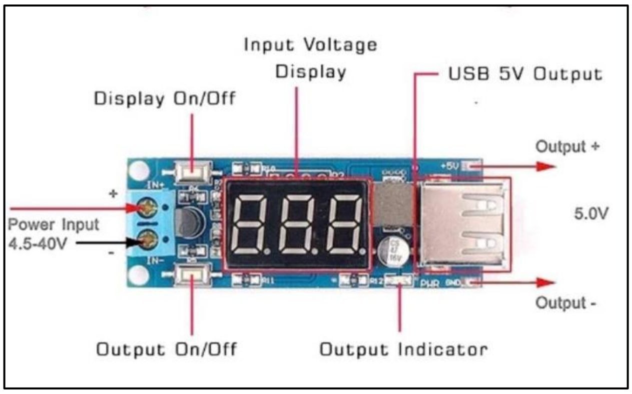


Figure 3.17: voltage regulator

F. Component Board Design

The electronic component uses a prototype board which is a Printed Circuit Board (PCB). The Hibiscus Sense board and motor driver are attached to the PCB by soldering the female header to the conductive pads on the board. Figure 3.18 shows the motor drivers attached to the PCB.

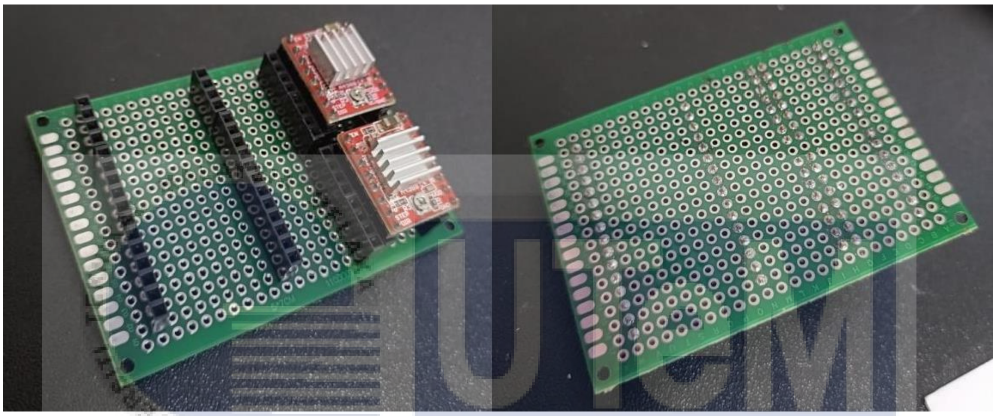


Figure 3.18: Printed Circuit Board (PCB)

3.3.3 Third Milestone: Simulation

This section explains the simulation process of the controller and the component that is used to control the spherical robot. The robot is controlled using MIT app inventor software to generate smartphone applications.

3.3.4 Designing GUI

Communication is important in order to allow the robot to communicate well with the smartphone. The interface was designed using MIT app inventor to ensure that the Wi-Fi module is capable of communicating with the smartphone. The GUI was designed to control the locomotion of the spherical robot. The flowchart of the GUI design is shown in Figure 3.19.

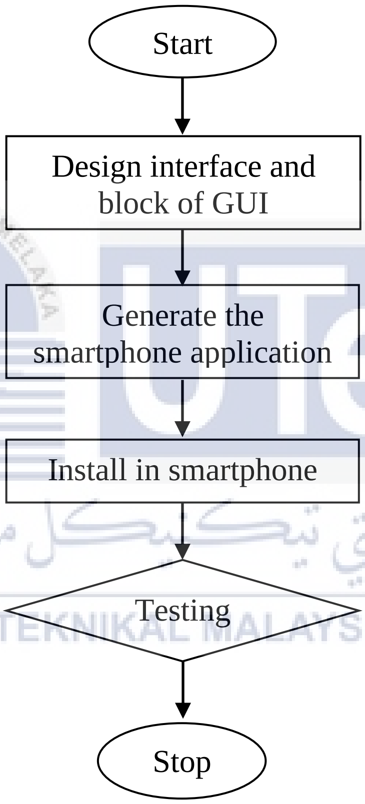


Figure 3.19: GUI design flowchart

3.3.5 Testing GUI interface

The Graphical User Interface (GUI) plays a crucial role in facilitating the interaction between a robot and a smartphone. It serves as the bridge that connects the user’s commands on the smartphone to the robot’s actions. To ensure seamless operation, the GUI undergoes some testing. The testing process is designed to verify the functionality of the GUI. It involves checking the responsiveness of the interface, the accuracy of commands sent to the robot, and the reliability of feedback received from the robot. This ensures that the user can effectively control the robot using their smartphone.

LED and RGB light on the board is tested in its functionality and controlled using a smartphone via a Wi-Fi connection. The purpose of this test was to verify the functionality and the connectivity of the LED and RGB light on the board. The RGB color code used in this testing is (255, 0, 255) for a magenta color and (255, 255, 153) for a blueish color.

After the connection was successful, the results showed that the light was responsive and adjustable to different colors. The analysis indicated that the board was working properly and met the requirements of the project. Figure 3.20 shows the board turning the light after command received.

Figure 3.20: The RGB light on the board changed color using a smartphone app.

3.3.6 Fourth Milestones: Testing

This section describes the process of implementing, testing, and analyzing the spherical robot system. There are several steps that aim to verify the functionality and performance of each component and the whole system .Testing the functionality of each component and data analysis are the first steps in the process.

3.3.6.1 Integration of The Mechanical and Electronic Hardware

The electronic components, battery, and controller will be installed within the spherical shell once the structure of the component and the spherical shell have been completed. Figure 3.21 provides a clear illustration of the implementation procedure.

