Simultaneous Localization and Mapping for Land Robot with Lidar and IMU Sensor

BY

DARREN PHANG REN YEE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2019

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____

____

____

Academic Session: _____________

I ____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

____

_________________________ _________________________

_________________________ Supervisor's name

Date: _____________________ **Date**: ____________________

Simultaneous Localization and Mapping for Land Robot with Lidar and IMU Sensor

By

Darren Phang Ren Yee

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF SIMULTANEOUS LOCALIZATION AND MAPPING FOR LAND ROBOT WITH LIDAR AND IMU SENSOR**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________________

Name : ____________________________

Date : ____________________________

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr. Lee Wai Kong who has given me this bright opportunity to engage in a mobile robot design project and unconditional support for my project. It is my first step to establish a career in mobile robot design field. A million thanks to you.

To a special person, Professor Nobuto Matsuhira, for his patience, support and provide me opportunities to learn more about current trending mobile robot during my research internship in Japan. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

The Robotic Operating System (ROS) provides user an environment similar to operating system to control or operate the robots. Simultaneous localization and mapping (SLAM) and autonomous navigation in an enclosed environment are nowadays popular autonomous robots’ issues. A robot in an unknown environment is quite challenging for autonomous navigation, but it also reveals many indoor environmental factors that may affect mapping and navigation performance. This presented work describes how an ROS-based control system is used with a self-designed mobile robot for indoor mapping, localization, and autonomous navigation. In addition, some factors that may affect mapping, localization, and automatic navigation indoor environments are also presented.

LiDAR is used as the main technique to perform SLAM in this project. However, LiDAR alone is not accurate enough when the traveling distance is far. We proposed to integrate IMU sensor into the mobile robot to assist LiDAR for more accurate odometry are also presented in this project. Most of the implementation was done using C and Python. The proposed technique able to reduce the localization error by 20 - 30%.

This project can be extended to support other advanced features. For example, the technique developed (LiDAR + IMU) can be used to track service robot in indoor environment more accurately. The result of this project was submitted to IEEE International Meeting for Future Electron Devices, currently under review process.

Tables of Contents

List of Tables

List of Figures

List of Abbreviations

SLAM	Simultaneous Localization and Mapping
ROS	Robotic Operating System
LRF	Laser Range Finder
LiDAR	Light Detection and Ranging
IMU	Inertial Measurement Unit

Chapter 1.0: Introduction

1.1 Problem Statement and Motivation

Monitoring situations or exploring environment that are dangerous to humans, using mobile robots are the ideal solution. Examples include search and rescue operations, such as the nuclear disaster near Fukushima, Japan (Michael et al., 2012). As we all know human beings possesses natural senses to help them get information on the environment for example eyes for vision, ears for hearing, nose for smelling, and feels objects through touching but a robot does not possess all these natural senses get information of the surrounding though these senses unless using sensors. Exploration of unknown environment for a mobile robot will be very difficult unless some sensing sources are provided to obtain environmental information. Odometers, sonars, laser range finders (LRF), inertial measuring units (IMU), global positioning system (GPS) and cameras are the famous current technology sensors to create a robot that can detect a broad range of environments. A mobile robot will need an environment map before performing indoor services such as moving from one coordinate to another, grabbing and picking an object from one place to another. To carry out this kind of action, the robot should be aware of its own environment location while having, for example, 2D map information about the environment. Thus, it is very important to have specific representation of each robot while in the 2D map in order to differentiate the robot location in the map.

The problem of determining the pose of a robot relative to a given representation (i.e. map) of the environment is the mobile robot location or pose estimation. More than two decades ago, Cox considered providing a mobile robot with autonomous capabilities to be the fundamental problem and it remains true to our day (Fu et al., 2016). Many robotic tasks require knowledge of the robot's location and the objects being manipulated. More and more attention been given to robot research and application. Mobile indoor services in order to make smart decisions, robots need to be aware of the indoor environment. While the robot's perception of the indoor environment is accurate and complete, robot intelligent decision accuracy is also determined. Autonomous mobile robot is one of the most common smart robots, the prerequisite for autonomous navigation and other functions is the "where I am" problem that can best reflect its autonomy (Yuan et al., 2016a).

Thanks to the progress in the design of sensors and processors. Besides, there more researchers been testing the combining of sensors with embedded system to execute some smart mapping feature. Therefore, the development of intelligent robot is very fast. Thus, some researchers also carried out real-time implementation of the embedded system SLAM algorithm. (Vincke et al., 2010) developed an architecture of hardware, co - design, a function detector, an algorithm for SLAM and a system architecture based on optimization methodology and (Vincke et al., 2012) also proposed an EKFSLAM Multi-Core Architecture Implementation Algorithm. However, due to limited amount of CPU memory consumption, running a mapping process in embedded system is not an easy task. I want to use an embedded processor (Raspberry Pi) to operate a mapping process and to create maps for the robot in my project and perform autonomous navigation by using the knowledge of combining various sensors integrate with embedded system to allow the robot to become more smart intelligent.

1.2 Project Scope

This project develops a mobile robot design which able to perform SLAM, autonomous navigation and a robotic arm for gripping and releasing object. This project involves a new design of mobile robot combining various sensors with an embedded system.

1.3 Project Objective

The objective of this study is:

1) To develop a low-cost mobile robot able to perform 2D mapping and localization in an indoor unknown environment.
2) To develop a mobile robot able to perform autonomous navigation.
3) To increase the accuracy of the mobile robot odometry when perform SLAM.

1.4 Impact, significance, and contribution

Control system is the most important section for any mobile robot, as it is like a core for mobile robot movement performance, accuracy, and stability. In addition, autonomous navigation is one of mobile robots ' essential functions. While line-tracking is the most commonly used autonomous navigation technology today. Although line-tracking is an easy and cheap way to solve this problem. It is not intelligent, however, and it is not well adapted to the environment. When the line path is blocked or disrupted, the precision and stability will drop. Therefore, SLAM will take over to make the mobile robot smarter.

1.5 Background Information

ROS is a robot's meta-operating system. It provides services expected from an operating system, including abstraction of hardware, low-level device control, commonly used implementation of functionality, message-passing processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code from multiple computers. It's called a meta-operating system because it's something between a middleware and an operating system. It not only provides standard operating system services (such as hardware abstraction), but also high-level features such as asynchronous and synchronous calls, centralized database, robot configuration system, etc. ROS can also be interpreted as an operating system software framework for the development of robot software.

Computer science and robotics development has received widespread attention from advancement of mobile robots today. It is difficult to write software for robots, especially as robotics scale and scope continue to grow. Various types of robots may have wildly different hardware, resulting in non - trivial code reuse. Furthermore, the sheer sizing of the prerequisite guidelines must be disheartening, as it may encompass a deep batch starting with driver - stage firmware and persisting thru preconception, representational argumentation, and beyond. Before the prerequisite scale of acumen extends far beyond any single researcher’s capacities, tech proprietary frameworks also ought to endorse exhaustive endeavors to incorporate apps.

To meet these obstacles, many tech academics have subsequently constructed a small repertoire of technologies to manage unpredictability and stimulate the rapid prototyping of instrumental computers, triggering in the successive conventional programming mechanisms today used in academia and economy (Kramer and Scheutz, 2007). Every of these paradigms was modeled for a specialized function, conceivably in response to interpreted foibles of other available paradigms, or to concentrate on facets that were contemplated many vital in the performance phase.

ROS is also the manifestation of compromises and prioritizations sourced during its performance phase, the framework described in this paper. I believe that its focus on large-scale integrative robotics research will be useful in a wide range of situations as robotic systems are becoming increasingly complex. Because every mobile robot is widely used in many areas like industrial, logistics and so on. Therefore, each mobile robot will have their own specific function.

Chapter 2.0: Literature Review

I. Mobile Robot setup for SLAM

SLAM has been an active studies sector for special mobile robots' implementations (Song et al., 2018). It solves the robot's positioning issue on the map while constructing the map (Song et al., 2018). The basic concept is to update the robot position and map the environment by scanning perception sensor information. (Bailey and Durrant-Whyte, 2006, Song et al., 2018). Multiple researchers postulated constructive instruments and computations for slam difficulties (Schroeter and Gross, 2008, Liu and Sun, 2012, Trivun et al., 2015, Song et al., 2018). Laser scanners and visual sensors are the popular SLAM sensors (Fossel et al., 2015, Yuan et al., 2016b). The visual sensing gadgets can supply a staggering level of knowledge in an picture, but the predictive load is very vast, and the behaviors of the picture are shaped by the condition of economic illumination (Song et al., 2018). By contrast, laser scanners (LiDAR) are more reliable, but in general they are more expensive (Song et al., 2018).

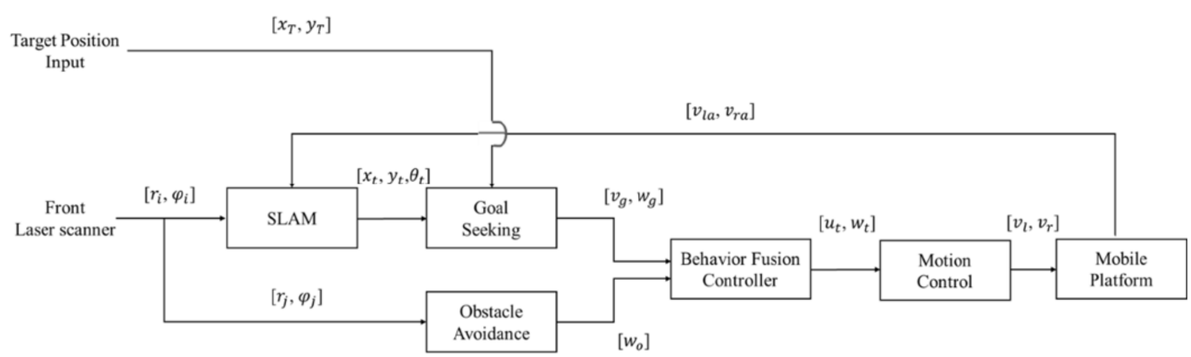


Figure 2.1: Architecture of navigation proposed by (Song et al., 2018).

As shown in Figure 2.1, proposed navigation control structure to interweave the evasion function with the LiDAR SLM- guidance transmitter focused on it (Song et al., 2018).

The proposed prototype embraces the cartographer SLAM to obtain independent translation and map creation (Hess et al., 2016, Song et al., 2018). Cartographer SLAM can calculate the robot's location in real time and create a LiDAR sensor - based environment map (Song et al., 2018). The resulting map is 5 cm in resolution. This algorithm is available in ROS (Nüchter et al., 2017). So, when the mobile robot moves, it will generate a map of the surrounding region. The global map is made up of a number of submaps with a number of laser scans (Song et al., 2018). The proposed design adopts the cartographer slam to retrieve independent passage and map adoption (Song et al., 2018).

Monte Carlo adaptive localization tactic (AMCL) to pinpoint the mobile robot on a acknowledged map is used in (Song et al., 2018) (Thrun et al., 2005). AMCL algorithm suites already exist in ROS. Monte Carlo's adaptive positioning method is a probabilistic model positioning system (Song et al., 2018). It uses the particle filter method of Monte Carlo localization (MCL) to locate the pose of the mobile robot on a known map and to simulate the sensor at each particle's position (Song et al., 2018). Compared to the observed information, the information gives the probability that each particle could exist and updates the particle number through KLD-sampling (Song et al., 2018). Since multiple phases, all particulates steadily converge to the position of the mobile automaton (Song et al., 2018). The procedure above allows positioning of the mobile robot on a known map (Song et al., 2018). However, such optimizations are not required in many scenarios under real world conditions as the approach is sufficiently accurate for robots to fulfill their mission.

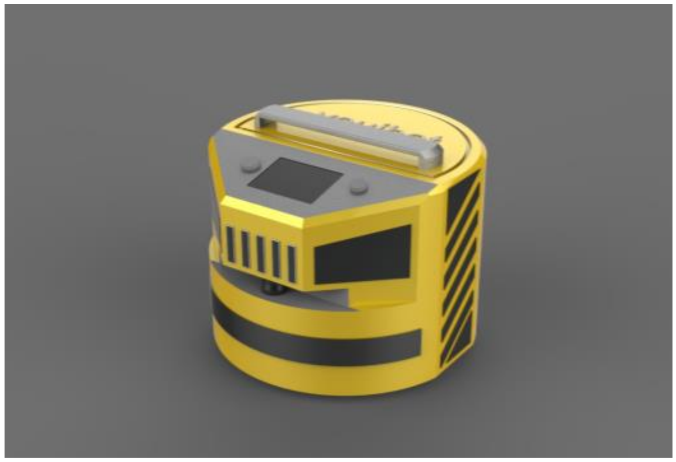


Figure 2.2: (Zhi and Xuesong, 2018) proposed mobile robot design

Mobile robot designed by (Zhi and Xuesong, 2018) as Figure 2.2. This is a differential drive mobile robot (Zhi and Xuesong, 2018). The front, back and turn movements of the robot are accomplished by controlling the rotation speed of the left and right wheels(Zhi and Xuesong, 2018). The researchers have developed and researched an open control system that includes mobile robots ' servo driver and controller (Zhi and Xuesong, 2018) . In the meantime, the researchers developed upper computer software for interaction between humans and computers, remote control, remote monitoring. SLAM is a key navigation component used to create an environment map (Zhi and Xuesong, 2018). With laser scanning, data from encoder and IMU, the researchers also used gmapping package to build the map (Zhi and Xuesong, 2018). However, that under real world conditions such optimizations are not needed in many scenarios as the approach is sufficiently accurate for robots to fulfill their mission. Autonomous navigation is the topic most concerned for mobile robot (Zhi and Xuesong, 2018). In this project, the robot's localization, navigation, and autonomous movement are all depends on the laser scanning data and the map have already created. Through ROS navigation stack, this function is finished.

Figure 2.3: SLAM mobile robot designed by (Ocando et al., 2017)

The LIDAR used in as shown in Figure 2.3 was a URG-04LX - UG01 from Hokuyo (Ocando et al., 2017). This sensor has a range of 4 meters and records data in a 240-degree semi-circumference. It is basically a 2D data plane, which with the sensor's proper rotations and translations can be converted into a 3D data volume. However, its every expensive. Therefore, in my project a 360-degree LIDAR with cheaper price will be used to lower down the cost.

In addition, in regard to this research, separate scientists used a Raspberry Pi to regulate low-cost robots (Pereira et al., 2014, Premkumar and Nigel, 2015, Horak and Zalud, 2016). Sending the robot into tough tasks and exploring unknown areas that could be hazardous for individuals to explore implies the robot itself will be in risk, making the price of such an operation a crucial factor for the robot (Hamzeh and Elnagar, 2015). Therefore, reducing the hardware and robot software costs is needed.

Mapping an indoor environment needs laser scanning, a mobile robot base, and a computer unit for wireless communication. For example, as stated in various implementations (Ghani et al., 2014), the robot will collect depth scans data from the Asus Xtion pro 3D sensor, creating a map, or making an indoor 3D model. Localization and mapping are essential in indoor settings for tele-operated mobile robots. Various method for mapping has been investigated, including the use of laser scanners, 3D vision and wireless strength, (Ghani et al., 2014, Rehman, 2013). However, a 2D LIDAR is sufficient for indoor mapping and the price of a Kinect sensor too expensive.

Figure 2.4: (Aagela et al., 2017) designed mobile robot for SLAM

As shown in Figure 4, to create a cost-effective robot platform that could create a map for a new environment by using low-cost equipment while maintaining performance in performing tasks while also reducing power consumption for the robot platform. The aim of (Aagela et al., 2017) is to integrate the Raspberry Pi 3 as an alternative to the laptop with the Turtlebot II robot. However, in this project, I used a self-design mobile robot to cut down the cost and the performance is same where as a Turtlebot II which very expensive.

