UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DESIGN AND DEVELOPMENT OF A TESTING RIG FOR POSITION LOCALIZATION USING ROBOT OPERATING SYSTEM

This report is submitted in accordance with requirement of the
Universiti Teknikal Malaysia Melaka (UTeM) for Bachelor Degree of
Manufacturing Engineering (Hons.)

by

ABDULLAH AZAM BIN HAIRUDIN

[redacted]

[redacted]

FACULTY OF MANUFACTURING ENGINEERING

2022

DECLARATION

I hereby, declared this report entitled "Design and Development of a Testing Rig for Position Localization Using Robot Operating System" is the results of my own research except as cited in reference.

-AZAM-

Signature : ....

Author's Name : ABDULLAH AZAM BIN HAIRUDIN

Date :23 Jan 2022

APPROVAL

This report is submitted to the Faculty of Manufacturing Engineering of Universiti Teknikal Malaysia Melaka as a partial fulfilment of the requirement for Degree of Manufacturing Engineering (Hons). The members of the supervisory committee are as follow:

..........SHARIMAN BIN ABDULLAH......
Pensyarah
Fakulti Kejuruteraan Pembuatan
Universiti Teknikal Malaysia Melaka

(Dr. Shariman bin Abdullah)

ABSTRAK

Penyelidikan projek ini menumpukan pada mereka dan menghasilkan robot bergerak sebagai platform rig pengujian untuk menjalankan proses penyetempatan dan navigasi kendiri. Robot akan dipasang dengan Sistem Operasi Robot (ROS), yang merupakan pakej yang merangkumi simpul ROS, *independent library* ROS, set data, fail konfigurasi, dan apa sahaja yang secara logik melakukan modul praktikal untuk penyetempatan dan navigasi. Konsep sistem pengangkutan yang tidak terkira banyaknya dalam industri semasa kebanyakannya tetap dan sukar diganti. Keperluan untuk sistem pengangkutan yang fleksibel sangat penting dalam meningkatkan kecekapan industri dan syarikat untuk menangani sistem pengangkutan barang mereka. Idea dengan menggunakan paket ROS, Adaptive Monte Carlo Localization (AMCL) dalam melakukan Penyetempatan dan Navigasi Serentak (SLAM) dan memperbaiki sistem navigasi dan penyetempatan robot bergerak yang digunakan untuk mengatasi masalah sistem pengangkutan industri. Badan rig pengujian robot bergerak akan membuat fabrikasi menggunakan pencetak 3 dimensi menggunakan bahan plastik polylactic acid (PLA) standard. Ia dilampirkan dengan Raspberry Pi 4, sebagai komputer mini utama untuk mengintegrasikan semua algoritma ROS untuk mengawal pergerakan robot, menghasilkan pemetaan, dan membaca pelbagai input data sensor. Fasa terakhir dalam projek ini akan dinilai dalam bab metodologi dengan membuat percubaan untuk mengendalikan robot mudah alih menggunakan perisian RViz yang dijalankan dalam Sistem Operasi Linux untuk robot mudah alih yang melakukan proses pergerakan dari lokasi ke lokasi lain yang dikehendaki menggunakan penyetempatan kendiri dan konsep navigasi. Hasil projek ini menunjukkan semua objektif tercapai kerana robot mudah alih menunjukkan prestasi yang baik dalam menunjukkan ketepatan yang tinggi dalam pemetaan dan melakukan proses pergerakan.

ABSTRACT

This project research focuses on designing and developing a mobile robot as a testing rig platform to carry out self-localization and navigation processes. The robot will be installed with Robot Operating System (ROS), which is a package that includes ROS nodes, a ROS-independent library, a dataset, configuration files, and anything else that logically does a practical module for localization and navigation. Innumerable transportation system concept in the current industry mostly is fixed and difficult to be alternate. The necessitate for a new flexible transportation system is very essential in increasing industry and company efficiency to handle transportation systems for their goods. An idea by using ROS package, Adaptive Monte Carlo Localization (AMCL) in performing Simultaneous Localization and Navigation (SLAM) and improving the mobile robot's navigation and localization system used for resorting to industry transportation system problems. The mobile robot testing rig body will fabricate using a 3-dimension printer using the standard polylactic acid (PLA) plastic material. It is attached with Raspberry Pi 4, as the main minicomputer to integrate all the ROS algorithms to control therobot movement, generate mapping, and read the various sensor data input The last phase in this project will be evaluated in the methodology chapter by experimenting to operate the mobile robot using the RViz software running in Linux Operating System for the mobile robot performing movement process from a location to another desired location using self-localization and navigation concept. The result of this project shows all the objectives are achieved as the mobile robot is performing well in showing high accuracy in mapping and performing movement processes.

DEDICATION

To my family,

My beloved father, Hairudin bin Ahmad

My beloved mother, Zuraina binit Ishak

My respectful brothers, Hafizudin, Ismail,

Muhsin, Rashdan, and Syamil

To my supervisor

Dr. Shariman bin Abdullah

Thank you for giving me moral support,

money, cooperation,

encouragement, and understandings

Thank You So Much & Love You All

ACKNOWLEDGEMENT

Alhamdulillah. The completion of this final year project really had me hustling throughout the whole 14 weeks and thanks to all those people who have helped me making it possible for me to finish the project.

First and foremost, I would like to give my deepest appreciation to my supervisor, Dr. Shariman bin Abdullah who have helped me tremendously by guiding, supervising, and supporting me while I finished this project. He really contributed a lot to my project and without him, this project would be impossible for me. I am also very thankful to my fellow course mates who have been supporting me throughout finishing this project by sharing together all the knowledge that they have regarding my topic.

Finally, to my family who have been supportive and very understanding, always supporting me morally even though we are not always meet each other. It was impossible to list out all the people that have helped me in various way, but I really appreciate your encouragement, support, and warm cooperation. I am making this project not only for degree but to also contribute to empowering humanity and increase my knowledge. Thanks again to all who helped me.

TABLE OF CONTENT

CHAPTER 1: INTRODCUTION

CHAPTER 2: LITERATURE REVIEW

CHAPTER 3: METHODOLOGY

CHAPTER 4: RESULTS AND DISCUSSION

CHAPTER 5: CONCLUSION AND RECOMENDATION

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGV	-	Automated Guided Vehicle
AMCL	-	Adaptive Monte Carlo Localization
DMR	-	Differential Mobile Robot
EKF	-	Extended Kalman Filter
IMU	-	Inertial Measurement Units
LiDAR	-	Light Detection and Ranging
OS	-	Operating System
PLA	-	Polylactide Acid
ROS	-	Robot Operating System
RGB-D	-	Red-Green-Blue-Depth
SLAM	-	Simultaneous Localization and Mapping

LIST OF SYMBOLS

A - Ampere
V - Volt
KV - Kilovolt
α - Angular angle
d - Distance
° - Degree
θ - Angle
$\boldsymbol{\omega}$ - Angular velocity
$\boldsymbol{v}$ - Linear velocity
k - Spring constant
ms - Millisecond
mm - Millimeter
Hz - Hertz

CHAPTER 1

INTRODUCTION

1.1 Research Background

Robot localization and navigation denotes the robot's ability to establish its position and do the orientation of movement within the unknown environment. This project's main point is to study similar research and perform robot localization and navigation. Robot navigation refers to the process of robot moving from the starting point to the target point. During this period, the robot determines its position according to the saved map and lidar, and then plans out the optimal path according to its own position and preset target point. Finally, the robot moves to the target point roughly according to this path.

Path planning is divided into global planning and local planning according to the scope of action. Local planning is to plan the specific moving speed of the robot according to the local target points and the environmental information obtained by the radar. Robot Operating System (ROS) is a package includes ROS nodes, a ROS-independent library, a dataset, configuration files, and anything else that logically does a practical module for localization. There are lot of others ROS package that free access to anybody that can be utilized if we have the right knowledge and good effort. Mobile robot navigation technology is now increase rapidly with the technology of 2D/3D mapping. Many studies have been done on the implementation of 2D/3D SLAM to determine the goal position respect to the current position of the robot in an area that has not been recognized previously. This research is determined to improving transportation concept system by implementing IR 4.0 for transporting system widely in our industry.

1.2 Problem Statement

Mostly in our current transportation system in industry are not flexible as they tend to be fix and cannot be manipulate or adjusted. In industry sector, people tend to have a carrier system in transporting their goods. Commonly these company will manually transfer and move their components using machinery or in intermediate company, they will transportation system such AGV. The problem came out when the need of changing of factory layout or transportation process. For this old manual transportation method, there will be need huge, fixed cost needed for creating new layout for their industry. In addition, for the AGV and similar sort of this transportation device, they demand for landmark as line guidance, barcode and others figure for performing their movement process to lead them in their path. For that, this study was needed to build and autonomous mobile platform testing rig for position localization analysis in assisting all the stated problem to make in a safer and efficiently.

This research will inquiry the efficiency of a Simultaneous Localization and Mapping (SLAM) based of robot model which are implemented in Robot Operating System (ROS) by measuring how the robot travel to reach their destination set by the operators need to be optimized. Much research from others study stated ROS algorithm that used in autonomous mobile robot platform for localization tend to have shortage and error which become problem for the autonomous mobile robot.

There are many resources needed for completing this task to achieve and meet the localization systems, that allow a robot to locate itself, whether there is a static map available or simultaneous localization and mapping is required. We can use different sensors like LIDAR, RGB-D camera, inertial measurement units (IMU) and sonar to give the sensing power. By using these sensors and mapping algorithms a robot can create a map of the surroundings and locate itself inside the map (Guimarães et al., 2016). The robot will be continuously checking the environment for the any environment changes that could happened.

Mobile robot navigation technology is now increase rapidly with the technology of 2D/3D mapping. Many studies have been done on the implementation of 2D/3D SLAM to determine the goal position respect to the current position of the robot in an area that has not been recognized previously. We need this mobile platform testing rig robot that enable operators to plan outages more precisely and more efficient. This increases the availability of facilities and makes them more profitable, as well as boosting the safety of people and the environment which eliminate human entry in (Priyandoko et al., 2017).

1.3 Objectives

The objectives are as follows:

(a) To develop a mobile robot platform testing rig programming framework and control in Robot Operating System (ROS).

(b) To perform data collection from the surrounding environment in creating mapping through Simultaneous Localization and Mapping (SLAM) process.

(c) To analyze the behavior of the mobile robot when moving from a position to an input desired position.

1.4 Scope

This project limitation is to fabricate the mobile robot platform with the dimension size of 45 cm in width from 3D printer process by standard Polylactide Acid (PLA) material using differential drive locomotive method with additional caster wheel. The Robot Operating System ROS used is Adaptive Monte Carlo Localization (AMCL) package for the Simultaneous Localization and Mapping (SLAM) process which installed in the microcomputer board of Jetson Xavier NX. Dual shaft brush DC motor will be used at both side tire powered with 24-volt Li-Po battery and controlled by Odrive V2.6 board for the movement speed and rotation. Odometry encoder sensor will be utilize increasing accuracy in rotation motor shaft while maneuver by Light Detection and Ranging (LiDAR) sensor to detect the surrounding environment. The speed of this mobile robot platform is determined to be as 0.5 meter per second.

1.5 Summary

This section provides an overview of the research that has led up to the development of the mobile robot. An aim and problem statement that are specific, observable, and attainable are defined. Finally, the project's scope is specified, and the report's structure is constructed to illustrate the report's flow in each of its sections.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses literature of how others journal and study on navigation of mobile robot using the Robot Operating System for navigation and localization. There are few studies that is relate to this topic on developing and design the mobile platform robot such using the mapping, localization, path planning, Simultaneous Localization and Mapping (SLAM) and the robot operating system (ROS). The needs of various sensor in the development of this project as laser sensor, odometry sensor and kinetic sensor is also being discussed in this chapter.

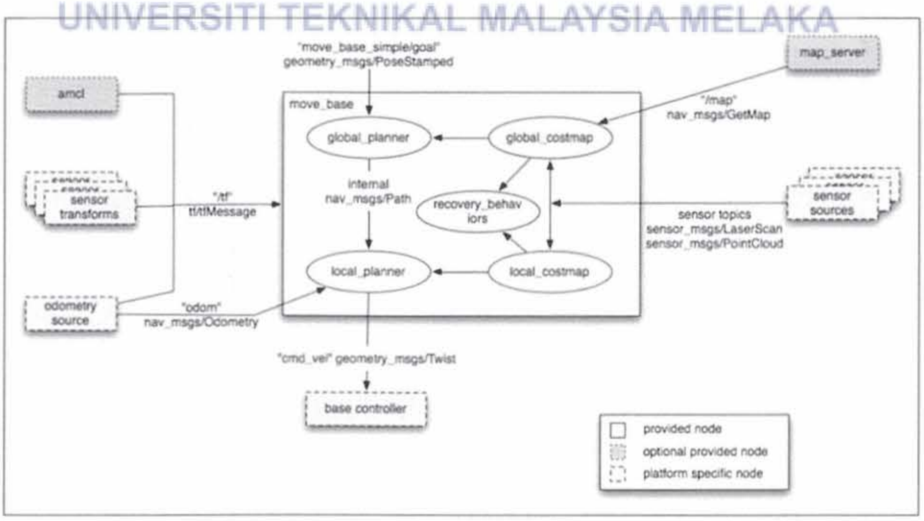


Figure 2.1: Navigation Stack (Guimarães et al., 2016)

Figure 2.1 shows the system overview of the navigation stack in the autonomous navigation system to operate. All the items block stated in this figure will be explain and analyze one by one in this sub chapter.

2.2 Autonomous Navigation Technologies

Autonomous navigation for a robot is achieved when the robot could do a moving task by its own by using mapping the environment process and plotting the odometry process (Akash et al., 2019). These two applications are important as a particle filtering algorithm which will resulting in Simultaneous Navigation and Mapping (SLAM) to work for the robot. Akash et al., (2019) said, by only use this filtering algorithm assist the robot to find the moveable way and path for moving to the input position inserted while avoid collision on surrounding.

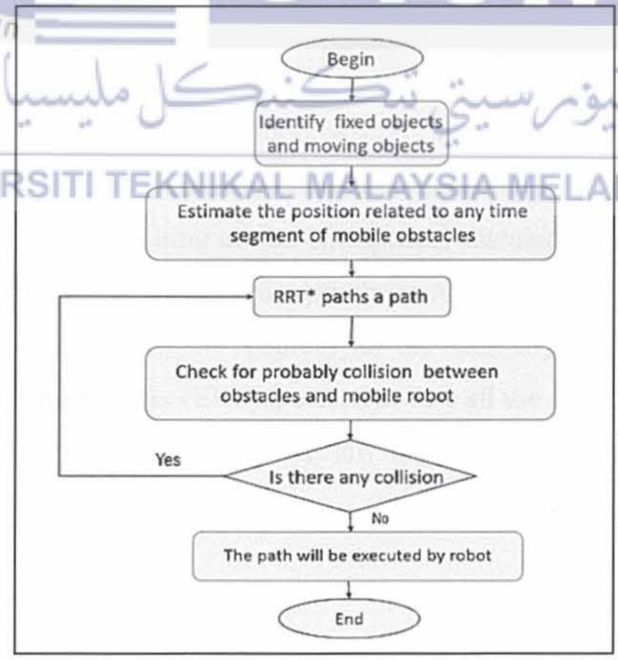


Figure 2.2: Autonomous Navigation Flow Chart (Akash et al., 2019)

The flow chart in figure 2.2 shows how the sequence process for the autonomous navigation process. As stated, the process starts by identifying surrounded object using attached sensor until then it could create mapping for localization process. Next process by undergoing algorithm filtering to process the input and output using the ROS libraries to move to the desired position.

2.2.1 Extended Kalman Filter (EKF)

In assisting local localization system process to works, EKF is used to help to operate in higher frequency to fuse the different sensors data obtained combined with odometry measures sensors. Robot localization package provided by ROS implement the EKF for use in this autonomous navigation application by developing such 2D mode concept, environment parameters as frequency and sensor timeout for tuning process(Valera et al., 2021).

The problems occurs when integration process of parts and the software implementation during autonomous navigation which is error in the odometry calculation. Köseoğlu et al., (2017) stated that odometry calculation in the navigation algorithm was just based on encoder ticks which resulting all the odometrical calculations were made through position change of motor shaft data obtained when the movement occurs. So, the solution to overcome this errors and to increase the accuracy for the odometry calculation to get more reliable, the Extended Kalman Filter (EKF) are used to fuse all the data obtain and solve the problem (Köseoğlu et al., 2017). Figure 2.3 below shows an example how graph when plotting position using EKF concept.