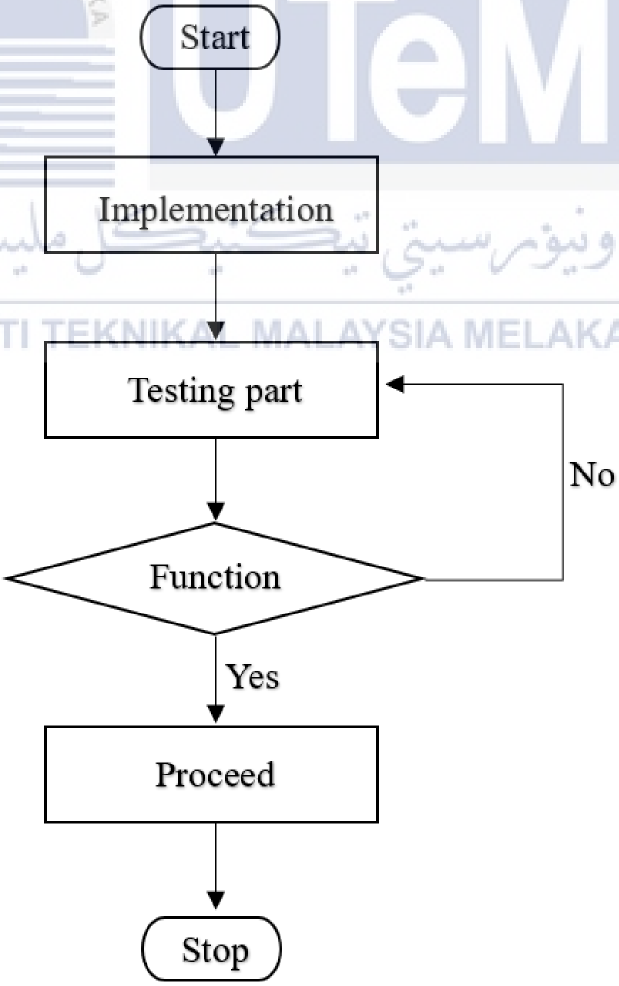


Figure 3.21: Implementation of component flowchart

A. Stepper Motor Testing

In this section, In the process of robot development, both stepper motors undergo rigorous testing to ensure optimal performance. The testing involves four distinct motor movements: forward, backward, turning left, and turning right. Each stepper motor is configured with a maximum speed of 400 and a maximum acceleration of 700. These settings are carefully calibrated to provide the robot with the agility and responsiveness needed for its tasks. Figure 3.22 show the illustration for robot's movement testing.

The testing of these movements is crucial as it verifies the motors' ability to execute commands accurately and efficiently. This ensures that the robot can navigate its environment effectively, contributing to its overall functionality and performance. The testing procedure was recorded and documented for in-depth analysis. A video that encapsulates the entire procedure is incorporated in this report. For viewing the video that illustrates the testing process, kindly follow the link included in this report.

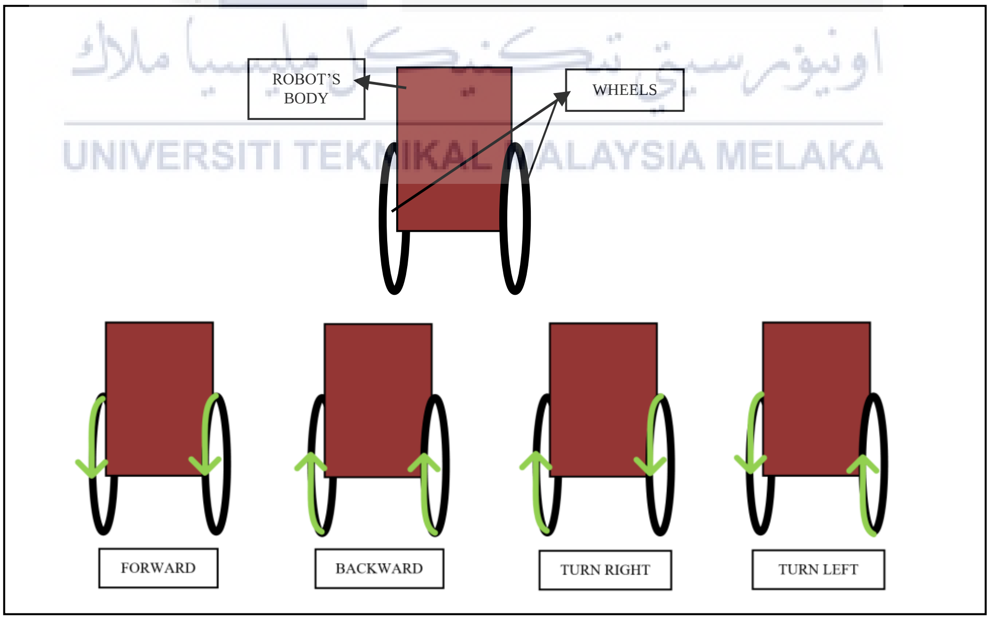


Figure 3.22: Illustration of robot's movement

i. Forward movement:

https://youtube.com/shorts/FpmA0wTGBcQ?si=kg76ZnEL7rOo6qPp

ii. Bacward movement:

https://youtube.com/shorts/8cjacw3E2KY?si=72L-RVxAmoiaKo8G

iii. Turn left movement:

https://youtube.com/shorts/HoBbL0kxX0A?si=vqYdzeZreooQDj7p

iv. Turn right movement:

https://youtube.com/shorts/UE1rPaMZ8l8?si=5BHNQxdxf5RIE_2u

B. Stepper Motor Controlled using Smartphone

The control of both stepper motors is facilitated through a smartphone application, utilizing Wi-Fi communication. This setup allows for remote operation of the robot, providing the user with the flexibility to control the robot's movements from a distance. The smartphone application sends commands to the stepper motors via Wi-Fi, directing the robot's movements. This wireless communication method ensures a seamless and responsive interaction between the user and the robot. The motor movement video link is below. Figure 3.18 shows the block diagram of the interface GUI.

Video link: https://youtube.com/shorts/h8dZhmqxais?si=eaxTbtExl-QYv2vk

3.3.6.2 PID Testing

The PID tuning of our pick-and-place machine has been successfully completed. The adjustments made to the Proportional, Integral, and Derivative parameters have enhanced the machine's accuracy in reaching its target position. Following the completion of the PID tuning, a test was conducted to observe the robot's behavior. The test involved the robot maintaining an upright position while being slightly pushed by hand. This test was designed to evaluate how the PID controller operates when the robot is in balancing mode. The results of the test will provide valuable insights into the stability of the robot and the effectiveness of the PID tuning. This is a critical step in ensuring that our pick-and-place machine operates reliably and accurately, contributing to the overall productivity and efficiency of our operations. The testing process was thoroughly documented and recorded for further analysis. A video capturing the entire process has been included in this report. Please refer to the provided link to view the video of the testing process.