Figure 2.5: Mobile Robot designed by (Ibáñez et al., 2017)

In (Ibáñez et al., 2017) proposed how to create a cost - effective platform with capabilities based on SLAM was examined shown in Figure 5 using Arduino to control the DC motor and a laptop to collect the data from the Kinect sensor. However, the laptop and the Arduino able to be replaced with a Raspberry Pi to save more cost since a Raspberry Pi able to perform as a computer and control a motor.

(Ibragimov and Afanasyev, 2017) created a human-operated UGV prototype for their experiments, which was equipped with a computer system and a set of sensors shown in Figure 6. The mobile robot recorded raw video data streams of industrial camera Basler acA2000-50gc GigE12 shown in Figure 6B mounted on UGV prototype to analyze ROS-based visual SLAM-related methods (Ibragimov and Afanasyev, 2017).

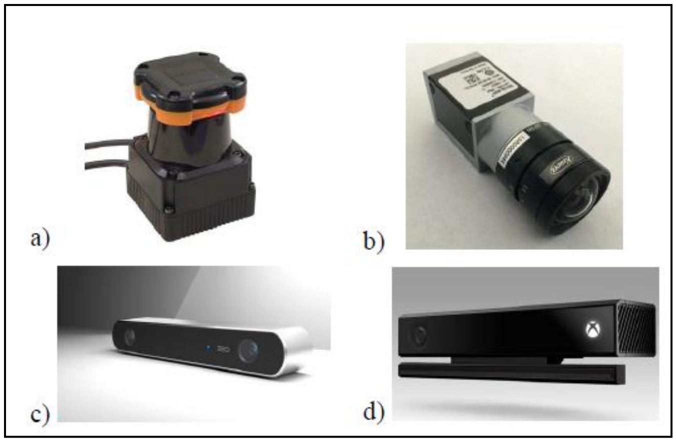


Figure 2.6B: Sensors used on UGV prototype: a) Laser rangefinder Hokuyo UTM-30LX. B) Camera Basler acA2000 - 50gc. C) ZED camera for stereolabs. D) Depth Sensor Microsoft Kinect 2.0 by (Ibragimov and Afanasyev, 2017). Courtesy of manufacturers of sensors

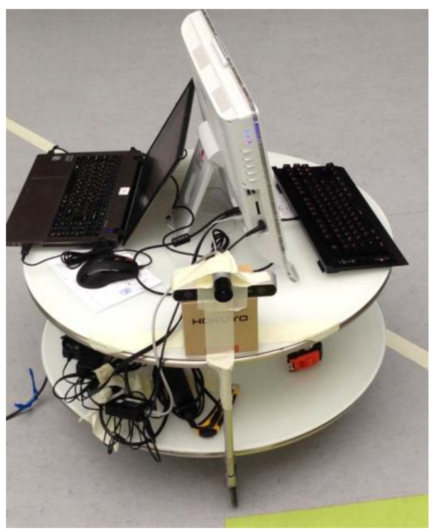


Figure 2.6: (Ibragimov and Afanasyev, 2017) designed mobile robot

The UGV shown in Figure 2.6 computing platform is based on the Intel Core i3 processor and the GeForce GT740 M graphics card that supports CUDA technology, producing parallel on-board computing (Ibragimov and Afanasyev, 2017). The sensors used is shown in Figure 2.6B (Ibragimov and Afanasyev, 2017). However, that under real world conditions such optimizations are not needed in many scenarios as the approach is sufficiently accurate for robots to fulfill their mission.

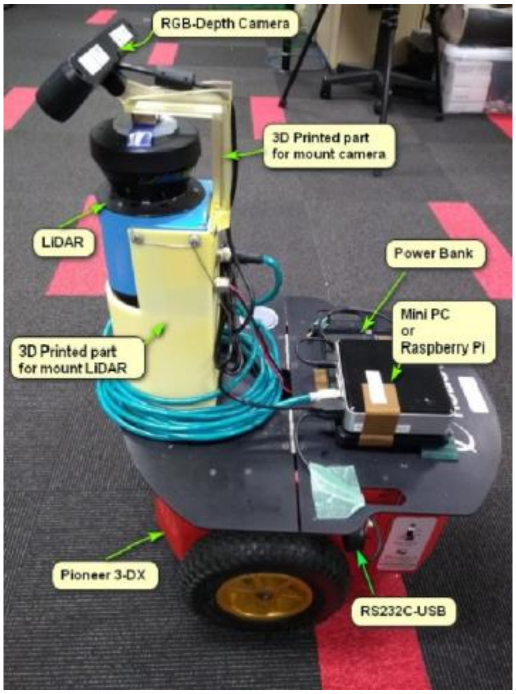


Figure 2.7: (Gatesichapakorn et al., 2019) mobile robot with equipment

According to mobile robot design shown in Figure 2.7, the mobile robot is mounted with "SICK LMS100-10000" 2D LiDAR and "ASUS Xtion PRO LIVE" RGB-D camera. Adept Mobile robot Pioneer 3-DX is the mobile robot body (Gatesichapakorn et al., 2019). However, in this project show that there is no need for such optimizations under real world conditions in many scenarios as the approach is sufficiently accurate for robots to fulfill their mission. Moreover, all the equipment used in (Gatesichapakorn et al., 2019) are too expensive especially Pioneer 3-DX.

Chapter 3.0: System Design

3.1 System Overview

Component Required

1. Mobile Robot (Slave)
2. LiDAR
3. Raspberry PI
4. Arduino
5. Adafruit Motor Shield
6. Battery pack for voltage supply
7. Laptop (Master)

Compiler / Software Program Required

1. Arduino Compiler
2. Ubuntu MATE 16.02
3. Robotic Operating System (Kinetic)

Hardware Design

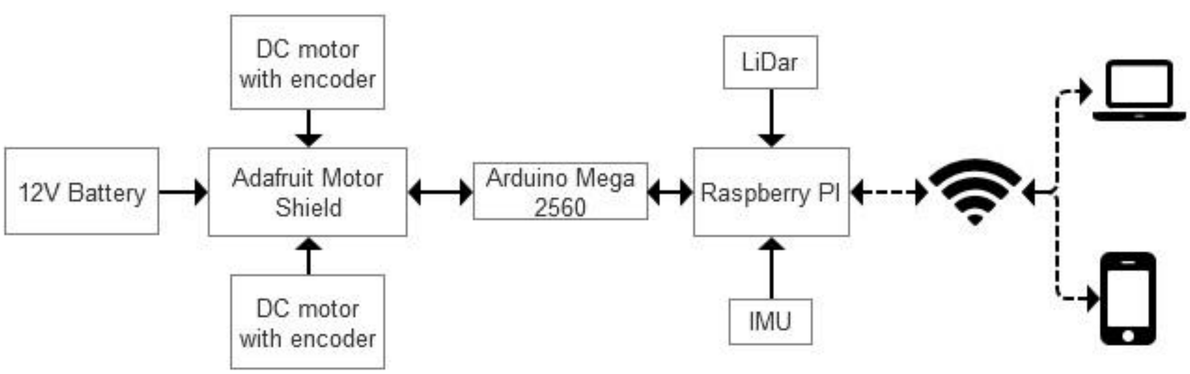


Figure 3.1.1: Hardware design for the mobile robot.

In Figure 3.2.1 shown the hardware design setup for the mobile robot and the components needed for the mobile robot to operate. The LiDAR sensor mounted on the mobile robot is

Delta-2B a Laser Range Finder. The Inertial Measurement Unit (IMU) used is BNO055 USB Stick from Bosch Sensortec. A 10000mAh power bank is used to power up both the Raspberry PI and the motor for the Delta-2B rotation. 12V AA battery pack is to power the Adafruit Motor Shield.

Generally, the Adafruit Motor Shield is the driver used to drive the mobile robot. Arduino Mega is being used to control the motor driver and perform low level functionalities. Raspberry Pi 3B is being used to communicate with PC and obtain sensor information from Lidar and IMU. Robot Operating System (ROS) is being used to control the robot movement, as well as providing simultaneous localization and mapping (SLAM) functionality. A Wi-Fi adapter is used to connect the mobile robot to the Wi-Fi network. Thus, laptop and smart phone both act as a Master node to control the mobile robot.

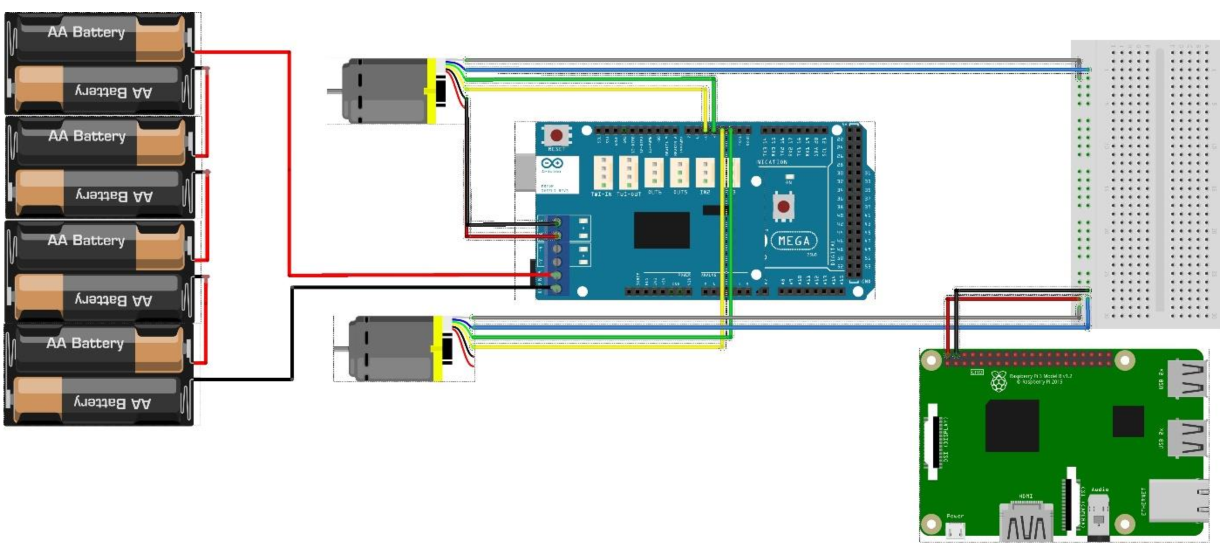


Figure 3.1.2: Visualization for hardware connection.

In Figure 3.2.2 shown the connection of the hardware using visualization tool. The motor control is control by an Arduino MEGA 2560 which correspond with an Adafruit Motor Shield. The Adafruit Motor Shield is attached on top of the Arduino MEGA as shown in Figure 3.1.2. The motor encoder V+ is connected to the Raspberry PI 5v pin. As for the motor encoder V- will connected to the Raspberry PI ground pin. Each of the motor's V+ and V- are connected specified terminal blocks on the Adafruit Motor Shield.

The LiDAR will be connected to the Raspberry PI through USB port to send the scan data from the LiDAR to the visualization tools in ROS to view. The IMU also connected to the Raspberry PI though USB port to send the Odometry data to the Raspberry PI for Odometry calculation.

For the movement of the mobile robot is depends on the serial communication of Arduino and Raspberry PI. Raspberry PI will send velocity command to the Arduino as a linear and angular velocity. The Arduino will translate the command from Raspberry PI into velocity command for each motor through Arduino Motor Shield. The Arduino drives the motors through PWM. The speed of each motor is calculated with the information of each motor encoder and sent back to the Raspberry PI.

Software Design

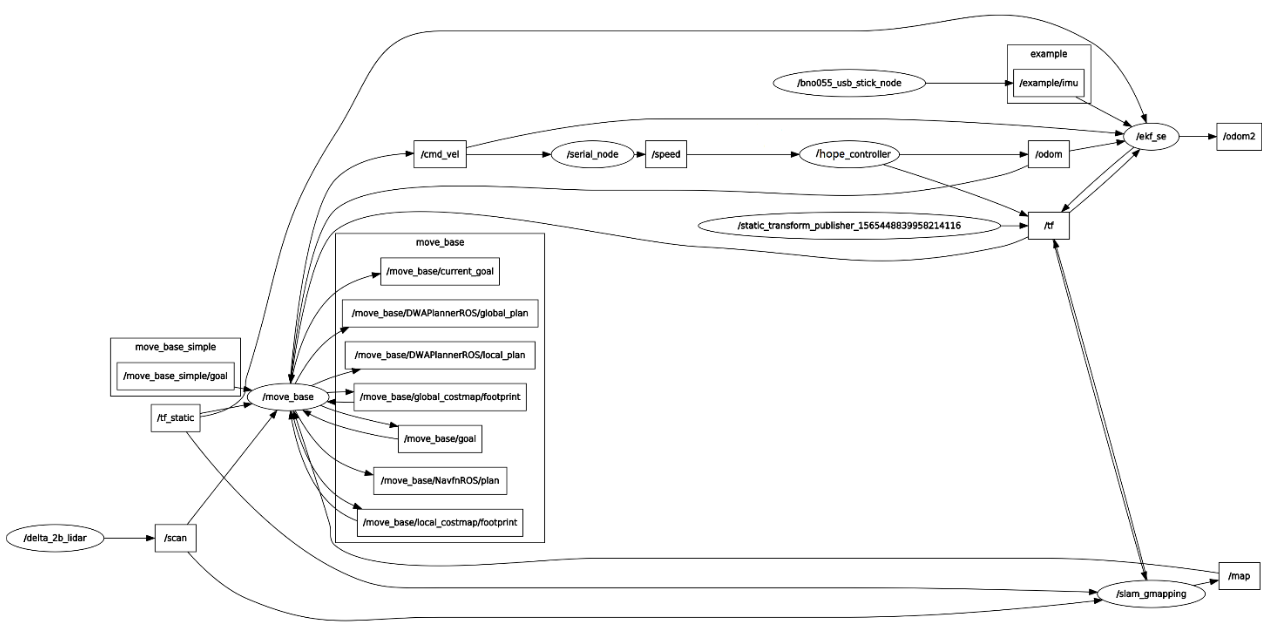


Figure 3.1.3: Actual software connection of the mobile robot

Node name	Function
/delta_2b_lidar	Driver for the delta-2B LiDAR to connect with the ROS.
/move_base	Node needed for the mobile robot to move in the Rviz.
/slam_gmapping	Node to perform simultaneous localization and mapping for the mobile robot.
/serial_node	Connection between the Arduino to ROS.
/hope_controller	To control the motor of the mobile robot.
/bno055_usb_stick_node	Driver for the IMU BNO055 usb stick.
/ekf_se	Extended Kalmann Filter node for combining the odom and the IMU data.

Table 3.1.0: A table of function for each node.

Topic name	Function
/scan	Consist of LiDAR scan data.
/tf_static	Consist of the original transform.
/tf	Consist of current transform / new transform
/map	Consist of 2D map data.
/odom	Consist of odometry data of the mobile robot.
/odom2	Consist of combination of odometry data of the mobile robot and IMU odometry data.
/example/imu	Consist of IMU odometry data.
/speed	Consist of speed of the mobile robot
/cmd_vel	• Consist of new speed for the mobile robot

	• Keep update the mobile robot on the new speed received from the ROS master.
/move_base/current_goal	Data of new goal.
/move_base/DWAPlannerROS/global_plan	Data of path planner in global map.
/move_base/ DWAPlannerROS/local_plan	Data of path planner in local map.
/move_base/global_costmap/footprint	Data of mobile robot footprint in global costmap.
/move_base/goal	Data of the current goal.
/move_base/NavfnROS/plan	Data of current path planner.
/move_base/local_costmap/footprint	Data of mobile robot footprint in local costmap.