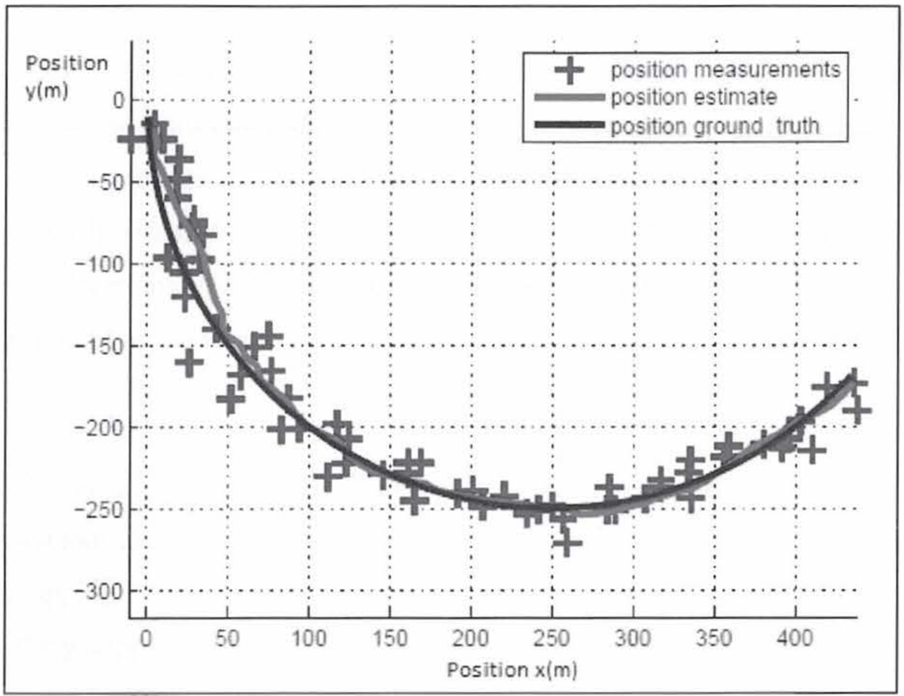


Figure 2.3: Estimates and position measurements using EKF (Akash et al., 2019)

2.2.2 Adaptive Monte Carlo Localization (AMCL)

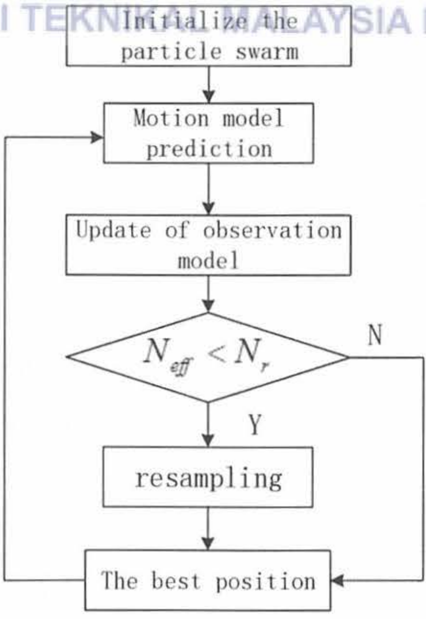


Figure 2.4: Monte Carlo flow chart (Wang et al., 2019)

Figure 2.4 shows the flow chart on how Adaptive Monte Carlo (AMCL) system proses works. AMCL is a same concept as EKF where is a tool for the robot in locate its position in the real-time at the environment through a static map or a previously created map (Guimarães et al., 2016). It is a localization system where is based on Monte Carlo localization approach obtained on the ROS libraries system. AMCL works as representing the robot possible locations by randomly distributes particle in a known generated or given map, so the particle filter used could determine the actual location for the robot pose.

Guimarães et al., (2016) discover that an anonymous navigation robot could not works greatly without using the AMCL system. He stated that the robot will get lost easily when using gmapping only without AMCL that will cause affect when odometry errors occurs. When the gmapping bases its mapping and localization, the odometry errors make the system confused but if the mapping created by gmapping and deliver to the AMCL node, it will trust the map and adapts to the odometry errors which resulting for a better result.

As mentioned earlier, the concept used is similar with EKF which these two-system assisting the robot in Simultaneous Localization and mapping (SLAM) process throughout interpreting the data to increase the robot accuracy by tracking the position and overcome the odometry problem caused when the robots tire starts to slip.

2.2.3 AGV Line guidance

Apart from autonomous mobile robot that using the ROS navigation such EKF and AMCL that used sensors to navigate an environment for movement, Autonomous Guided Vehicle (AGV) is a programmed robot that use a physical guidance to navigate in a pre-defined environment through on a pre-defined path (Köseoğlu et al., 2017). Nowadays, AGV robot are frequently being used in the industry as a movement mechanism for their product because its suitable in performing a repetitive task and a specific designed task.

AGV robots are typically used in transportation of raw-materials, finished goods, and work-in-process in a manufacturing production line in the industry. Combination of software and sensor-based guidance system are used in the AGV to move on a predictable and the set path with precisely controlling its acceleration and deceleration. There are several examples of guidance used by the AGV such floor-surfaced mounted magnetic tape and bars, optical sensors, lasers, and magnetic based inertial guidance.

Figure 2.5: AGV robot with floor-surfaced mounted magnetic tape (Köseoğlu et al., 2017)

Figure 2.5 above shows example of a AGV floor-surfaced mounted magnetic tape robot used in an industry. The problem for AGV is its works according to a preset systems and processes only which resulting difficulty when a need of sudden rapid changed. As summary, the inexecution of dynamically changing tasks is the main disadvantage for the AGV robot.

2.3 ROS Navigation Concept

ROS or Robot Operating System is a set of package resources that capable for a robot to perform a movement and navigation through an identified or unidentified environment. This set of system assist the robot for undergoes process such planning, tracking, sensing, and mapping a path while deviates from surround obstacle that lay on its path of movement (Guimarães et al., 2016).

2.3.1 Navigation Stack

Navigation stack is a package that can be retrieve from the ROS navigation system which is crucial for mobile robots to move from a place to another reliably (Zheng, 2017). Zheng stated that the navigation stack's job is to create safe path for the robot to move preventing any obstacle by processing the data obtained from the sensors, odometry and the environment mapping created. All these inputs will be tune based on the desired set parameters which will need lot of work to obtain the maximum results in comply the required robot movement process.

Guimarães et al., (2016) study shows that localization system is one of many resources that present in navigation stack to do planning and following path while deviates from any obstacle appeared on the robot's path. This localization systems are the same as mentioned in autonomous navigation sub-chapter as the EKF and AMCL where works to allow the robot for locating its real time position when there are given static mapping or a previously created map from the environment.

There are several problems being highlighted by the author, where when the static map is used, all the environment around the robot cannot have any change or modification. This is because of for every change, a new map will be needed to regenerate and would consume extra time and effort. The solution given is to bypass the lack flexibility of static map by adding two others more localization systems which are gmapping and hector_mappping. This two tools system could be retrieved also from the navigation stack package and could assists by applying a technique that combining process mapping of the environment while at the same time moving the robot. It means while the robot navigate to face a new environment, it gather all the needed data information from the surrounding through its sensor and generate a map at the same time (Guimarães et al., 2016).

2.3.2 Transform Frames (TF)

Transform frames or stated as “tf” is a tool offered by ROS navigation which preform in adjusting or recalibrate the sensor position in relation of the robot’s body adequate the measurement of the robot navigation (Guimarães et al., 2016). To generate a mapping for a robot, as mentioned we need to use various of sensor. There is one indispensable data that is crucial to sense the distance travel for the robot and serve as the reference which named odometry sensor. Before the odometry sensor could be used, its need to be adjusted and tuned before can be operate by the robot. The author finds that there will need adjustment for the sensor because the sensor measures the environment in relation to itself, not in the relation to the robot’s body. There will need be a geometric conversion for solving this issue.

Jusuf (2016) Stated that when used the navigation stack, we need to know the position of each sensor, all joints and each wheel using the “tf” for understanding the relation between the whole robot frame constraint with the robot’s position. This shows that Jusuf and Guimarães have the same idea and concept in assiting the navigation stack to generate mapping for the robots.

There is a author define that stated position vector and quartenion decribes the concept of the “tf” to transform data to show the real position and posture between two object of sensor position and the robot’s body (Li & Mei, 2018). Figure 2.6 below shows direction and location of robot can be calculated in x,y direction and its angle. The authors also stated that to get the exactly position of the robots, odometry can be calculate using this formular.

$$X = \int dX = \int (V_x \cos\theta - v_y \sin\theta)dt$$
$$Y = \int dY = \int (V_x \sin\theta + v_y \cos\theta)dt$$
$$\theta = \int d\theta = \int \omega dt$$

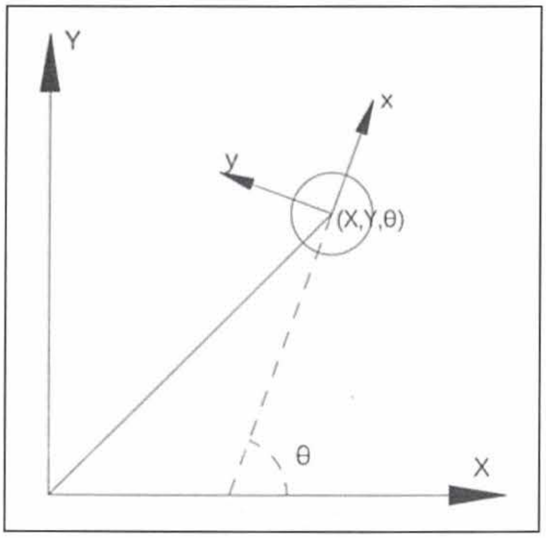


Figure 2.6: Odometry calculate figure (Li & Mei, 2018)

2.3.3 Odometry Source

As mentioned in earliest sub chapter, odometry sources being used in the navigation stack as a method in determining the momentary position of a robot. Odometry source is tool to retrieve the data from odometry sensors as input to be tune and combine with sensor transform "tf" in generate the local planner or mapping for the robot navigation path. Odometry sources assist in providing easily accessible real-time positioning information for the navigation stack algorithm system between periodic absolute position measurement (Köseoğlu et al., 2017). Combined with Simultaneous Localization and Mapping (SLAM) concept such Extended Kalman Filter (EKF) and Adaptive Monte Carlo Localization (AMCL), odometry sources could counter the previous mentioned problem as the odometry calculation was based on the change sensed in encoder tick from the odometry sensor which in some cases, error occurs due to slip on the rotation of the robot's wheels.

2.3.4 Command (Base Controller)

Base controller can be defined as a block or a specific platform node to receive the robot's data such velocity data retrieve from sensor and convert it into motor commands for the robot (Jusuf, 2016). This means the base controller is needed for the robot execute its movement throughout the overall navigation stack. The sensor that used for determining the robot velocity is the odometry sensor by calculate every rotation of the wheels.

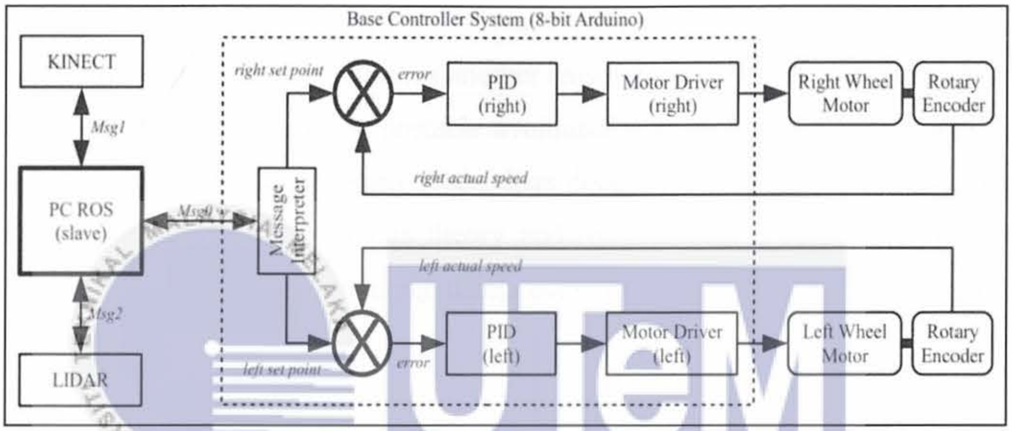


Figure 2.7: Base Controller System (Priyandoko et al., 2017)

Figure 2.7 shows the flow chart process in the base controller system where in this ROS, the authors used LIDAR and KINECT as the sensor to transmit the signal to the base controller in their research (Priyandoko et al., 2017). In others word, the base controller will receive two type of input velocity from the sensors which are 3-axis linear velocity and 3-axis angular velocity and then interpreted the data into two correspondence values of speed where for left wheel and the right wheel. There are two wheels being used because the researcher used Differential Mobile Robot (DMR) type which are the commonly being used. This design of locomotive method will be explained later in next sub-chapter.

2.3.1 Path Planning

In order to achieved a smooth motion for the robot movement with real-time replanning system, path planner is an important element of research in the field of mobile robots path planning in creating its global trajectory planners (Cybulski et al., 2019). Basically, the path planning will begin as the robots start to move or to start making its own new mapping for the environment as a dummy mapping. From the inserted dummy mapping, the robot using SLAM algorithm will execute new real-time mapping while its move.

Safety consideration for the robot is another reason for path planning process. Wang et al., (2019) said from the process, obstacle avoidance can be achieved to prevent any damage on the robot body. But as many researchers done the path planning process, there are plenty improvement needed in this theory and methods. For an autonomous robot navigation, this system of path planning using autonomous positioning and navigation technology is the basis needed.

Wang et al., (2019) also stated that for improvement in the path planning process, he used the ant colony algorithm in which is a heuristic intelligent path finding algorithm stimulating ants foraging behavior so the robot can search its optimal path in the known environment mapping. The flow chart of the ant colony algorithm is stated as figure 2.8 below:

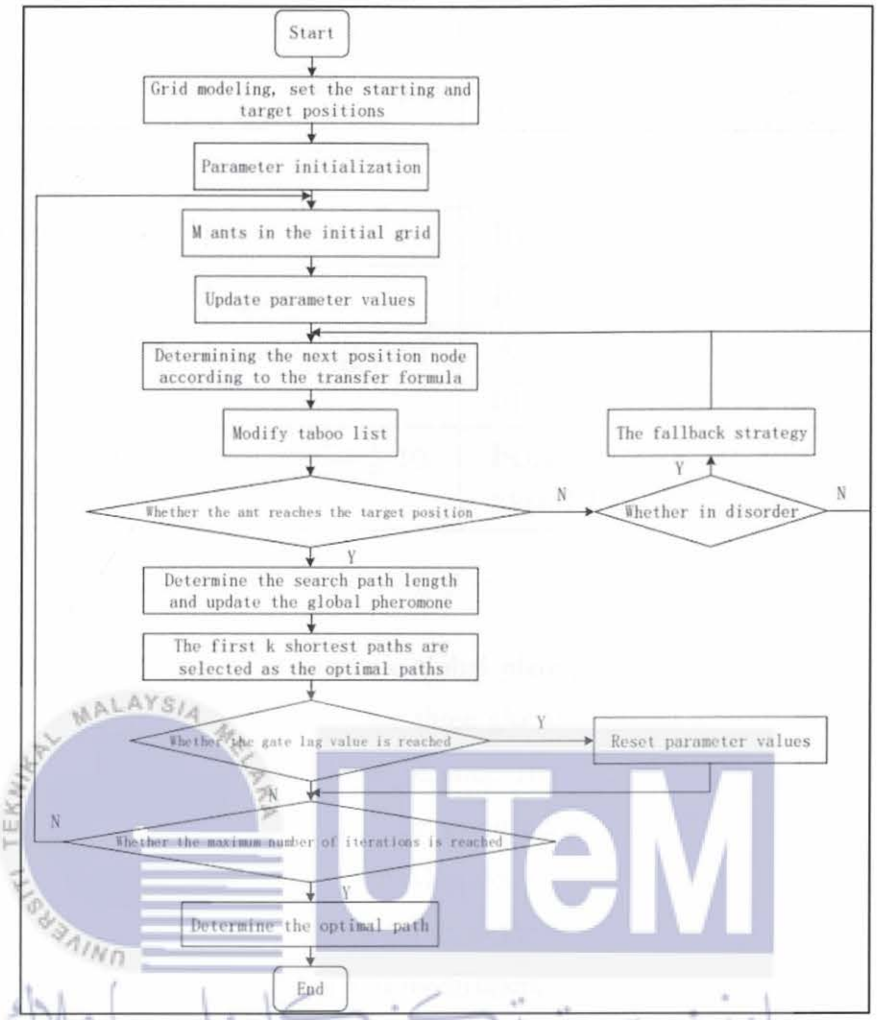


Figure 2.8: Ant Colony Algorithm (Wang et al., 2019)

2.3.2 Global Path Planner and Local Path Planner

Global planner is a base of static map which being set up into the algorithm for the robot to create path from the early start as the current robot position to the desired set position destination point. Only after that, the local path planner will subscribe with the global planner path plan and started to created its new mapping based on the robot's kinematic and robot's sensor reading in order to achieve its goal moving to the desired location point (Cybulski et al., 2019). The different between Global Path planner and Local Path planner is stated in the table 2.1 below.