Video link: https://youtube.com/shorts/PWovqx3IyNA?si=SDGGuhRhp3Fh713p

3.3.6.3 Testing and analysis of the robot locomotion

The aim of testing and analysis is to confirm the robot's stability on various surfaces and paths while monitoring surrounding areas. Testing the robot's locomotion is shown in Figure 3.14. Each path's performance and stability by the robot will be watched and observed. To enhance the robot's performance during locomotion, the data will then be examined.

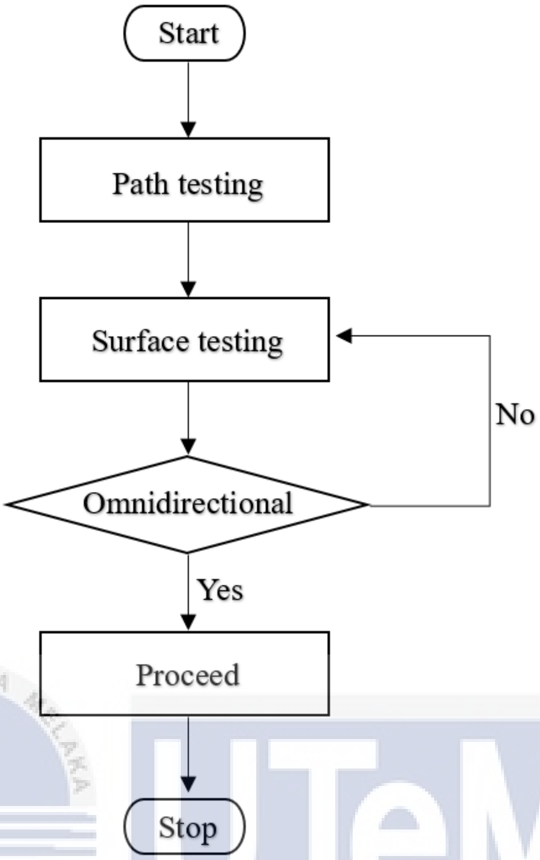


Figure 3.23: Testing algorithm

The primary objective of these tests is to evaluate the robot’s ability to navigate towards a predetermined target, set approximately 2 meters from its initial position. The robot is controlled using a smartphone, which adds another layer of complexity to the testing process. This comprehensive testing approach ensures that the robot can effectively maneuver across different surfaces, respond accurately to the smartphone controller, and successfully reach its target, thereby demonstrating its robustness and versatility. Figure 3.24 shows the illustration of how the spherical robot’s testing.

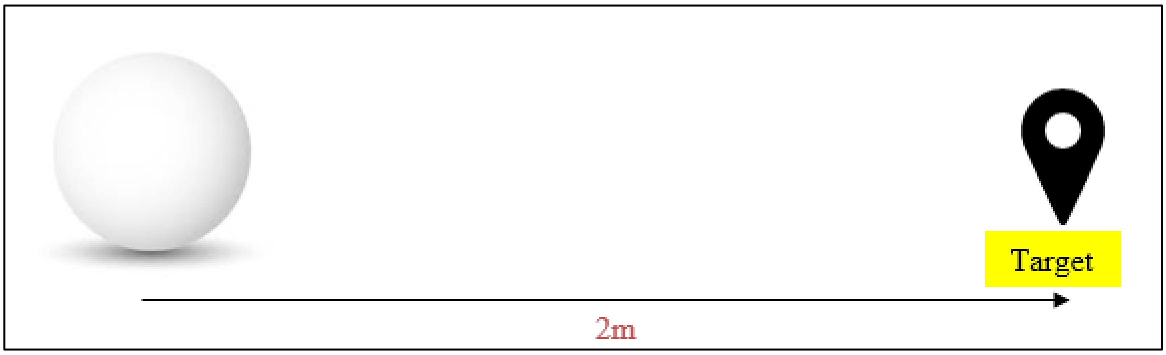


Figure 3.24: Spherical robot testing

3.3.6.4 Camera Testing

The functionality of the wireless camera on the spherical robot can be tested by examining the quality and stability of the video feed on the smartphone controller. This involves observing the clarity, resolution, and smoothness of the video stream under various conditions. It's crucial to ensure that the camera can effectively capture the robot's surroundings, providing a comprehensive view of the environment. This test should be conducted under different lighting conditions - bright daylight and night litght.The goal is to confirm that the camera can adapt to these changes and still provide clear and stable footage. This will enable the robot to accurately perceive its surroundings and navigate effectively, regardless of the lighting conditions. This rigorous testing will ensure the robot's robust performance and safe operation.

3.4 Summary

This section outlines the process of implementing, testing, and analyzing a spherical robot system. The process involves integrating mechanical and electronic hardware into the robot's shell, testing the stepper motors' performance and the robot's locomotion on various surfaces, and assessing the functionality of the wireless camera. The robot is remotely controlled via a smartphone application, and the comprehensive testing ensures its ability to effectively manoeuvre, respond accurately to the controller, and successfully reach a target, demonstrating its robustness and versatility.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This section presents a comprehensive evaluation of a spherical robot's performance across various terrains. The robot's adaptability, efficiency, maneuverability and stability assessed on surfaces ranging from smooth, flat areas to challenging terrains like carpet, sand, grass, cement, and tar roads. The data collected during these tests is used to determine the robot's optimal operating surface. A camera mounted on the robot captures videos of the testing process, providing real-time monitoring and facilitating one-way communication between the robot and a smartphone. The findings from these tests are crucial in understanding the interaction between the robot and its environment.