Table 3.1.1: A table of function for each topic.

In Figure 3.1.3 shown the software design of the mobile robot. In Table 3.1.0 shown the function of each node shown in Figure 3.1.3. In Table 3.1.1 shown the function of each topic shown in Figure 3.1.3.

Operation Steps

Gmapping:

1) Specified the ROS_MASTER_URI IP address for the master (laptop).
2) Show the ROS_IP of the master in the local server.
3) Run roscore.
4) Continue step 1 – 2 in the slave (mobile robot)
5) Run hope_controller either in mobile robot command prompt. Alternative, using SSH section through MobaXterm. Once connected the LED beside the mobile robot will change from fast blinking to slow blinking.
6) Run delta_2d_lidar launcher.
7) Run gmapping package with corresponding parameters that suit the mobile robot itself.
8) Run rviz for visualization.

9) Add new settings to rviz for viewing the 2D map capture or scan by the LiDAR on top of the mobile robot itself.
10) Run teleop_twist_keyboard.
11) Use the keyboard, ‘i’ = move forward, ‘j’ = move left, ‘l’ = move right, ‘<’ = move backward and ‘k’ = stop all movement, work either in capitalize or non- capitalize form.
12) Run rosbag to record down the 2D map data.
13) Move the mobile robot around to get data or scan the surrounding to create a full 2D map of the unknown area.
14) After a clear map can be visualize in rviz.
15) Stop the rosbag command.
16) Export the .bag file into .yaml and jpg file for future reference of the 2D map.

Simultaneous Localization and Mapping:

1) Specified the ROS_MASTER_URI IP address for the master (laptop).
2) Show the ROS_IP of the master in the local server.
3) Run roscore.
4) Continue step 1 – 2 in the slave (mobile robot)
5) Run hope_controller either in mobile robot command prompt. Alternative, using SSH section through MobaXterm. Once connected the LED beside the mobile robot will change from fast blinking to slow blinking.
6) Run delta_2d_lidar launcher.
7) Run gmapping package with corresponding settings that suit the mobile robot itself.
8) Roslaunch the move_base launcher. Which from the self-created package hope_nav which contain all the mobile robot suitable parameter for example, the update_frequency, publish_frequency, transform_tolerence, inflation_radius, footprint_padding, etc. All the mobile robot parameters are coded in the hope_nav package.
9) Rviz visualization will run after move_base successful launched.
10) Setup the rviz setting for better visualization.

11) Use the current_goal to pinpoint a new goal/location for the mobile robot to relocate.

Simultaneous Localization and Mapping (With IMU implementation):

1) Specified the ROS_MASTER_URI IP address for the master (laptop).
2) Show the ROS_IP of the master in the local server.
3) Run roscore.
4) Continue step 1 – 2 in the slave (mobile robot)
5) Run hope_controller either in mobile robot command prompt. Alternative, using SSH section through MobaXterm. Once connected the LED beside the mobile robot will change from fast blinking to slow blinking.
6) Run delta_2d_lidar launcher.
7) Run the bno055_usb_stick.
8) Run the ekf launcher.
9) Run gmapping package with corresponding settings that suit the mobile robot itself.
10) Roslaunch the move_base launcher. Which from the self-created package hope_nav which contain all the mobile robot suitable parameter for example, the update_frequency, publish_frequency, transform_tolerence, inflation_radius, footprint_padding, etc. All the mobile robot parameters are coded in the hope_nav package.
11) Rviz visualization will run after move_base successful launched.
12) Setup the rviz setting for better visualization.
13) Use the current_goal to pinpoint a new goal/location for the mobile robot to relocate.

Chapter 4.0: Specification
4.1 Analysis Specifications
This project was planned to offer an affordable operational indoor mapping system that can be used across all situations, especially in environments that could be considered a hazard to humans.

A. Robotic Operating System (ROS)

It is an open-source platform that provides the robot functions with a number of packages and software (Quigley et al., 2009). It also controls the switching between the robots and the controller, where each device becomes a node in the ROS environment. A master node is responsible for the interconnection protocol between the nodes, using the sensors and the actuator as topics that would be available for subscription or publishing all the communications as shown in Figure 3.1. The messages contain different data types such as video and odometry data transferred between nodes (Robotics, 2017).

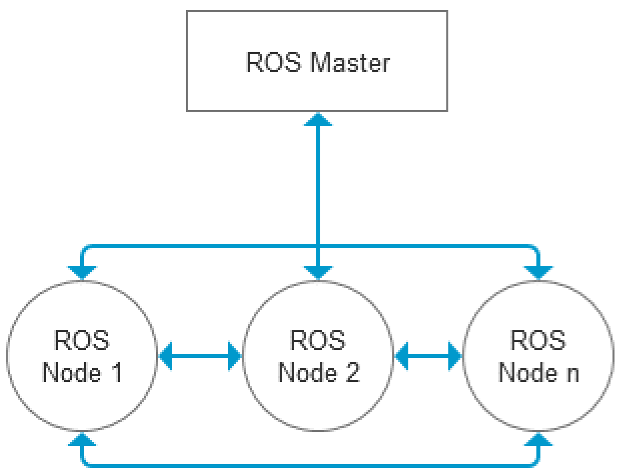


Figure 3.1.1: ROS system

ROS supports mainly Linux operating system such as Ubuntu and Debian. There large variety of selection for hardware, sensors and actuators that are supported by ROS. ROS defines the architecture of the hardware as well as a low level of control to manage the connected devices, be it a robot or an individual sensor connected to a computer. ROS has flexibility with existing algorithms that can be used to improve the performance of complex tasks by robots. SLAM is an example used to map and navigate the robot (Ghani et al., 2014). To run the ROS packages, ROS master should be active or at least one ROS master is running in the system as shown in Figure 3.1. The simple ROS connection system enables sending and receiving messages from other nodes to the ROS node. Once ROS nodes established connection with the master, message could be sent directly between the nodes (Robotics, 2017).

B. Mobile Robot

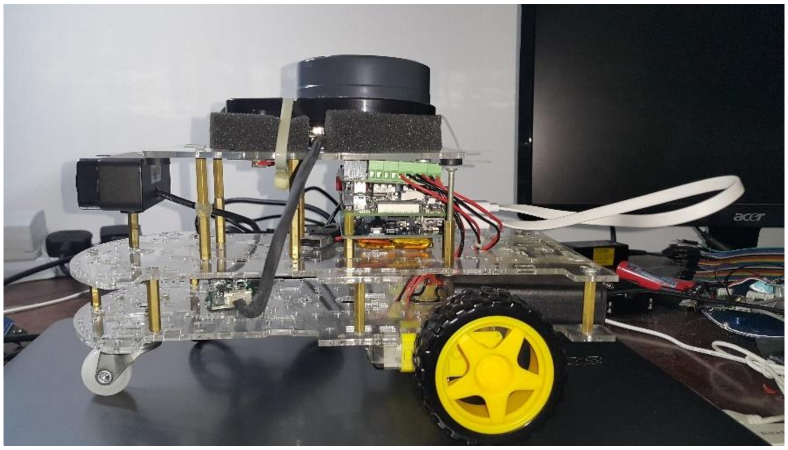


Figure 3.1.2: Mobile robot used in project I.

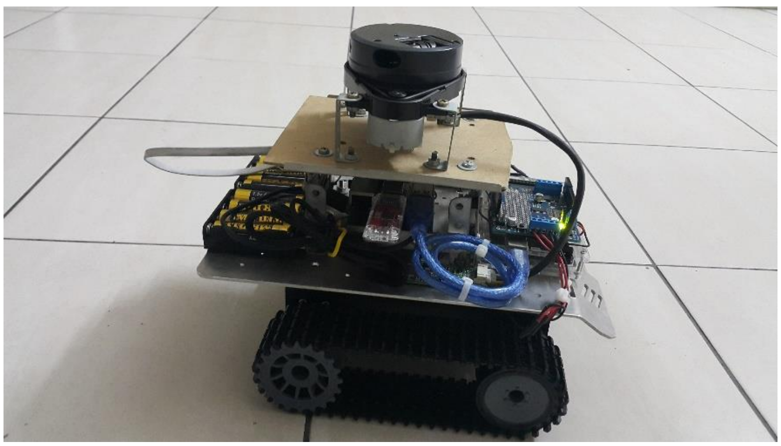


Figure 3.1.3: Mobile robot use in this project II.

In Figure 3.1.2 shown the older version of self-made mobile robot used in Project I which is motorized wheels base. In this project II an improved version of self-made mobile robot shown in Figure 3.1.3 is used and running on a continuous track system without using any wheels. The Odometry can be obtained by using a sensor called the Delta 2B sensor mounted on the robot to produce 2D image of the map. This mobile robot includes several features such as low-cost DC motors, large batteries, individual power pack for powering the Raspberry Pi and a USB camera.

C. Raspberry Pi

A single board computer which has most of the computer basic components built-in a small size. The Raspberry Pi uses a 1.2 GHz ARM processor for the Raspberry Pi 3 used in this project. The Raspberry Pi 3 runs on Ubuntu 16.04 Mate Operating System installed on a Micro SD card of 16 GB. In addition, ROS with its basic packages has been installed. The Raspberry Pi 3 connects via USB cables to the Delta-2B Laser Range Finder Sensor and connect to the motorized base through Raspberry Pi Robot Board v3.0, with the Delta-2B sensor is mounted on the robot's top as shown in Figure 3.2 to get clear view of the surrounding without any components blocking the sensor view range.

The Raspberry Pi collects data from the sensor namely:

1. 2D image scan from the Delta-2B LRF sensor.
2. Receive video stream from the USB camera.
3. Receive information of the robot odometry.

Below provides an example description of the application or the handlers running on it for a Raspberry Pi 3 that has ROS installed on it:

1. Delta_2B is the driver that is used in the connection between Raspberry Pi and Delta 2B LRF sensor for sending data collect from the Delta 2B to Raspberry Pi.
2. Motion_Topic is the driver for Raspberry Pi to control the DC motors by sending command to move forward, move backward, move left and move right.

3. The hector_mapping application is referred to as the ROS application tasked with processing all the information within Odometry or laser scanning data to create a grid map on the map sequence.
4. The rviz application allow user to view the map.

D. Master Node (PC)

This personal computer also runs on the Ubuntu MATE 16.04 operating system and ROS kinetic desktop installed on it, which comes with variety of software that enables maps to be viewed, as well as being able to view video streams using the following components:

1. The main application runs on the master node of the ROS. This ROS core node communicates with the ROS master node so that all nodes available in the ROS environment can be accessed. This makes it the main database where all nodes and the path to access them are listed.
2. The node applied to connecting the ROS system and sending control commands to the robot to be able to control its movements is known as the tele-operating application. A keyboard connected to the corresponding PC is used to guide the mobile robot operation.
3. The RVIZ is called the 3D visualizer used to display both the sensor data and the ROS state information. Basically, RIVZ is a visualizer. Used to view the model of the robot, the stream of video, and create map. Additionally, as shown in Figure 3.3, the software can deal directly with robot sensor topics such as measurements of infrared distance, camera data, laser data, sonar, and etc.

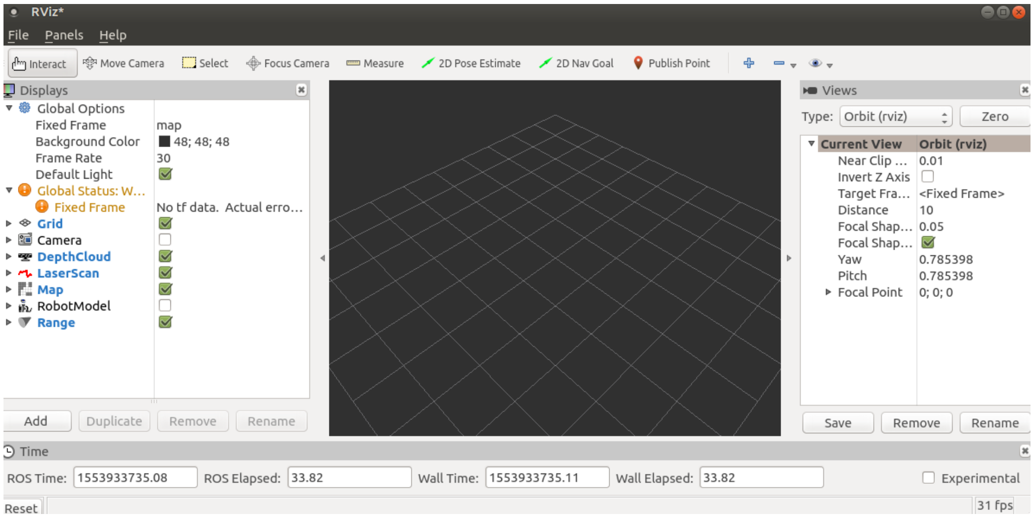


Figure 3.1.4: Rviz visualizer

E. Network

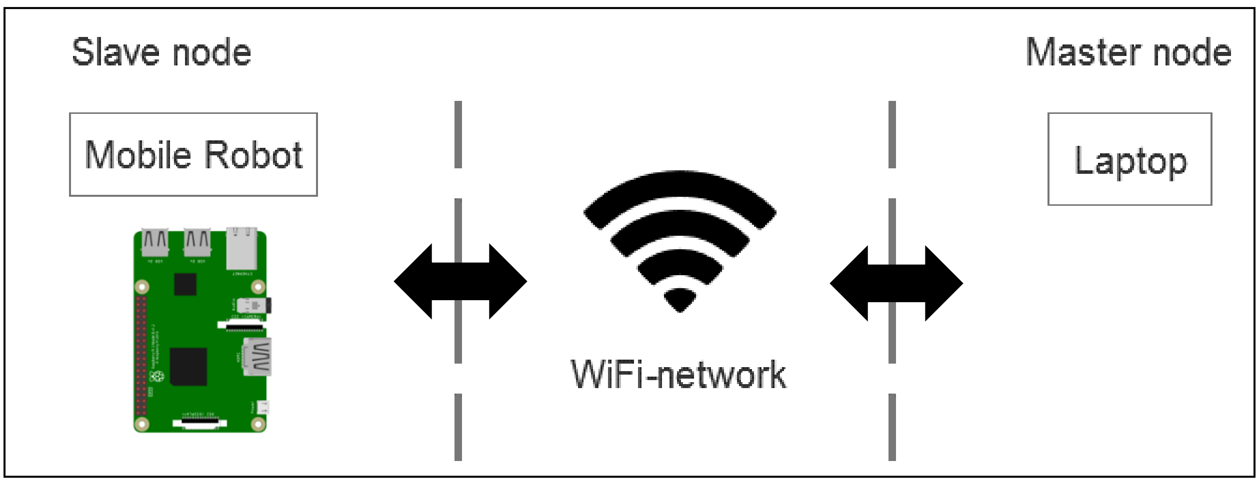


Figure 3.1.5: Network architecture.

The network architecture used in the experiment is shown in Figure 3.1.4. A local wireless network is used in the project for the mobile robot to connect with the master node.

To configure an ROS environment over the network, this project setup IP addresses for the master node PC and Raspberry Pi as well as the hostnames which to be quite important for ROS communication systems if didn't setup hostname also able to proceed using IP address.

It is necessary to connect to a wireless network to make the Raspberry Pi mobile, which uses a router that connects it to the main master node. This connection is maintained intact as long as the Raspberry Pi can be reached by the router.

Another important factor would be latency when it comes to a network. Since the mobile robot's operation is performed on a remote base, if the mobile robot fails to receive the right commands and on time could result in the mobile robot being faced with obstacles as control commands reaching the Raspberry Pi may be possible late. Moreover, the simulation time update will be fatal, then the map will unable to create perfectly. Network must be constantly stable when carryout mapping process. It is typically found that latencies of up to 400ms are acceptable.