Table 2.1: Difference between Global and Local Path (Cybulski et al., 2019)

Global Path Planning	**Local Path Planning**
Map based	Sensor-based system
Deliberative system	Reactive system
Relatively slower response	Fast response
Assume complete knowledge of the workspace area	Assume incomplete knowledge of the workspace area
Obtain a feasible path leading to goal	Follow path to the target while avoiding obstacles

To execute navigation stack process, global planner and local planner is needed. Zheng, (2017) find in his research, there are three global planners that he basically used which are carrot_planner, navfn and global_planner. He stated that carrot_planner is the simplest one which operate in checking if the given goal is an obstacle, it will find an alternative goal close to the original goal by moving along the robot and the goal point vector. So, this type of global planner is not very accurate as the robot will only move closer to the stated point. The next global planner which is navfn operate by using Dijkstra's algorithm in finding the global path while having the most minimum cost between starting point and the end point of the robot path. While global_planner is created for more flexible replacement for navfn with added of more options. The author stated that the options are as toggling quadratic approximation, support, and toggling grid path.

2.4 Sensors Type

Working with robots necessitates many sensors, and each operation must be done in real time. We need a form of operating system that allows us to use sensors and actuators that need to be modified every 10-50 milliseconds in order to use them (Kannan Megalingam et al., 2018). From various of researcher study, many of author have used few common sensors in building their autonomous robot navigation such for Kannan said he used the sensing power is provided by LIDAR, RGB-D camera, IMU and sonar to creating the surrounding map for the environment. Newman et al., (2020) stated that a robot must be able to feel and adapt to its surroundings to navigate without supervision. This method necessitates the robot being fitted with the appropriate sensors and gaining situation knowledge by real-time algorithms.

2.4.1 Laser Sensor

Laser sensor used to create the environment chart, but it is often used to detect and clear complex obstacles (Valera et al., 2021). Laser Measurement Systems (LMS) is a stand-alone non-contact measurement device (NCSD) for industrial applications. Using infrared laser rays, the device scans the surroundings in two dimensions with a radial field of vision. For evaluating locations and other estimation operations, the evaluated data may be independently analyzed in real-time with external assessment tools.

The laser sensor's basic theory is that the laser beam is transmitted in a set direction, that the emitted laser is mirrored as it meets an obstacle, that the reflected laser is captured by the receiving device, and that the distance of the target is calculated by the correlation algorithm measurement (Wang et al., 2019). The author stated an example on how the laser sensor operated to measure things as angle between the laser sensor and the robot orientation is α , the distance of the obstacle determined by the laser sensor is d and by using the formular below, the actual robot position from the obstacle can be known as (x_0,y_0). Then we will know where the obstacle is on the grid map, as well as where the robot is using this given formula.

Li & Mei, (2018) said that sensors such as laser scanners, sonar sensors, and cameras are commonly used to create a map of any environment. The author also used the laser sensor for their navigation and control system of mobile robot based on ROS project to obtain environment condition. By transmitting laser beams, laser scan may detect the position of a target, environment, or obstacles. Figure 2.9 below shows type of laser sensor used by the authors.

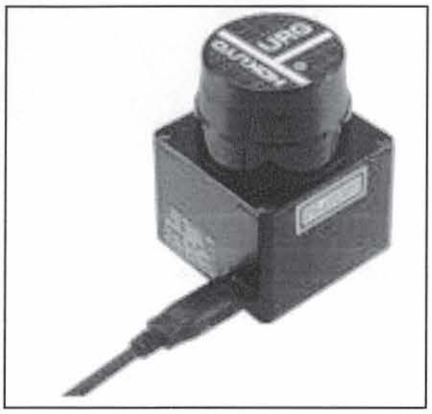


Figure 2.9: Laser Sensor (Li & Mei, 2018)

2.4.2 Odometry

Odometry is data from motion sensors, such as wheel encoders which it is used to measure the estimated change in the robot's location and heading over time, relative to the launching point in terms of an x and y position, as well as an orientation along the z axis (Housein & Xingyu, 2021). The authors also mentioned that there is problem in which the odometry data become less accurate when the robot starts moving. As mentioned before, the inaccuracy may be caused by wrong wheel diameters, irregular driving surfaces, or the wheel slipping, all of which cause the wheel encoders to emit inaccurate results. For this problem, they overcome with adding extra sources of information by installing rplidar. By this device, it gives the absolute distance and orientation calculated from the robot's current position. Figure 2.10 shows an example of rplidar sensor device.

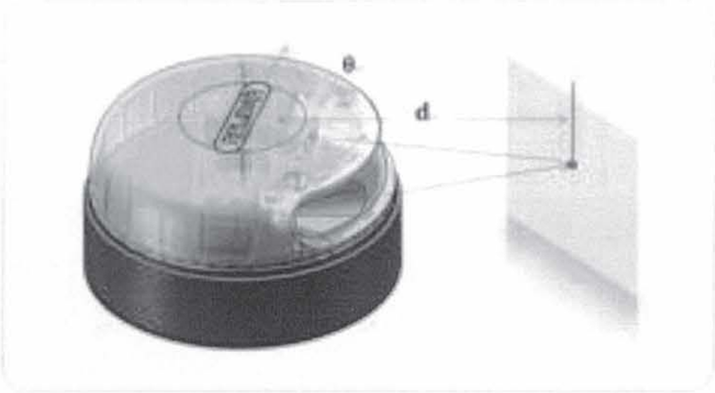


Figure 2.10: Rplidar device (Housein & Xingyu, 2021)

As explained in the Extended Kalman Filter (EKF) topic, odometry data needs in increase the accuracy to estimate the location and position of the robot when combining with laser sensor through the SLAM concept (Li & Mei, 2018). Köseoğlu et al., (2017) stated that Odometry is the most used technique for calculating a mobile robot's current location. Odometry offers readily accessible real-time positional information between periodic absolute location measurements in most practical applications.

2.4.3 Kinect Sensor

Kinect is a multi-sensor system that includes a depth sensor, an RGB monitor, and microphones (Guimarães et al., 2016). 3D RGB-D sensor is a Kinect sensor with providing a depth data and the color image data. It equipped with IR emitter and IR depth sensor which will resulting in capturing the depth information. The robot must rely on its sensors for mapping and localizing itself in its operational area while navigating in an indoor environment. This is where the need of 3D sensor such Kinect Sensor play its role (Mishra & Javed, 2018). The author uses this type of sensor for their project in creating ROS based service robot platform which is the Microsoft Kinect XBOX 360. It come with view of 43° degrees vertically and 57° degrees horizontally with tilt range of 27° degrees which very suitable for their project requirement. From this sensor they could use the Kinect to collect

3D point cloud data from the target area and be used for indoor navigation as well. Figure 2.11 and 2.11 below shows on how input data from the sensor. (Mishra & Javed, 2018)

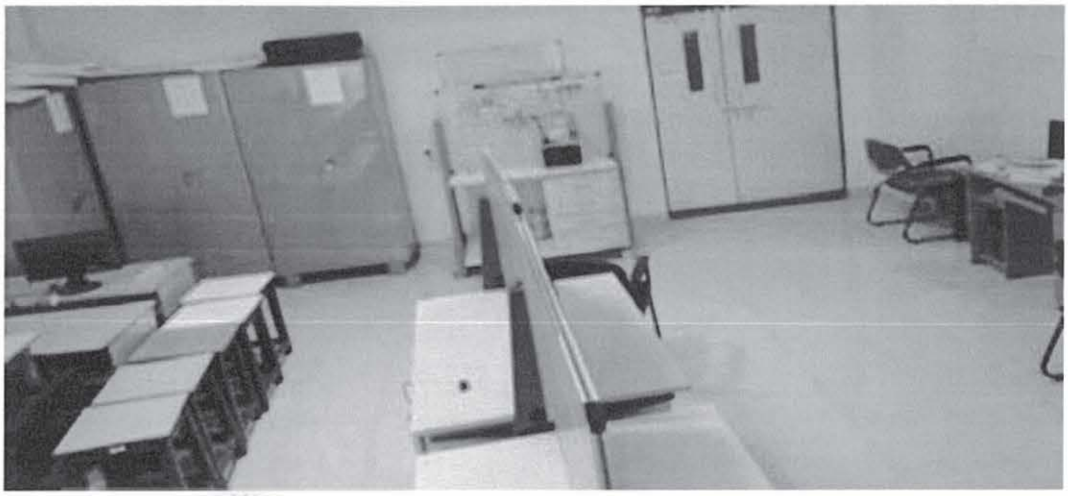


Figure 2.11: Image of actual environment (Mishra & Javed, 2018)

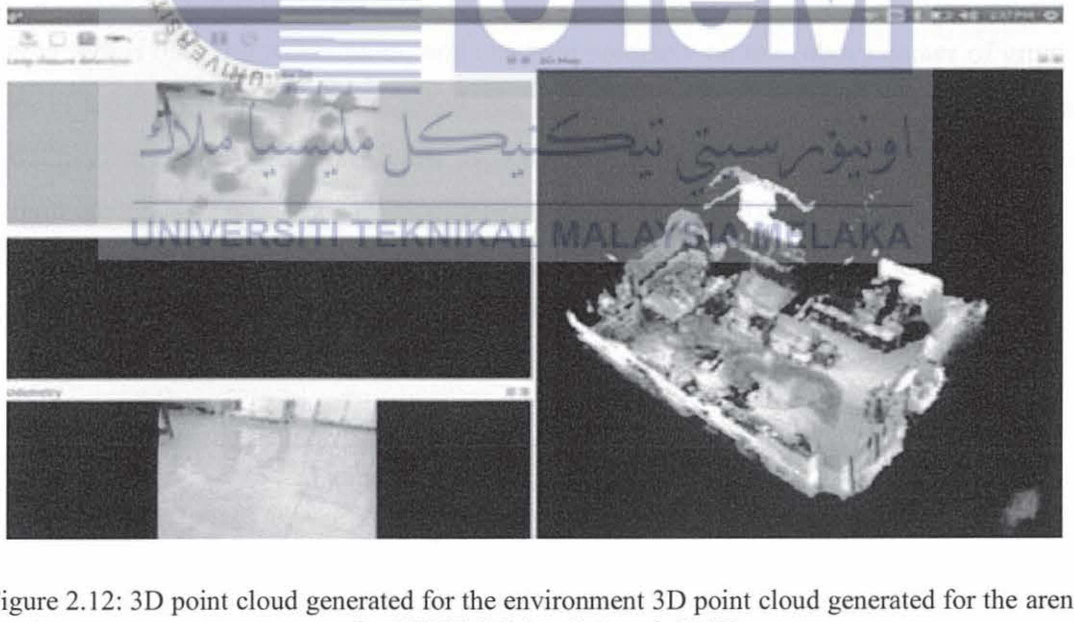


Figure 2.12: 3D point cloud generated for the environment 3D point cloud generated for the arena using RVIZ (Mishra & Javed, 2018)

2.5 Design Locomotive Method

A typical method of generating trajectories for differential drive robots is to assume a motion in which the robot travels straight, slows, and then turns because of its simplicity on the spot (Lauer et al., 2020). Design locomotive for a mobile robot can be in many ways such differential drive robot (DMR), skid steering and Ackerman steering. All this type of locomotive will be resulting movement in their various ways.

2.5.1 Differential Drive

The differential mobile robot (DMR) is used in addition to the Ackermann model, which is widely used by researchers and used by Priyandoko in his project on mobile robot that used for mapping on industrial plant environment (Priyandoko et al., 2017). The DMR has two separate wheel drive systems that can be operated independently, as well as a wider turning radius than the Ackermann, and can spin precisely on the robot's center of gravity. Also, in terms of tight space, the DMR has greater maneuverability than the Ackermann model to clear obstacles. Figure below depicts the kinematic model of the real robot, which accommodates two inputs, linear velocity (v) and angular velocity (ω) toward the target spot, as seen in the figure 2.13. In general, the DMR model assumes the unicycle kinematic model seen in Equation (1-3):

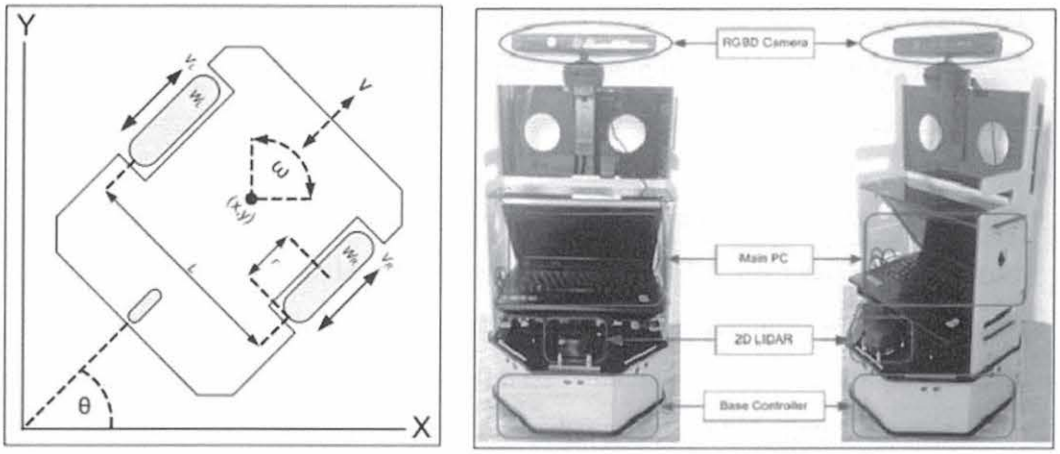


Figure 2.13: DMR model used by the author (Priyandoko et al., 2017)

$$\dot{x} = v\cos\theta \qquad Equation\ 2.6$$
$$\dot{y} = v\sin\theta \qquad Equation\ 2.7$$
$$\dot{\theta} = \omega \qquad Equation\ 2.8$$

2.5.2 Skid-Steering

A skid-steer vehicle can produce a significant amount of traction force, which is particularly useful for navigating rugged terrain (Kim et al., 2017). Since there are no steering systems on a skid-steer robot, it turns by the difference in tire force between the left and right wheels. This feature has several benefits, including a strong traction force and a basic mechanical structure. The authors did also mention many studies in skid-steer vehicle operation have been conducted to achieve autonomous navigation. However, designing a controller for a skid-steer vehicle remains a difficult problem due to the vehicle's high proneness to slippage when producing turning motion.

Ordonez et al., (2017) also find that modeling the motion of a skid-steered robot is difficult because slippage and skidding are intrinsic to this type of platform, and curvilinear motion involves high torques. Motion planners will sometimes create trajectories that are not practicable by the robot if the ground–robot contact, and torque specifications are not adequately captured. The difficulty of this model stems from the fact that that skid-steered platform's wheels or tracks must slip or when doing a turning maneuver, it can skid.

The solution to increase the accuracy of this type of terrain model is by using a learning- based approach. In particular method, we use the Gaussian method (GP) to describe the skid-steer vehicle's velocity model (Kim et al., 2017). The nonlinear characteristic or uncertainty of the model will be described by the GP which is a Bayesian non-parametric. Figure 2.14 below show an example of mobile robot that using the kinematic of a skid-steer type locomotive system.

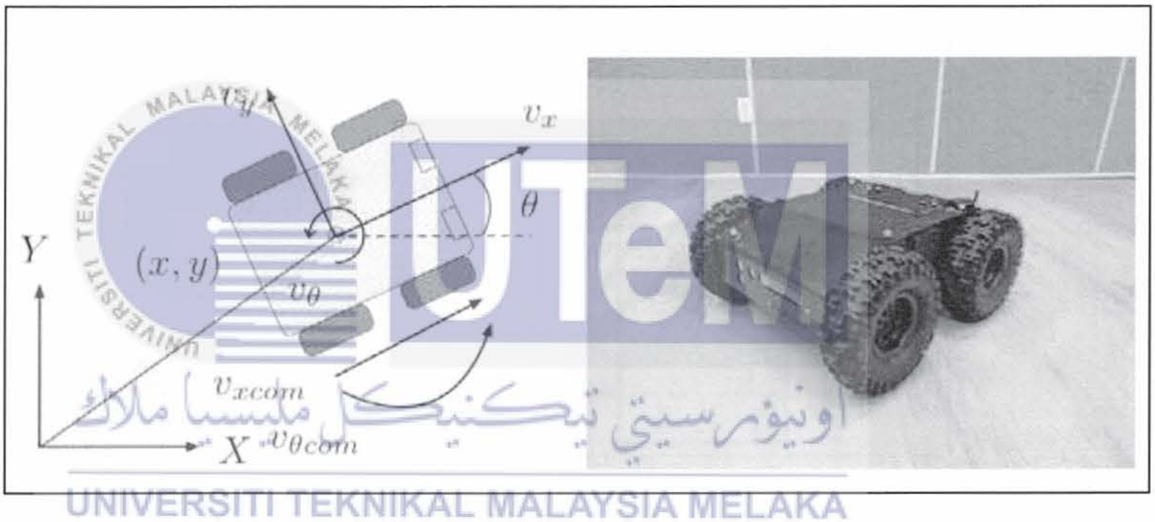


Figure 2.14: Kinematics of a skid-steer vehicle and mobile robot (Kim et al., 2017)

2.5.3 Ackermann Steering

Zhao et al., (2011) stated that when an automotive wheel moves or spin on the same center point, the situation is named as the Ackermann steering principle. Meeting the Ackermann steering concept allows the wheels to rotate freely without sideslip, reducing tire wear. But the Ackermann steering concept cannot always be met by the most widely used four-bar linkage steering trapezoid.