4.2 Spherical Robot Performance in Locomotion

Upon the completion of the spherical robot's development, a comprehensive evaluation of its locomotion performance is conducted across a variety of terrains. This testing phase is crucial to assess the robot's adaptability and efficiency in different environmental conditions. The robot is subjected to surfaces ranging from smooth, flat areas to more challenging terrains such as carpet, cement, sand, grass, and tar road environments. The objective is to observe and analyze the robot's maneuverability and stability under these varying conditions. This data is then used to determine the robot's optimal operating surface and the terrain in which the robot exhibits the highest performance metrics.

4.2.1 The Locomotion Performance on Different Terrains

The testing phase of the spherical robot’s development is a critical stage where its ability to navigate towards a target across various terrains is observed. This involves setting a designated target and monitoring the robot’s movement as it attempts to reach this goal in different environmental conditions. Figure 4.1 shows the illustration of robot, performance testing. These tests are not just about reaching the target, but also about how effectively and efficiently the robot can do so. The observation of robot performance is tabulated as shown in Table 4.1. The data collected during these tests and tabulated as shown in Table 4.2.

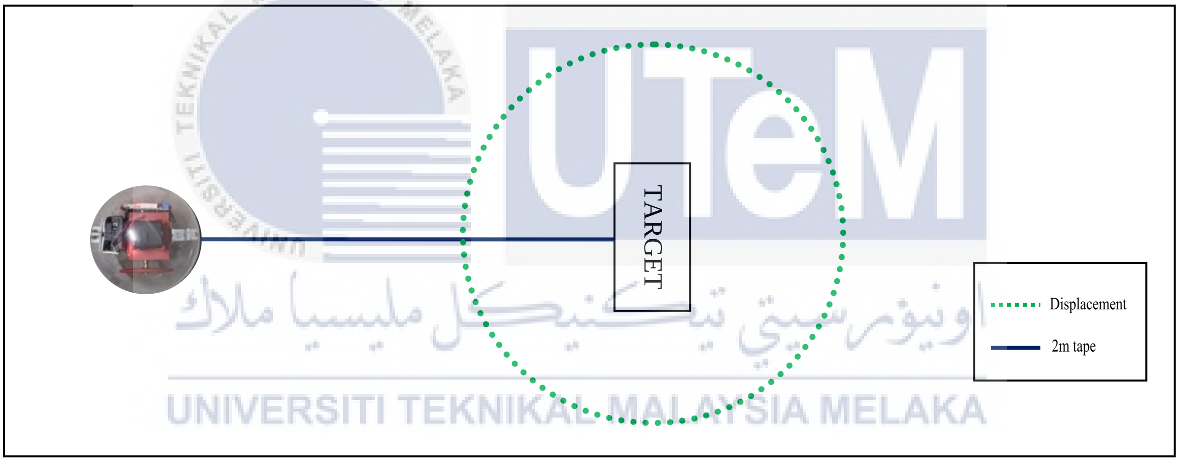


Figure 4.1: Robot performance testing procedure

Table 4.1: The performance of the robot to reach target

	Terrain type				
No. of test	**Carpet**	**Cement**	**Grass**	**Sand**	**Tar road**
1	✓	✗	✗	✗	✗
2	✓	✗	✗	✗	✗
3	✓	✓	✗	✓	✗
4	✗	✓	✗	✓	✓
5	✓	✓	✗	✗	✗

Table 4.2: The displacement of the robot from the target

	Displacement from target (m)				
No. of test	**Carpet**	**Cement**	**Grass**	**Sand**	**Tar road**
1	0.10	0.50	2.00	1.10	0.80
2	0.20	0.80	2.00	1.20	1.10
3	0.20	0.10	2.00	0.30	1.00
4	0.50	0.20	2.00	0.40	0.40
5	0.00	0.20	2.00	1.10	1.20

In this robot’s testing, the success of a robot is determined by its ability to get close to the target. Specifically, if the displacement of the robot from the target is less than 0.5 meters, the robot is deemed successful. Any displacement greater than this is considered a failure. However, there’s a special case to consider, if the robot does not move from its initial position, the displacement value is set as 2 meters, given that the target is positioned two meters away from the robot’s starting point. This setup ensures that the robot’s performance is accurately measured, taking into account both its movement and proximity to the target. A video capturing some of the testing process has been included in this report. Please refer to the provided link to view the video of the testing process.

Video link: https://youtube.com/watch?v=jvfXkzhOehU&feature=shared

4.3 Preferable Surface of The Robot

Based on previous testing, it has been observed that The type of terrain significantly impacts the movement of spherical robots. The surface texture and material can either facilitate or hinder the robot's ability to reach its target. These findings highlight the importance of considering the interaction between the robot and its environment in the design and operation of spherical robots. Five different surfaces, namely carpet, cement, grass, sand, and tar, were tested to evaluate their impact on the locomotion of a spherical robot

Carpet surfaces have proven to be the most conducive environment. The unique texture of carpets provides the necessary grip that these robots require for effective movement. This is largely due to the interlocking mechanism between the carpet fibers and the robot's surface, which significantly reduces slippage. Through rigorous testing, it has been observed that carpet surfaces yield the best results in terms of robot performance. Therefore, carpet surfaces are highly recommended due to their optimal characteristics. This preference is backed by evidence gathered from testing and analysis.

Following carpet, cement surfaces have also shown to be a viable terrain for spherical robots. The inherent hardness and flatness of cement surfaces facilitate smooth rolling, enabling the robot to navigate effectively. However, it's important to note that the rough texture of cement can pose a challenge by increasing the rolling resistance. This, in turn, can impact the robot's performance by slowing its movement or requiring more energy for locomotion. Despite these challenges, cement surfaces are still considered a suitable terrain for spherical robots, second only to carpet surfaces in terms of conduciveness.

Sand is ranked third, following carpet and cement. Despite the challenges it presents, such as the potential for sinking or slipping and the difficulty in halting movement, sand is still considered a more manageable surface for spherical robots compared to some other terrains. However, it’s important to note that both carpet and cement surfaces provide more stability and predictability, making them more suitable for such tests. This ranking underscores the varying degrees of difficulty robots face across different surfaces and the ongoing need for advancements in robotic design and control mechanisms.

