Localization and Autonomous Navigation of a Mobile Robot: Robomuse 5.0

PAVAN SAI TEJA (2017ME10583)

SHASHANK (2017ME10572)

CHAITANYA RAM (2017ME10576)

Prof. Subir Kumar Saha

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, DELHI

FEB 2021

ABSTRACT

In manufacturing plants and warehouses, one of the most important job is to move objects from one location to the other. Traditionally this job was done manually using human labour and mechanical tools. This logistics inside these plants and warehouses cost a significant amount of expenditure. This handling and dispatch can be made efficient and the expenditure can be reduced by automating the process with robots. A simple solution is to design a line follower robot that travels to the workshop/warehouse floor in a predefined path. But the problem arises when the warehouse floor becomes highly dynamic with multiple robots and people moving simultaneously on a single path. In this project we are working on the 5th version of IIT Delhi's indigenous Autonomous Ground Robot: Robomuse 5.0. The objective is to build a robot that can perform SLAM and later Autonomous Navigation in the dynamic warehouse environment. Our design process is guided by the specifications required for moving Barrels in the Textile Department. As part of BTP1 we worked on Mapping of an environment using Lidar Scan Data. During BTP2, we will implement our learnings into a real AGV Robomuse 5.0.

CHAPTER 1

Introduction

Mobile robotics

Mobile robotics is the field of robotics concerned with robots that can move around in an environment. Today, it is an area with rapid research potential. These include robot assistants, personal assistant robots, distribution robots, warehouse robots, and so on. The major research areas in this field are Navigation, motion planning, Context driven exploration, Simultaneous Localization, and Mapping (SLAM). Mobile robots are usually controlled using software like ROS and equip sensors to detect obstacles in their surroundings while moving. Accuracy of the robot navigation and mapping directly depends upon the sensor package and the odometry techniques used inside the robot. The work on improving the robustness of the robot is critical to this field because reliability is essential when working in dynamic and unstable environments.

ROS

Robot Operating System (ROS) is a software platform that simplifies robot creation. It includes a plethora of software packages and drivers that help to speed up the robot creation phase and eradicate the need for constant reinvention of the wheel, which was common in robotics prior to ROS's popularity. It provides collaboration methods between various programmes known as nodes and helps them to make these relations in a much more comfortable manner. It's very modular, and the programming is very reusable. It has evolved into an indispensable instrument for roboticists. It allows programmers to use several languages of their choosing to programme the robots, such as Python, C++, Matlab, and so on, and it also allows for widespread collaboration with different team members due to individual packages that can be used along with each other. It is also supported on a variety of hardware architectures, including amd64 and arm processors. It is also cross-platform in terms of device compatibility with Windows and Linux. It is also really available to all programmers because it is open source and has an active user and enthusiast community. ROS has grown year after year, with more packages being released and more packages being more effective.

CHAPTER 2

Literature Survey

Simulations for Testing

Conducting simulations before implementing on a real robot always helps. It will help us utilize the time that will be wasted in component procurement and helps in testing our algorithms beforehand. There is a large collection of simulators available for designing SLAM using simulation; however, presenting all of them exhaustively is outside the reach of this article. Here are a few common choices that are related to our job.

1. Gazebo [1] has been one the most popular simulation platforms for the research work. It has a modular design that allows it to use different physics engines, sensor models and create 3D worlds. Gazebo goes beyond monolithic rigid body vehicles and can be used to simulate more general robots with links-and-joints architecture such as complex manipulator arms or biped robots. While Gazebo is fairly feature rich it has been difficult to create large scale complex visually rich environments

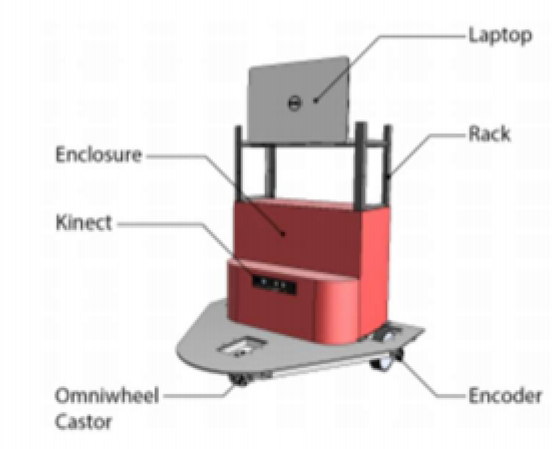
2. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles that are closer to the real world and it has lagged various advancements in rendering techniques made by platforms such as Unreal engine or Unity.