To meet the Ackermann steering requirement, a concept automotive steering mechanism with pure rotation transfers is proposed. Figure below illustrates the theory. In the car definition, steering mechanism, the rotational feedback from the steering wheel is transmitted to the. A center pair of bevel gears connects the right and left steering shafts while right pair of bevel gears transfers the rotation of the right steering shafts to the front right wheel, then a pair of noncircular gears transfers the rotation of the left steering shafts to the left steering shafts. A left pair of bevel gears is then transferring the rotation of the left steering shafts to the front left wheel while the pair of constant velocity joints make up both the left and right steering shafts.

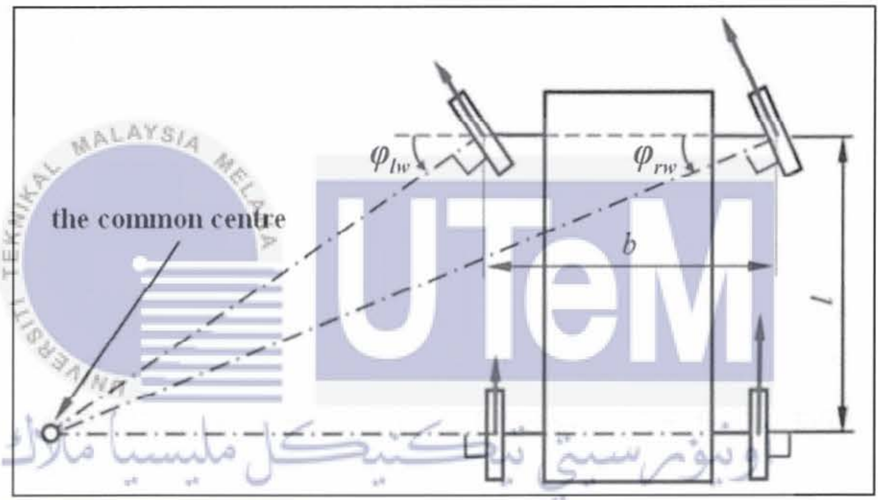


Figure 2.15: Ackermann Steering Principle (Tang et al., 2017)

Figure 2.15 above show an example on how Ackermann Steering Principle works in real-life. According to many studies about the design locomotive for navigation mobile robot, it is quite hard to find any author or researcher using the Ackermann steering design in their project as the mechanism used need high complexity in design and production.

2.5.4 Suspension

Springs, linkages, tires, air in the tires, shock absorbers, joints, and bushings make up the suspension system structure, which is situated between the frame and the wheels. It oversees filtering the wheel's excitation as it meets the field (Poojari et al., 2021). Suspension is very important in improving the efficient design of any steering wheel design in maneuverability and movement of a mobile robot. It is almost impossible for any wheeled mobile robot to move in uneven surface since there is no instantaneous core consistent for both wheels, two wheels connected by a traditional rigid axle cannot roll on rough ground without slipping (Appala & Ghosal, 2015). Plus, if any slip occurs and the on-board odometer is used, it causes localization errors and wastes power at the wheel-ground touch points.

Thus, to overcome this wheel slip issue, the distance between the wheel and ground contact points must be adjusted to provide a typical instantaneous core such installing the suspension on the wheel linkage design. The authors solution to overcome this issue is to setup the balance-rocker mechanism which is one type of reliable suspension system which consist of two rockers, two rods, and a balance link make up the assembly. Each rocker's flexible side frame is attached to it as a locking mechanism and a balance-rocker mechanism, respectively.

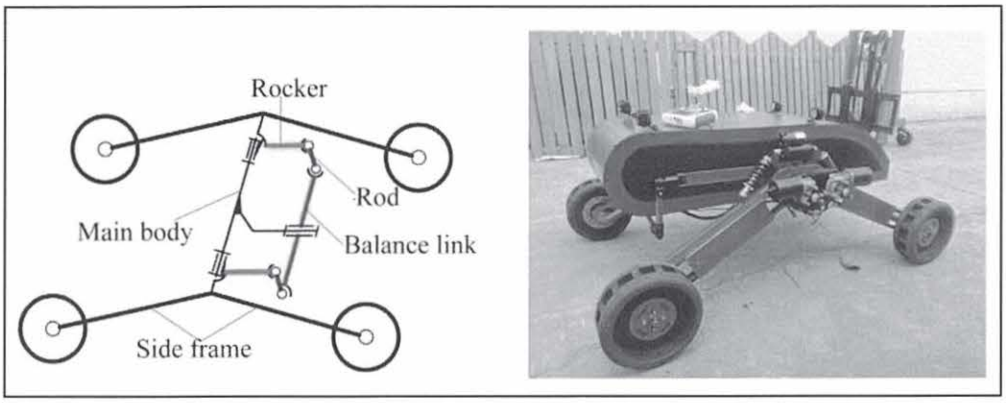


Figure 2.16: Mobile robot model using balance-rocker system (Jiang et al., 2019)

Figure 2.16 show how balance-rocker suspension system work on some mobile robot. The balance-rocker system is a passive feature that provides a mechanical escape that allows the two side frames to pivot three touch points with the ground independently and efficiently. It is also known as a differencing linkage. On rocky surfaces, the mobile robot's center of mass is located halfway between the side panels, resulting in fair loading on all four wheels (Jiang et al., 2019).

2.6 Summary

This literature review topic defines there are many ways and concept could be amendment in developing mobile robot using the ROS algorithm. Different of concept used will be affecting in different performance result. By observing and studying from this topic, this project will determine the best way to develop mobile robot with maximize its performance in production phases and its result in achieving this project objective.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the topic will be discussed on how to design and develop a mobile robot platform testing rig programming framework and control in Robot Operating System (ROS). This topic covers all the needed element in the idea to complete this project. Every part and component used in completing this project are stated specifically with their every specification. Generally, there are three phases in methodology to fabricate this project which consist of design and develop, data collection, and finally project analysis. The flow process for whole project will be stated in Gantt Chart in appendices.

3.2 Design and develop

Based on the study and research done during literature review topic, the best concept in design the body of the mobile platform testing rig have been develop and will set up with the suitable Robot Operating System (ROS) algorithm to implementing the objective of this project. The overview of method in design and development of this mobile robot will being explain and stated clearly according to their subtopic. Figure 3.1 shows design planning on mobile robot platform rig. Left picture is the finished prototype of the actual mobile robot. The design has been develop using the Computer Aided Diagram (CAD) software which is

Fusion 360. By considering all the needs for electrical components and as rig platform, the design for mobile robot has design to be fit and compact for increasing its mobility and at the same time could be as transportation platform.

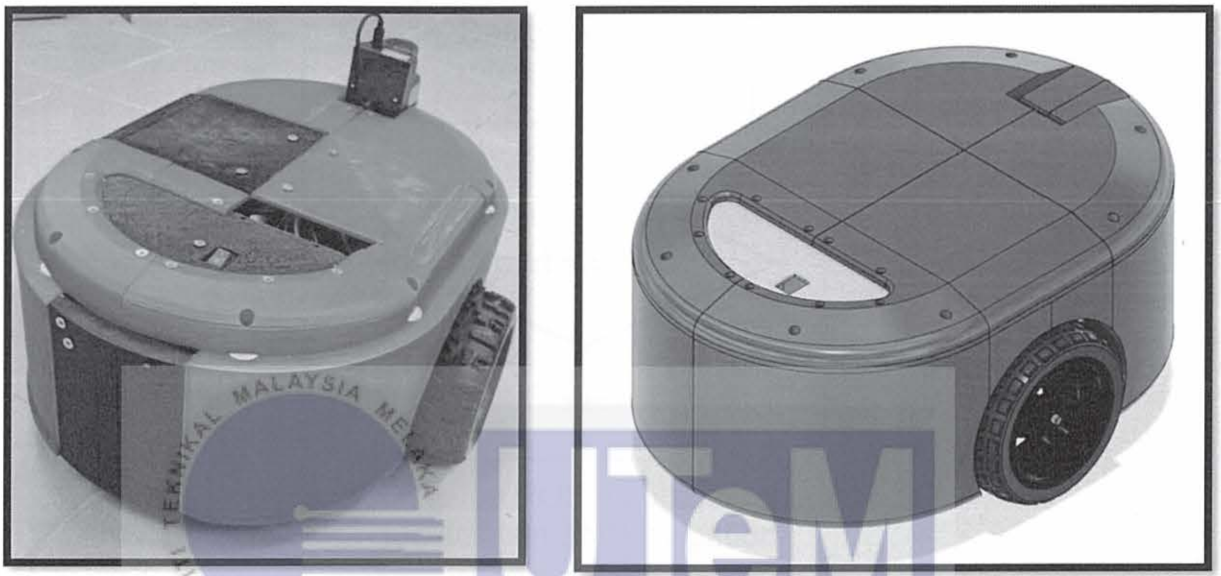
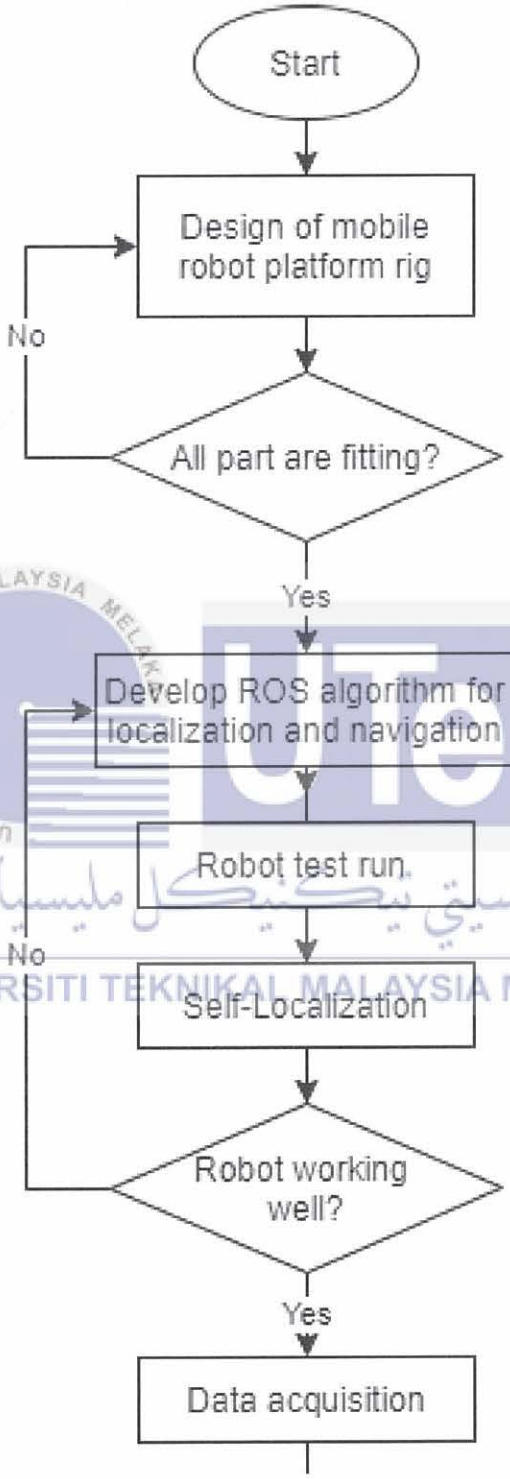


Figure 3.1: General view of mobile robot

3.2.1 Methodology Flow Chart



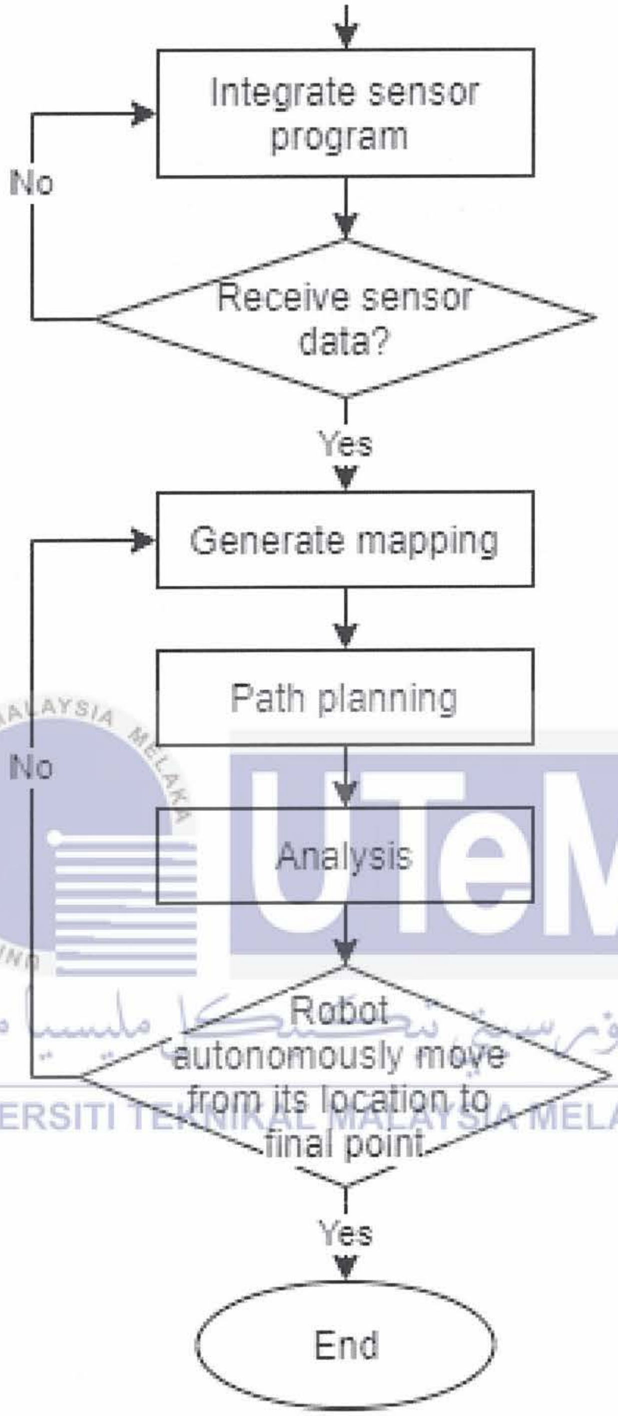


Figure 3.2: Project flow chart

3.2.2 Design

The deisgn for mobile robot is drawn base on the robot project scope where the platform size is 45 cm x 45 cm attached with differential drive as the design locomotive method used. Draftinng for the mobile robot platform in this project is as shown. Figure 3.3 below shows tha details design of mobile robot where generally the size of its will be aproximately 480mm in length and 340mm in thicnkness. The height of mobile robot will be aorund 195 mm when attached with the both side tyre.