In the ranking of surfaces for robot locomotion testing, tar surfaces take the fourth position, following carpet, cement, and sand. Despite the challenges posed by tar’s temperature variations and the presence of small pebbles or cracks, spherical robots generally perform better on tar than on grass. This is due to the relative predictability and stability of tar surfaces compared to the irregular and slippery nature of grass. However, maintaining a straight path on tar remains a challenge, indicating room for further improvements in robotic design and control mechanisms. This ranking underscores the complexities of robot locomotion across different terrains

Navigating grass surfaces presents a unique set of challenges for spherical robots and are thus placed at the bottom of the ranking. The irregularity and slipperiness of grass, coupled with the mini obstacles posed by individual grass blades, significantly hinder the movement of robots, particularly those of a spherical design. These robots are limited to turning left and right, with forward and backward movements proving to be a considerable challenge. Consequently, grass is often the least preferred surface for testing robot locomotion due to these complexities.

Figure 4.2: Robot placed on different terrains

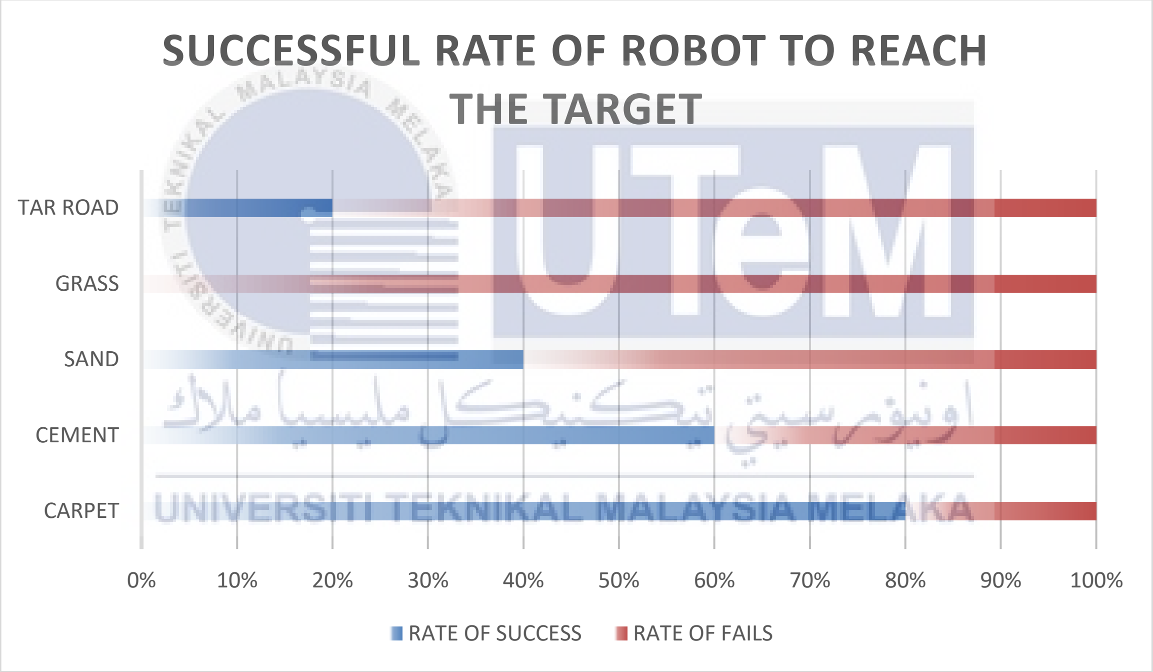


Figure 4.3: Graph of robot's performance

4.4 Camera

During the testing phase of the robot, a camera is strategically placed on it to capture videos and monitor the robot's surroundings. This camera is rigorously tested under various lighting conditions, including daylight and night light, to ensure optimal performance. Figure 4.3 shows the image captured during the day. In low-light environments, the camera automatically switches to infrared mode, enhancing its ability to capture clear images despite the lack of light. This feature is particularly useful in ensuring the robot can function effectively in all lighting conditions. Figure 4.4 shows the image captured during testing process in the night. One of the key features of this setup is the ability to share live video feeds that can be monitored using a smartphone. This real-time monitoring allows for immediate response to any issues that may arise during the robot’s operation. The recorded video, captured from the robot’s point of view, can be viewed using the provided link.

Video link: https://youtube.com/watch?v=p6VF4noighQ&feature=shared

Figure 4.4: Robot’s point of view during the day

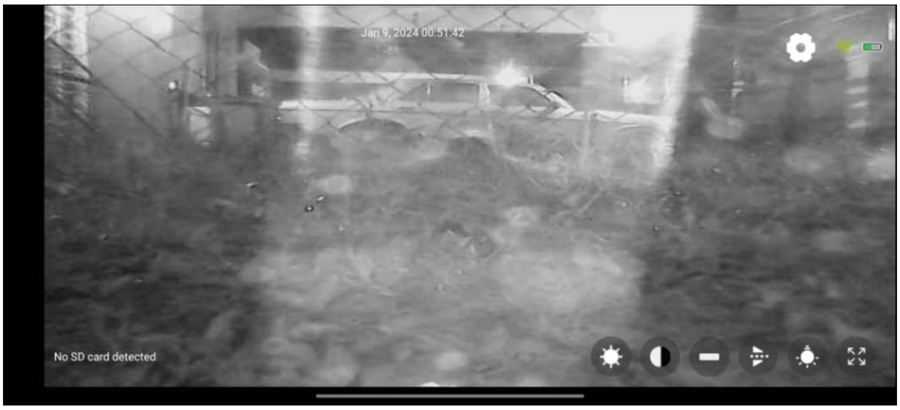


Figure 4.5: Robot’s point of view during the night

Additionally, the camera has a unique feature that enables one-way communication between the smartphone and the camera. Since the camera is mounted on the robot, this feature essentially allows for one-way communication between the robot and the smartphone. This can be particularly useful in situations where immediate feedback or instructions need to be relayed to the robot. Overall, the integration of the camera into the robot not only enhances the robot’s functionality but also provides a means for real-time monitoring and communication, making it a valuable tool in the robot development process.