4.2 Design Specifications

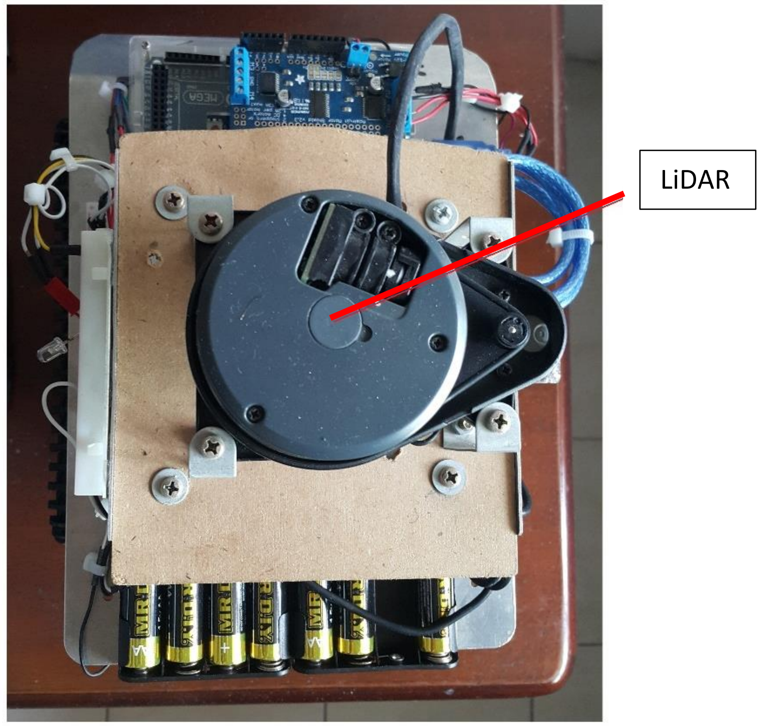


Figure 4.2.1: Top view of the mobile robot design

In Figure 4.2.1 shown the top view of the mobile robot design. The LiDAR must be installed on top of the mobile robot. The view of the LiDAR must be in clear view which to prevent any obstacles obstructed it line of sight and able to obtain more accurate scan data for building the 2D map.

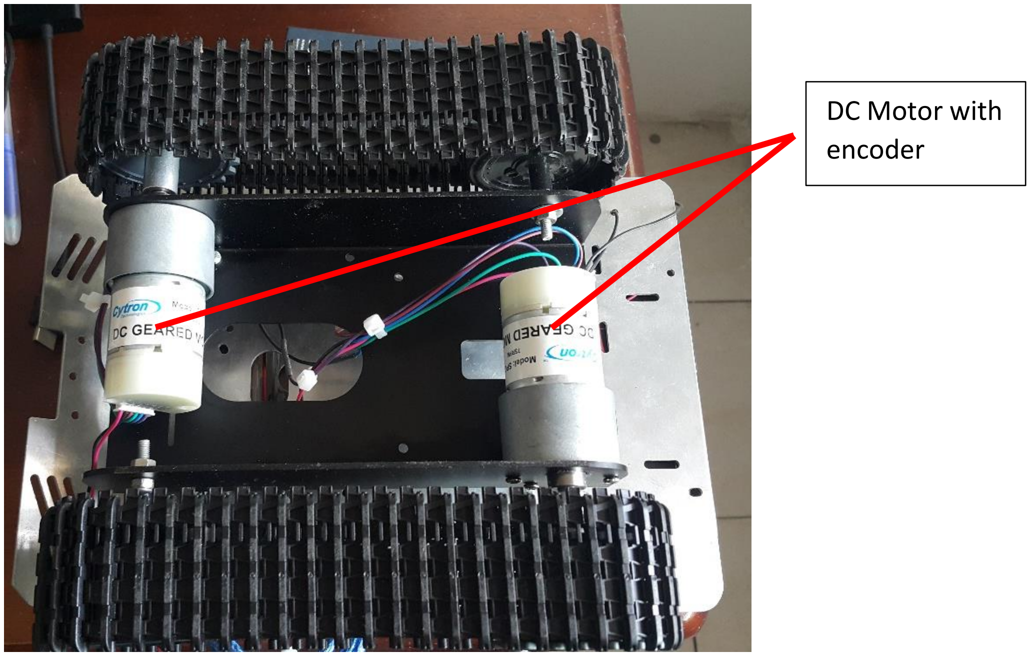


Figure 4.2.2: Bottom view of the mobile robot design.

In Figure 4.2.2 shown the bottom view of the mobile robot design. Continous track system were proposed in this project to overcome moving on slippery surface which was one of the problem in past project. The DC motors were installed at the bottom of the mobile robot which allow the continuous tracks to move and control the direction of the mobile robot. The DC motors used were attached with encoder to get better odometry data from the motors. The encoder will be connected to the Adafruit Motor Shield.

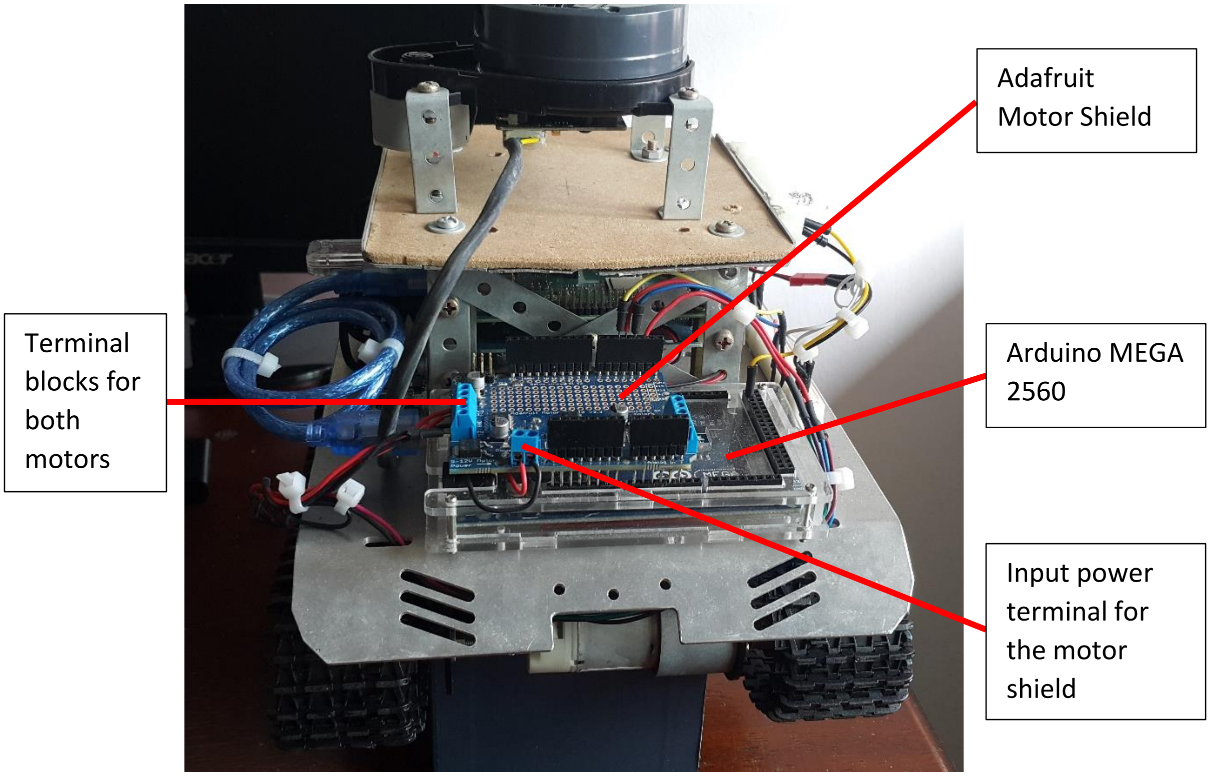


Figure 4.2.3: Front view of the mobile robot design.

In Figure 4.2.3 shown the front view of the mobile robot design. The Arduino MEGA was placed on the front of the mobile robot for easier wire connection from both DC motor encoder. The Adafruit Motor Shield was atteached on top of the Arduino MEGA through direct connection. The Arduino MEGA connected with the Raspberry PI in serial connection.

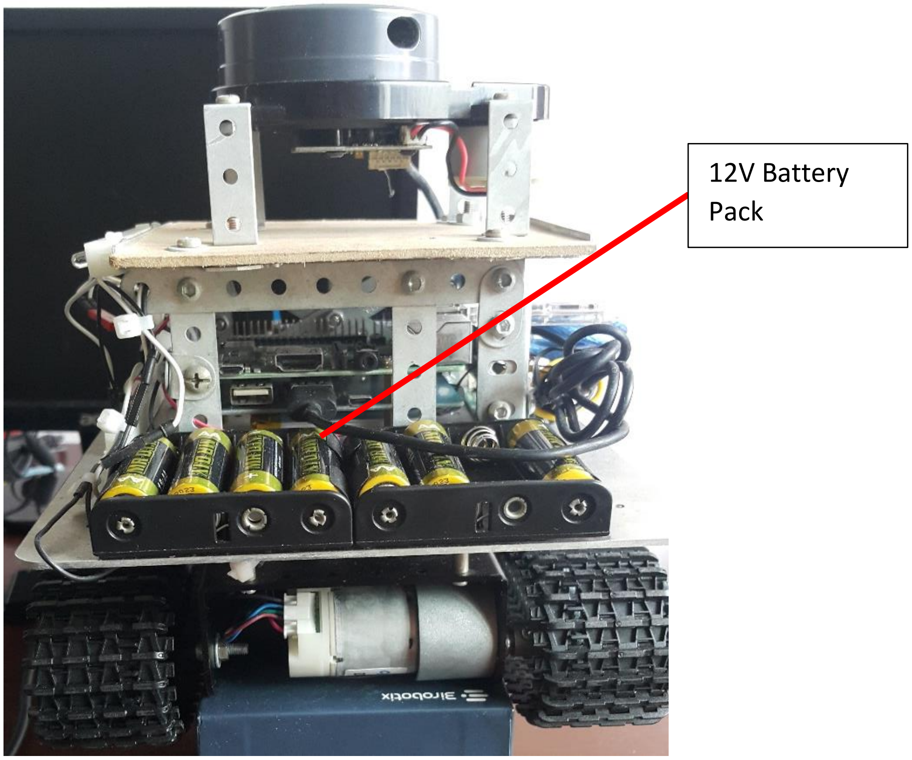


Figure 4.2.4: Rear view of the mobile robot design

Figure 4.2.4 shown the rear view of the mobile robot design. The 12V battery pack was stored at the rear of the mobile robot to increase the rear weight of the mobile robot. Therefore able to balance the front weight of the mobile robot and the mobile robot's continuous track system able to in contact with the ground firmly. Moreover, the 12V battery pack is connectd to the Adafruit motor shield to power up the board and to perform equipvalent distribution of power toward both DC motors.

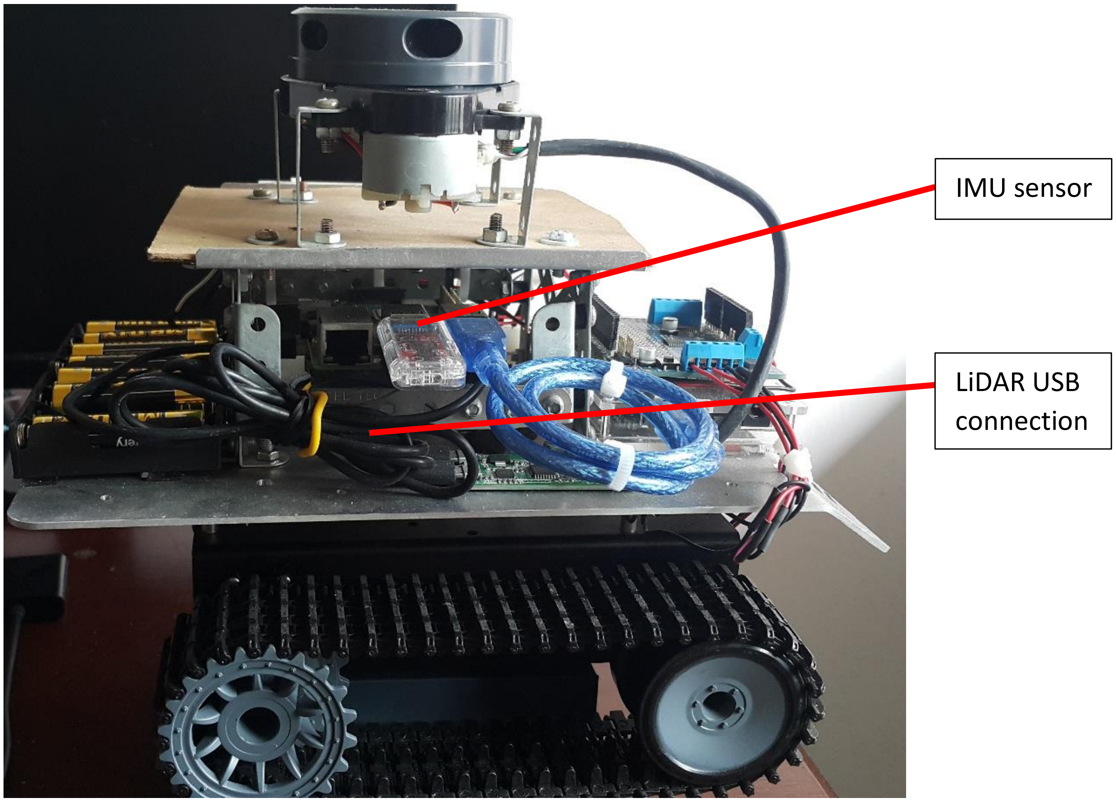


Figure 4.2.5: Side view (right) of the mobile robot design.

In Figure 4.2.5 shown the right side of the mobile robot. Right side of the mobile robot will be the USB ports section. Which all the sensors used in this project for example IMU sensor and the LiDAR sensor were connected. Thus, the IMU sensor connected to the Raspberry PI through serial connection. Moreover, the LiDAR sensor connected to the Raspberry PI through USB connection. Lastly, the Arduino MEGA connected to the Raspberry PI though serial connection.

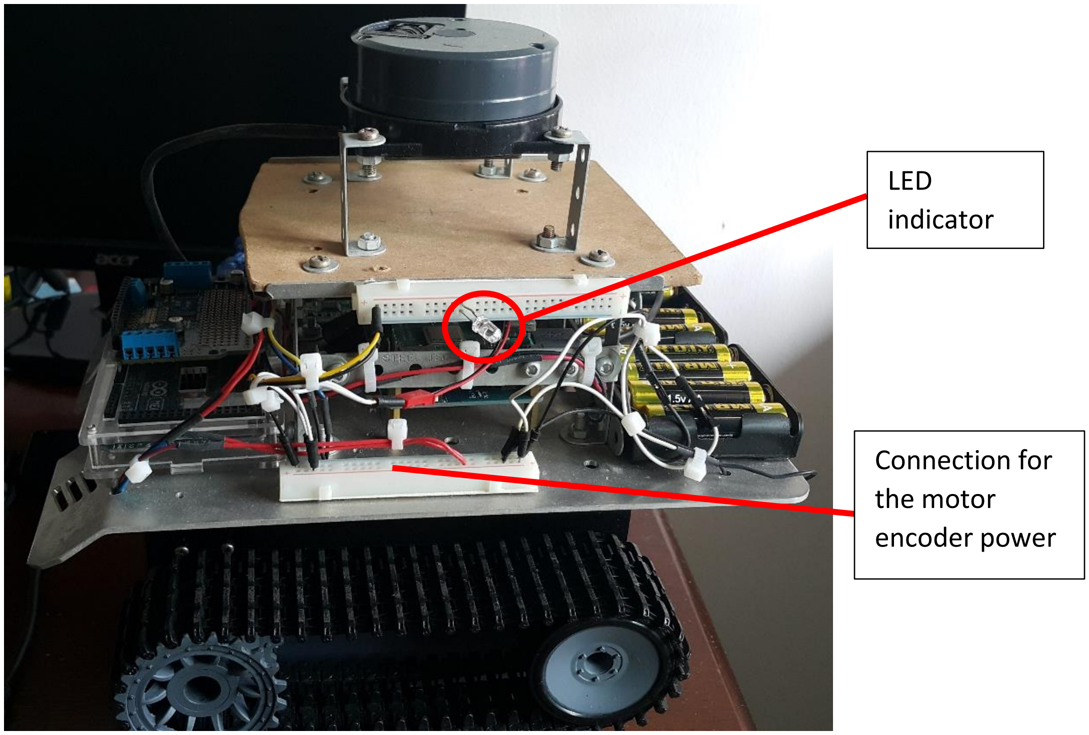


Figure 4.2.6: Side view (left) of the mobile robot design.