3. Other notable efforts include Hector [2] that primarily focuses on tight integration with popular middleware ROS and Gazebo. Similarly, Rotors [3] provides a modular framework to design Micro Aerial Vehicles and build algorithms for control and state estimation that can be tested in simulators. Finally, jMavSim [4] is an easy to use simulator that was designed with a goal of testing PX4 firmware and devices.

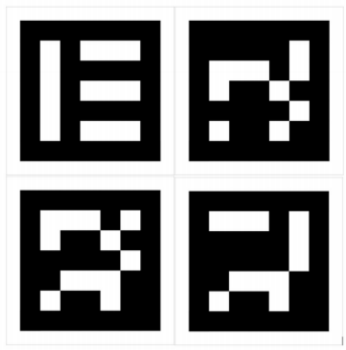
Last time we used AirSim in combination with MATLAB to implement SLAM as part of BTP1. In this course, we are implementing Navigation on a real robot. This required us to shift the platform from MATLAB|AIRSIM to ROS.

Mobile Robot: Robomuse

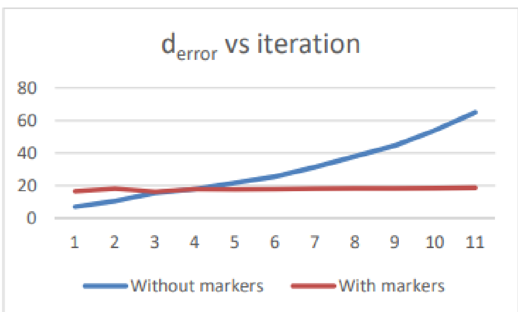
IIT Delhi has its own robot series called RoboMuse. RoboMuse 1 started as a line-following mobile robot capable of moving from point A to point B while automatically charging. It was inspired by one of the robots built by the IIT Delhi team for Robocon 2008. The next update, RoboMuse 2, focused on improving the mission's reliability and robustness by using white wooden straps for line following instead of plastic tapes, which were often damaged. It also came with upgraded circuitry for quicker charging. The same technology was used by RoboMuse 3, which was tasked with picking up a plastic container and depositing it in a basket placed a distance away. Robomuse 4.0 used the Kinect(Vision) Sensor to map and explore its surroundings, as well as Aruco Markers for odometry error correction that builds over time. Aruco markers are not used in Robomuse 5 because SLAM techniques such as GMapping use (Lidar) Scan Matching as an additional step in determining the robot's odometry. It comes with two Lidar sensors for measuring depth. In contrast to the Kinect sensor, which produces unreliable pseudo-laser scan results, Lidar sensors provide precise scan data.



Robomuse 4



Aruco Markers



Error in Odometry:
With vs Without Marker

In case of RoboMuse 4.0, they used Aruco marker to correct the robot's estimated pose. The robots pose is updated every time a new Aruco Marker is in the field of Kinect Sensor.

CHAPTER - 3

PROJECT OBJECTIVES AND WORK PLAN

Problem Definition/Motivation:

Factory and Warehouse automation is being used in the modern supply chain in the form of autonomous driving vehicles like automated stackers, forklifts, trucks, and rack-carrying robots. These vehicles are proving to be the safest alternative to factory workers handling labour intensive tasks. Security, adaptability, cost savings, increased productivity, and precision are few advantages of using (AGVs) auto-guided vehicles in the warehouse. In this project we are mainly focused on solving this automation in the textile industries where barrels containing fabric materials are transported from one place to the other. We are working with the Textile Dept at IIT Delhi under the guidance of Prof SK Saha Sir.

OBJECTIVES OF THE WORK

In this project, our goal is to move the ground robot Robomuse-5.0 from one place to another autonomously in an environment similar to that of a Textile Industry.

Our main goals:

1. Autonomous Navigation of Robot Muse 5.0
2. Design & Structural Analysis of a Textile Industrial Robot

At the beginning of the project, RoboMuse 5.0 frame is available. So, we will utilize the structure and integrate hardware/software stack to make it autonomous. We will test the system in a real environment. We can divide the project into 3 parts

Mechanical Component: For Robomuse 5.0, the structure is already available. Robomuse 5.0 is a small robot with load capacity <150 Kg. We have the CAD design of a blown-up ground robot with higher load capacity that can be used in the Textile industry directly. In this project we will only conduct Structural analysis on the CAD file. Fabrication can be done in a later stage.