CHAPTER 5

CONCLUSION

5.1 Conclusion

In conclusion, the development and testing of the spherical robot has provided valuable insights into its performance across various terrains. The robot demonstrated adaptability and efficiency, with its performance metrics varying based on the surface it navigated. The most conducive surfaces for the robot's movement were found to be carpet and cement, while it faced challenges on more irregular and loose terrains like sand, grass, and tar roads.

A significant achievement in this project was the successful implementation of sensor fusion in the Inertial Measurement Unit (IMU) sensor. This advancement effectively mitigated the drift from the IMU sensor, enhancing the accuracy and reliability of the robot's control. This has enabled the robot to perform efficiently across various terrains, underscoring the potential of sensor fusion in improving the adaptability and performance of robotic systems.

The integration of a camera on the robot enhanced its functionality, allowing for real-time monitoring and one-way communication between the robot and a smartphone. This feature proved particularly useful in low-light conditions, where the camera automatically switched to infrared mode.

Overall, the project highlighted the importance of considering the interaction between the robot and its environment in the design and operation of spherical robots. The data collected will guide future improvements to optimize the robot's performance. Despite the challenges faced, the project was a significant step forward in the field of spherical robot design and testing. The learnings from this project have the potential to contribute to the broader field of robotics, paving the way for more advanced and adaptable robotic systems in the future.

5.2 Potential and commercialization

The spherical robot, with its adaptability and efficiency across various terrains, has significant commercialization potential. Its ability to navigate different surfaces makes it suitable for a variety of industries, including surveillance, exploration, and entertainment. The successful implementation of sensor fusion in the IMU sensor further enhances its market value, as it improves the robot's control and efficiency.

The practical applications of spherical robots are vast. In surveillance, the robot's camera and one-way communication feature can provide real-time monitoring of a location. In exploration, the robot's adaptability to different terrains can be useful in navigating challenging environments. In entertainment, the robot's unique design and movement can provide a novel and engaging experience for users.

The spherical robot can address several community needs. For instance, in disaster-stricken areas, the robot can be used for search and rescue operations, navigating through debris and providing real-time visuals to the rescue team. In elderly care, the robot can serve as a monitoring device, providing caregivers with real-time updates about the elderly person's whereabouts and well-being.

In conclusion, the spherical robot project has successfully demonstrated the potential of sensor fusion in enhancing robotic performance across various terrains. The project has also highlighted the robot's wide range of applications in industries such as surveillance, exploration, and entertainment, as well as its ability to address community needs.

5.3 Recommendation for future works

i. Terrain Adaptability: Given the challenges faced by the robot on irregular and loose terrains like sand, grass, and tar roads, future work could focus on enhancing the robot’s adaptability to these surfaces. This could involve designing new mechanisms or materials for the robot’s exterior to improve its grip and maneuverability.

ii. Sensor Fusion: The successful implementation of sensor fusion in the IMU sensor has significantly improved the robot’s control and efficiency. Future work could explore the integration of additional sensors and the development of more sophisticated sensor fusion algorithms to further enhance the robot’s performance.

iii. Camera Functionality: The integration of a camera on the robot has proven particularly useful, especially in low-light conditions. Future improvements could include enhancing the camera’s capabilities in various lighting conditions, perhaps through the integration of advanced light sensors or adaptive image processing algorithms.

iv. Two-Way Communication: The current one-way communication between the robot and a smartphone could be expanded to two-way communication in future iterations. This would allow the robot to send feedback or alerts to the smartphone based on its environment or status.

REFERENCES

[1] S. Kuswadi, M. N. Tamara, B. T. Brahmantio, D. Sugiharto, and N. U. H. Mohammad, “Disaster Robot Navigation using Behavior-based Systems,” in *3rd International Conference on Applied Science and Technology, iCAST 2020*, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 196–202. doi: 10.1109/iCAST51016.2020.9557734.

[2] Institute of Electrical and Electronics Engineers. Indonesia Section, Universitas Syiah Kuala, and Institute of Electrical and Electronics Engineers, *2017 International Conference on Electrical Engineering and Informatics : “Advancing Knowledge, Research, and Technology for Humanity” : proceedings : Banda Aceh, Aceh, Indonesia, October 18-20, 2017*.

[3] M. Bujňák *et al.*, “Spherical Robots for Special Purposes: A Review on Current Possibilities,” *Sensors*, vol. 22, no. 4. MDPI, Feb. 01, 2022. doi: 10.3390/s22041413.

[4] R. Chase and A. Pandya, “A review of active mechanical driving principles of spherical robots,” *Robotics*, vol. 1, no. 1. MDPI AG, pp. 3–23, Dec. 01, 2012. doi: 10.3390/robotics1010003.

[5] *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[6] S. Firlej, “Design, construction and control of a spherical rolling robot with internal two-wheel cart,” *Automatyka/Automatics*, vol. 19, no. 2, p. 63, 2015, doi: 10.7494/automat.2015.19.2.63.

[7] IEEE Robotics and Automation Society and Institute of Electrical and Electronics Engineers, *IEEE ROBIO 2017 : 2017 IEEE International Conference on Robotics and Biomimetics : December 5-8, 2017, Macau SAR, China*.

[8] M. Nagai, “MASTER’S THESIS Control System for a Spherical Robot”.

[9] I. Rangapur, B. K. S. Prasad, and R. Suresh, “Design and Development of Spherical Spy Robot for Surveillance Operation,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1212–1220. doi: 10.1016/j.procs.2020.04.130.

[10] Muhammad Affan Arif; Aibin Zhu; Han Mao; Xu Zhou; Jiyuan Song; Yao Tu; Peifeng Ma ,“Design of an Amphibious Spherical Robot”, IEEE/ASME Transactions on Mechatronics.2023, doi: 10.1109/TMECH.2023.3247641

[11] S. S. Ahn and Y. J. Lee, “Novel Spherical Robot with Hybrid Pendulum Driving Mechanism,” *Advances in Mechanical Engineering*, vol. 2014, 2014, doi: 10.1155/2014/456727.