In Figure 4.2.6 shown the left side of the mobile robot. On the left side of the mobile robot had the LED indicator for connection purpose. If the LED blink rapidly or fast mean that the connection of rosserial is not successful. If the LED blink slowly mean that the connection of rosserial is successful. The power supply for both motors encoder were also located on the left side of the mobile robot.

4.3 Verification Plan

The mobile robot is setup as shown in Figure 4.2.1, Figure 4.2.2, Figure 4.2.3, Figure 4.2.4, Figure 4.2.5, and Figure 4.2.6 according to the proposed method.

The main result of this study is to develop a low-cost mobile robot which able to do autonomous navigation in an unknown area with more accurate odometry data. The verification plan take place in two closed unknown area. As shown in Figure 4.3.1 Westlake House 1st floor and Figure 4.3.2 UTAR FICT FYP lab.

Figure 4.3.1: Westlake House 1st floor

Figure 4.3.2: UTAR FICT FYP lab

This project carries out by running the ROS system on four type of datasets containing LiDAR data compatible with ROS. Each dataset consists of with or without IMU implementation onto the mobile robot. The four datasets are mobile robot moves in vertical line, mobile robot moves in horizontal line, mobile robot moves in diagonal line with obstacles and mobile robot moves in diagonal line without obstacles. These four datasets will be carrying out at the location shown in Figure 4.3.1. One of the datasets, mobile robot moves in diagonal line without obstacles will be carrying out at the location shown in Figure 4.3.2. Moreover, the experiment will run total 30 times from starting point to destination for each dataset to get more accurate data of the distance different from actual destination compare to visualization destination. Thus, the average result from each dataset will be used to calculate the percentage of improvement after IMU implementation.

Chapter 5.0: Testing

5.1: Test Plans

A. Mobile robot moves in vertical line.

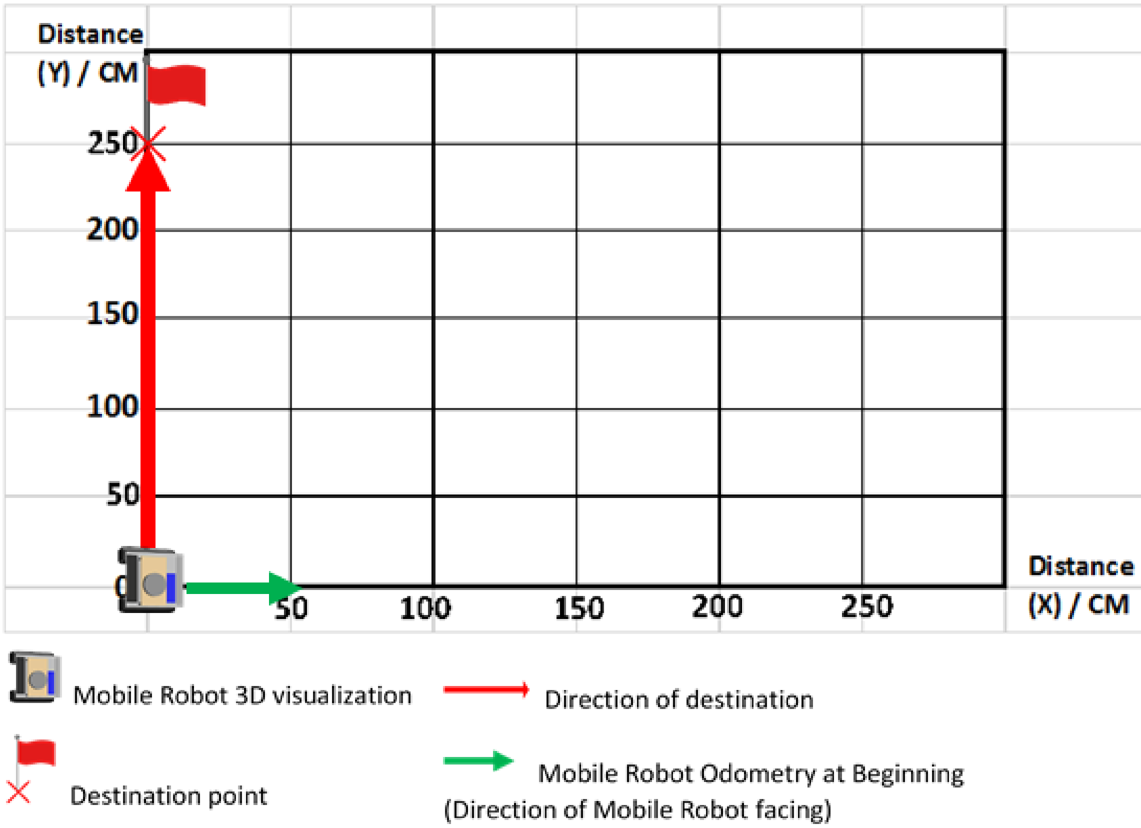


Figure 5.1.1: Visualization for mobile robot moves in vertical line path plan

In Figure 5.1.1 shown the visualization for the mobile robot moves in vertical line path plan. The destination point was set at 250cm away from the mobile robot vertically. The mobile starting position will be facing horizontally.

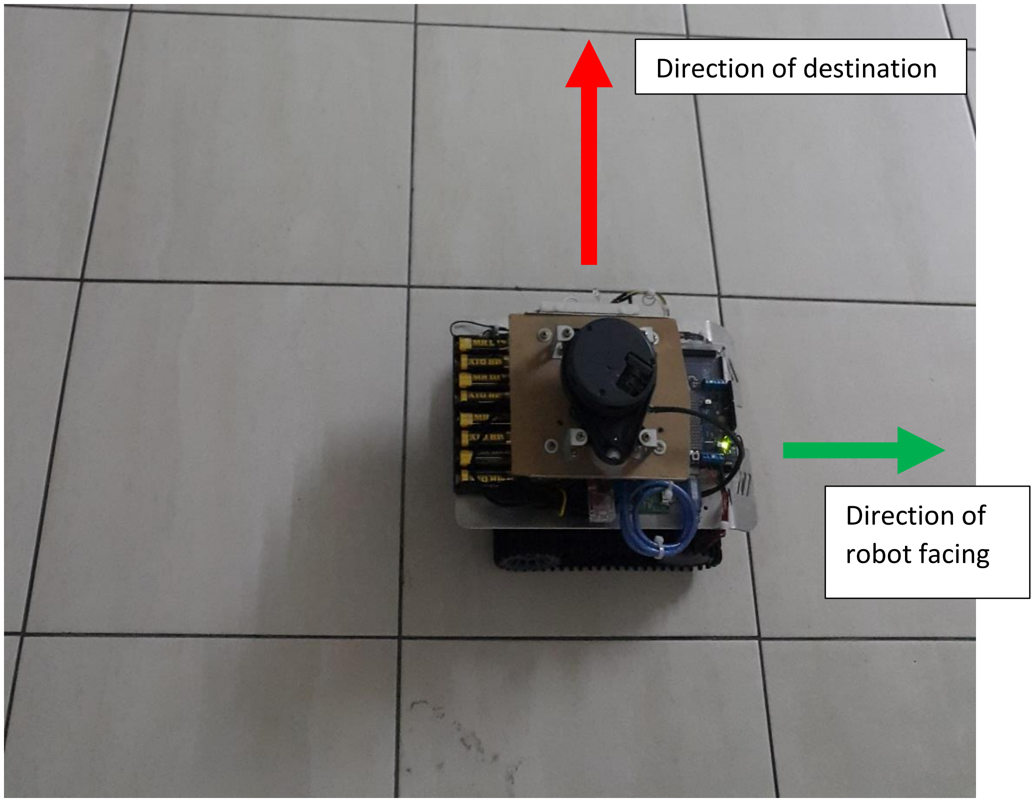


Figure 5.1.2: Actual setup for mobile robot moves in vertical line path plan

The experiment carried out as shown in Figure 5.1.2 in actual environment according as shown in Figure 5.1.1. The experiment carried out 30 times as mentioned in verification plan in order to get more accurate result for the different in distance from actual destination and virtualization destination.

B. Mobile robot moves in horizontal line.

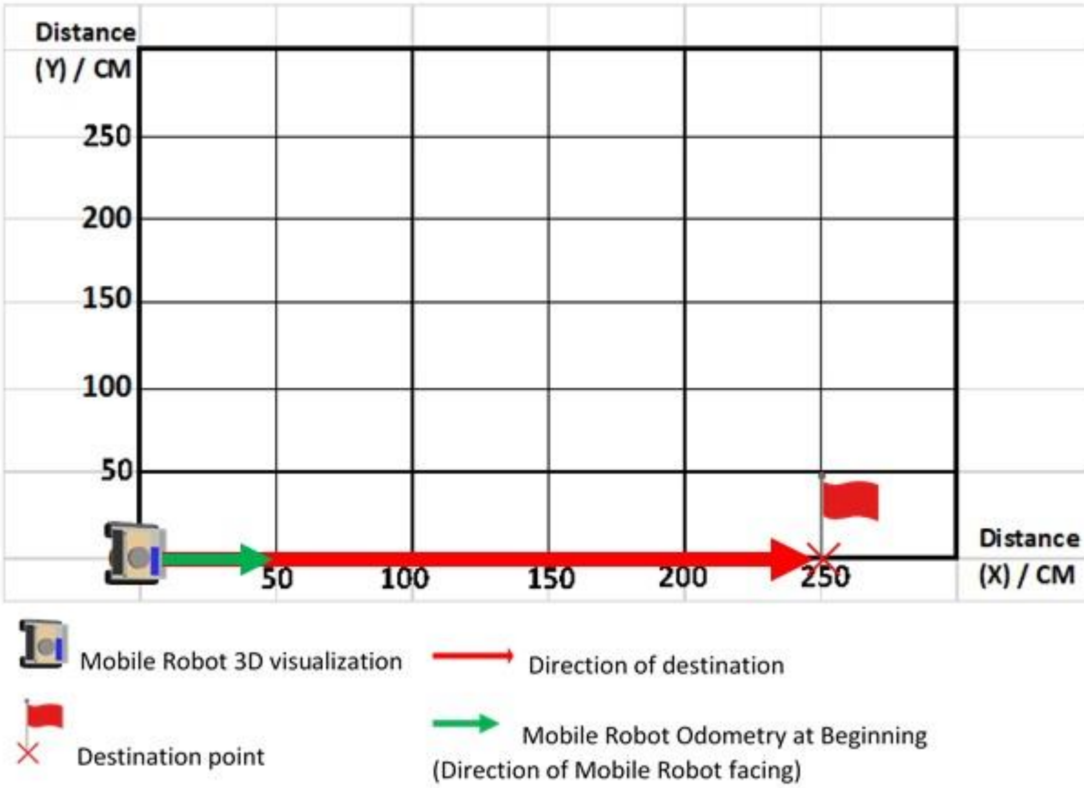


Figure 5.1.3: Visualization for mobile robot moves in horizontal line path plan

In Figure 5.1.3 shown the visualization for the mobile robot moves in horizontal line path plan. The destination point was set at 250cm away from the mobile robot horizontally. The mobile starting position will also be facing horizontally.

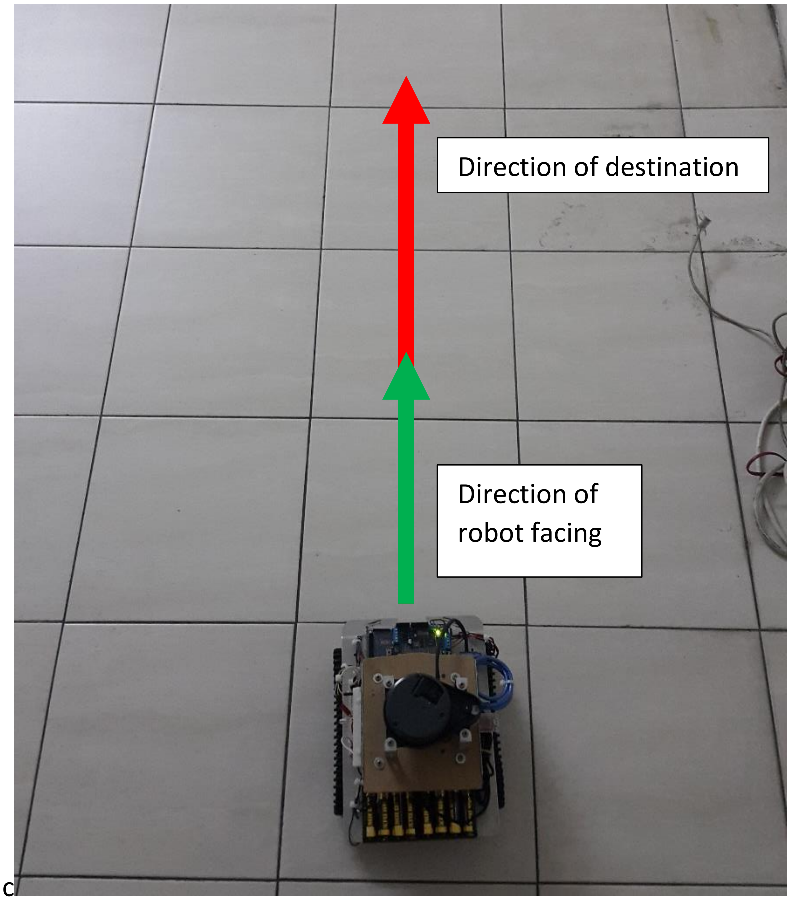


Figure 5.1.4: Actual setup for mobile robot moves in horizontal line path plan

The experiment carried out as shown in Figure 5.1.4 in actual environment according as shown in Figure 5.1.3. The experiment carried out 30 times as mentioned in verification plan in order to get more accurate result for the different in distance from actual destination and virtualization destination.