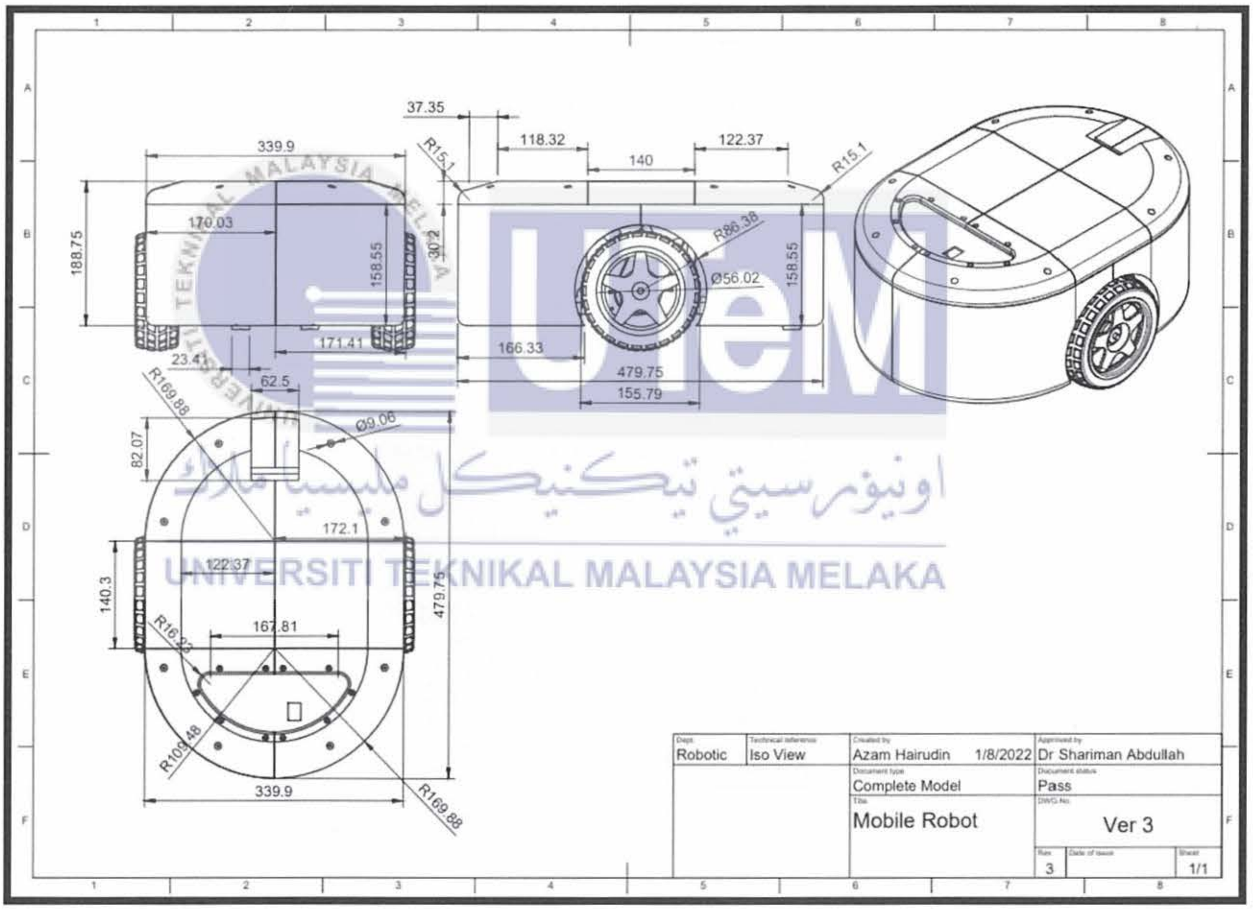


Figure 3.3: Drafting of mobile robot

The total part for fabricate the mobile robot are approximately 60 parts as its body excluding the microcomputer, sensor, and other board. For 3D printing part, there are approximately 50 parts to be fabricate as exclude the tire, motor, shaft, and the aluminum plate which will as the main body. Material used for 3D printing process is using the standard polylactic acid (PLA) which is a biodegradable material and can be fully renewable. This make the standard PLA material is environmentally friendly and its advantages has faster performance when changing physical state form as molten to solid which enabling to been printing with more details than others normal plastic. Figure 3.4 below shows for a complete mobile robot device, there will be 4 major components for its physical body part. From above, which is mobile robot's lid, then housing body, while in the housing body is where the motor component placed and at the bottom part which is the tire part. Figure below shows the whole major part in the mobile robot.

Figure 3.4: Major component part

Figure 3.5 below shows the exploded view for the mobile robot with all the 46 components is as below. Generally, there are many types of material used as the mobile robot's body and housing as example the main structure for it using the hollow aluminum bar. This aluminum bar has been selected as the best materials and type because of its performance in maximizing strength while minimize the total weigh of the mobile robot. Other's materials used from the 3D printing PLA materials, there are all the components used for mobile robot mechanism as steel rod bar, rubber tire, steel pulley, spring and lots of screw and nuts. The details of materials will be shows in the Table 3.1.

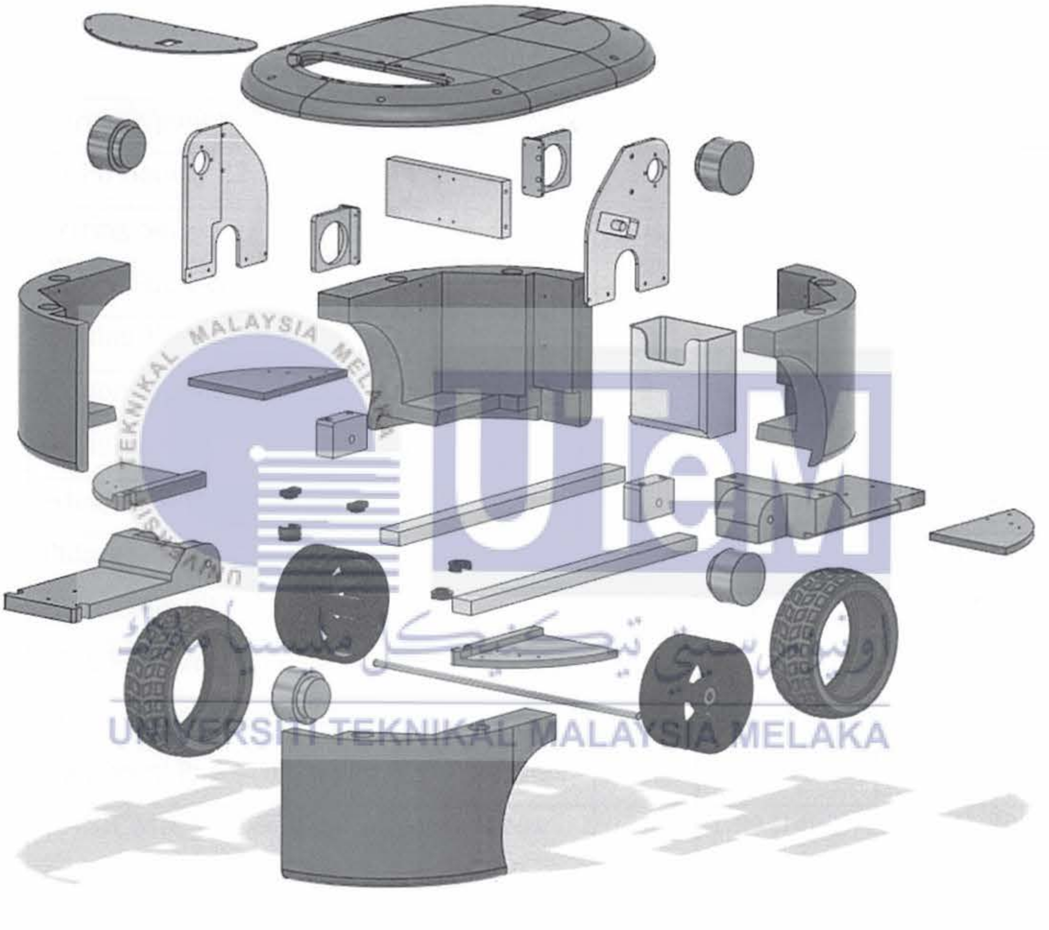


Figure 3.5: Exploded view

3.2.1 Bill of Materials (BOM)

In completing mobile robot platform, there are many part and electrical components needed for this project. As the duration for this project development progress is short, additive manufacturing technologies have been chosen to produce the body for the mobile robot. The additive manufacturing technologies use is the Fused Deposited Modelling by 3D printing process. For the 3D printing material, as mentioned before where standard polylactic acid (PLA) material is selected, and the total materials needed is approximately around 3kg which is 3 units of PLA Filament 1KG. Hence, all the material and part needed are as table 3.1 below.

Table 3.1: Bill of materials

No	Part component	Unit	Cost 1 Unit	Total Cost
1	Li-Po Battery 2200mAh 11.1V	1	RM53	RM53
2	Wiring Set	1	RM10	RM10
3	Robot Tire Gear Motor	2	RM15	RM30
4	Casting Wheels	2	RM7	RM14
5	Custom Spring	2	RM20	RM40
6	Stainless Steel Rod 8mm x 1m	1	RM10	RM10
7	Aluminum Bar	1	RM8	RM8
8	Pulley 8m-20mm 22T	4	RM73	RM292
9	Belt 20mm 22T	2	RM25	RM50
10	PLA Filament 1KG	3	RM50	RM150
11	Screw Nut Set 4mm	1	RM45	RM45
12	Raspberry Pi 4 4GB	1	RM700	RM700
13	BASICMICRO MCP263 Advance Motor Controller	1	RM1400	RM1,400
14	HOKUYO LiDAR Sensor UBG-04LX-F01	1	RM9900	RM9,900
15	12V 63RPM 20kgfcm Planetary DC Geared Motor with Encoder	2	RM250	RM500
16	KP08 Ball Bearing Pillow Block Mounted 8mm	4	RM7	RM28
17	Custom Pull-up Board 5V	1	RM40	RM40
18	Main Switch Board	1	RM20	RM20
			TOTAL	RM13,290

3.2.3 Suspension

A spring will be placed at both left and right tire at the back place of the robot body (red arrow). Suspension concept of the spring will act as damper for the whole body when the front tire moves through an uneven surface. Stiffness of the spring will determine on how much weight of the robot can support. The size of spring used for this project is 14mm in Outside Diameter (OD), 42mm of Free Length (FL) and size wire of 1mm. Position of the spring will be attached in vertical line as to overcome the mobile robot weigh when moving across any uneven surface. The stiffness of the spring can be determined by following formular.

$$k = \frac{F}{x}$$

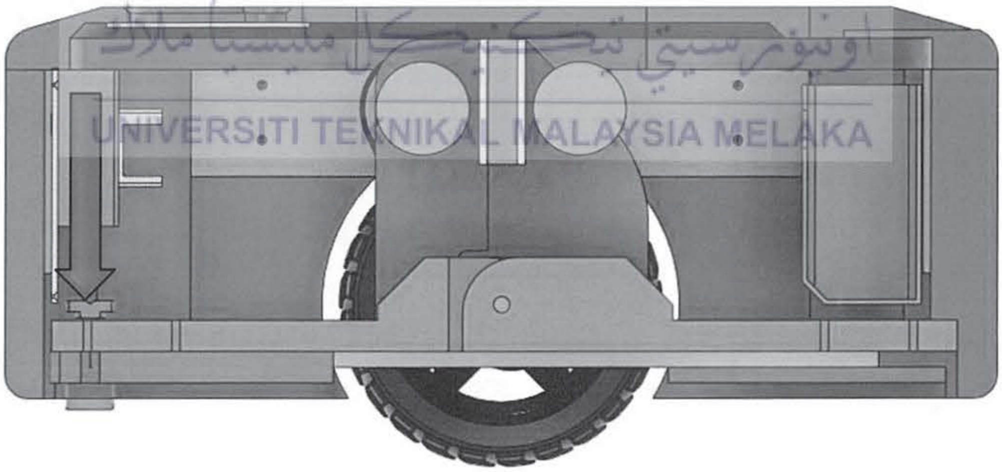


Figure 3.6: Cross sectional view of robot

The red arrow mark shows exactly the position where the expandable spring will be placed for completing the suspension system.

3.2.4 ROS Package

To achieve this project objective where to be able performing robot localization and execute robot navigation, there are few Robots Operating System (ROS) package required to be optimize in the robot micro-controller. For localization process, Simultaneous Localization and Mapping (SLAM) concept will necessitate by using the Adaptive Monte Carlo Localization (AMCL) algorithm. Adaptive Monte Carlo Localization (AMCL) is one from other thousands of Robots Operating System (ROS) packages used to localization system for the robot moving in 2D. It demands in filtering the particle to track the pose of the mobile robot at a new environment or unknown map.

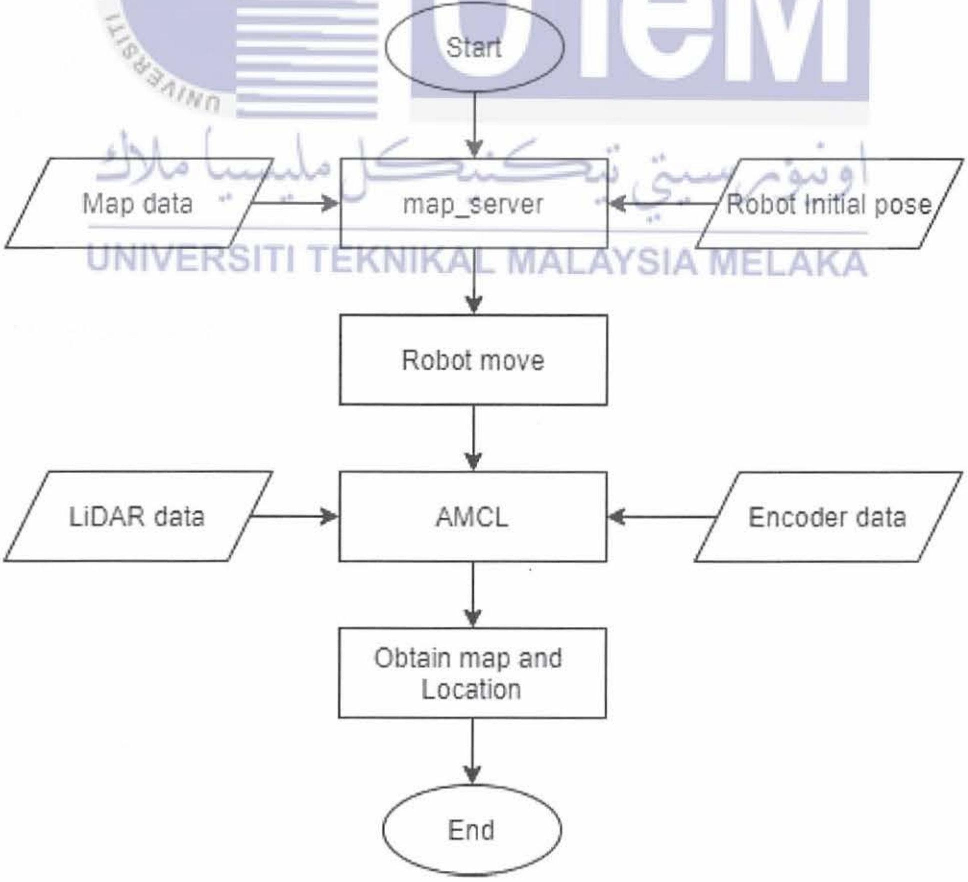


Figure 3.7: AMCL flow chart

Figure 3.7 above shows on how the concept of AMCL process in the mobile robot. By obtaining initial map data, the mobile robot will start moving to execute new mapping and stimulate the localization process. Input data from Light Detection and Ranging (LiDAR) laser sensor and odometry encoder data also will transferred into the AMCL package to be analyze. Then finally a new map and new set of location for the mobile robot will be obtained. The model of the laser and odometry sensor will be explain in the next sub chapter.

3.3 Component Integration

There are few main components needed in designing the mobile robot platform to perform all the process as a main control, motor controller, sensor input and the encoder for the motor. This all components will be fit in the body of mobile robot and linked with each other by using the Phyton software which its Operating System of Ubuntu 18.04. Apart connected using the software, these components will be connected using USB type wire and simple wiring. The wiring diagram will be stated in the Chapter 4.

3.3.1 Raspberry Pi 4

The main controller will be the Raspberry Pi 4 which will be as the main minicomputer to integrate all the ROS algorithm as the Adaptive Monte Carlo Localization (AMCL) package, control the robot movement, generate mapping, and read the sensor data input. Figure 3.8 below shows Raspberry Pi 4 minicomputer that will be used in this project. This Raspberry Pi 4 is choosing as its most compatible and stable operating with Ubuntu 18.04 Operating System (OS) which will be mainly used in operating the ROS package.

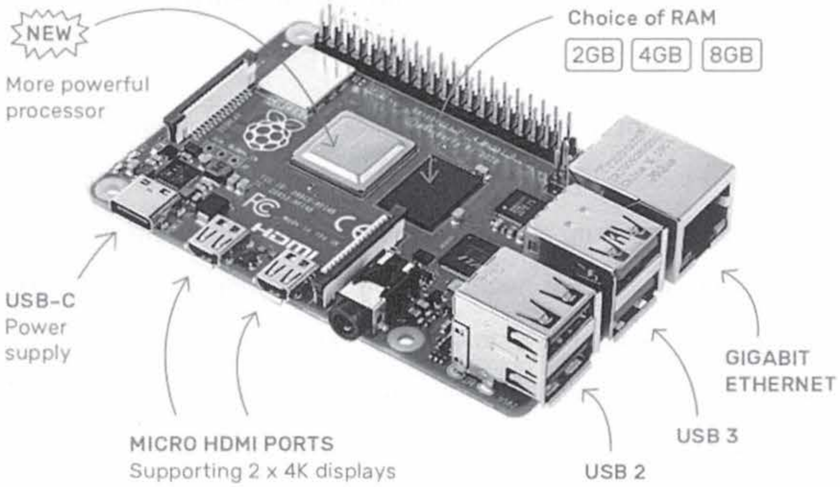


Figure 3.8: Raspberry Pi (Barkstrom, 2019)

Raspberry Pi microcomputer will be powered by the 5V micro-USB from normal 5V sources as power bank. The power bank will act as its power sources replacing its wired power sources as we will need to be freely mobile so can-do movement freely. The operating system (OS) used for this microcomputer will be Ubuntu OS as the platform to running the RViz in controlling and navigate the robot movement as well generate the mapping for the environment. In table below, stated all the characteristic description for this Raspberry Pi.

Table 3.2: Raspberry Pi (Barkstrom, 2019)

SoC	BCM2873
CPU	Quad Cortex A53 @1.2GHz
Instruction Set	ARMv8-A
GPU	400MHz VideoCore IV
RAM	1GB SDRAM
Storage	Micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
GPIO	40

3.3.3 IG-42GM DC Carbon-brush Motors

Planetary geared motors have long been recognized for their minimal backlash and high efficiency characteristics, which enable them to fulfil greater expectations. It is made by a reputable Taiwanese motor firm renowned for its quality and endurance. This 42mm diameter variant is significantly more powerful than the 32mm ones, making it ideal for applications requiring a lot of torque. The motor is equipped with a 49:1 gearhead and produces 120 revolutions per minute and 1.76N.m torque. It is well suited for ordinary automation tasks for the mobile robot. Additionally, it has a 5 pulses per rotation encoder for real-time input on the rotating position which will needed for mapping process. Figure 3.10 below shows picture of brushed DC motor that will be used in mobile robot for movement process. The parameter and characteristic for IG-42GM DC Carbon-brush Motors are listed in Table 3.3 below.

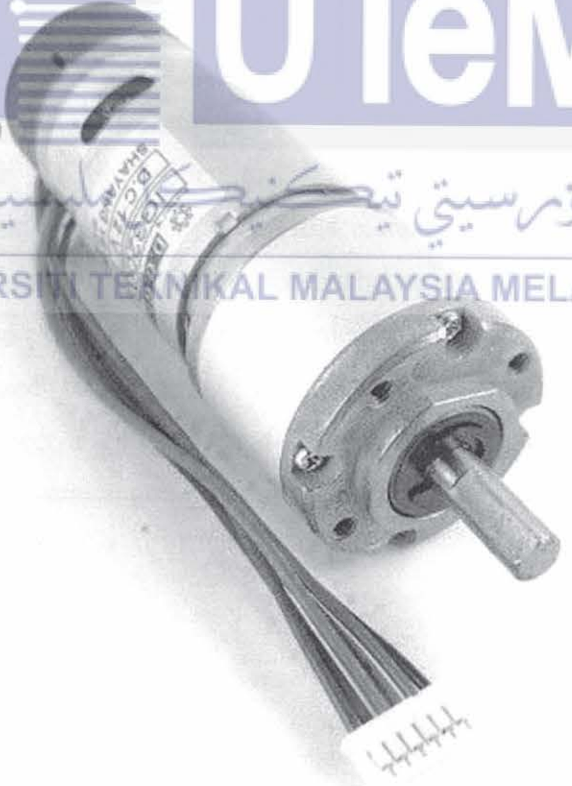


Figure 3.9: Brushed DC Motor with encoder (Sha Yang Ye, 2019)

Table 3.3: Motor specification (Sha Yang Ye, 2019)

IG-42GM DC Carbon-brush Motors	
Voltage	12VDC
Speed (RPM)	101-200
Torque (kgf.cm)	10.01-20.00
Gear Ratio	49:1
Current	5.5A
Diameter Motor	42mm/45mm
Encoder	Yes

3.3.4 HOKUYO UBG-04LX-F01 LiDAR Sensor

The main sensor for this project will be using the LiDAR sensor which is version of HOKUYO UBG-04LX-F01 a high-speed laser range finder (Rapid URG) which is one of the latest generations for low cost 2D laser scanner (LiDAR) with 240-degree point of its sense range. The UBG-04LX-F01 scanning laser rangefinder is a compact, high-precision instrument designed for robotic applications. This device has both RS232 and USB communication interfaces and can measure up to 4 meters with millimeter precision. This scanner may be utilized on battery-powered systems because to its low power consumption (HOKUYO, 2018).When performing scanning process, the obtained 2D point cloud data collected will be utilized for mapping, localization, and modelling of objects and environments in this project. Figure 3.11 and Table 3.4 below show the UBG-04LX-F01 LiDAR sensor from Japan version and its complete specification.

Figure 3.10: UBG-04LX-F01 laser sensor (HOKUYO, 2018)

The UBG-04LX-F01 by HOKUYO specification are as follow:

Table 3.4: Laser sensor specification (HOKUYO, 2018)

Model No.	UBG-04LX-F01
Power source	12VDC±10%
Detection Range	20 to 5,600mm (White Square Kent Sheet 80mm
	240°
Accuracy	Nominal Range 60 to 1,000mm: ±10mm,
	1,000 to 4,095mm:1% of Distance
Angular Resolution	0.36° (360°/1,024 steps)
Light source	Semiconductor laser diode(λ=785nm)
	Laser safety Class 1(FDA)
Scan Time	28msec/scan
Sound level	Less than 25dB
Interface	USB2.0(Full Speed),

	RS-232C (19.2k, 57.6k, 115.2k, 250k, 500k, 750kbps)
Output	Photo-coupler/NPN open-collector output (30VDC 50mA or less)
	Synchronous output and malfunction output
Command system	Exclusively designed command SCIP Ver.2.0
Connection	Power and Synchronous output:2m flying lead wire
	USB: 2m cable with type-A connector
Amblent illuminance	Halogen/mercury lamp: 10,000lx or less, Florescent: 6,000lx (Max)
Amblent(Temperature/Humidity)	-10 to +50 degrees C, less than 85%RH (without dew and frost)
Vibration Resistance	Double amplitude 1.5mm 10 to 55Hz, 2 hours in each X, Y and Z direction
Impact Resistance	196m/s^2, 10 times in each X, Y and Z direction
Weight	Approx. 260g (with cable attachment)

3.3.5 MCP263 Dual 60A, 34VDC Advanced Motor Controller

The final component used for this project is motor controller by BASICMICRO where the version of MCP263 Dual 60A, 34VDC for motion control. This motor controller will be attached with the microcomputer which is Raspberry Pi as mentioned earlier in supplying control ability for the phyton software in it by using ROS control and RViz software. Figure below shows the motor controller. Figure 3.12 below shows the BASICMICRO motor controller used in this project. This motor controller will need to control the movement and read the encoder from the Brushed DC motor install with encoder as mentioned before.

Figure 3.11: MCP263 Dual 60A, 34VDC Motor controller (BASICMICRO, 2019)

Others named for this device is Motor Control Panel (MCP) which offers a wealth of features and is highly configurable. They integrate a powerful programming language for motion control. The MCP's scripting language is based on a simple-to-use BASIC variation. The scripting language provides comprehensive control over the motor controller's functionality, including the ability to make on-the-fly modifications to the I/O and system settings. The scripting language supports integer and floating-point arithmetic operations, as well as 32-bit variables, arrays, and conditional expressions. Because the MCP's I/O is scriptable, it can be easily adapted to any application. Scripts are generated and debugged entirely inside Motion Studio's free interface. Due to the MCP's great degree of flexibility, it may operate as a stand-alone controller or integrate seamlessly into any system (BASICMICRO, 2019). Table 3.5 below shows the specification for MCP263 Dual 60A Motor Control Panel (MCP).

Table 3.5: Motor Control Panel (BASICMICRO, 2019)

Motor Channels2	Motor Channels2
Bridge Channels	Yes
Continuous Current per Channel	160A
Bridged Channels Continuous Current	120A
Max Voltage	34 VDC
BEC	5 VDC 300mA
3.3V Safe Outputs	Yes
5V Tolerant Inputs	Yes
Quadrature Encoders	Yes
Absolute Encoders	Yes
Encoder Auto Tune	Yes
Thermal Protection	Yes
Current Sense	Yes
User Selectable Peak Current	Yes
Battery Protection	Yes
Regenerative Braking	Yes
Over Voltage Protection	Yes
Under Voltage Protection	Yes
Self-Diagnostics	Yes

3.4 Data Collection

The data that to be obtain in this project is to generate the mapping and execute self-navigation for mobile robot through Simultaneous Localization and Mapping (SLAM) process. Result from data collection will be presented in the next result chapter. From the data, the mobile robot will generate the map in the RViZ software and acknowledge its exact location in the map. Only then, the final objective of this project can be achieve if the mobile are able to perform navigation process with the generated map in the new environment.

3.4.1 Mapping

As to achieve this project objective by performing localization for the mobile robot platform, mapping is one of the major data need to obtain in the process. The data acquire from RP Lidar sensor will be filter by AMCL package to generate mapping of mobile robot surrounding environment named costmap. By launching RViz which is as 3-deimensional visualization tool for ROS, the mapping for the environment can be obtain and visualize the position of mobile robot in the map. Raspberry Pi microcomputer came with ROS Melodic on Ubuntu 18.04 by Linux, so the RViz simulation opened from this platform.

3.4.2 Navigation

After derive mapping data, navigation feature will acquire from the RViz, and the position (x,y) of the robot will be determined. Data from mapping process saved in the process and will used to perform the navigation. By using RViz simulation, when the mobile robot instructed to move into another location, it will create a new path in the movement process which named planned path. The acquired path will be determined and saved in the process.

3.5 Project Analysis

To accomplish the successfulness of this project, analysis will be done as last phase of the project. This project objective will consider a success if all the objectives are achieved and give positive feedback in all process. For analysis process, there are few factors that will be determine such the accuracy of the mobile robot perform movement process in real life compared to in the RViz simulation. All the analysis will be done when the mobile robot is completed fabricated.

3.5.1 Efficiency of Robot

To determine the effectiveness and efficiency of this mobile robot, comparison of this mobile robot performance need to be evaluated among others current mobile robot. Comparison from sort of ROS package used in every different mobile robot which will be outcome in least error from the actual movement of robot and the movement data in the system. As explain in the literature review part where most of author mentioned the problem of slip tire cause the movement reading errors and affect the position in mapping not synchronize with the actual position of robot.

3.5.2 Simulation

Another analysis in this project is through the execution from simulation and the real time respond position of the mobile robot. As using the RViz simulation, the planned path generates when the new location input is stated for the robot. The movement of the actual robot moving will be determine its precision compared to the path planned in the simulation. The more the accuracy, the more precise this mobile robot is.

3.5.3 Experiment

An experiment to evaluate the main objective of this project which the mobile robot should perform localization and perform movement by the desire stated input location will be test by the robot. This experiment will be conducted in any new environment for the mobile robot so that the project is able to determine the capability of the mobile robot to generate mapping of the surrounding environment and to generate path for perform movement from location A to location B as the experiment objective. Figure 3.13 above shows example on how the map will be generated in RViz and the position that the robot position that will be tested to move to Point B location.

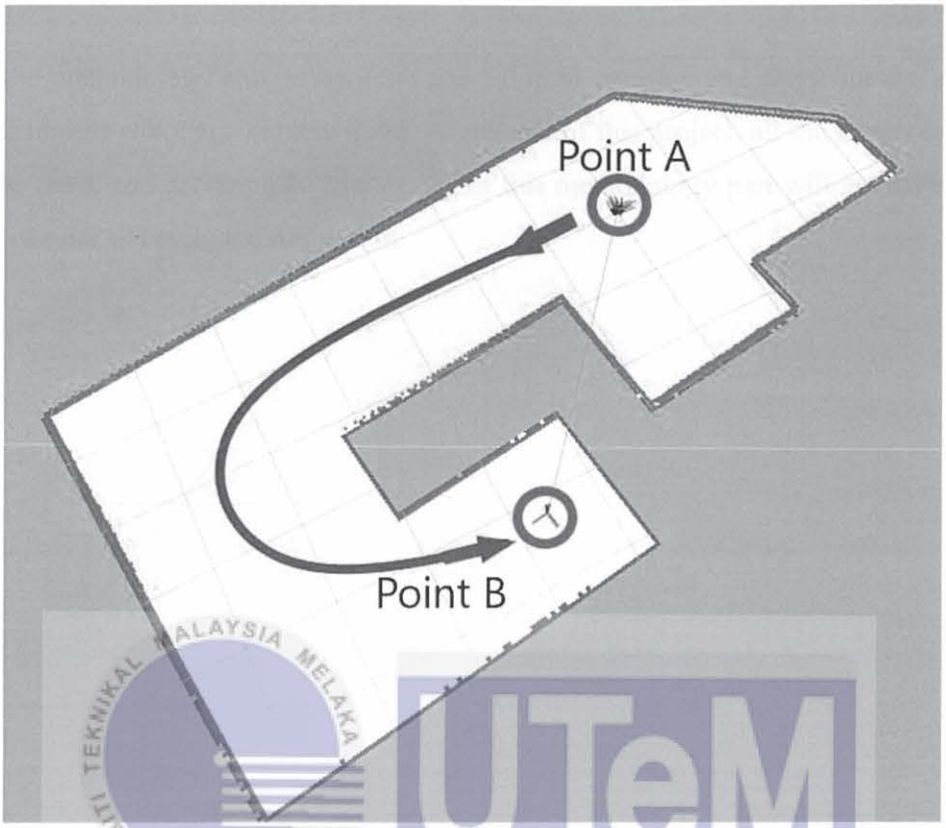


Figure 3.12: Example simulation mapping in RViz

The apparatus and material need for this experiment is the complete mobile robot rig platform, RViz simulation in Phyton operate in Ubuntu 18.04 operating system and lastly the marking position and the new environment for mapping process. The first procedure that will be used in this experiment is to mark the location at new environment which A for starting place and B for desired location point for movement. Next is place the Mobile Robot Rig Platform at location any location and turn on it. After that run the RViz simulation and perform movement to generate mapping in the new environment. When the mobile robot finished the mapping process, the generated mapping is saved in specific folder. Next, once again place the mobile robot at marked A location using the RViz simulation. After that, command the mobile robot to move to location B using the RViz and record and observe the movement of the mobile robot. This step is repeated to 10 time to observed and analyzed the mobile robot's behavior. All the data and observation are recorded to the project report.

3.6 Summary

All the methodology should be done according to the plan and carry out the analysis when the mobile robot is complete build. At the end of this project, all the three objectives should be reach and accomplish. The result for this methodology part will be presented in the next chapter of result and discussion.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The results of an experiment involving the use of a mobile robot for indoor mapping will be discussed in this chapter. This chapter will show the final design of the robot model that was created using the Fusion 360 software program. The result of the experiment will also be shown in this chapter, where it will be shown using the RViz software program. Then, the data from the experiment will be collected and evaluated in the right manner. Finally, from the result, the data will be discussed with the performance and behavior of mobile robot which will determined if this project objective will be achieve or vice versa.

4.2 Final design of mobile robot

Figure 4.1 below shows the final of develop mobile robot after finished printed with 3D Printing prototype and assembled with all the electrical components. The inside space of mobile robot where its inner space has been improved in space capacity for electrical components attachment. This modification also needs for compacting the installment of two brushless DC motor with encoder at the center of mobile robot. The lid on top of mobile robot have been modified into smaller form the actual size of design as to minimize the usage of PLA materials and most importantly to reduce the time needed to printing phase. As limited of 3D Printing machine, size of materials and time management of production set-up for this project must be reduced and maximized.

Figure 4.1: Final complete design of mobile robot

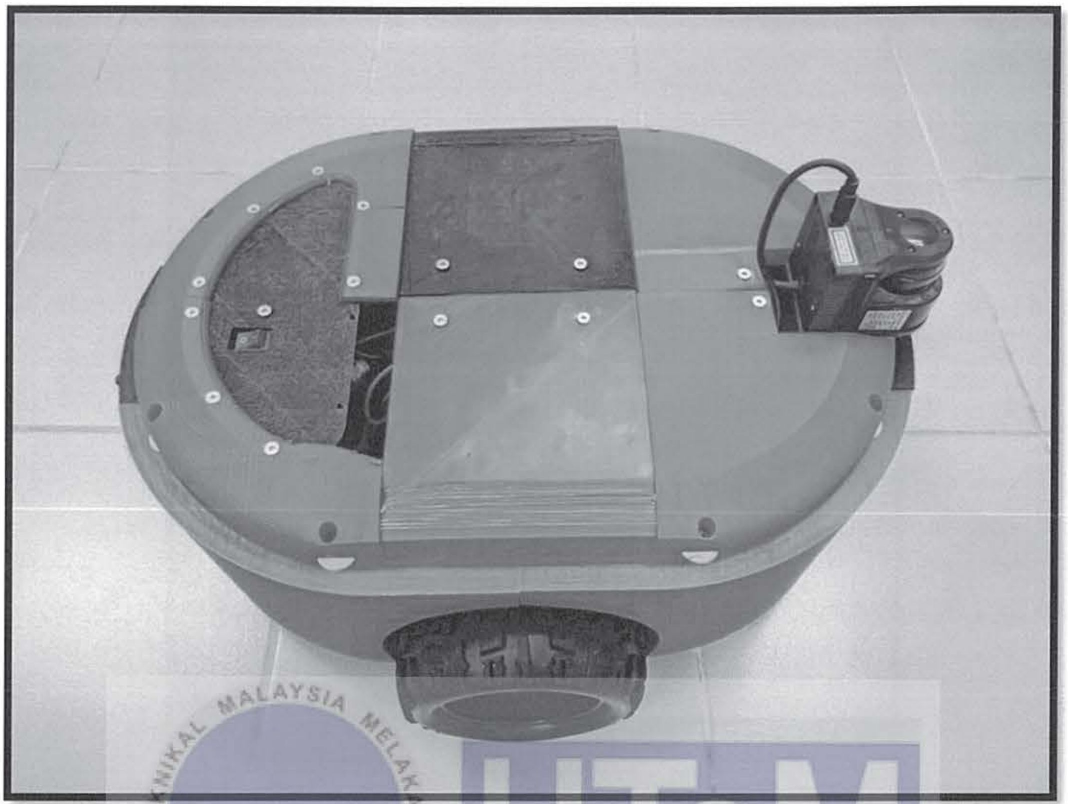


Figure 4.2: Side view of complete mobile robot

Figure 4.2 show the view of mobile robot from side view. The top right which blue color is the position of attached HOKUYO LiDAR sensor which will be the front face of the mobile robot. This HOKUYO LiDAR sensor attached at the front corner so that it can maximize the range of scanned region. As mentioned in Chapter 3, this HOKUYO LiDAR laser sensor can scan up to 240° deg of Point of View (POV). So, preventing the laser sensor to scanning the body of mobile robot, it will be best position. At the left position which is will be the back of this mobile robot, it is attached with the switch for the main micro computer power sources. It will act as the main switch to on or off the mobile robot.

Figure 4.3: Top view of complete mobile robot

Figure 4.3 show the top view of mobile robot and its back position. From this position it can be see half side of tire is exposed in both side of mobile robot. This is because of in fitting and at the same time minimizing the size of robot, the tire thickness will be resulting in exposed from the mobile robot's body. As mentioned before, the time for this project is limited and its resources in term of material supply is also limited. To amend these two factors, the mobile robot body will be minimized and fitted at its maximum scale. Finally, all the others electrical component has been fitted and install inside the body of mobile robot.

4.3 Wiring Connection

Figure 4.4 below shows the full wiring connection diagram for mobile robot. This wire connection consists of 6 components in total which are the main switch, pull-up board for encoder, brushless DC motor controller, battery source and 2 brushed DC motor. Pull-up board is designed to step down the 12V voltage from the battery to 5V output sources. The wiring diagram shows from the power sources, which is the Li-Po 12V battery will power up with board from main switch to both Brushes DC motor, motor driver and the microcomputer Raspberry Pi 4which will connected with the motor driver using the micro-USB cable. From Figure 4.4 also show that the encoder from the brushed DC motor will be connected to pull-up board encoder as to power up with 5V and connected to the motor driver.

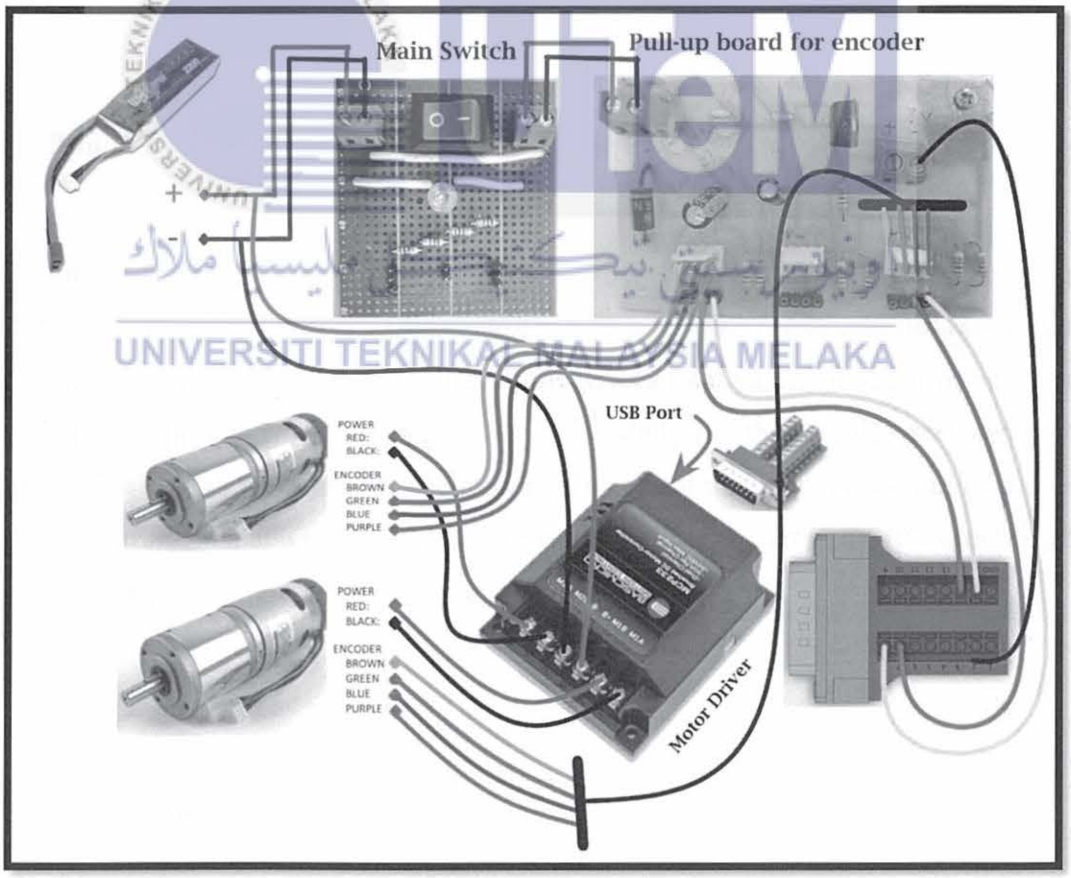


Figure 4.4: Wiring diagram

4.4 ROS Package setup

To performing ROS package to operate the mobile robot, Ubuntu 18.04 operating system will be used as the software extracting all the command needed. All the ROS command is already prepared in the wiki.ros.org website by the developer, from the sources will be selected according to this project plan and objective to execute for mobile robot. By using this Phyton operating software and its terminal, these are command for running the mobile robot. As this project is using the Phyton software to operate the ROS, all the setup and command will be needed to manually be added and applied into the Phyton terminal. Generally, there are 10 major steps in completing ROS package command from starting to its final stage where to move the mobile robot through the navigation process. The next subchapter below will explain in detail on the step and method to run the all the ROS package for mapping and navigation of mobile robot.

4.4.1 Run ROS package on Mobile Robot

First step into open ROS package is by running the roscore itself and then running the ROS launch controller check. This controller check in ROS is essential because it will refer to the ROS topic that broadcasts the mobile robot's joint values in real time and then view the robot's present status in RViz. After successfully run the roscore, next step is to launch the Basicmicro and link it to the roscore. To control the input and output of motor, the motor controller panel which is Basicmicro will be needed to start in the ROS command. Below shows all the command used for the step explained.

```
$$ roscore

$$ echo "source $HOME/catkin_ws/devel/setup.bash" >> ~/.bashrc

$$ cd catkin_ws/

$$ source devel/setup.bash
```

```
$$ roslaunch mobile_robot_autonomous_navigation controller_chec k.launch
$$ roslaunch roboclaw_node roboclaw.launch
```

After successfully run roscore with Basicmicro, the next step is to launch the teleop twist keyboard. It will be necessary to utilize a teleop twist keyboard to maneuver around the robot model. The teleop twist keyboard package is used to verify and identify the movement of the mobile robot. After that, to access the HOKUYO laser scan, the Phyton will need to open access for its USB port as In Phyton, all the access or input will need to manually configure to be used. This all is accomplished by opening a second terminal for the USB port. Then, the mobile robot will be testing the laser scan component for verification. To do this step, USB hub access is opened, and the Hokuyo LiDAR sensor will be launched and accessed with ROS package.

```
$$ roslaunch teleop_twist_keyboard teleop_twist_keyboard.py
$$ sudo chmod a+rw /dev/ttyACM0
$$ rosrun urg_node urg_node
```

The next step is to run the RViZ software. RViZ will be the main platform in performing mapping and navigation process for the mobile robot. After successfully running the RViZ, the setting for global fixed frame need to be changed to "laser" setting. This step will be achieved by adding by topic for the laser scan marker. Only then the "laserscan" will be appear in the RViZ setting interface. To use two different types of physical input in ROS, there will need to setup the 2nd port for the Hokuyo laser sensor. USB port for HOKUYO laser will be needed to change its port according to laser wire USB. For the "chmod", its need to change the "py" file to executable, then send the twist command manually as shown. All the command for execute RViZ and running the HOKUYO laser as mentioned is as below.

```
$$ rosrun rviz rviz
```

```
$$ ls /dev/ttyACM*

$$ rosrun urg_node urg_node _serial_port:=dev/ttyACM0

$$ rostopic echo /topic_name

$$ rostopic list

$$ rosnode list

$$ rosrun tf view_frames

$$ evince frames.pdf

$$ chmod +x simple_topic_publisher.py

$$ rostopic pub -r5 /cmd_vel geometry_msgs/Twist "linear:
```

Finally, after the RViZ and laser sensor are launched in ROS, the next stage is to launch the gmapping package and perform the localization and navigation process. In assisting to do mapping process, the roscore is launched with hector slam package which will be being used for generating the map for the environment in the RViZ. By using the gmapping and hector slam package, the mapping process will be started and on the RViZ software will show the map generated in real-time. As a follow-up, after creating a detailed map of an area, the map saver command saves the map in both .pgm and .yaml formats. To use the gmapping and hector slam package, the position of the laser sensor according to the mobile robot body will need to be define which is the “tf” is set from base_foorprint to laser running command. To align the map with the laser scan, it will need to launch the hector slam. If the hector slam is successfully launched, then the map generated will be in high accuracy and smooth.

```
$$ echo "source $HOME/catkin_ws/devel/setup.bash" >> ~/.bashrc

$$ cd catkin_ws/

$$ source devel/setup.bash

$$ roslaunch mobile_robot_autonomous_gmapping.launch
```

```
$$ rosrun map_server map_saver -f mymap

$$ rosrun azamrobot_setup_tf tf_broadcaster

$$ roslaunch hector_slam_launch tutorial.launch
```

Lastly, after launching the gmapping and hector slam, “map” topic is added in the RViZ and the “fixframe rviz” is changed to “map” also. When all the command is successful, the mobile robot will be move manually to around the new environment to perform the mapping process. When the mapping process is done and the RViZ shows the map result, it will be saved to the computer in the navigation folder. After the mapping file are successfully to saved and generated, the map will be retrieved to performing the navigation process. Then the map file is moved to move_base.launch. By launching the navigation stack, the mobile robot will be able to move in any position desired in the map by pointing in the RViZ. This method also will be used in the mentioned experiment before to move the mobile robot from position A to position B. To increase the accuracy of navigation process, AMCL packages must be launched to get complete map data in the mapping process. The AMCL and odometry data will be obtained using the produced map. The following command will be used for all the step explained.

```
$$ rosrun map_server map_saver -f namamapfile

$$ move_base.launch file

$$ echo "source $HOME/catkin_ws/devel/setup.bash" >> ~/.bashrc

$$ cd catkin_ws/

$$ source devel/setup.bash

$$ roslaunch mobile_robot_autonomous_navigation.launch
```

4.4.2 Requirement and test results

In confirming all the ROS package that need for localization and navigation of mobile robot, all the launch result is recorded as reference. The table 4.1 below summaries the simulation requirements and test results, proving that all criteria for performing mapping analysis have been satisfied and passed. All the packages will be needed to pass in performing the main task which for robot to undergo localization and navigation process. The all command describes and mentioned may be differ from time to time as all the command belong to the ROS developer in the wiki.ros.org. Lastly, all these packages must be performing on Phyton that running the Ubuntu 18.04 operating system installed with ROS melodic only.

Table 4.1: Requirement and test result

No.	Requirement	Test Results
1.	ROS Launch Controller check	PASS
2.	ROS Launch Basicmicro	PASS
3.	ROS Launch teleop twist keyboard	PASS
4.	ROS Launch USB Access	PASS
5.	Launch testing for Hokuyo laser scan	PASS
6.	ROS Launch RViz	PASS
7.	ROS Launch Teleop twist keyboard	PASS
8.	ROS Launch gmapping	PASS
9.	ROS Run Hector SLAM	PASS
10.	ROS Run Navigation stack	PASS

4.5 Experiment result

In observing the mobile robot behavior and determining its performance in accuracy, it's have been analyzing with the mobile robot abilities in do mapping process for new environment and lastly to measure of mobile robot's distance length in the x-y direction error on final experiment from position A to position B when performing the navigation process through the RViZ. Figure 4.5 below shows mapping generate by the mobile robot in a new environment where the mapping that generated by manually moving the mobile robot on a new environment area. The map shows high accuracy and smoothness of mapping process with only few undefine space where the extra white line with black dot in outer space of it. This phenomenon occurs as there are reflective component or things in the map environment which preventing the laser from receive back its signal.

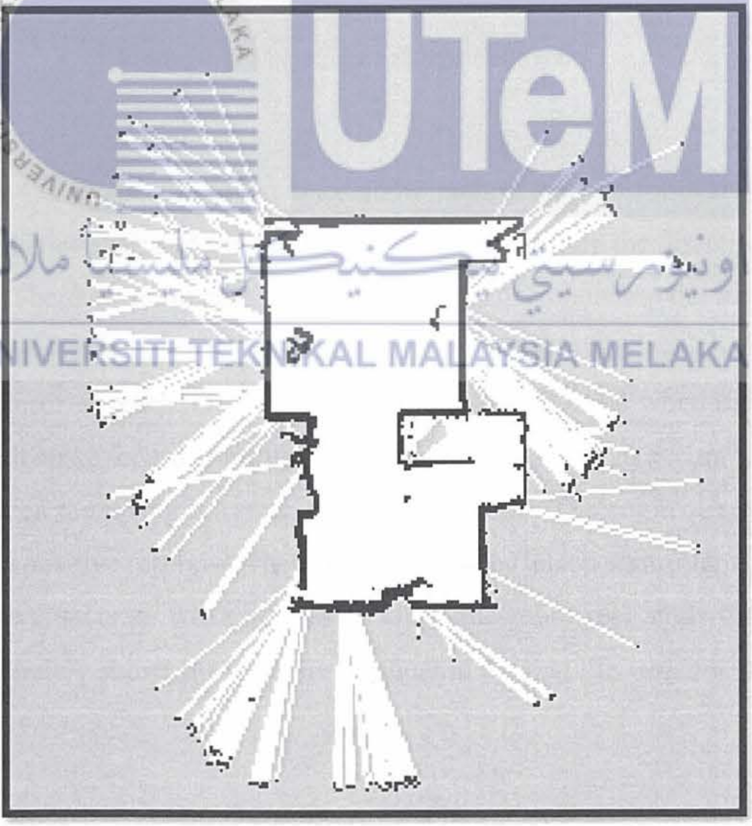


Figure 4.5: Mapping of new environment

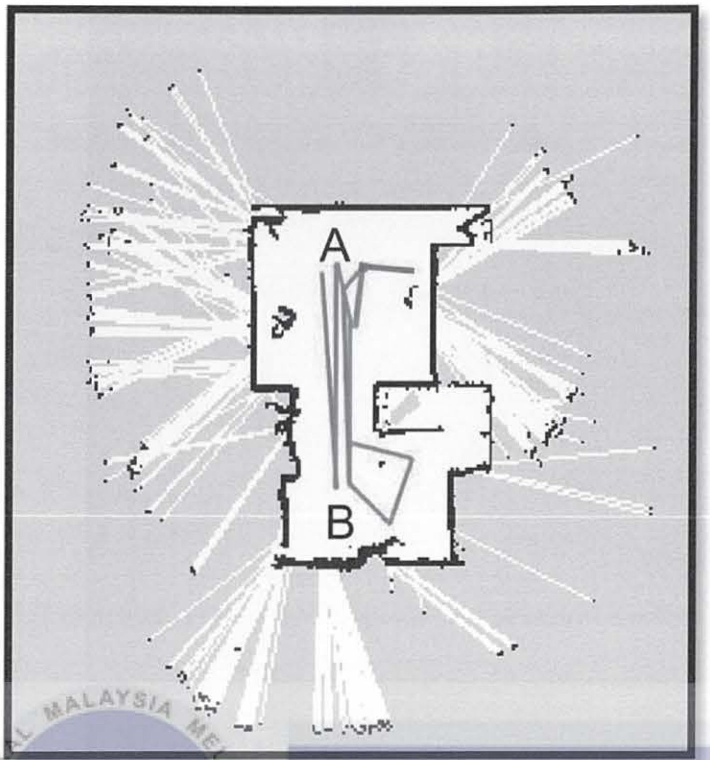


Figure 4.6: Mapping track of mobile robot

In Figure 4.6 above, the green lines show path of mobile robot movement when performing mapping prosses and the red line with blue dot define the direction of real-time when its moving. There are quite number of paths needed in completing the mapping process for the new environment as the space contain multiple confines spaced which will be difficult for the LiDAR sensor to reach. Overall, by observing the real-time when the mobile robot move using the teleop keyboard controller, its movement shows high accuracy in term of its localization between the position in map and the real world. The angle direction of mobile robot also shows a positive response when controlled by the teleop controller where the angle in the RViZ is very accurate when compared to mobile robot real angle direction in real world. Figure 4.7 below shows the new environment in the real life which being mapped by the mobile robot.

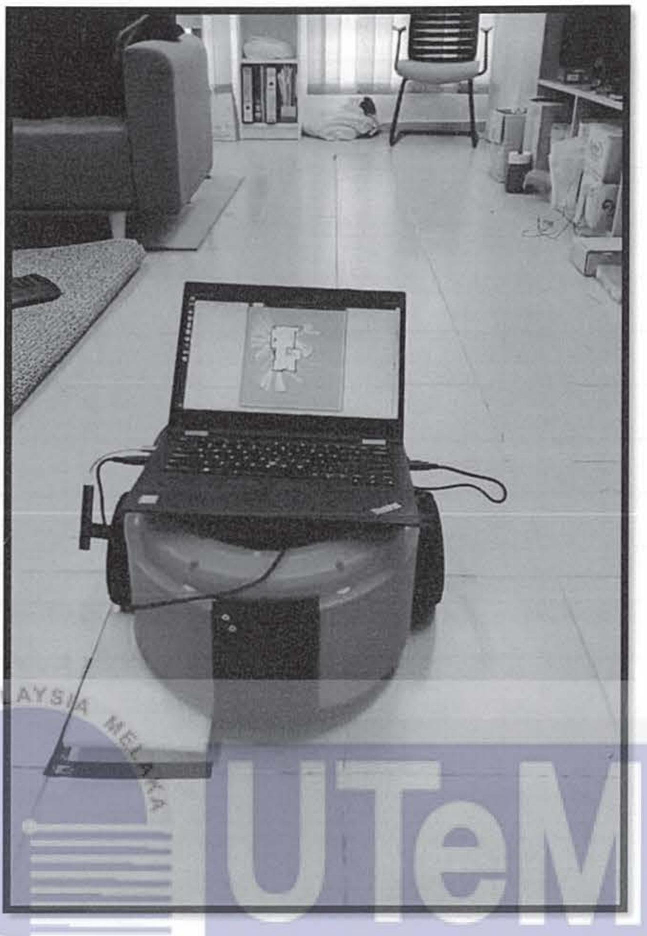


Figure 4.7: New environment for mapping

As mentioned in the experiment setup in Chapter 3, Table 4.2 below shows the result for experiment where describe the value of distance error of mobile robot last position at position B from the marked placed. The value is calculated by measured between corner of white cardboard placed under mobile robot's body ant the marked black tape location at point B. Figure 4.8 below describes how the x and y direction error are calculated during the experiment process. The distance error is only calculated in this experiment as the angle direction of mobile robot is very accurate with mobile robot's in RViZ software compared to the real mobile robot's angle position. The speed of this mobile robot in undergoing the experiment is same for all the 10 times testing process to only find the average of position error. Real computer was attached on the mobile robot to shows the screen of RViZ process in real-time.

Table 4.2: Position error

No	X-Direction Error (mm)	Y-Direction Error (mm)
1	13	4
2	7	5
3	12	7
4	11	6
5	13	5
6	9	6
7	15	8
8	4	3
9	8	7
10	9	5
Average	10.1	5.6

Figure 4.8: x-y measured direction

The result in Table 4.2 shows average of 10.1mm of final mobile robot's position error for x-direction and 5.6mm for y-direction. These position error define this mobile robot are mostly accurate in performing the navigation process and only have only around 1 cm misplaced in the final position place. Position error is because of there is occurrence of slippage during the mobile robot movement. After being observed and analyze, this problem may cause of the slippage occurs when the mobile robot's caster wheels are carried by the tile's drain as the size of their contact area is small. To overcome this problem, the type of the casting wheels will be changed in the future to increase its contact area size touched with the ground avoiding for the mobile robot to be carried away from its origin path when performing movement process.

The speed of the mobile robot is being observed by performing run testing before beginning the experiment. This speed optimizing is needed for the mobile robot's consistency in experiment. As for safety and precaution, the speed of mobile robot is limited to 40mm/s to preventing unwanted accident scheme during this experiment process. The standard speed that has been applied on the mobile robot which this experiment speed is 40mm/sec or 144m/hour while the maximum speed for mobile robot can accelerate is 80mm/sec which around 288m/h. The duration needed for one complete movement from position A to position B in the experiment is approximately 9.75 second. Total length for one complete path between position A and position B is 390cm which is for 13 tiles time by 30cm per tile length. Figure 4.9 below shows the tiles floor diagram which the mobile robot is moved for ten time from position A to position B.

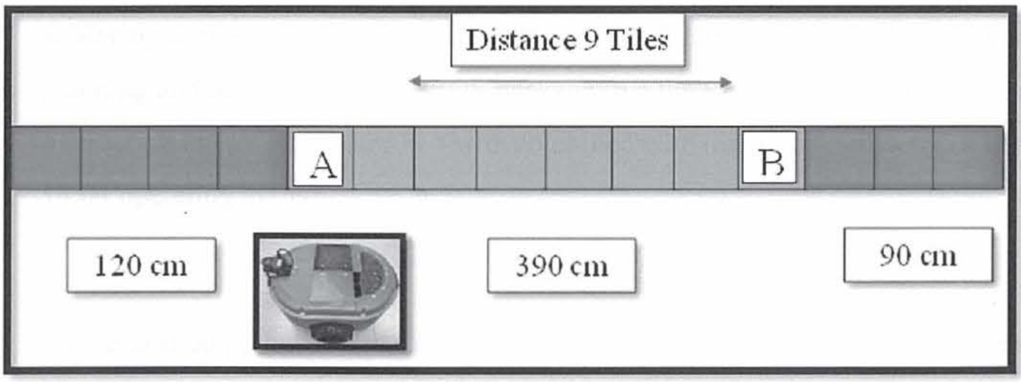


Figure 4.9: Floor diagram of experiment setup

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This project main purpose is to develop a mobile robot platform rig for transportation of goods in industry using the Robot Operating System (ROS) package. From the results and discussions that have occurred in the previous chapter can be used to form a conclusion that summarizes all the findings of the research and can be represented in accordance with the objective that have been defined.

i. The first objective is to develop a mobile robot platform testing rig programming framework and control in Robot Operating System (ROS). This objective has been achieving as the mobile robot platform have been successfully designed based on the planning and being install with ROS package. All the ROS have been attached in the microcomputer of Raspberry Pi 4 with combined with the Python software of Ubuntu 18.04 operating system.

ii. The second objective is to perform data collection from the surrounding environment in creating mapping through Simultaneous Localization and Mapping (SLAM) process. Based on the result on Chapter 4, the mobile robot has generated a map of the new environment smoothly and accurately using the mapping process through the SLAM. The path that's shows mobile robot's movement while performing mapping shows that it can locate and state its position on to the map although while generating the map.

iii. The final objective state is to analyze the behavior of the mobile robot when moving from a position to an input desired position. During the experiment, the mobile robot movement path accuracy has been measured and analyze. Result shows that there are slight few positions error in the x-direction which is 10.1mm and 5.6mm for y-direction. But above all, the result also stated that this mobile robot platform has high accuracy where the position of mobile robot in real life is similar in RViZ mapping.

5.2 Recommendation for future work

This project has shown the successful development of mobile robot platform using the ROS algorithm. As the purpose of this mobile robot in replacing the exist AGV mobile robot in the industry, extra feature could be improvise as attaching the Pick and Place robot technologies. So that while performing the autonomous navigation, the mobile robot could function in inserted and delivered the goods on industry efficiently. In increasing the safety of the mobile robot, it is advice to installing a manual fuse at the electrical component board and controller. By this method, it will lower the risk of possible damage occurrence for the component which will be adding extra cost in the development process.

5.3 Sustainability

As this project used the rapid prototyping process which consisting of Additive Manufacturing by 3D printing process, the additive manufacturing itself already represents a more sustainable means of production. But for massive production, by using the rapid prototyping technique will be more load and not efficient.

So, the making of this mobile robot product more competitive requires manufacturing sectors to consider factors such as economic efficiency, environmental sustainability, and social benefits when developing processes and utilizing new manufacturing technologies to enhance a product's competitive position. Not limit to production stage, by extending the product lifetime with the correct life cycle analysis could helping in lowering resources consumption and preventing worsting environmental effect.

5.4 Complexity Engineering

Systems, components, or processes that combine critical areas and satisfy established goals while taking public health and safety, cultural, social, and environmental factors into account should be addressed in design initiatives. Real complexity, time-independent imagination, time-dependent combinatoriality, and time-dependent periodicity all derive from the phenomenon known as complexity. It's important to keep engineering systems like products and manufacturing processes as simple as possible while creating them in order to maximize their long-term robustness, stability, dependability, and cost efficiency.

REFERENCE

Akash, U., Ananya, N., Sandeep, S., & Deepti, R. (2019). Autonomous Navigation with Collision Avoidance using ROS. *Journal of Remote Sensing GIS & Technology*, *5*(2), 14–18.

Appala, T., & Ghosal, A. (2015). A mobile robot with a two-degree-of-freedom suspension for traversing uneven terrain with minimal slip: Modeling, simulation and experiments. *Mechanism and Machine Theory*, *93*, 83–97. https://doi.org/10.1016/j.mechmachtheory.2015.06.009

Barkstrom, J. (2019). What is a Raspberry Pi? In *Introduction to the Raspberry Pi* (p. 1). https://doi.org/10.1007/978-1-4842-5571-1_1

BASICMICRO. (2019). *MCP263 Dual 60A, 34VDC Advanced Motor Controller*. Motor Control Device. https://www.basicmicro.com/MCP263-Dual-60A-34VDC-Advanced-Motor-Controller_p_40.html

Cybulski, B., Wegierska, A., & Granosik, G. (2019). Accuracy comparison of navigation local planners on ROS-based mobile robot. *12th International Workshop on Robot Motion and Control, RoMoCo 2019 - Workshop Proceedings*, 104–111. https://doi.org/10.1109/RoMoCo.2019.8787346

Guimarães, R. L., de Oliveira, A. S., Fabro, J. A., Becker, T., & Brenner, V. A. (2016). ROS navigation: Concepts and tutorial. *Studies in Computational Intelligence*, *625*(May 2018), 121–160. https://doi.org/10.1007/978-3-319-26054-9_6

HOKUYO. (2018). *Scanning Rangefinder Distance Data Output/URG-04LX-UG01 Product Details | HOKUYO AUTOMATIC CO., LTD.* https://www.hokuyo-aut.jp/search/single.php?serial=164

Housein, A. A., & Xingyu, G. (2021). Simultaneous Localization and Mapping using differential drive mobile robot under ROS. *Journal of Physics: Conference Series*, *1820*(1), 012015. https://doi.org/10.1088/1742-6596/1820/1/012015

Jiang, H., Xu, G., Zeng, W., & Gao, F. (2019). Design and kinematic modeling of a passively-actively transformable mobile robot. *Mechanism and Machine Theory*, *142*, 103591. https://doi.org/10.1016/j.mechmachtheory.2019.103591

Jusuf, F. (2016). *Auto-Navigation For Robots . Implementation of. March*, 1–24.

Kannan Megalingam, R., Ravi Teja, C., Sreekanth, S., & Raj, A. (2018). ROS based Autonomous Indoor Navigation Simulation Using SLAM Algorithm. *International Journal of Pure and Applied Mathematics*, *118*(7), 199–205.

Kim, T., Kim, W., Choi, S., & Jin Kim, H. (2017). Path Tracking for a Skid-steer Vehicle using Model Predictive Control with On-line Sparse Gaussian Process. *IFAC-PapersOnLine*, *50*(1), 5755–5760. https://doi.org/10.1016/j.ifacol.2017.08.1140

Köseoğlu, M., Çelik, O. M., & Pektaş, Ö. (2017). Design of an autonomous mobile robot based on ROS. *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*. https://doi.org/10.1109/IDAP.2017.8090199

Lauer, A. P. R., Uchiyama, N., & Sawodny, O. (2020). Design and Experimental Verification of Two-Dimensional Rate Limiters in Trajectory Generation for Differential Drive Robots. *IFAC-PapersOnLine*, *53*(2), 9664–9669. https://doi.org/10.1016/j.ifacol.2020.12.2614

Li, Z., & Mei, X. (2018). Navigation and Control System of Mobile Robot Based on ROS. *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2018*, *Iaeac*, 368–372. https://doi.org/10.1109/IAEAC.2018.8577901

Mishra, R., & Javed, A. (2018). ROS based service robot platform. *Proceedings - 2018 4th International Conference on Control, Automation and Robotics, ICCAR 2018*, 55–59. https://doi.org/10.1109/ICCAR.2018.8384644

Newman, A., Yang, G., Wang, B., Arnold, D., & Saniie, J. (2020). Embedded Mobile ROS Platform for SLAM Application with RGB-D Cameras. *IEEE International Conference on Electro Information Technology*, *2020-July*, 449–453. https://doi.org/10.1109/EIT48999.2020.9208310

Ordonez, C., Gupta, N., Reese, B., Seegmiller, N., Kelly, A., & Collins, E. G. (2017). Learning of skid-steered kinematic and dynamic models for motion planning. *Robotics* *and Autonomous Systems*, *95*, 207–221. https://doi.org/10.1016/j.robot.2017.05.014

Poojari, M., Shreyas, B. S., Muddaiah, R., Raj, A., & Babu, B. S. (2021). Design, analysis of steering system and front suspension for an Electric All-Terrain vehicle. *Materials Today: Proceedings*, *xxxx*. https://doi.org/10.1016/j.matpr.2021.03.102

Priyandoko, G., Ming, T. Y., & Achmad, M. S. H. (2017). Mapping of unknown industrial plant using ROS-based navigation mobile robot. *IOP Conference Series: Materials Science and Engineering*, *257*(1). https://doi.org/10.1088/1757-899X/257/1/012088

Sha Yang Ye. (2019). *12V 63RPM 20kgfcm Planetary DC Geared Motor with Encoder*. Cytron. https://my.cytron.io/p-12v-120rpm-18kgfcm-planetary-dc-geared-motor-with-encoder?r=1

Tang, S., Yuan, S., Hu, J., Li, X., Zhou, J., & Guo, J. (2017). Modeling of steady-state performance of skid-steering for high-speed tracked vehicles. *Journal of Terramechanics*, *73*, 25–35. https://doi.org/10.1016/j.jterra.2017.06.003

Valera, Á., Valero, F., Vallés, M., Besa, A., Mata, V., & Llopis-Albert, C. (2021). Navigation of autonomous light vehicles using an optimal trajectory planning algorithm. *Sustainability (Switzerland)*, *13*(3), 1–23. https://doi.org/10.3390/su13031233

Wang, S., Wang, K., & He, H. (2019). An autonomous navigation method for mobile robot based on ROS. *WRC SARA 2019 - World Robot Conference Symposium on Advanced Robotics and Automation 2019*, 284–290. https://doi.org/10.1109/WRC-SARA.2019.8931954

Zhao, S., Zhao, J. S., & Feng, Z. J. (2011). Design of a pair of noncircular gears meeting Ackermann steering principle. *2011 International Conference on Consumer Electronics, Communications and Networks, CECNet 2011 - Proceedings*, 217–221. https://doi.org/10.1109/CECNET.2011.5768797

Zheng, K. (2017). ROS Navigation Tuning Guide. *ArXiv*, *2019*, 1–23.

APPENDICES

GANTT CHART FOR FYP 1

Activities	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9	WK10	WK11	WK12	WK13	WK14	WK15	WK16
Title Selection	■															
Registration of FYP Title	■															
Endorsement of FYP Title		■														
Lecture 1: Chapter 1 (Introduction)		■														
Problem Statement & Objectives		■	■													
Project Scope		■	■													
Submission of Chapter 1 draft				■												
Lecture 2: Chapter 2 (Literature Review)					■											
Submission of Chapter 2						■										
Amendments on Chapter 2						■	■									
Lecture 3: English Technical Talk						■										
Lecture 4: Chapter 3 (Methodology)							■									
Discussion on Methods to be used								■	■							
Submission of Logbook to Supervisor										■	■					
lecture 5: Reference and Formatting											■					
Amendments on Chapter 3												■	■	■		
Submission of FYP 1 Report															■	■

GANTT CHART FOR FYP 2

Activities	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9	WK10	WK11	WK12	WK13	WK14	WK15	WK16
Methodology Confirmation	■															
Design Setup		■	■	■												
Printing Setup		■	■	■	■											
Programming for Localization					■	■										
Programming for Navigation						■	■									
Whole Component Setup							■	■	■	■						
Program Testing										■	■					
Program Amendments										■	■					
Final Testing										■	■					
Amendments on Chapter 3											■					
Results Writing											■					
Conclusion Writing											■					
FYP 2 Presentation												■				
Submission of Logbook to Supervisor												■				
Submission of Report to Panel and Supervisor												■				
Amendments on Report													■	■		
Resend Report															■	