[12] American University of Sharjah., *2012 8th International Symposium on Mechatronics and its Applications (ISMA) : 10-12 April 2012*. [IEEE], 2012.

[13] M. Yang *et al.*, “Design and Analysis of a Spherical Robot with Two Degrees of Freedom Swing,” in *Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020*, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 4913–4918. doi: 10.1109/CCDC49329.2020.9164196.

[14] Shanghai Dianji University. School of Electronics & Information, M. IEEE Systems, and Institute of Electrical and Electronics Engineers, *2016 3rd International Conference on Systems and Informatics : 19-21 November 2016 : Shanghai Jingyue International Aviation Hotel, Shanghai, China : ICSAI 2016*.

[15] V. A. Joshi, R. N. Banavar, and R. Hippalgaonkar, “Design and analysis of a spherical mobile robot,” *Mech Mach Theory*, vol. 45, no. 2, pp. 130–136, Feb. 2010, doi: 10.1016/j.mechmachtheory.2009.04.003.

[16] K. M. Easow, “Design and Implementation of Spherical Robot Actuated by Omni-Wheels.” [Online]. Available: www.ijert.org

[17] AKMAL, KARLSSON, TARA, MOHAMMED-AMIN “Holonomic Spherical Mobile Robot.”, Mechanical Engineering, at the Basic Level, 15 HP Stockholm Sweden 2020.

[18] IEEE Robotics and Automation Society and Institute of Electrical and Electronics Engineers, *IEEE ROBIO 2017 : 2017 IEEE International Conference on Robotics and Biomimetics : December 5-8, 2017, Macau SAR, China.*

[19] N. Bun-Athuek and P. Laksanacharoen, “A new locomotion concept using shell transformation for a Reconfigurable Spherical Robot New navigation concepts using Conversion of spherical motor".”

[20] Y. Zhai, Z. Ding, Y. Liu, and S. Jin, “Design and Research of Underwater Spherical Robot,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1742-6596/1549/2/022067.

[21] G. Welch and G. Bishop, “An Introduction to the Kalman Filter.” [Online]. Available: http://www.cs.unc.edu/~gb

[22] Mody University of Science and Technology. School of Engineering and Technology and Institute of Electrical and Electronics Engineers, *2020 International Conference on Emerging Trends in Communication, Control and Computing : copyright information, proceedings : February 21-22, 2020.*

[23] Y. Tang, J. Wang, M. Nguyen, and I. Altintas, “PEnBayes: A multi-layered ensemble approach for learning bayesian network structure from big data,” *Sensors (Switzerland)*, vol. 19, no. 20, Oct. 2019, doi: 10.3390/s19204400.

[24] Z. Deng and J. Wang, “A novel evidence conflict measurement for multi-sensor data fusion based on the evidence distance and evidence angle,” *Sensors (Switzerland)*, vol. 20, no. 2, Jan. 2020, doi: 10.3390/s20020381.

[25] P. Narkhede, S. Poddar, R. Walambe, G. Ghinea, and K. Kotecha, “Cascaded complementary filter architecture for sensor fusion in attitude estimation,” *Sensors*, vol. 21, no. 6, pp. 1–18, Mar. 2021, doi: 10.3390/s21061937.

[26] A. Noordin, M. A. M. Basri, and Z. Mohamed, “Sensor fusion algorithm by complementary filter for attitude estimation of quadrotor with low-cost IMU,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 2, pp. 868–875, Apr. 2018, doi: 10.12928/TELKOMNIKA.v16i2.9020.

[27] B. Mccarron, “Low-Cost IMU Implementation via Sensor Fusion Algorithms in the Arduino Environment,” 2013.

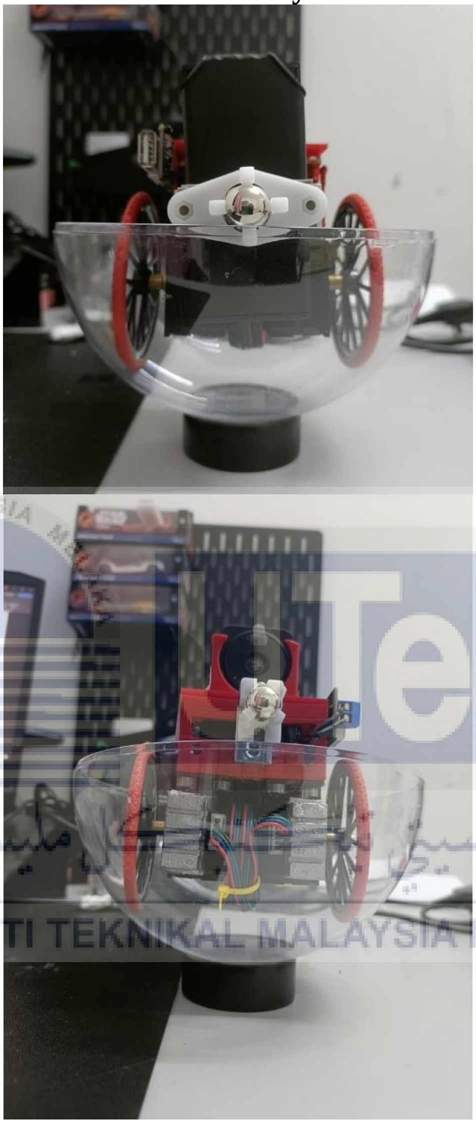
[28] R. R. Slavescu, Universitatea Tehnică din Cluj-Napoca. Computer Science Department, IEEE Romania Section, and Institute of Electrical and Electronics Engineers, *Proceedings, 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP) : Cluj-Napoca, Romania, September 6-8, 2018.*

[29] D. G. (Donald G. Bailey *et al.*, *ICARA 2015 : proceedings of the 2015 6th International Conference on Automation, Robotics and Applications : Rydges Lakeland Resort, Queenstown, New Zealand, February 17 - 19, 2015.*

[30] C. C. Tsai, H. C. Huang, and S. C. Lin, “Adaptive neural network control of a self-balancing two-wheeled scooter,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1420–1428, Apr. 2010, doi: 10.1109/TIE.2009.2039452.