C. Mobile robot moves in diagonal line without obstacles.

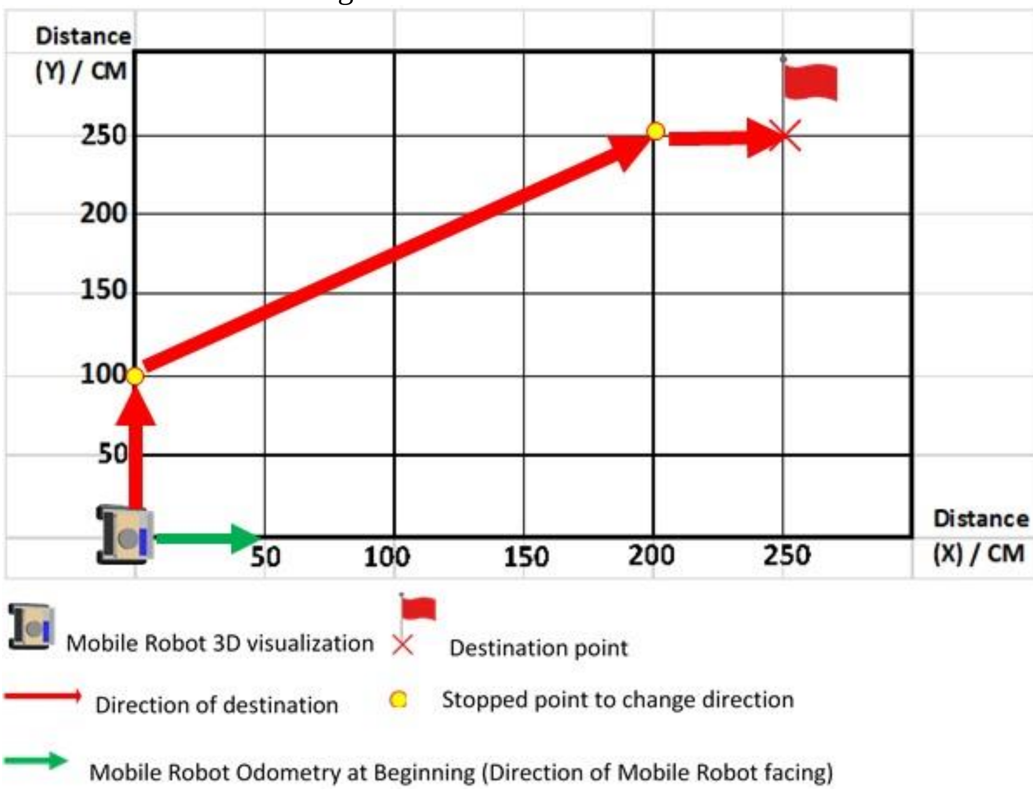


Figure 5.1.4: Visualization for mobile robot moves in diagonal line without obstacles path plan.

In Figure 5.1.4 shown the visualization for the mobile robot moves in diagonal line without obstacles path plan. The destination point was located 353 cm away from the mobile robot diagonally. The mobile starting position will also be facing horizontally.

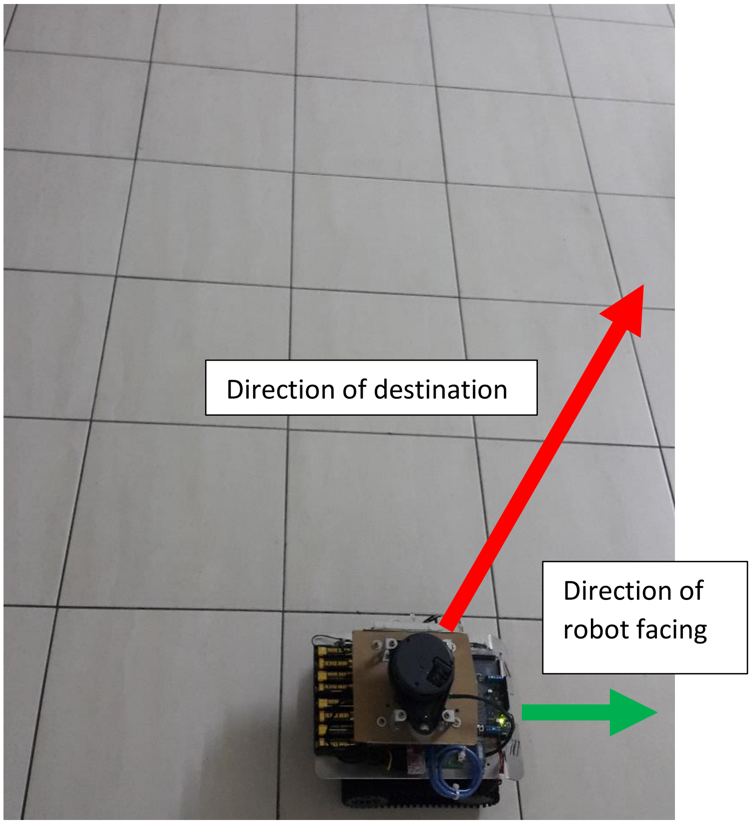


Figure 5.1.5: Actual setup for mobile robot moves in diagonal line without obstacles path plan

The experiment carried out as shown in Figure 5.1.5 in actual environment according as shown in Figure 5.1.4. The experiment carried out 30 times as mentioned in verification plan in order to get more accurate result for the different in distance from actual destination and virtualization destination.

D. Mobile robot moves in diagonal line with obstacles

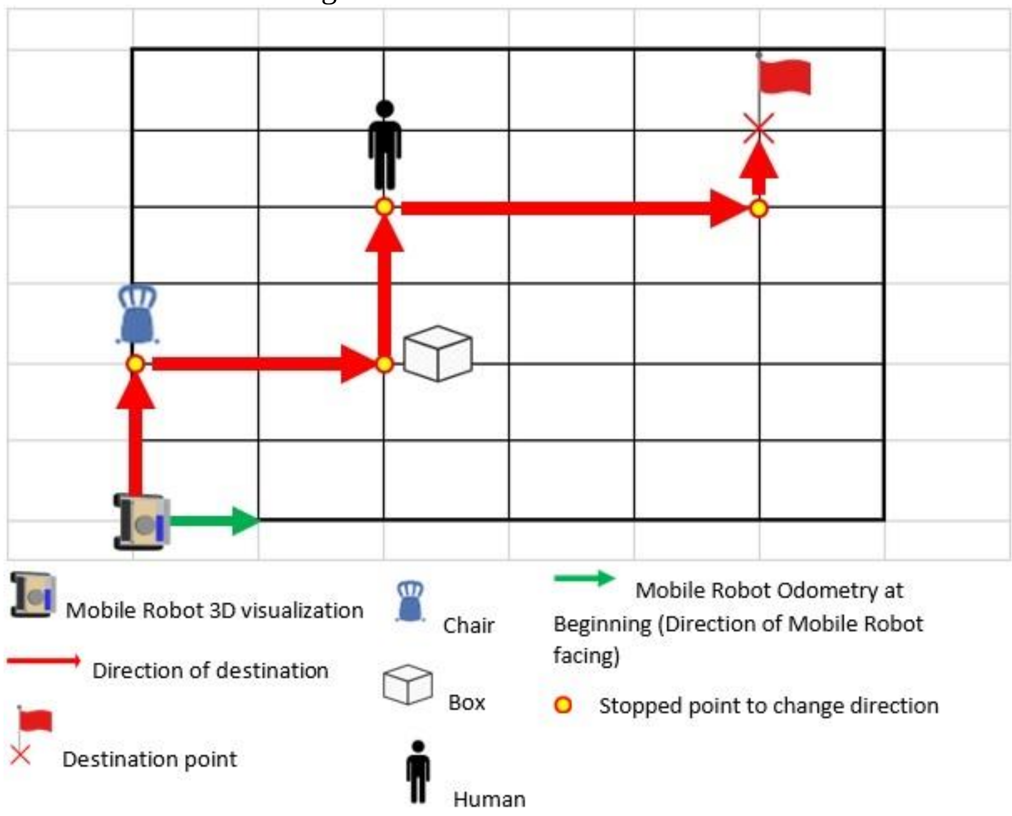


Figure 5.1.6: Visualization for mobile robot moves in diagonal line with obstacles path plan.

In Figure 5.1.6 shown the visualization for the mobile robot moves in diagonal line with obstacles path plan. The obstacles will be box, chair, and human. Through this experiment able to acquire data which closer to actual environment. The destination point was located diagonally from the mobile robot origin position. The mobile starting position will also be facing horizontally.

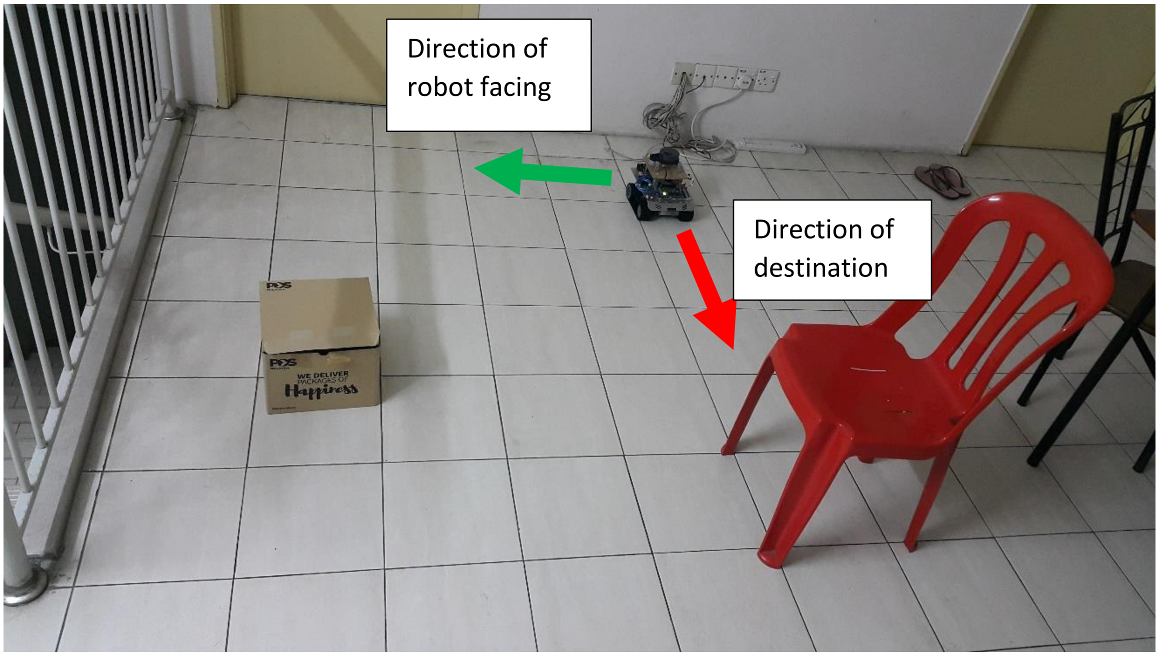


Figure 5.1.7: Actual setup for mobile robot moves in diagonal line with obstacles path plan.

The experiment carried out as shown in Figure 5.1.7 in actual environment according as shown in Figure 5.1.6. The experiment carried out 30 times as mentioned in verification plan in order to get more accurate result for the different in distance from actual destination and virtualization destination.

E. Mobile robot moves vertically with obstacle in FYP lab.

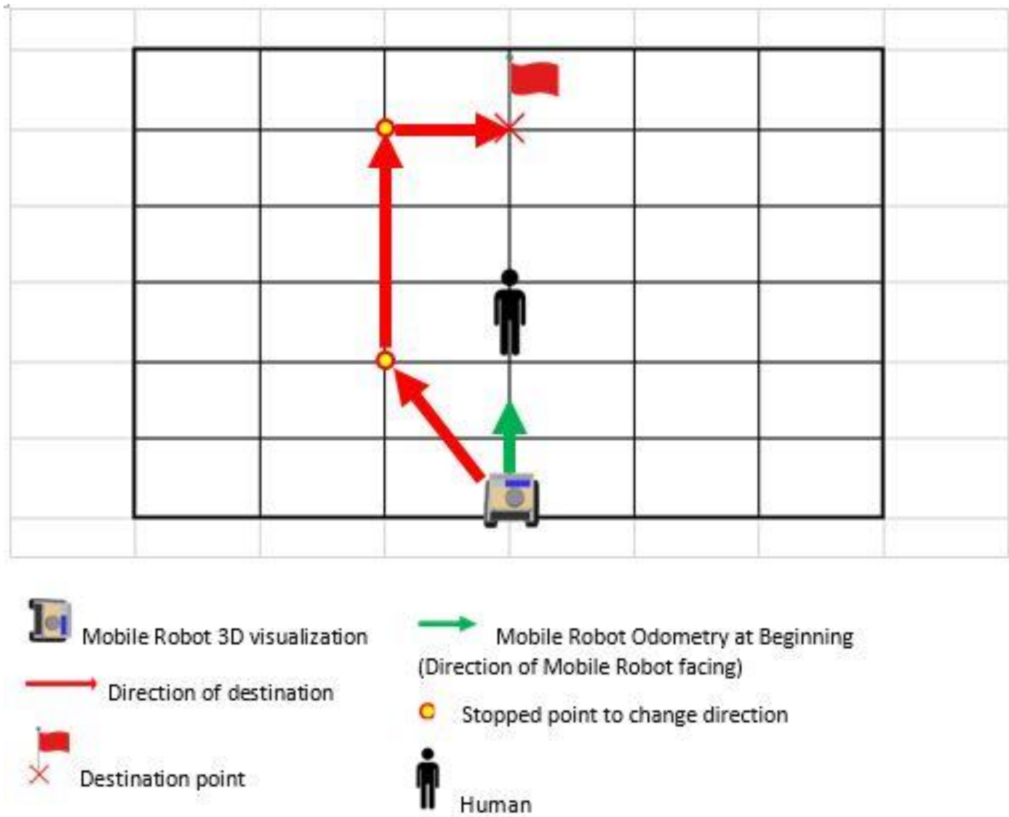


Figure 5.1.8: Visualization for mobile robot moves vertically with obstacle in FYP lab path plan.

In Figure 5.1.8 shown the visualization for mobile robot moves vertically with obstacles in FYP lab path plan. The obstacle is a human. Through this experiment also able to acquire data which closer to actual environment but not as close as in (D) because in (D) will have more obstacles which in actual environment there will have more obstacles. The destination point was located vertically from the mobile robot origin position. The mobile starting position will also be facing vertically.

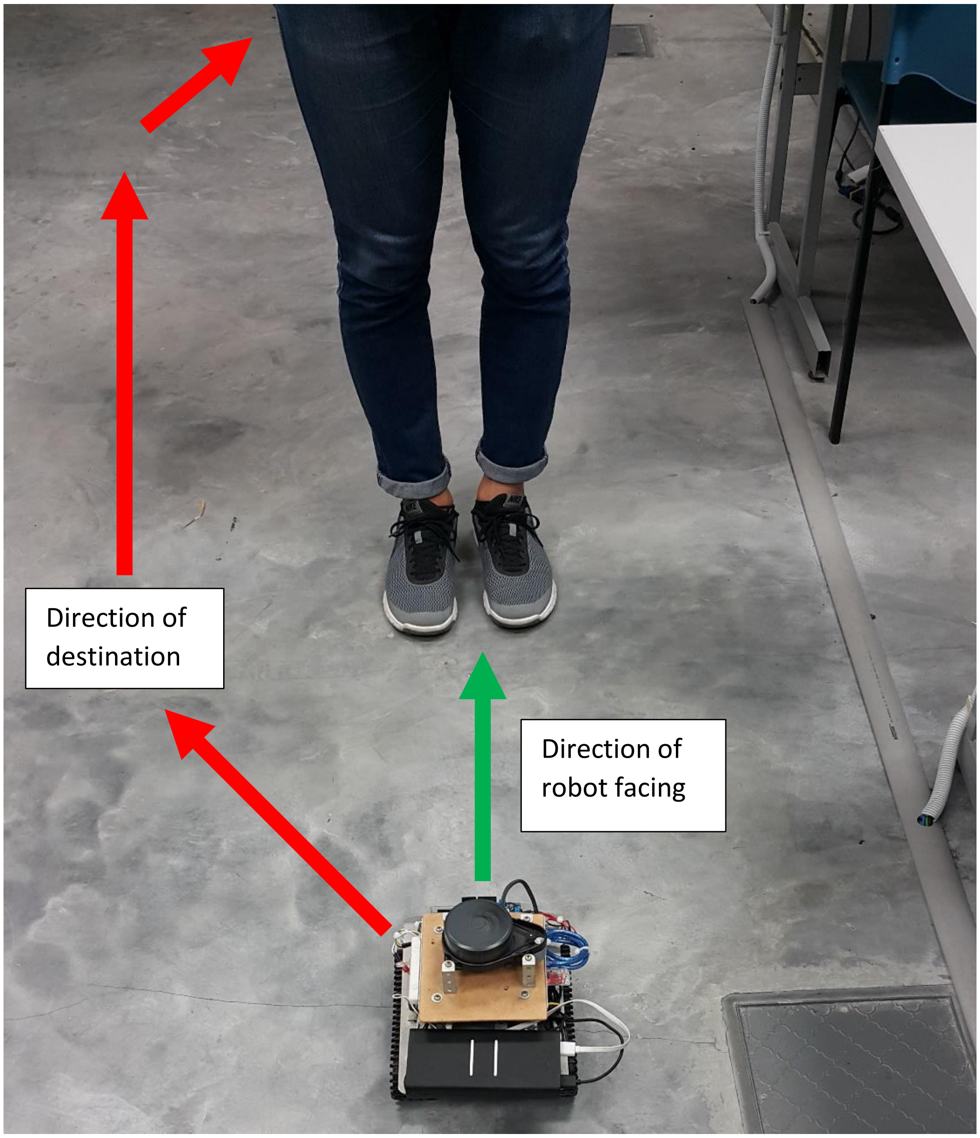


Figure 5.1.9: Actual setup for mobile robot moves vertically with obstacle in FYP lab.