Electrical/Electronics: Selecting the required components and building the circuit. This process will require us to make a custom PCB.

Software Stack: We will use ROS for running the robot. Till now, we have implemented open source GMapping (SLAM tool) and AMCL (Navigation Stack) on the TurtleBot3 robot using Gazebo

METHODOLOGY

- Simulating TurtleBot robot on Gazebo
- Testing Mapping and Navigation Stack on Turtlebot3 robot using ROS packages
- We use GMapping packages for implementing SLAM, and AMCL package for implementing Navigation
- Designing the Electric Circuit and required PCB
- Building the complete RoboMuse 5.0
- Testing RoboMuse 5.0 for Autonomous Navigation in real environment.

GANTT CHART

TASK	Week 1-2	Week 3-4	Week 5-6	Week 7-8	Week 9-10	Week 11-12	Week 13-14
Literature Review about Lidar-SLAM & Visual Inertial- SLAM							
Learning ROS & Simulating Turtlebot 3 Robot							
Implementing Lidar based SLAM & Navigation in Gazebo & Rviz							
Started Working on RoboMuse 5.0 PCB Design | AGV Control System							
Implement SLAM & Navigation in Robomuse 5.0							
Conducting Navigation in Real Environment							
Report Making							

CHAPTER 4

WORK PROGRESS

RELEVANT THEORY, EQUATIONS

SLAM (Simultaneous Localization and Mapping):

- SLAM addresses the solution for autonomous navigation of mobile robot in an unknown environment. While navigating the environment, the robot seeks to acquire a map of the environment, and at the same time it wishes to localize itself using its map.

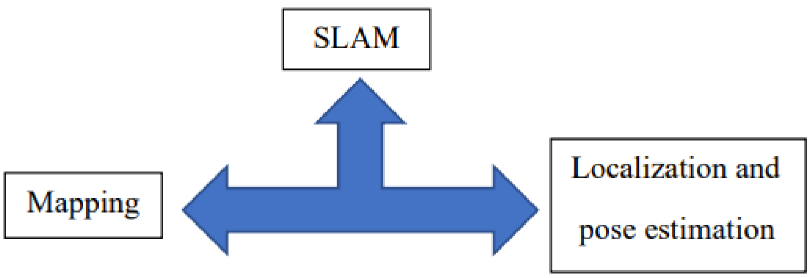
  Mapping : Constructing the map of the environment using a mobile sensor platform
  Localization : Estimating the robot location

- SLAM is a Chicken-Egg problem and is a fundamental problem for truly autonomous robot i.e.
  a) A map is needed for localization and
  b) Robot pose estimate/Robot location is needed for Mapping

- Hence, Simultaneous localization and mapping refer to the navigation of mobile robots, constructing a map of the unknown world in the absence of external referencing systems such as GPS, and simultaneously localizing within this map (SLAM).

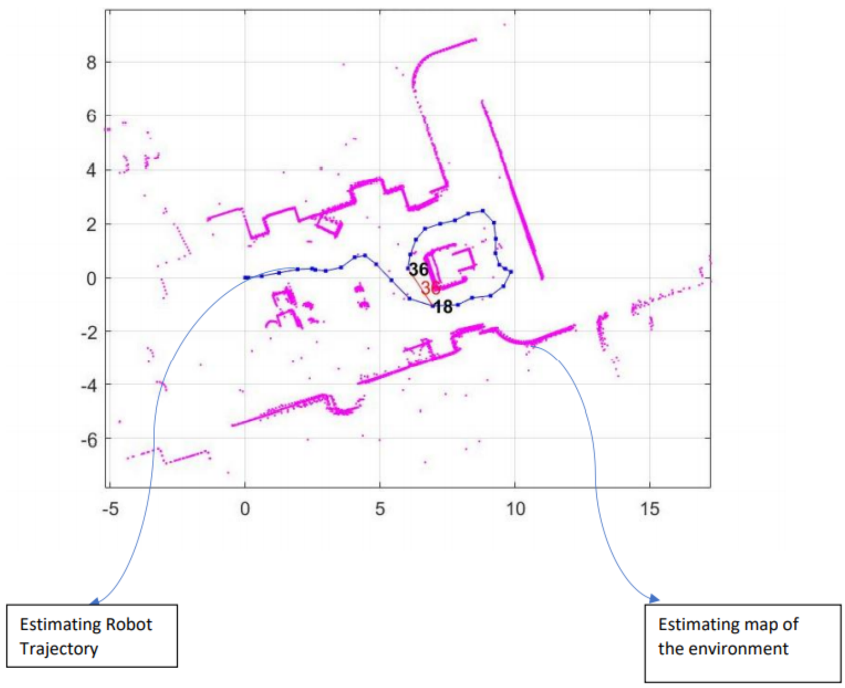
- Learning maps under robot pose uncertainty is also often referred to as simultaneous localization and mapping.

- SLAM is an essential skill for navigation of the mobile robots in an unknown environment.



- SLAM uses “mapping algorithms” to build a map and “Localization algorithms” to localize your vehicle in that map at the same time.

Mapping and Localization of autonomous robot



2D grid maps using lidar-based SLAM-MATLAB

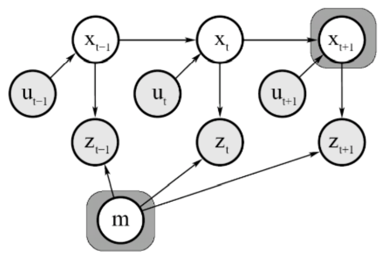
Picture : Taken from MATLAB SLAMBOX TUTORIALS

MATHEMATICAL DESCRIPTION OF SLAM PROBLEM:

GIVEN:

Sensor Observations: $u_{1:T} = \{u_1, u_2, u_{3\ldots}, u_T\}$

Odometry Measurements: $z_{1:T} = \{z_1, z_2, z_{3\ldots}, z_T\}$



$p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1})$

SLAM PROBLEM IS TO FIND :

Given a series of sensor observations and odometry measurements over discrete time steps t, the SLAM problem is to compute an estimate of the robot’s state(x_t) and a map of the environment(m_t).

All quantities are usually probabilistic, so the objective is to compute $p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1})$

Map: m (Occupancy grid maps are used)

Path of the Robot: $x_{0:T} = \{x_1, x_2, x_{3\ldots}, x_T\}$

ALGORITHMS:

Statistical techniques for estimating the robot’s pose and map of the environment include “Kalman Filters” and “Particle Filters” (aka. Monte Carlo methods)

ROS SLAM GMapping:

- This package contains GMapping, from OpenSlam, and a ROS wrapper.
- The gmapping package provides laser-based SLAM, as a ROS node called slam_gmapping.
- Slam gmapping is used to create a 2-D occupancy grid map (which is like a building floor plan) from laser and pose data collected by mobile robots.
- The map is then saved as a .pgm file.

Adaptive Monte-Carlo Localization (AMCL):

- AMCL, which is a particle filter based is used to localize the mobile robot in an indoor environment.
- Specifically, it refers to estimation of the position and orientation of a robot within the previously generated map.

Description of the Problem:

For Autonomous Navigation, we need

1. SLAM

2. Navigation

3. Motion Planning

But, in this project we are implementing SLAM and Navigation on the mobile robot RoboMuse 5.0. This project is intended to create an Autonomous Mobile Robot capable of localization and Navigation using Lidar sensors in factory setups.

Fig 1: Picture Courtesy: Peer Robotics

Experimental Setup

Simulation in ROS:

1. Spawned TurtleBot3[9] in Gazebo as a Node
2. Controlled Turtlebot3 using Teleop Node
3. Implement GMapping Stack using TurtleBot3 | moved robot using teleop Node