APPENDIX

A. Castor wheels attached to the robot body.



B. Progamming code

```
#include <Wire.h>
#include <MPU6050_light.h>
#include <Kalman.h>
#include <AccelStepper.h>
#include <PID_v1.h>
#include <WiFi.h>
#include <Adafruit_NeoPixel.h>

#define LED_PIN 2

AccelStepper stepper1(AccelStepper::DRIVER, 14, 12);
AccelStepper stepper2(AccelStepper::DRIVER, 25, 26);
Adafruit_NeoPixel rgb(1, 16);

MPU6050 mpu(Wire);
unsigned long timer = 0;

Kalman kalmanX;
Kalman kalmanY;
Kalman kalmanZ;

double setPoint, input, output;
double Kp = 6, Ki = 1.2, Kd = 1.5;
PID pid(&input, &output, &setPoint, Kp, Ki, Kd, DIRECT);

const char* ssid = "the beast";
const char* password = "0987654321.";

WiFiServer server(80);
String currentCommand = "stop";
WiFiClient client;

 void setup() {
  Wire.begin();
  Serial.begin(115200);

  byte status = mpu.begin();
  Serial.print(F("MPU6050 status: "));
  Serial.println(status);

  if (status != 0) {
    Serial.println(F("Error: Could not connect to MPU6050"));
    return;
  }

  Serial.println(F("Calculating offsets, do not move MPU6050"));
  delay(1000);
```

```
  mpu.calcOffsets();
  Serial.println("Done!\n");

  stepper1.setMaxSpeed(300);
  stepper1.setAcceleration(300);
  stepper2.setMaxSpeed(300);
  stepper2.setAcceleration(300);

  setPoint = 0;
  pid.SetMode(AUTOMATIC);
  pid.SetOutputLimits(-255, 255);

  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    digitalWrite(LED_PIN, !digitalRead(LED_PIN));
  }

  digitalWrite(LED_PIN, HIGH);
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  server.begin();
}

void handleCommands(String request) {
  if (request.indexOf("forward") != -1) {
    currentCommand = "forward";
    setPoint = 0.10;
  } else if (request.indexOf("backward") != -1) {
    currentCommand = "backward";
    setPoint = -0.10;
  } else if (request.indexOf("left") != -1) {
    currentCommand = "left";
    setPoint = 0.05;
  } else if (request.indexOf("right") != -1) {
    currentCommand = "right";
    setPoint = 0.05;
  } else if (request.indexOf("stop") != -1) {
    currentCommand = "stop";
    setPoint = 0;
```

```
 }
}
void handleWifi() {
WiFiClient client = server.available();
if (client) {
  Serial.println("New Client.");
  String currentLine = "";
  while (client.connected()) {
   if (client.available()) {
    char c = client.read();
    Serial.write(c);
    if (c == '\n') {
     if (currentLine.length() == 0) {
      client.println("HTTP/1.1 200 OK");
      client.println("Content-type:text/html");
      client.println();
client.print("Click <a href=\"/forward\">here</a> to move forward.<br>");
      client.print("Click <a href=\"/backward\">here</a> to move
backward.<br>");
      client.print("Click <a href=\"/left\">here</a> to move
left.<br>");
      client.print("Click <a href=\"/right\">here</a> to move
right.<br>");
      client.print("Click <a href=\"/stop\">here</a> to stop.<br>");
      client.println();
      break;
     } else {
      currentLine = "";
     }
    } else if (c != '\r') {
     currentLine += c;
    }
    if (currentLine.endsWith("GET /forward")) {
     currentCommand = "forward";
     setPoint = 0.20;
    } else if (currentLine.endsWith("GET /backward")) {
     currentCommand = "backward";
     setPoint = -0.20;
    } else if (currentLine.endsWith("GET /left")) {
     currentCommand = "left";
     setPoint = 0.05;
    } else if (currentLine.endsWith("GET /right")) {
     currentCommand = "right";
     setPoint = 0.05;
    } else if (currentLine.endsWith("GET /stop")) {
     currentCommand = "stop";
     setPoint = 0;
    }
   }
```

```
    }
    client.stop();
    Serial.println("Client disconnected");
  }
}
void handleMotorControl() {
  double dt = (double)(micros() - timer) / 1000000;
  timer = micros();
  mpu.update();
double pitch = kalmanY.getAngle(mpu.getAngleY(), mpu.getGyroY(), dt) *
DEG_TO_RAD;
  input = pitch;
  pid.Compute();

  if (currentCommand == "forward") {
    stepper1.setSpeed(200);
    stepper2.setSpeed(-200);
    stepper1.runSpeed();
    stepper2.runSpeed();
  } else if (currentCommand == "backward") {
    stepper1.setSpeed(-200);
    stepper2.setSpeed(200);
    stepper1.runSpeed();
    stepper2.runSpeed();
  } else if (currentCommand == "left") {
    stepper1.setSpeed(-200);
    stepper2.setSpeed(-200);
    stepper1.runSpeed();
    stepper2.runSpeed();
  } else if (currentCommand == "right") {
    stepper1.setSpeed(200);
    stepper2.setSpeed(200);
    stepper1.runSpeed();
    stepper2.runSpeed();
  } else {
    // No command or stop
    if (abs(input) > 0.05) {
      output = map(output, -255, 255, -400, 400);
      stepper1.move(-output);
      stepper2.move(output);
      stepper1.run();
      stepper2.run();
    } else if (abs(input) < -0.40) {
      output = map(output, -255, 255, -400, 400);
      stepper1.move(output);
      stepper2.move(-output);
      stepper1.run();
      stepper2.run();
    } else {
```

```
      stepper1.setSpeed(0);
      stepper2.setSpeed(0);
      stepper1.runSpeed();
      stepper2.runSpeed();
    }
  }
}

void loop() {
  handleWifi();

  double dt = (double)(millis() - timer) / 1000; // Calculate delta time in
seconds
  timer = millis();

  double pitch = kalmanY.getAngle(mpu.getAngleY(), mpu.getGyroY(), dt) *
DEG_TO_RAD;
  handleMotorControl();
}
```