The experiment carried out as shown in Figure 5.1.9 in actual environment according as shown in Figure 5.1.8. The experiment carried out 30 times as mentioned in verification plan in order to get more accurate result for the different in distance from actual destination and virtualization destination.

5.2 Test Result

A. Mobile robot moves in vertical line.

NO.	Distance different from destination (Without IMU) / m	Distance different from destination (With IMU) / m
1	0.18	0.06
2	0.15	0.06
3	0.11	0.07
4	0.10	0.04
5	0.23	0.06
6	0.22	0.11
7	0.19	0.09
8	0.27	0.08
9	0.18	0.08
10	0.11	0.09
11	0.13	0.04
12	0.18	0.07
13	0.22	0.04
14	0.20	0.03
15	0.16	0.01
16	0.15	0.05
17	0.18	0.04
18	0.17	0.03
19	0.15	0.05
20	0.13	0.08
21	0.26	0.10
22	0.21	0.09
23	0.18	0.04
24	0.23	0.04
25	0.11	0.03
26	0.10	0.04
27	0.14	0.05
28	0.23	0.06
29	0.12	0.04
30	0.22	0.05
Average	0.17	0.06

Table 5.2.1: Testing result for mobile robot moves in vertical line.

In Table 5.2.1 shown the testing result for the vertical path plan of the mobile robot.

B. Mobile robot moves in horizontal line.

NO.	Distance different from destination (Without IMU) / m	Distance different from destination (With IMU) / m
1	0.16	0.04
2	0.20	0.08
3	0.13	0.12
4	0.20	0.13
5	0.21	0.12
6	0.22	0.02
7	0.24	0.07
8	0.21	0.02
9	0.24	0.08
10	0.24	0.05
11	0.28	0.09
12	0.16	0.10
13	0.29	0.09
14	0.14	0.12
15	0.10	0.08
16	0.12	0.06
17	0.11	0.06
18	0.14	0.07
19	0.16	0.05
20	0.10	0.06
21	0.16	0.07
22	0.14	0.04
23	0.22	0.05
24	0.18	0.06
25	0.21	0.08
26	0.22	0.07
27	0.19	0.04
28	0.20	0.01
29	0.18	0.02
30	0.15	0.03
Average	0.18	0.06

Table 5.2.2: Testing result for mobile robot moves in horizontal line.

In Table 5.2.2 shown the testing result for the horizontal path plan of the mobile robot.

C. Mobile robot moves in diagonal line without obstacles.

NO.	Distance different from destination (Without IMU) / m	Distance different from destination (With IMU) / m
1	0.12	0.01
2	0.13	0.03
3	0.18	0.04
4	0.19	0.05
5	0.30	0.07
6	0.16	0.08
7	0.13	0.01
8	0.17	0.01
9	0.14	0.04
10	0.20	0.05
11	0.18	0.05
12	0.24	0.06
13	0.22	0.03
14	0.13	0.04
15	0.17	0.09
16	0.26	0.04
17	0.18	0.02
18	0.17	0.03
19	0.17	0.08
20	0.16	0.07
21	0.16	0.03
22	0.17	0.09
23	0.22	0.07
24	0.20	0.04
25	0.26	0.04
26	0.22	0.05
27	0.19	0.07
28	0.15	0.08
29	0.17	0.08
30	0.18	0.05
Average	0.18	0.05

Table 5.2.3: Testing result for mobile robot moves in diagonal line without obstacles.

In Table 5.2.3 shown the testing result for the diagonal path plan of the mobile robot without obstacles.

D. Mobile robot moves in diagonal line with obstacles

NO.	Distance different from destination (Without IMU) / m	Distance different from destination (With IMU) / m
1	0.18	0.06
2	0.22	0.07
3	0.16	0.05
4	0.17	0.08
5	0.15	0.03
6	0.19	0.05
7	0.18	0.08
8	0.17	0.09
9	0.21	0.10
10	0.16	0.09
11	0.14	0.11
12	0.12	0.10
13	0.25	0.10
14	0.24	0.12
15	0.19	0.08
16	0.21	0.04
17	0.28	0.04
18	0.17	0.05
19	0.18	0.04
20	0.21	0.05
21	0.20	0.05
22	0.21	0.02
23	0.20	0.07
24	0.19	0.09
25	0.15	0.07
26	0.17	0.12
27	0.18	0.09
28	0.14	0.08
29	0.20	0.11
30	0.19	0.09
Average	0.19	0.07

Table 5.2.4: Testing result for mobile robot moves in diagonal line with obstacles.

In Table 5.2.4 shown the testing result for the diagonal path plan of the mobile robot with obstacles.

E. Mobile robot moves vertically with obstacle in FYP lab.

NO.	Distance different from destination (Without IMU) / m	Distance different from destination (With IMU) / m
1	0.12	0.06
2	0.18	0.07
3	0.21	0.11
4	0.17	0.08
5	0.19	0.09
6	0.15	0.12
7	0.22	0.01
8	0.16	0.05
9	0.22	0.08
10	0.14	0.09
11	0.28	0.10
12	0.22	0.10
13	0.25	0.09
14	0.22	0.10
15	0.22	0.11
16	0.18	0.12
17	0.13	0.09
18	0.14	0.07
19	0.18	0.05
20	0.16	0.02
21	0.14	0.04
22	0.21	0.06
23	0.22	0.08
24	0.25	0.07
25	0.16	0.09
26	0.16	0.05
27	0.21	0.02
28	0.22	0.10
29	0.23	0.06
30	0.15	0.06
Average	0.19	0.07

Table 5.2.5: Testing result for mobile robot moves vertically with obstacle in FYP lab.

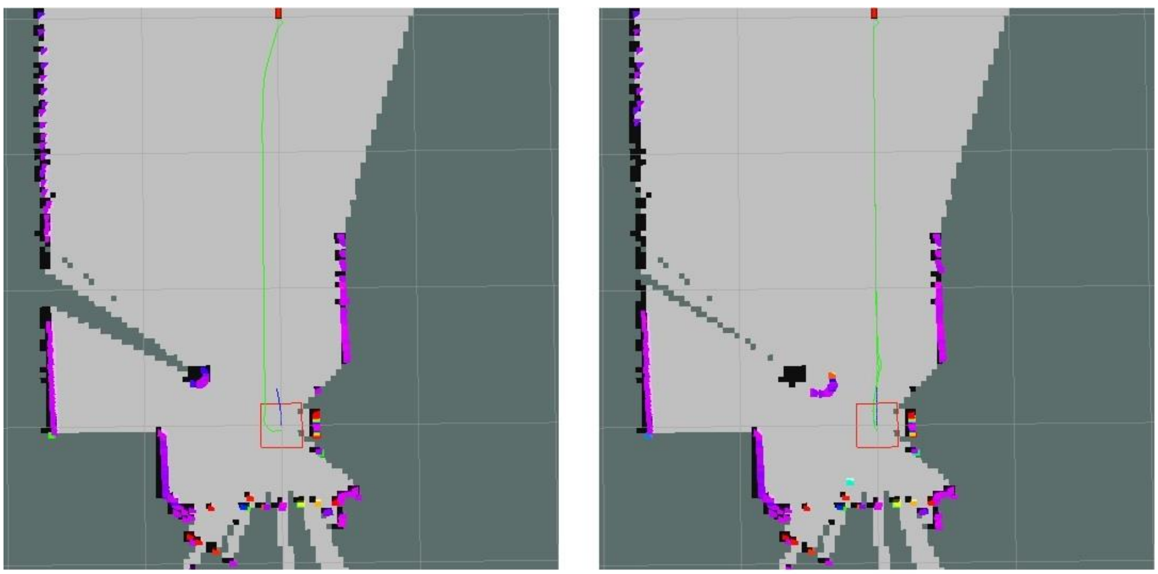
In Table 5.2.5 shown the testing result for the mobile robot moves to a destination vertically with obstacle in FYP lab.

5.3 Calculation Result

A. Vertical and horizontal path

Average distance apart from destination point after 30 tests carried out / meter				Percentage of improvement/ %	
Vertical Line		Horizontal Line		Vertical Line	Horizontal Line
With IMU	Without IMU	With IMU	Without IMU	64.7	64.7
0.17	0.06	0.18	0.07		

Table 5.3.1: Percentage of improvement in odometry for mobile robot move in vertically and horizontally after IMU implementation.



(a) (b)

Figure 5.3.1: Rviz visualization (a) without IMU implementation and (b) with IMU implementation.

Figure 5.3.1 and Table 5.3.1, shown that the localization error is reduced by 64.7% for both vertical and horizontal movement, after the implementation of IMU.

B. Diagonal path plan with and without obstacles

<table>
<tr><td colspan="4">Average distance apart from destination point after 30 tests carried out / meter</td><td colspan="2">Percentage of improvement/ %</td></tr>
<tr><td colspan="2">Moving in diagonal without obstacles</td><td colspan="2">Moving in diagonal with obstacles</td><td rowspan="2">Moving in diagonal without obstacles</td><td rowspan="2">Moving in diagonal with obstacles</td></tr>
<tr><td>Without IMU</td><td>With IMU</td><td>Without IMU</td><td>With IMU</td></tr>
<tr><td>0.18</td><td>0.05</td><td>0.19</td><td>0.08</td><td>72.2</td><td>57.9</td></tr>
</table>

Table 5.3.2: Percentage of improvement in odometry for mobile robot move in diagonally with and without obstacles after IMU implementation.

Referring to Table 5.3.2, the introduction of IMU reduces the localization error by 72.2% when it travels in diagonal line. If there are obstacles present, the proposed method is less effective with only 57.9% error reduction.

C. Mobile robot moves vertically with obstacle in FYP lab.

<table>
<tr><td colspan="2">Average distance apart from destination point after 30 tests carried out / meter</td><td>Percentage of improvement/ %</td></tr>
<tr><td colspan="2">Moving toward the destination with obstacle in FYP lab.</td><td rowspan="2">Moving toward the destination with obstacle in FYP lab.</td></tr>
<tr><td>Without IMU</td><td>With IMU</td></tr>
<tr><td>0.19</td><td>0.07</td><td>63.2</td></tr>
</table>

Table 5.3.3: Percentage of improvement in odometry for mobile robot move vertically with obstacle after IMU implementation.

Referring to Table 5.3.3, the introduction of IMU reduces the localization error by 63.2% when it travels in vertical line with obstacle in front.

Chapter 6.0 Conclusion

To conclude, the experimental result show that the localization is improved by 57.9% – 72.2% with the proposed technique. Moreover, this project provides an affordable solution for using the ROS - based SLAM algorithm to generate 2D indoor map of the environment models. In addition, the Raspberry Pi 3B+ also proved to be usable and compatible with the mobile robot. Moreover, ROS provides most of the functionality of a mobile robot. This concept will enable us to develop a cheap robot for a complex task, where the robot's overall cost is a vital aspect of scanning and monitoring unknown areas which there is a high risk of losing the mobile robot. There are few limitations of this project as follows:

- Robot localization with Lidar usually suffers from accumulated errors after long travelling distance.
- Robot localization with Lidar also suffers from accumulated errors after avoided the obstacles.
- The RAM on the Raspberry Pi was very limited which will be lagging during updating the local costmap with new scans data.
- With limited RAM also cause the movement of mobile robot not smooth due to local costmap is not updated properly.
- The real-time update for the mobile robot localization.

Integrate new sensors for example RFID reader, Global Positioning System module, and Kinect sensor will able to solve the robot localization limitation more efficiently. Furthermore, the single board computer technology is constantly improving, therefore using a more advanced single board computer which have higher processing power and RAM will be able to overcome the current limitation mentioned above. Moreover, must always have Wi-Fi connection with good connection strength and fast speed to solve the real-time update of the mobile robot localization.

Bibliography

AAGELA, H., AL-NESF, M. & HOLMES, V. An Asus_xtion_probased indoor MAPPING using a Raspberry Pi with Turtlebot robot Turtlebot robot. 2017 23rd International Conference on Automation and Computing (ICAC), 2017. IEEE, 1-5.

AMBROSE, R. O., ALDRIDGE, H., ASKEW, R. S., BURRIDGE, R. R., BLUETHMANN, W., DIFTLER, M., LOVCHIK, C., MAGRUDER, D. & REHNMARK, F. 2000. Robonaut: NASA. *IEEE Intelligent Systems*, 57-63.

ASFOUR, T., REGENSTEIN, K., AZAD, P., SCHRODER, J., BIERBAUM, A., VAHRENKAMP, N. & DILLMANN, R. ARMAR-III: An integrated humanoid platform for sensory-motor control. Humanoid Robots, 2006 6th IEEE-RAS International Conference on, 2006. IEEE, 169-175.

BAILEY, T. & DURRANT-WHYTE, H. 2006. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine,* 13**,** 108-117.

BORST, C., FISCHER, M. & HIRZINGER, G. Calculating hand configurations for precision and pinch grasps. Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, 2002. IEEE, 1553-1559.

BUTTERFAß, J., GREBENSTEIN, M., LIU, H. & HIRZINGER, G. DLR-Hand II: Next generation of a dextrous robot hand. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, 2001. IEEE, 109-114.

DOSHI, M. A., PAREKH, S. J. & BHOWMICK, M. 2015. Wireless Robotic Hand Using Flex Sensors. *International Journal of Scientific and Engineering Research,* 6.

FOSSEL, J.-D., TUYLS, K. & STURM, J. 2D-SDF-SLAM: A signed distance function based SLAM frontend for laser scanners. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015. IEEE, 1949-1955.

FU, M., ZHU, H., YANG, Y., WANG, M. & DENG, Z. A navigation map building algorithm using refined RBPF-SLAM. 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), 2016. IEEE, 2483-2487.

GATESICHAPAKORN, S., TAKAMATSU, J. & RUCHANURUCKS, M. ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera. 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2019. IEEE, 151-154.

GHANI, M. F. A., SAHARI, K. S. M. & KIONG, L. C. Improvement of the 2D SLAM system using Kinect sensor for indoor mapping. 2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS), 2014. IEEE, 776-781.