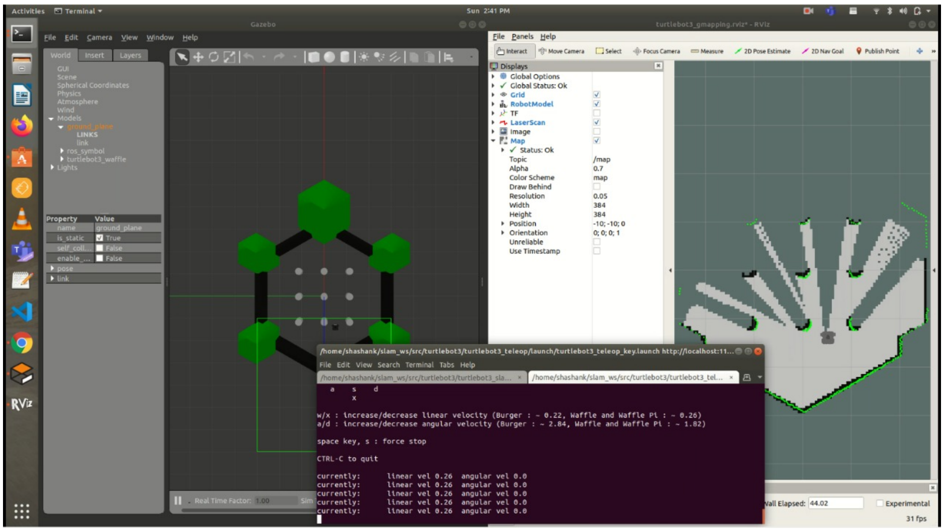


Fig: 2 _ On the left : gazebo simulator node | On the right : GMapping Node

4. Saving the generated GridMap of the Environment
5. Loading the GridMap and implementing Navigation stack on TurtleBot

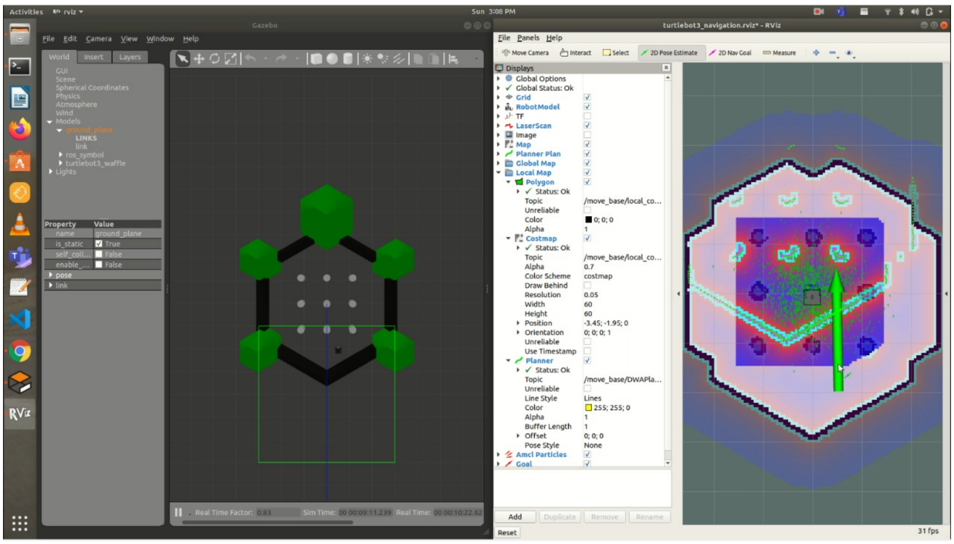
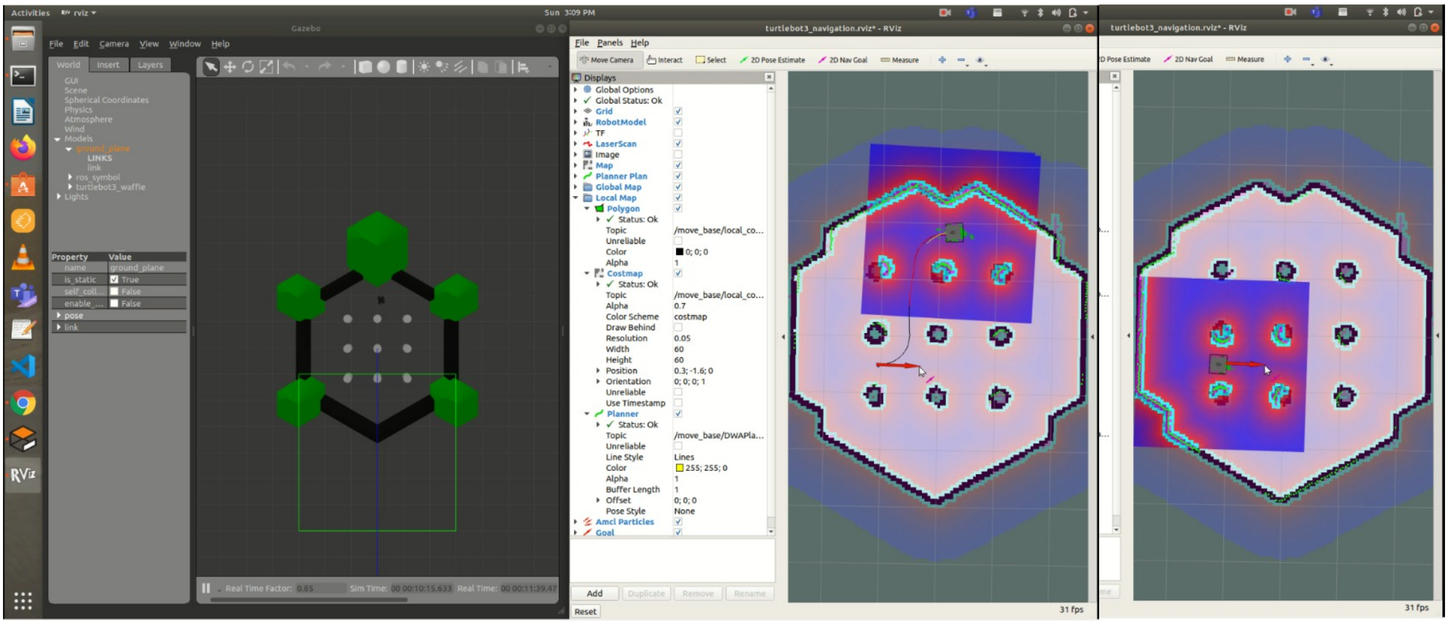


Fig: 3 _ On the left : Gazebo and Turtlebot3 | On the right : Initializing the robots current location in RViz (Navigation stack Node)



Navigation of robot in the simulation: On the left we can see the global path planning and on the right we can see the

Robot Body:

Currently, we have a RoboMuse 5.0 which is not completely built, our next task is to build the robot completely with all electronics integrated.

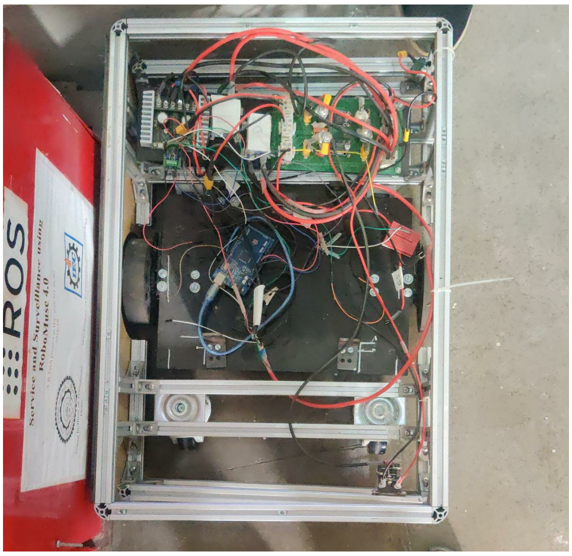


Fig: 4_ Structure of RoboMuse 5.0

PCB designing with Eagle Software:

A **printed circuit board**, or **PCB**, is used to mechanically support and electrically connect electronic components using conductive pathways, tracks or signal traces etched from copper sheets laminated onto a non-conductive substrate. We are currently learning the software to build our custom PCB for the RoboMuse 5.0. We are using Autodesk Eagle Version 9.6.2

CONCLUSION

Study of SLAM algorithms in simulation environments is completed. Now we are going to Study the hardware- Electrical components needed for robots like Arduino, Motor Drivers - Sabertooth, PCB designing and installing the algorithm developed in the simulation. Redesign the Robomuse 5.0 and build it completely. After building the robot, we have to test the Robomuse 5.0 to move autonomously in Our campus (Real world Environment).

REFERENCES

1. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IROS (2004)

2. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., Von Stryk, O.: Comprehensive simulation of quadrotor uavs using ros and gazebo. In: SIMPAR, pp. 400–411. Springer (2012)

3. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: Rotorsa modular gazebo mav simulator framework. In: Robot Operating System (ROS), pp. 595–625. Springer (2016)

4. Babushkin, A.: Jmavsim. https://pixhawk.org/dev/hil/jmavsim

5. https://microsoft.github.io/AirSim/ (AirSim Documentation)

6. Cadena, Cesar & Carlone, Luca & Carrillo, Henry & Latif, Yasir & Scaramuzza, Davide & Neira, Jose & Reid, Ian & Leonard, John. (2016). Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age. IEEE Transactions on Robotics. 32. 10.1109/TRO.2016.2624754.

7. The rise of Robot Operating System (ROS) software | Machine Design

8. Peer Robotics – Robots designed to work with humans

9. TurtleBot3 (robotis.com)

10. wiki.ros.org/gmapping