GÓMEZ, G., HERNANDEZ, A. & HOTZ, P. E. An adaptive neural controller for a tendon driven robotic hand. IAS, 2006. 298-307.

HAMZEH, O. & ELNAGAR, A. A Kinect-based indoor mobile robot localization. 2015 10th International Symposium on Mechatronics and its Applications (ISMA), 2015. IEEE, 1-6.

HESS, W., KOHLER, D., RAPP, H. & ANDOR, D. Real-time loop closure in 2D LIDAR SLAM. 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016. IEEE, 1271-1278.

HORAK, K. & ZALUD, L. 2016. Image processing on raspberry pi for mobile robotics. *International Journal of Signal Processing Systems,* 4**,** 1-5.

IBÁÑEZ, A. L., QIU, R. & LI, D. A Simple, Cost-Effective and Practical Implementation of SLAM Using ROS and Arduino. 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017. IEEE, 835-840.

IBRAGIMOV, I. Z. & AFANASYEV, I. M. Comparison of ros-based visual slam methods in homogeneous indoor environment. 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), 2017. IEEE, 1-6.

JACOBSEN, S., IVERSEN, E., KNUTTI, D., JOHNSON, R. & BIGGERS, K. Design of the Utah/MIT dextrous hand. Robotics and Automation. Proceedings. 1986 IEEE International Conference on, 1986. IEEE, 1520-1532.

JOHANSSON, A. J. & LINDE, J. Using simple force feedback mechanisms as haptic visualization tools. Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE, 1999. IEEE, 820-824.

KLUG, S., MÖHL, B., VON STRYK, O. & BARTH, O. 2005. Design and application of a 3 DOF bionic robot arm. *regulation,* 11**,** 12.

KRAMER, J. & SCHEUTZ, M. 2007. Development environments for autonomous mobile robots: A survey. *Autonomous Robots,* 22**,** 101-132.

LIU, Y. & SUN, Y. Mobile robot instant indoor map building and localization using 2D laser scanning data. 2012 International Conference on System Science and Engineering (ICSSE), 2012. IEEE, 339-344.

MANE, A. D., ARIF, S. M. & RAHIMAN, W. A. 2013. An Advanced Robot-Robin Heart (A Surgeon without Hand Tremor).

MICHAEL, N., SHEN, S., MOHTA, K., MULGAONKAR, Y., KUMAR, V., NAGATANI, K., OKADA, Y., KIRIBAYASHI, S., OTAKE, K. & YOSHIDA, K. 2012. Collaborative mapping of an earthquake ‐ damaged building via ground and aerial robots. *Journal of Field Robotics,* 29**,** 832-841.

NAMIKI, A., IMAI, Y., ISHIKAWA, M. & KANEKO, M. Development of a high-speed multifingered hand system and its application to catching. Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, 2003. IEEE, 2666-2671.

NÜCHTER, A., BLEIER, M., SCHAUER, J. & JANOTTA, P. 2017. Improving Google's Cartographer 3D mapping by continuous-time slam. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* 42**,** 543.

OCANDO, M. G., CERTAD, N., ALVARADO, S. & TERRONES, Á. Autonomous 2D SLAM and 3D mapping of an environment using a single 2D LIDAR and ROS. 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), 2017. IEEE, 1-6.

PAIK, J. K., SHIN, B. H., BANG, Y.-B. & SHIM, Y.-B. 2012. Development of an anthropomorphic robotic arm and hand for interactive humanoids. *Journal of bionic engineering,* 9**,** 133-142.

PEREIRA, V., FERNANDES, V. A. & SEQUEIRA, J. Low cost object sorting robotic arm using Raspberry Pi. 2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS), 2014. IEEE, 1-6.

PREMKUMAR, K. & NIGEL, K. G. J. Smart phone based robotic arm control using raspberry pi, android and Wi-Fi. 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015. IEEE, 1-3.

QUIGLEY, M., CONLEY, K., GERKEY, B., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R. & NG, A. Y. ROS: an open-source Robot Operating System. ICRA workshop on open source software, 2009. Kobe, Japan, 5.

REHMAN, U. A. 2013. Using Robots and SLAM for Indoor Wi-Fi Mapping in Indoor Geolocation [Revised and Extended Version].

ROBOTICS, C. 2017. ROS 101: Intro To The Robot Operating System| Robohub. *Robohub. org. Np*.

SCHROETER, C. & GROSS, H.-M. A sensor-independent approach to RBPF SLAM-Map Match SLAM applied to Visual Mapping. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IEEE, 2078-2083.

SONG, K.-T., CHIU, Y.-H., KANG, L.-R., SONG, S.-H., YANG, C.-A., LU, P.-C. & OU, S.-Q. Navigation Control Design of a Mobile Robot by Integrating Obstacle Avoidance and LiDAR SLAM. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018. IEEE, 1833-1838.

THRUN, S., BURGARD, W. & FOX, D. 2005. *Probabilistic robotics*, MIT press.

TRIVUN, D., ŠALAKA, E., OSMANKOVIĆ, D., VELAGIĆ, J. & OSMIĆ, N. Active SLAM-based algorithm for autonomous exploration with mobile robot. 2015 IEEE International Conference on Industrial Technology (ICIT), 2015. IEEE, 74-79.

VARGHESE, B. & THILAGAVATHI, B. Design and wireless control of anthropomorphic robotic arm. Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on, 2015. IEEE, 1-4.

VINCKE, B., ELOUARDI, A. & LAMBERT, A. Design and evaluation of an embedded system based SLAM applications. 2010 IEEE/SICE International Symposium on System Integration, 2010. IEEE, 224-229.

VINCKE, B., ELOUARDI, A., LAMBERT, A. & MERIGOT, A. Efficient implementation of EKF-SLAM on a multi-core embedded system. IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, 2012. IEEE, 3049-3054.

YUAN, L., QIN, S., YI, Z. & XIAOFENG, Z. 2016a. Improved Rao-Blackwellized $H\infty$ filter based mobile robot SLAM. *The Journal of China Universities of Posts and Telecommunications,* 23**,** 47-55.

YUAN, W., LI, Z. & SU, C.-Y. RGB-D sensor-based visual SLAM for localization and navigation of indoor mobile robot. 2016 International Conference on Advanced Robotics and Mechatronics (ICARM), 2016b. IEEE, 82-87.

ZHI, L. & XUESONG, M. Navigation and Control System of Mobile Robot Based on ROS. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018. IEEE, 368-372.

Poster

Simultaneous Localization and Mapping for Land Robot with Lidar and IMU Sensor

DARREN PHANG REN YEE   Supervisor: DR. LEE WAI KONG

Abstract

The Robotic Operating System (ROS) provides user an environment like operating system to control or operate the robots. Simultaneous localization and mapping (SLAM) and autonomous navigation in an enclosed environment are nowadays popular autonomous robots' issues. Lidar technology is widely used in performing robot localization. However, when only Lidar is being used, the accuracy of localization is not high. In this project, present a technique to integrate IMU sensor into a mobile robot equipped with Lidar module. With Extended Kalman Filter, the localization error introduced by Lidar can be corrected from time to time. Experimental results show that the robot can perform better localization with the introduction of IMU sensor.

Project Objective

- To develop a low-cost mobile robot able to perform SLAM in an indoor unknown environment.
- To develop a mobile robot able to perform autonomous navigation.
- To increase the accuracy of the mobile robot odometry when perform SLAM.

System Overview

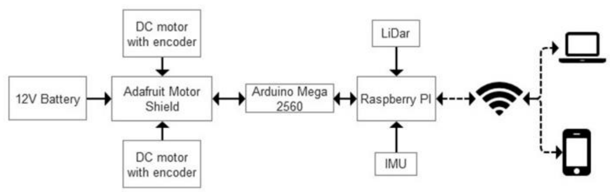


Figure 1: The hardware design of the SLAM mobile robot

Methodology

SLAM and Extended Kalman Filter (EKF)

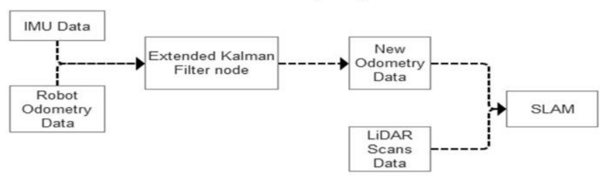


Figure 2: Visualization how EKF work

Discussion

The mobile robot is setup as shown in Figure 4 according to the proposed method. The main result of this study is to improve the robot localization accuracy in an unknown environment while using a low-cost mobile robot.

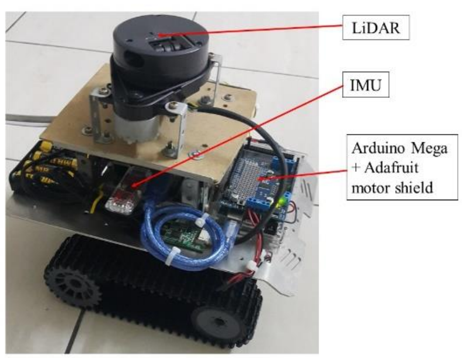


Figure 4: Overview of the mobile robot

Result

Average distance apart from destination point after 30 tests carried out / meter				Percentage of improvement/ %	
Vertical Line		Horizontal Line		Vertical Line	Horizontal Line
With IMU	Without IMU	With IMU	Without IMU	64.7	64.7
0.17	0.06	0.18	0.07		

Table 1: Percentage of improvement in odometry for mobile robot move in vertically and horizontally after IMU implementation.

Average distance apart from destination point after 30 tests carried out / meter				Percentage of improvement/ %	
Moving in diagonal without obstacles		Moving in diagonal with obstacles		Moving in diagonal without obstacles	Moving in diagonal with obstacles
Without IMU	With IMU	Without IMU	With IMU		
0.18	0.05	0.19	0.08	72.2	57.9

Table 2: Percentage of improvement in odometry for mobile robot move in diagonally with and without obstacles after IMU implementation.

Average distance apart from destination point after 30 tests carried out / meter		Percentage of improvement/ %
Moving toward the destination with obstacle in FYP lab.		Moving toward the destination with obstacle in FYP lab.
Without IMU	With IMU	
0.19	0.07	63.16

Table 3: Percentage of improvement in odometry for mobile robot move vertically with obstacle after IMU implementation.

Conclusion

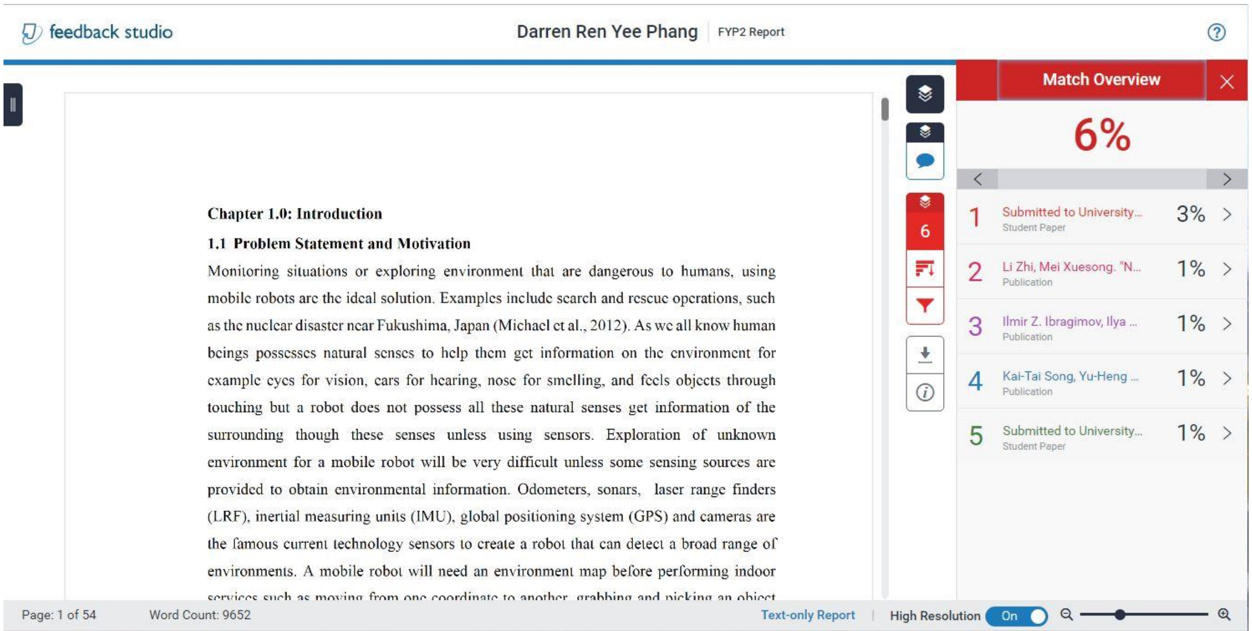
Robot localization with Lidar usually suffers from accumulated errors after long travelling distance and after avoiding obstacles. In this project, the proposed technique improves the accuracy of robot localization. The experimental results show that the robot localization is improved by 57.9 – 72.2%.

Future Work

- Integrate more sensors into the system to produce more accurate robot localization. For example, RFID reader, GPS, Kinect, etc.

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORATION AND COMMUNICATION TECHNOLOGY

Plagiarism Check Result



Turnitin Originality Report

Processed on: 23-Aug-2019 10:04 +08
ID: 1162283629
Word Count: 9652
Submitted: 4

FYP2 Report By Darren Ren Yee Phang

Similarity Index	Similarity by Source
6%	Internet Sources: 0% Publications: 3% Student Papers: 4%

2% match (student papers from 11-Jul-2019)
Submitted to University of Huddersfield on 2019-07-11

1% match (publications)
Li Zhi, Mei Xuesong. "Navigation and Control System of Mobile Robot Based on ROS", 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018

1% match (publications)
Ilmir Z. Ibragimov, Ilya M. Afanasyev. "Comparison of ROS-based visual SLAM methods in homogeneous indoor environment", 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), 2017

1% match (student papers from 28-Apr-2017)
Submitted to University of Huddersfield on 2017-04-28

1% match (publications)
Kai-Tai Song, Yu-Heng Chiu, Li-Ren Kang, Shao-Huan Song, Cheng-An Yang, Pei-Chun Lu, Song-Qing Ou. "Navigation Control Design of a Mobile Robot by Integrating Obstacle Avoidance and LiDAR SLAM", 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018

1% match (student papers from 29-Aug-2012)
Submitted to University of Leeds on 2012-08-29

Chapter 1.0: Introduction 1.1 Problem Statement and Motivation Monitoring situations or exploring environment that are dangerous to humans, using mobile robots are the ideal solution. Examples include search and rescue operations, such as the nuclear disaster near Fukushima, Japan (Michael et al., 2012). As we all know human beings possesses natural senses to help them get information on the environment for example eyes for vision, ears for hearing, nose for smelling, and feels objects through touching but a robot does not possess all these natural senses get information of the surrounding though these senses unless using sensors. Exploration of unknown environment for a mobile robot will be very difficult unless some sensing sources are provided to obtain environmental information. Odometers, sonars, laser range finders (LRF), inertial measuring units (IMU), global positioning system (GPS) and cameras are the famous current technology sensors to create a robot that can detect a broad range of environments. A mobile robot will need an environment map before performing indoor services

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index:________ % **Similarity by source** Internet Sources:_______________% Publications:____________ % Student Papers:______________ %	
Number of individual sources listed of more than 3% similarity: __________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

______________________________ ______________________________

Signature of Supervisor

Name: __________________________

Date: ___________________________

Signature of Co-Supervisor

Name: __________________________

Date: ___________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	
Student Name	
Supervisor Name	

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. ______________________ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ______________________ (Signature of Supervisor) Date: