VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JnanaSangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)**

ON

"MOBILE DRIVE ROBOT"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

ANURODH VERMA (1CR16CS024)

KARTHIK K (1CR16CS068)

KIRAN S M (1CR16CS072)

PANCHAL VIRAL ASHOK (1CR16CS109)

Under the Guidance of,

Prof. Sherly Noel

Assistant professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"MOBILE DRIVE ROBOT"** carried out by **Mr**. **ANURODH VERMA**, USN **1CR16CS024**, **Mr. KARTHIK K**, USN **1CR16CS068**, **Mr. KIRAN S M**, USN **1CR16CS072, Mr. PANCHAL VIRAL ASHOK**, USN **1CR16CS109,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of Bachelor **of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

.

Prof. Sherly Noel	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Assistant Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**MOBILE DRIVE ROBOT**" has been successfully completed under the guidance of Prof. Sherly Noel, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:

Date:

Team members:

ANURODH VERMA (1CR16CS024) ___________________

KARTHIK K (1CR16CS068 ___________________

KIRAN S M (1CR16CS072) ___________________

PANCHAL VIRAL ASHOK (1CR16CS109) ___________________

ABSTRACT

In this fast-paced world, anyone who can't keep up gets left behind, there are no exceptions made for anyone. The graph of the number of differently-abled people that have entered the industry and are scaling new heights is positively pacing, but these individuals require additional help to keep up with their peers. Autonomous robots are helpful in busy environments, like industries, hospitals etc. Instead of employees leaving their posts, an autonomous robot can take over the work so that other workers can focus on more important things rather than picking and moving objects.

Industries has to increase their productivity to keep up with the pace of today’s world, our robot will help in maximizing the productivity.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,**Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide
Prof. Sherly Noel, Assistant Professor, Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

In this fast-paced world, anyone who can't keep up gets left behind, there are no exceptions made for anyone. The graph of the number of differently-abled people that have entered the industry and are scaling new heights is positively pacing, but these individuals require additional help to keep up with their peers.

Autonomous robots are helpful in busy environments, like industries, hospitals etc. Instead of employees leaving their posts, an autonomous robot can take over the work so that other workers can focus on more important things rather than picking and moving objects. Without traditional guidance, these robots can navigate the industrial rooms, hospital hallways etc, and can even find alternate routes when another is blocked. They will stop at pick-up points, and collect the object and move it to the desired place.

1.1 Problem Statement

In today's world if a brand or an industry wants to grow rapidly and effectively, they need to automate couple of things so that the workers in industries can focus more on important things related to products rather than moving and placing objects from one place to another. For this a material handling robot needs to be designed. This also helps the company to optimize the labor cost and can save their workers from tedious and hazardous jobs and can optimize their productivity.

1.2 Objectives

Our objective is to design a sturdy material handling robot that can take the load of objects that needs to be moved from one place to another. We are focusing on designing a gripper that can pick and keep objects without dropping or damaging them. Our main focus is on designing a robot that can navigate itself in any environment by creating a 3D map of that environment. The robot should be able to avoid obstacles and should navigate purposefully to reach the destination.

1.3 Methodology

The main part of this project is ROS. Everything is build around ROS. Software in ROS is organized in packages. A package might contain ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module. The code is written in the form of packages. When we run our ROS nodes, they perform computations and obtain results. But they require results from other nodes in order to perform some other functions. The ROS Master provides naming and registration services to the rest of the nodes in the ROS system. It tracks publishers and subscribers to topics as well as services. The role of the Master is to enable individual ROS nodes to locate one another. Once these nodes have located each other they communicate with each other peer-to-peer.The transfer of data takes place via topics. If you want to send your data, you publish it to topics and whoever needs it, can subscribe to it using publishers.

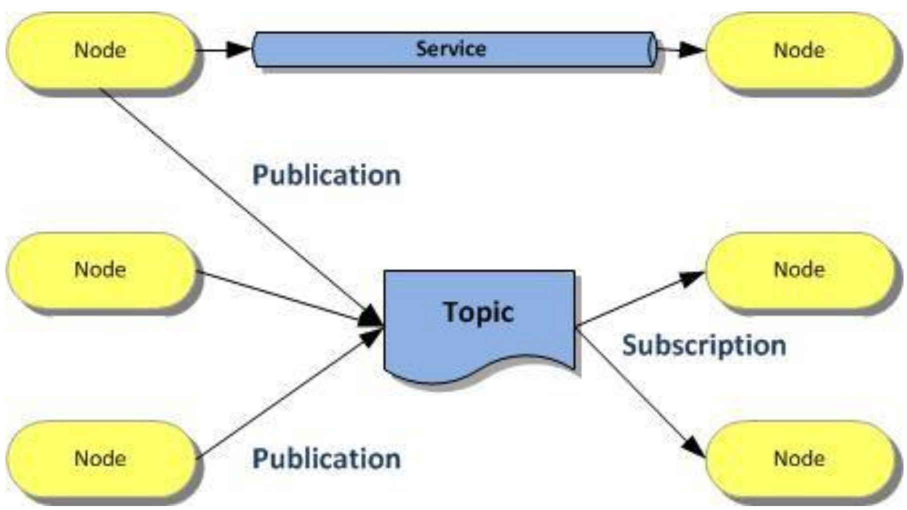


Fig.1 ROS Topic & Nodes

We have used Gazebo which is a simulation platform to simulate a working robot. The robot will be setup with different sensors and actuators. While running these sensors, we may need to visualize their data. We use RViz for this. RViz is a 3D Visualization tool for ROS. It takes in a topic as input and visualizes that based on the message type being published. It lets us see the environment from the perspective of the robot.

Chapter 2
LITERATURE SURVEY

Paper- 1

2.1 Design and Implementation of Robot Arm Control Based on MATLAB with Arduino Interface

By- T.Rajesh, M. Karthik Reddy, Afreen Begum, D.Venkatesh

In the present days, a number of situations exist where it is not possible for a human operator to do an activity on his/her own, due to a level of danger or difficulty involved. They may involve taking readings from an active volcano, entering a building on fire, diffusing a bomb, or collecting a radioactive sample. Rather than compromising on human lives, it is better to employ robotic systems for performing difficult tasks. Robotic systems are far superior in ensuring the accuracy of the system under adverse circumstances wherein a human operator may lose his/her composure and focus. Here we propose to build a robotic arm controlled by Matlab/Simulink interfacing with Arduino Uno. The development of this arm is based on the Arduino platform and Matlab A servo motor is a combination of DC motor, position control system, gears. The position of the shaft of the DC motor is adjusted by the control electronics in the servo, based on the duty ratio of the PWM signal. Servo is proposed for low speed, medium torque and accurate position application. These motors are used in robotic arm machines, flight controls and control systems. This project presents an interactive module for learning both the fundamental and practical issues of servo systems interface with ARDUINO UNO. This project, developed using Matlab coding tool, is used to control robotics applications. The objective of this project is to control the servo by using ARDUINO UNO with MATLAB & SIMULINK.

This paper proposes to build a robotic arm controlled by MATLAB/Simulink interfacing with Arduino Uno. The development of this arm is based on the Arduino platform and MATLAB. A servo motor is a combination of DC motor, position control system, gears. The position of the shaft of the DC motor is adjusted by the control electronics in the servo, based on the duty ratio of the PWM signal. Servo is proposed for low speed, medium torque and accurate position application.

These motors are used in robotic arm machines, flight controls and control systems. This project presents an interactive module for learning both the fundamental and practical issues of servo systems interface with ARDUINO UNO. This project developed using MATLAB coding tool, is used to control robotics applications. The objective of this project is to control the servo by using ARDUINO UNO with MATLAB & SIMULINK.

Paper - 2

2.2 Design of a mobile robot with a robotic arm utilizing microcontroller and wireless communication

By- I.B.Alit Swamardika

The purpose of this study is to design a prototype of a mobile robot equipped with a robotic arm which can be controlled by wireless technology. In this scheme, the mobile robot in the form of 6 Wheel Drive Robot equipped with robotic arm 6 Degree of Freedom and is controlled wirelessly through remote control based on XBee Pro Series 1. Data on the remote is sent serially via XBee transmitter, processed on the receiver, and then used as a reference to control the robot. The tests on the forward, backwards, turn left, turn right, stop and linear movement of the robot was performed successfully, which elevates the robot deployment potentials. The Six Degree of Freedom (6 DOF) has enabled the robotic arm to perform the designed movements very well. Furthermore, the mobile robot successfully follows the command from each input variable resistor via the remote control to move the robotic arm. The mobile robot and the robotic arm movement can successfully be done simultaneously, which elevates its potentials for deployment.

Hardware:

- Microcontroller Arduino Mega 2560 as a data processor and robot controllers.
- Wireless communications module XBee-PRO as sender and recipient of the data instructions.

Software:

- The Arduino IDE is used to create the program in the microcontroller.

Paper - 3

2.3 Pick and Place Robot with Wireless Charging Application

by N. Firthous Begum, P

Mankind has always strived to give life-like qualities to its artifacts in an attempt to find substitutes for himself to carry out his orders and also to work in a hostile environment. The popular concept of a robot is of a machine that looks and works like a human being. The industry is moving from the current state of automation to Robotization, to increase productivity and to deliver uniform quality. One type of robot commonly used in industry is a robotic manipulator or simply a robotic arm known as a pick and place robot. It is an open or closed kinematic chain of rigid links interconnected by movable joints. In this paper pick and place, Robot is designed which performs its operation by using android via object detection application and PIC microcontroller. This application has been programmed in java language. In the transmitter part, the voice input is given by using HM2007 to the microcontroller by using RF module. In the receiver section, the RF receiver will receive this voice input and it will be given to the microcontroller. Simultaneously the object to be picked will be done by using an android application where the camera of the android mobile will capture the objects. The output from the mobile will be sent through Bluetooth to the microcontroller and that will allow the motor to move in order to pick the object. In this paper, wireless charging application is also implemented by using electromagnetic induction concept that allows the robot to charge itself whenever the onboard battery goes low. Keywords: robotics, pick and place robots, wireless charging application, android object detection application.

In the proposed system, a humanoid robot is implemented which performs the task initiated by the user without human assistance by voice input. The pick and place robot eliminates the need for sensors which are used to detect object and object detection application is developed using java language. The input voice is given to the microcontroller and sent to the receiver using an RF transmitter. Simultaneously the object picking is done by using an android application where the camera of android mobile will detect and capture the image of the object.

The output from the android mobile is sent to the microcontroller through Bluetooth and it allows the motor to move in order to pick the object by using the motor drive. Wireless charging is implemented on the principle of electromagnetic induction.

Paper - 4

2.4 Development of real-time tracking and control mobile robot using video capturing feature for unmanned applications

By-P. Velrajkumar, S. Solai Manohar, C. Aravind, A. Darwin Jose Raju and R. Arshad

A wireless tracking and controlling mobile robot using video capturing features (VCF) for unmanned applications is presented. It is wheel-based and Radiofrequency (RF wireless communicated) is used for communication between the controller and the robot. The available motions of the robot are forward, backward, right, left and the combination of these movements. Besides, a camera is built in the robot for tracking. The video captured by the camera is displayed in the computer or Laptop by using the Window Media Encoder (WME) software that is able to be controlled by using Windows GUI remote control. It is built for the purpose of viewing the places that humans cannot reach. Same with other robots which are built to work in dangerous environments, it can be used to explore the situation of dangerous places that humans cannot reach, for example, the natural disaster area, cave, underground and so on. On the other hand, it can also serve as an investigation robot in the military field. It suits the task of searching for an ambush or sneaking into an enemy base to gather information.

This research presented the controlling and tracking of mobile robots using Video Broadcast Feature and Audio Broadcasting Feature for the unknown environment applications. It is an android based autonomous robot with Bluetooth and Wi-Fi that is used for communication between the robot and a remote controller. The robot is self -powered with 9V supply to power up the microcontroller, motors and transmitter for audio broadcast. The microcontroller used to control the robot remotely in the Arduino, it has built-in input and output ports. For the real-time audio transmitter, it will detect voice and any kind of audio using a mini electric microphone to transmit a signal when detected through a transmitter circuit that is designed to transmit the sound with the specified frequency range.

This robot has real-time video broadcasting capability to watch the activities that take place in other places. The real-time video coverage is broadcast by using smartphones that use broadcasting software. The video is transmitted using the internet medium for longer range receiving destination. The designed mobile robot can be used for surveillance purposes or military purposes for spying.

CHAPTER 3
SOFTWARE REQUIREMENT

The implementation phase of the project is where the detailed design is actually transformed into working code. Aim of the phase is to translate the design into the best possible solution in a suitable programming language. This chapter covers the implementation aspects of the project, giving details of the programming language and development environment used. It also gives an overview of the core modules of the project with their step by step flow.

The implementation stage requires the following tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding the selection of the platform
- Appropriate selection of the language for application development

3.1 SOFTWARE USED

3.1.1 ROS: Robot Operating System

Robot Operating System (**ROS or ros**) is robotics middleware (i.e. collection of software frameworks for robot software development). Although ROS is not an operating system, it provides services designed for a heterogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor data, control, state, planning, actuator, and other messages. Despite the importance of reactivity and low latency in robot control, ROS itself is *not* a real-time OS (RTOS). It is possible, however, to integrate ROS with real-time code.

Software in the ROS Ecosystem can be separated into three groups:

- language-and platform-independent tools used for building and distributing ROS-based software;
- ROS client library implementations such as roscpp, rospy, and roslisp
- Packages containing application-related code which uses one or more ROS client libraries.

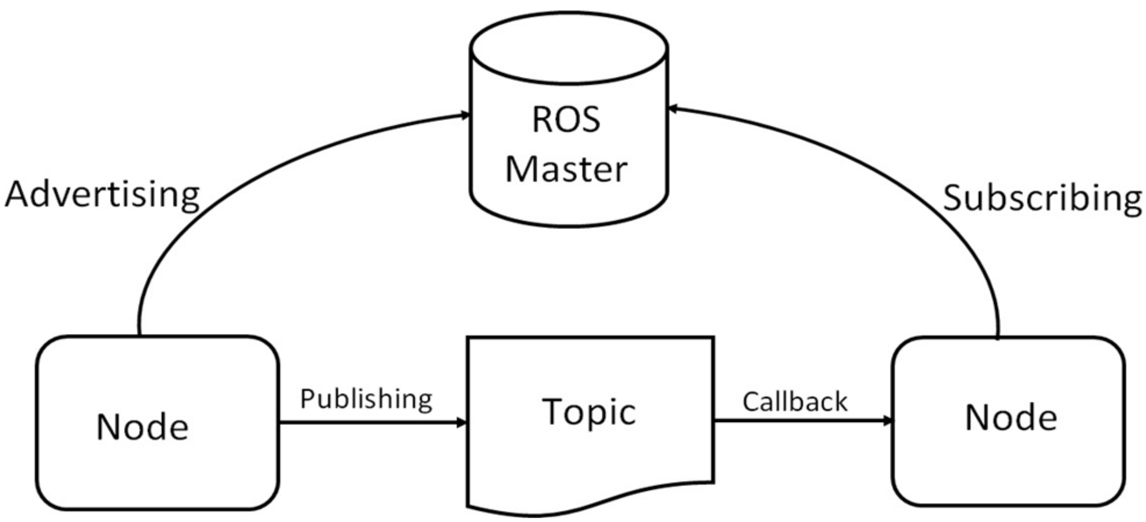


Fig.2 ROS

3.1.2 OpenVSLAM

OpenVSLAM is based on an indirect SLAM algorithm with sparse features, such as ORB-SLAM, ProSLAM, and UcoSLAM. One of the noteworthy features of OpenVSLAM is that the system can deal with various types of camera models, such as perspective, fisheye, and equirectangular. If needed, users can implement extra camera models (e.g. dual fisheye, catadioptric) with ease. Visual SLAM systems are essential for AR devices, autonomous control of robots and drones, etc. Simultaneous localization and mapping (SLAM) systems have experienced a notable and rapid progression through enthusiastic research and investigation conducted by researchers in the fields of computer vision and robotics. OpenVSLAM can accept images captured with perspective, fisheye and equirectangular cameras.

The main contributions of OpenVSLAM are:

- It is compatible with various types of camera models and can be customized for optional camera models.
- Created maps can be stored and loaded, then OpenVSLAM can localize new images using prebuilt maps.
- A cross-platform viewer running on web browsers is provided for the convenience of users

3.1.3 OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. The OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

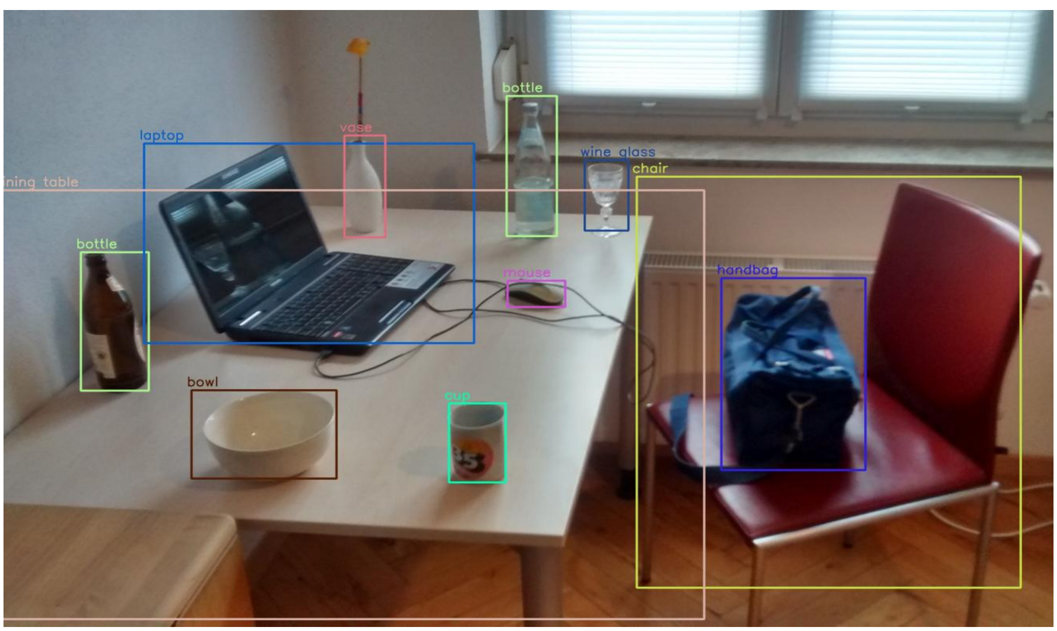


Fig.3 Object detection using OpenCV python

3.1.4 GAZEBO

Robot simulation is an essential tool in every roboticist's toolbox. A well-designed simulator makes it possible to rapidly test algorithms, design robots, perform regression testing, and train AI system using realistic scenarios. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. At your fingertips is a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces

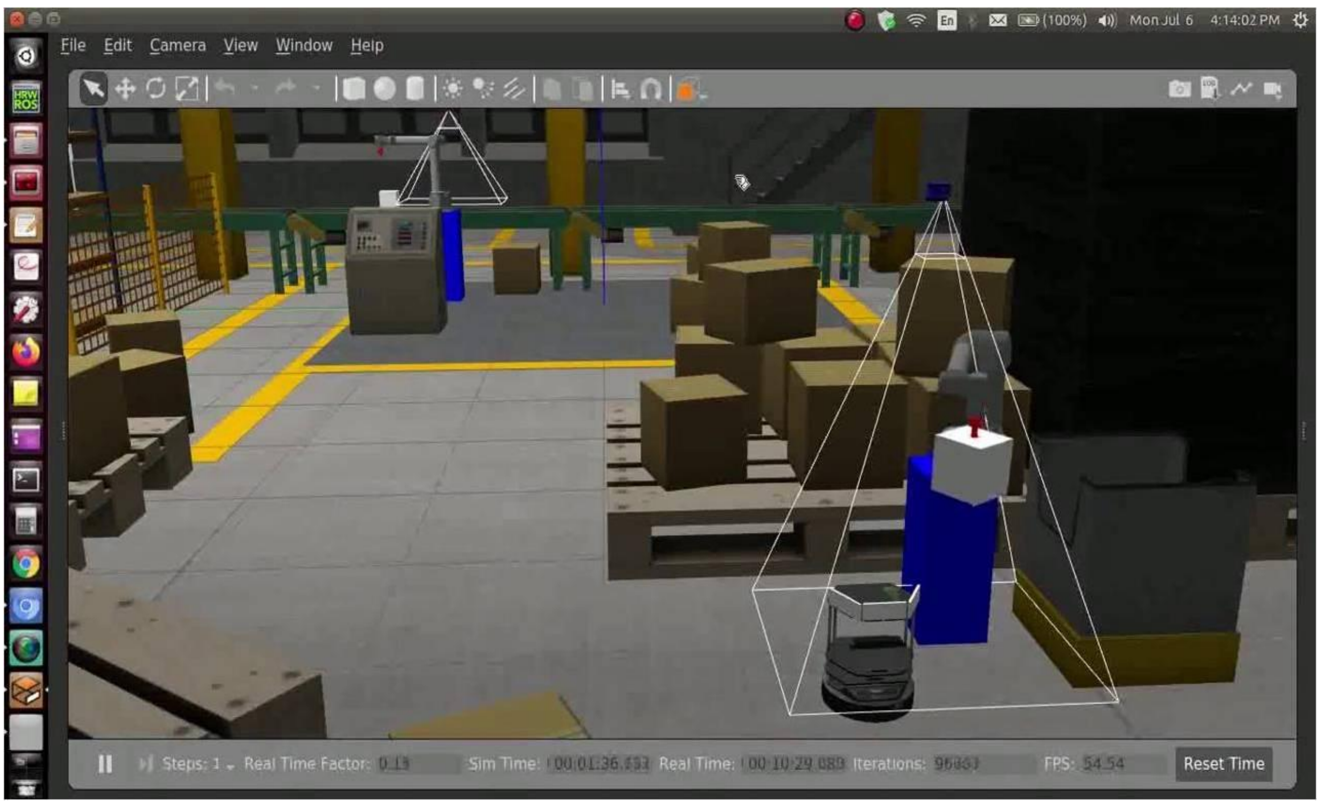


Fig.4 Simulation Using Gazebo

3.15 ROS MoveIt!

MoveIt is the most widely used software for manipulation and has been used on over 100 robots. It provides an easy-to-use robotics platform for developing advanced applications, evaluating new designs and building integrated products for industrial, commercial, R&D, and other domains.

By incorporating the latest advances in motion planning, manipulation, 3D perception, kinematics, control and navigation, MoveIt is state of the art software for mobile manipulation.

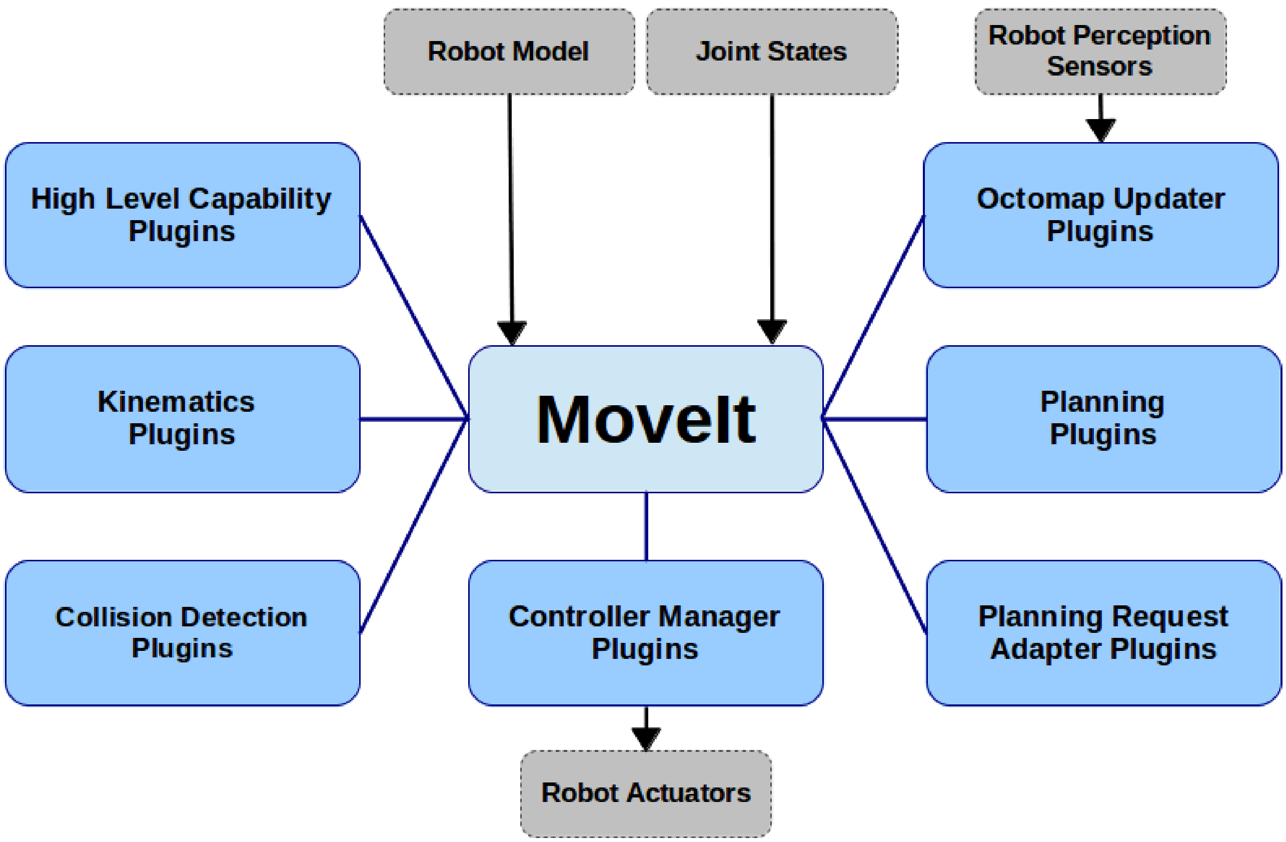


Fig.5 MoveIT

3.1.6 FlexBE

FlexBE helps you to create complex robot behaviors without the need for manually coding them. Based on basic capabilities, which interface standard functionality or your own system-specific features, state machines can easily be composed via the provided drag&drop editor. Afterwards, using the same graphical interface, execution can be started and monitored. Based on the concept of collaborative autonomy, the operator is able to influence the execution during runtime, e.g., by forcing transitions and the robot can request help or confirmation from the operator, if required. Even complete modification of the behavior's structure is possible during runtime.

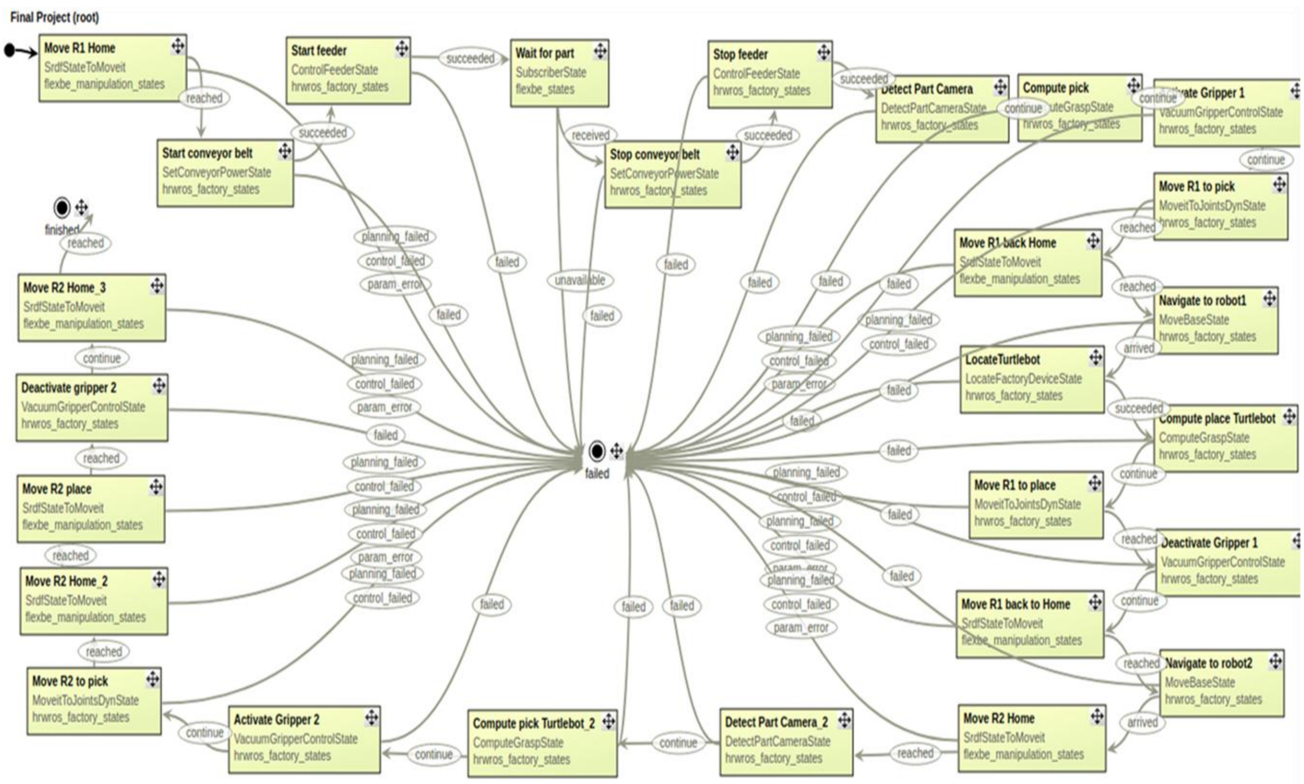


Fig.6 FlexBE States

Chapter 4
DESIGN AND IMPLEMENTATION

Our simulation consists of a robot (Turtlebot) and a 3D environment to replicate an industrial room with a conveyer belt and various objects. It also consists of a gripper and few cameras which helps in detecting and picking the objects. This simulation is done by using Gazebo.

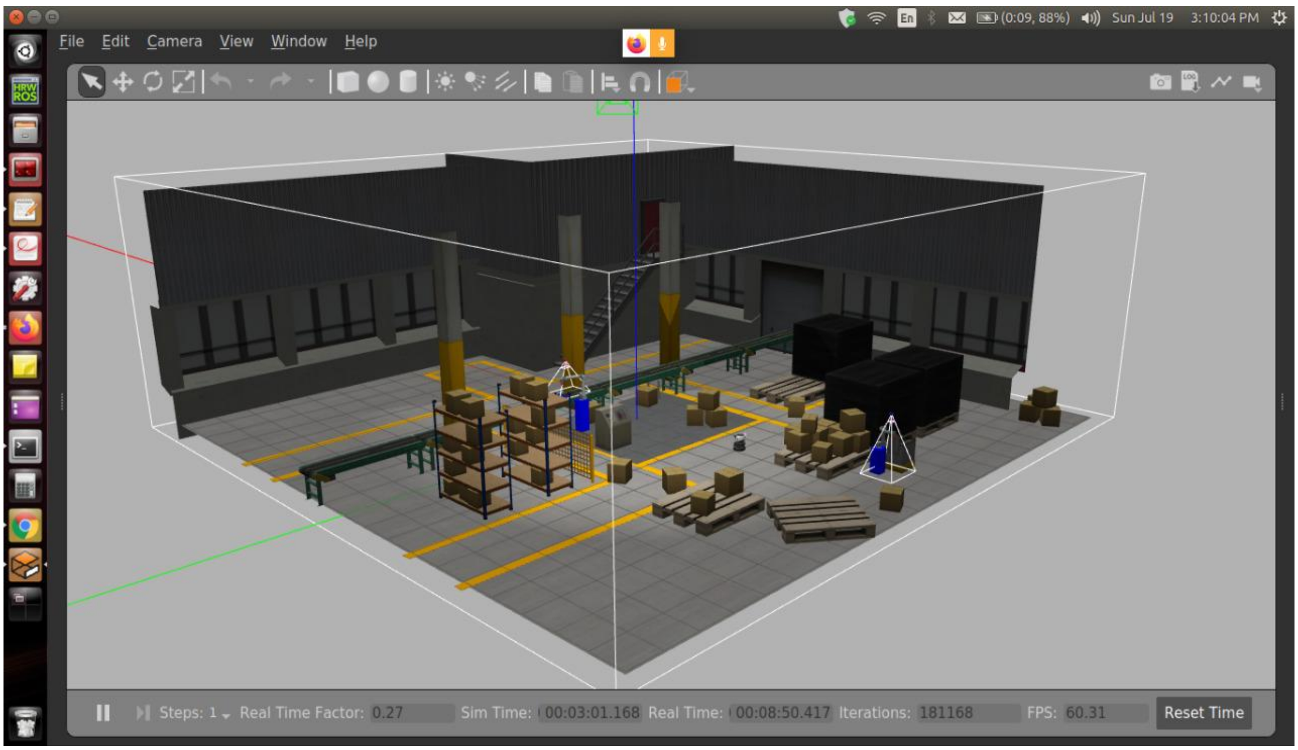


Fig.7 Model Design

4.1 Implementation

4.1.1 ALGORITHM - Room Mapping

The software of OpenVSLAM is roughly divided into three modules, as shown in Figure 5.2.1 tracking, mapping, and global optimization modules. The *tracking module* estimates a camera pose for every frame that is sequentially inputted to OpenVSLAM via key-point matching and pose optimization. This module also decides whether to insert a new keyframe (KF) or not. When a frame is regarded as appropriate for a new KF, it is sent to the mapping and the global optimization modules. In the *mapping module,* new 3D points are triangulated using the inserted KFs; that is, the map is created and extended. Additionally, the windowed map optimization called local bundle adjustment (BA) is performed in this module. *Loop detection, pose-graph optimization,* and *global BA* are carried out in the global optimization module. Trajectory drift, which often becomes a problem in SLAM, is resolved via pose-graph optimization implemented with g2o Scale drift is also canceled in this way, especially for monocular camera models.

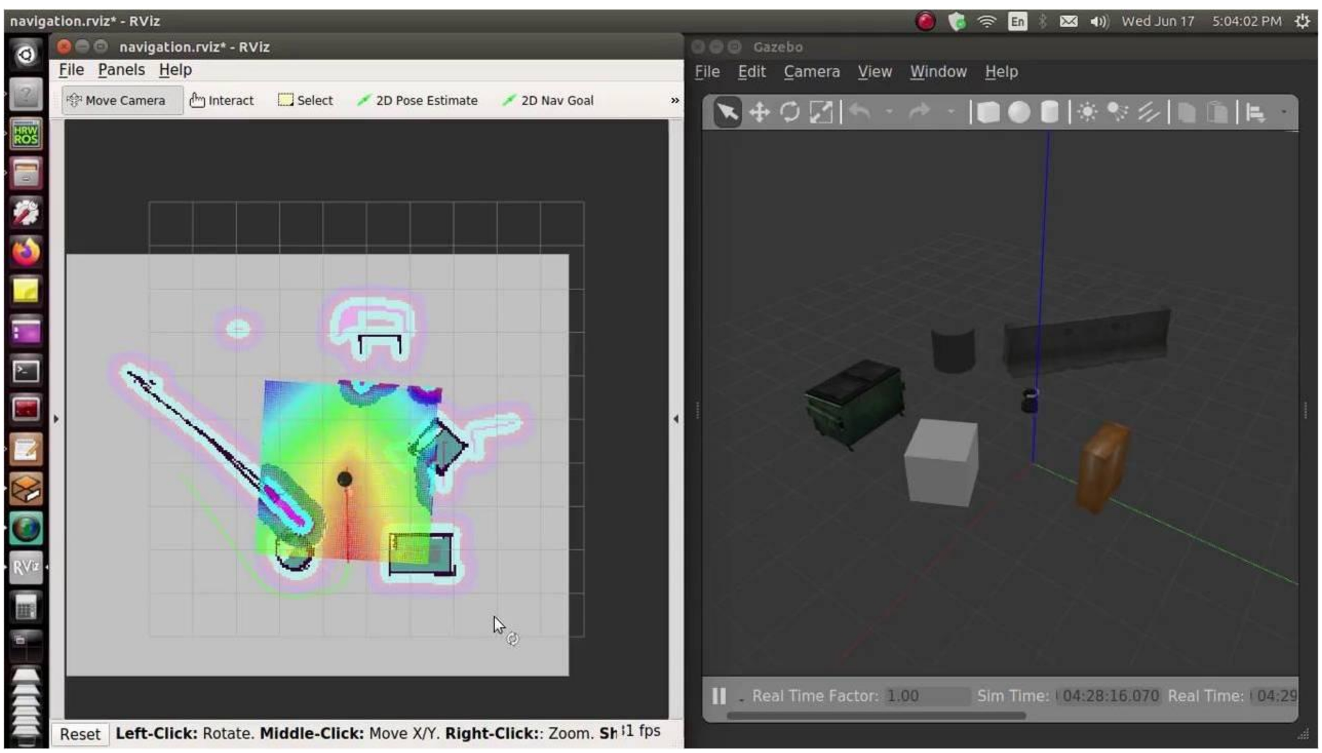


Fig.8 Room Mapping

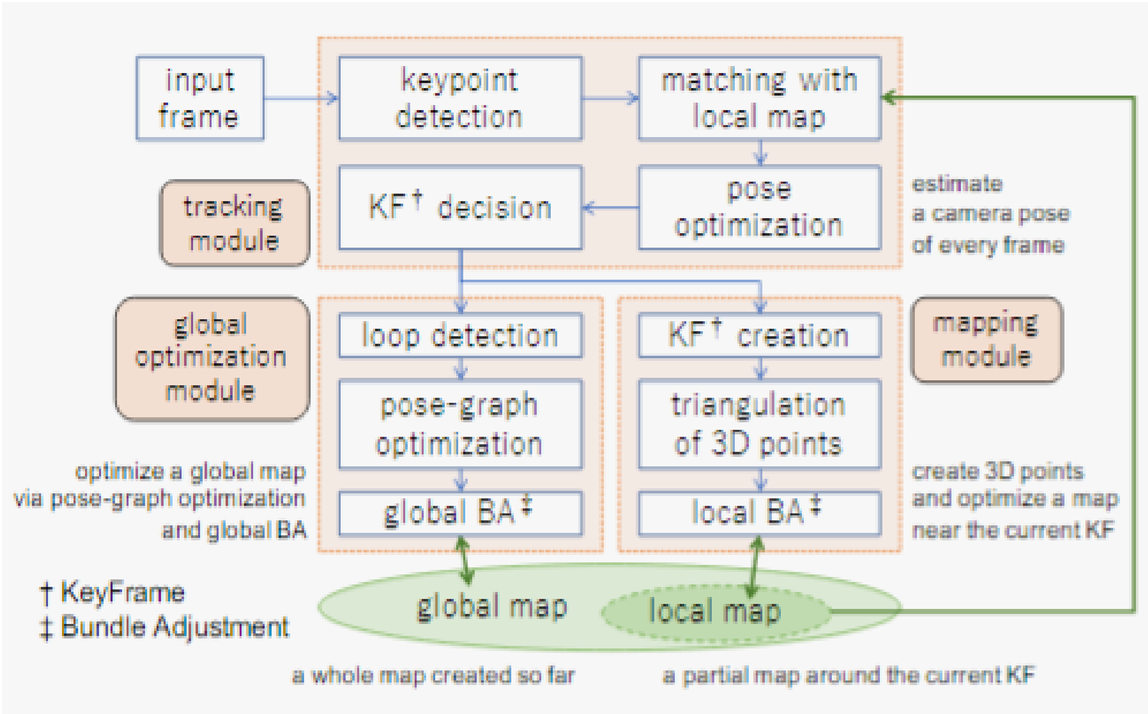


Fig.9 Flow Chart of Robot Navigation

4.1.2 ALGORITHM-Object Detection

We have used OpenCV and its built-in libraries for object detection. It contains algorithm for 3D object detection. Model algorithms are based on how humans perceive the visual data. It uses a logical camera (in our case) which is mounted on top of the gripper. Once any object comes in the field of view of the camera, the camera will capture the view of it and compares it with the stored database to identify the type and nature of the object. Once this is done and object position is identified, the gripper settings will be modified accordingly to pick the object without dropping or damaging it.

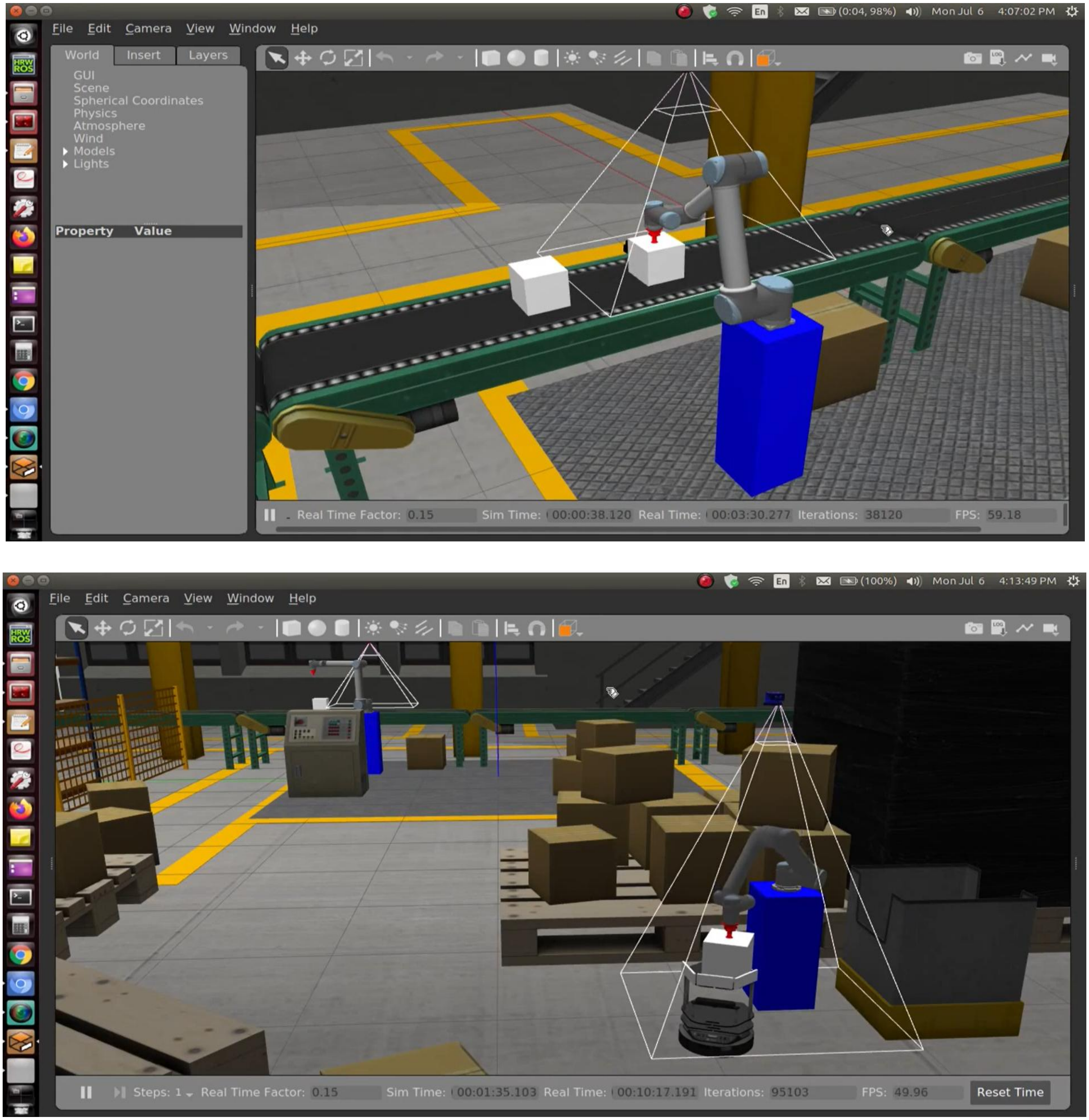


Fig.10 Object Detection

4.1.3 Simulation Using Gazebo

We have simulated our robot using Gazebo. We have used dynamic and kinematic physics, and a pluggable physics engine that are available in Gazebo so that the working of robot is realistic.

We have integrated ROS and Gazebo using the plug-ins which supports many existing robots like turtlebot and various sensors.

We have also created a 3D environment inside Gazebo to replicate an industrial room with various objects and a conveyer belt.

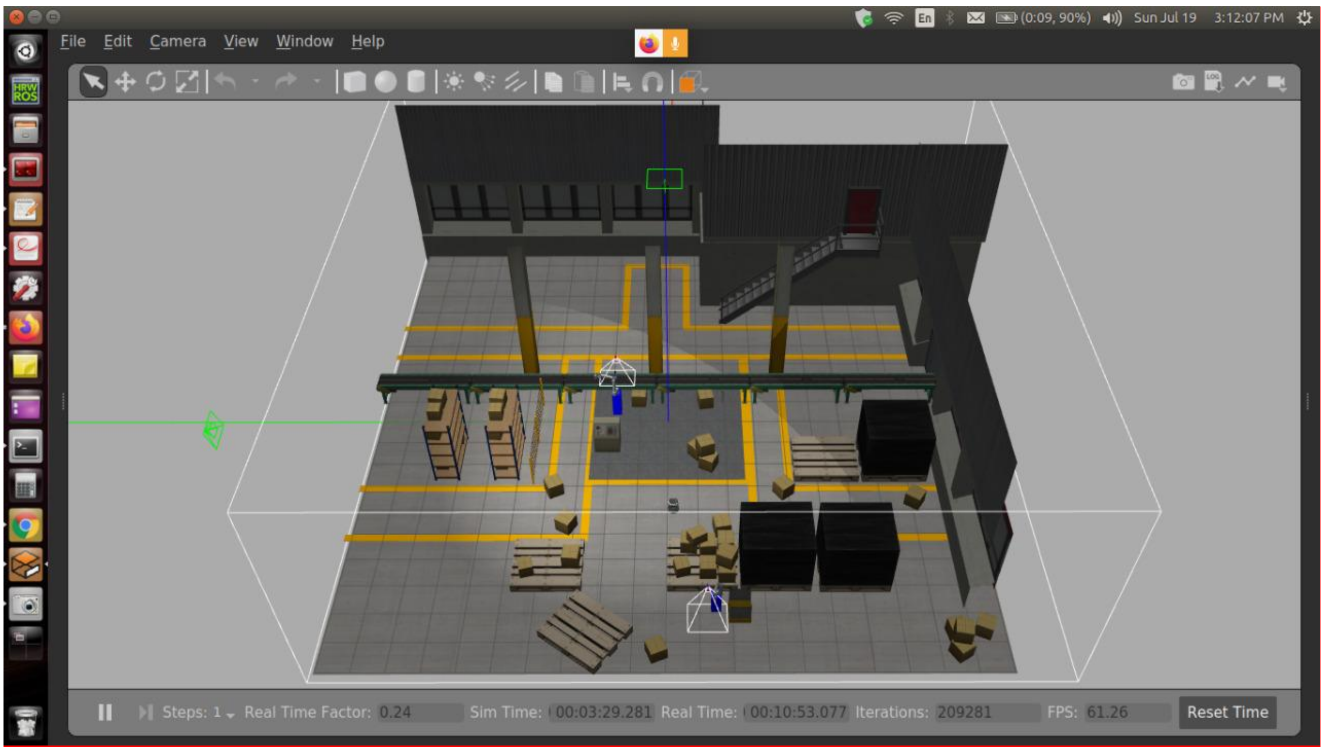
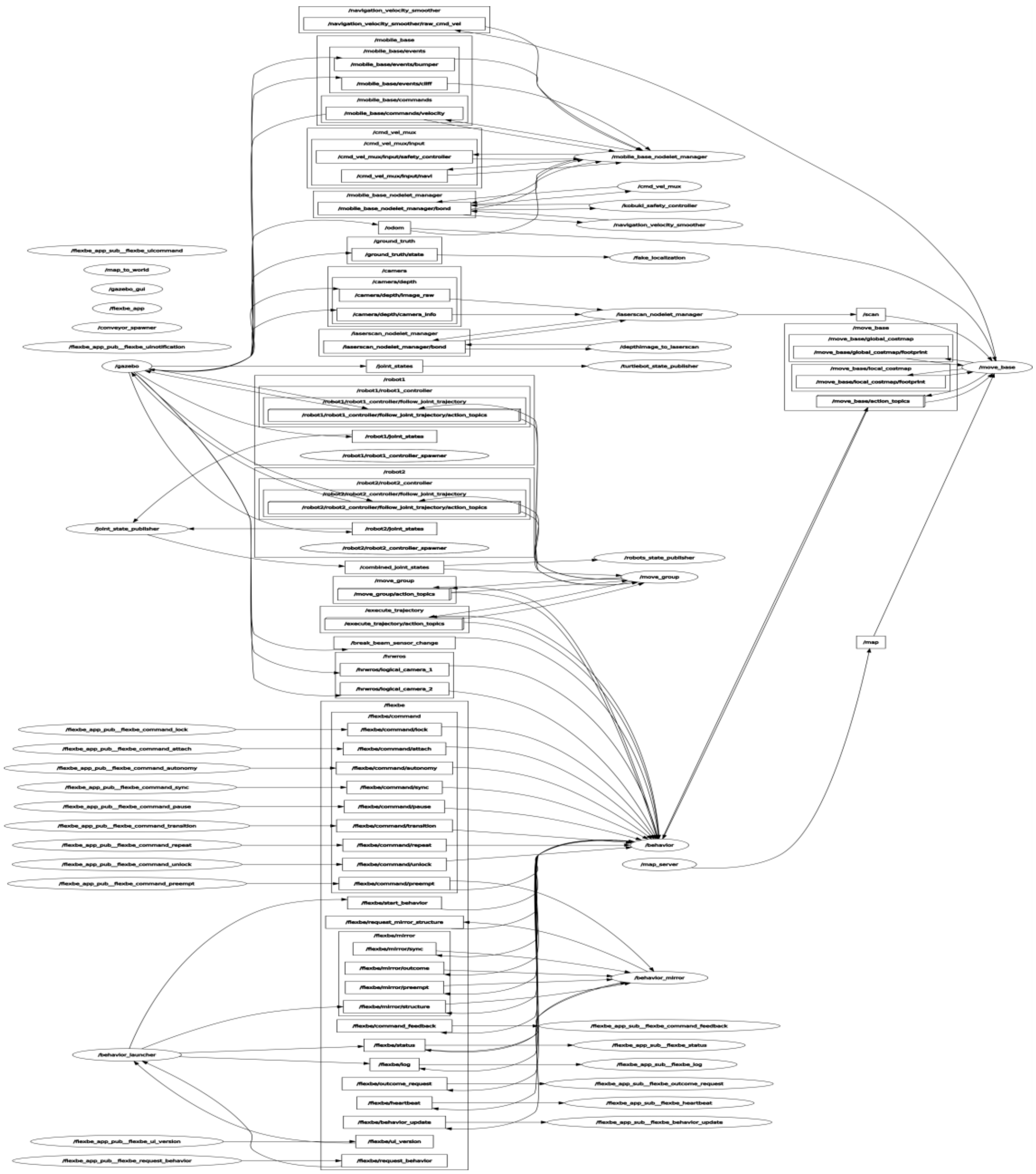


Fig.11 Simulation Using Gazebo

RQT GRAPH



RQT graph is a very useful tool to see what's going on in your ROS graph. It's a GUI plugin from the RQT tool suite. It gives a global overview of your system. This is really useful so you can take better decisions for the future new parts of your application.

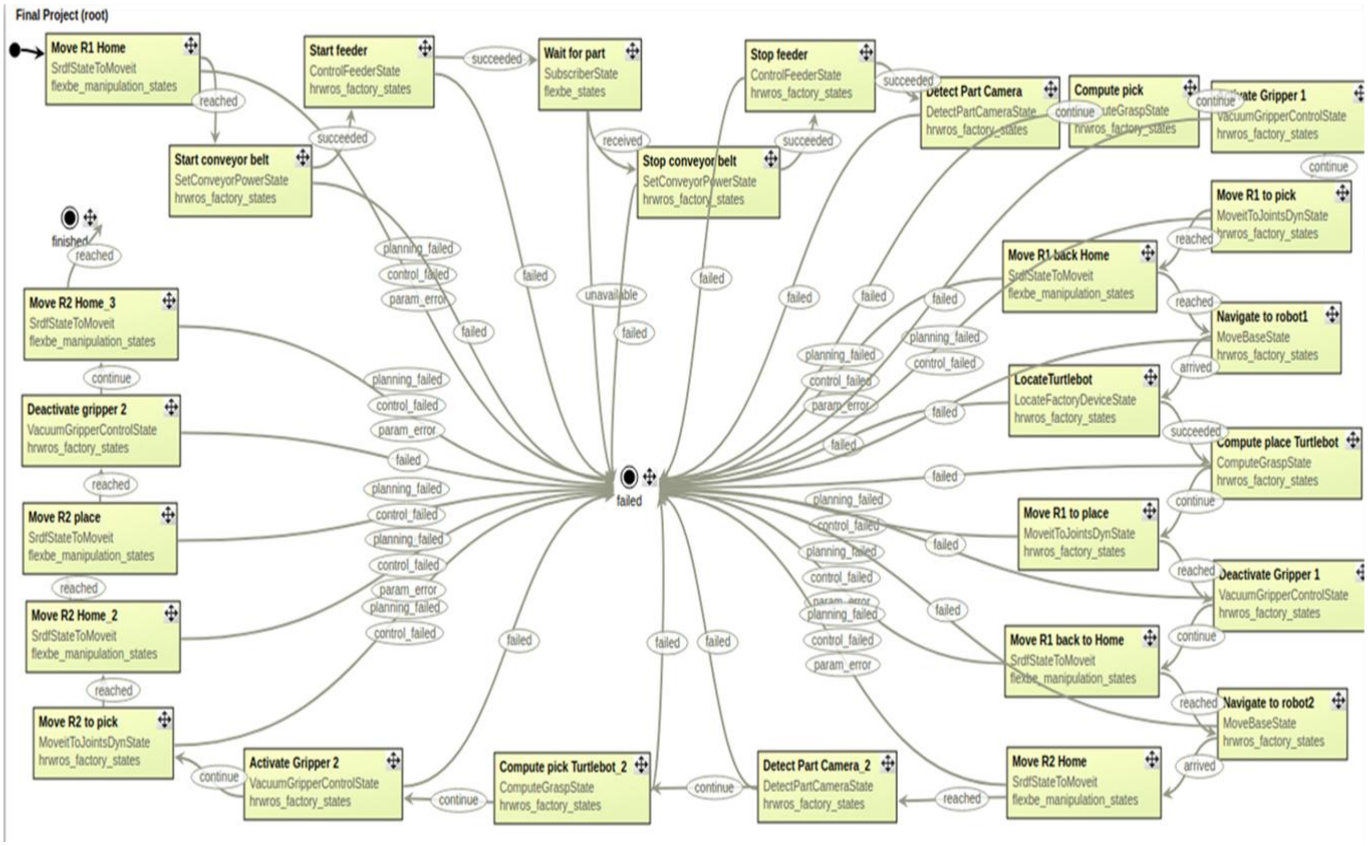
RQT graph have 4 main components which are Gazebo, FlexBE, main nodes and behavior.

FlexBE determines which node needs to be activated at what time and determine their action.

These nodes publish on topic and Gazebo subscribes them.

These nodes will perform certain actions and Gazebo will take live data from these nodes as input and give the certain output.

FLEXBE STATES



FlexBE States is a state transition diagram which is used to describe the behavior of robot.

Once the conveyer belt is started, it will indicate the system to start the feeder. Once the feeder is stared, it will wait for the object to arrive. As soon as the object is arrived and is detected by camera and the beam sensor, it will generate the command to stop the conveyer belt. Pick function will be activated which will command the gripper to pick the object from the conveyer belt. The gripper will adjust according to the size and nature of the object to avoid damaging or dropping the object while picking it up. Robot (R1) will be initialized to move towards the conveyer belt (PICK). Once the robot (R1) is at PICK, the gripper will place the object on the robot. After this is done, the robot will move towards the destination (PLACE). Once the robot reaches PLACE and is detected by the camera, gripper 2 will be activated and it will pick the object from the robot to place at the specified position.

The robot (R1) will be instructed to head back to it's initial position (HOME) and the grippers will be deactivated.

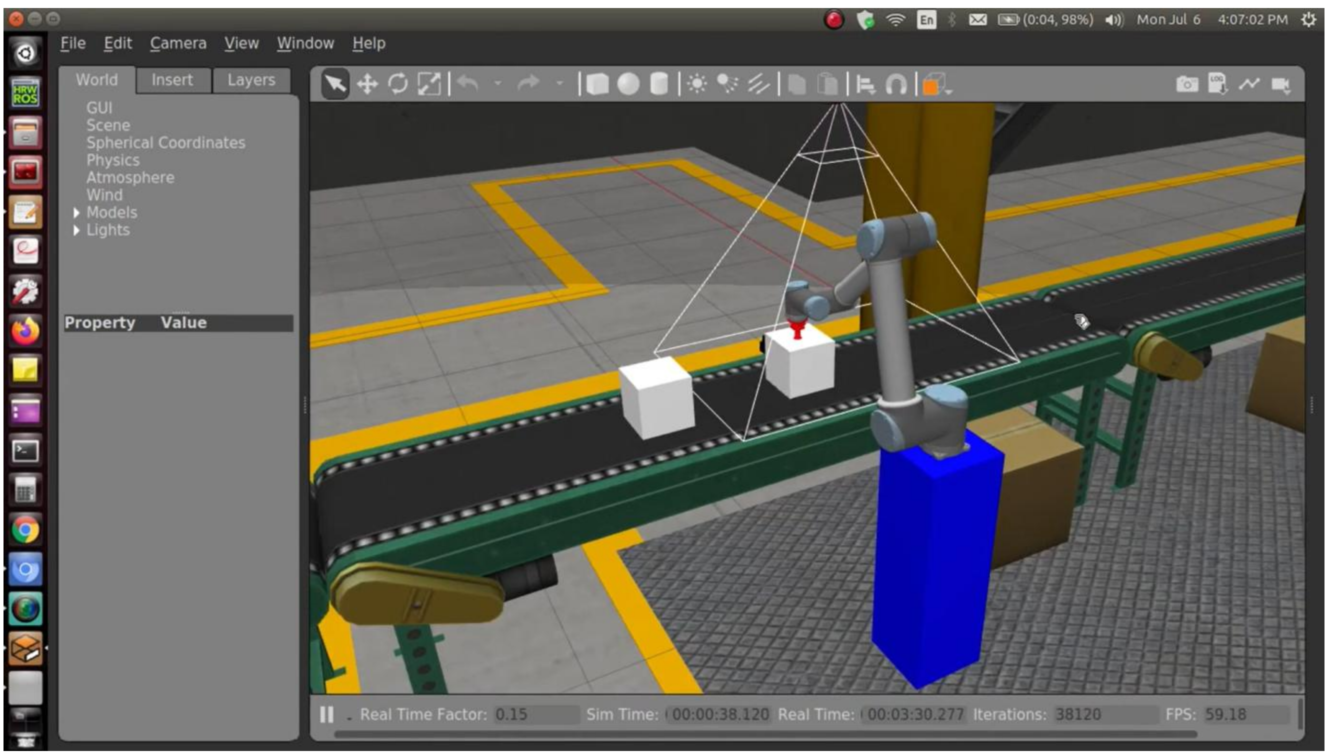
Chapter 5

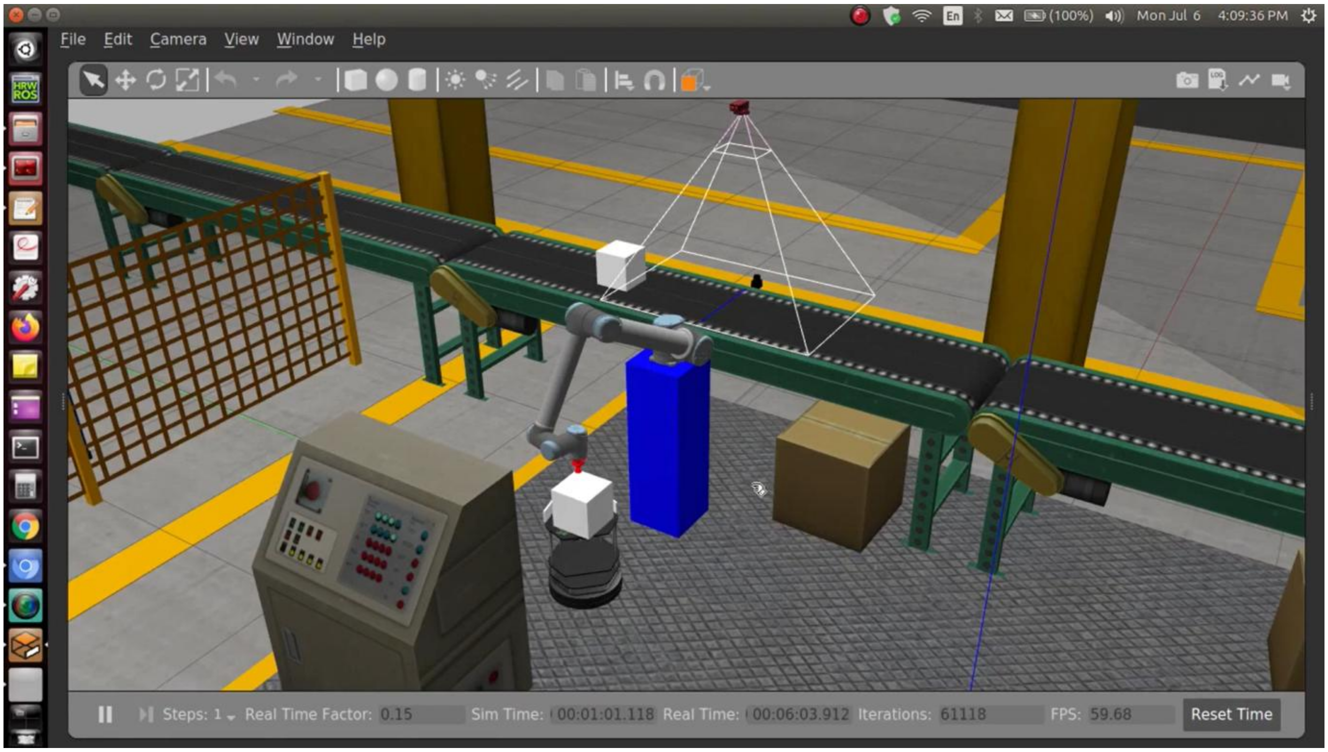
RESULTS

The robot is able to localize and develop a map for the environment and travel from point A to B . It is also able to detect the desired object and use a gripper to pick it up and move it from one position to the other. The movement of the robot's arm and body should be coordinated according to the readings given by the cameras.

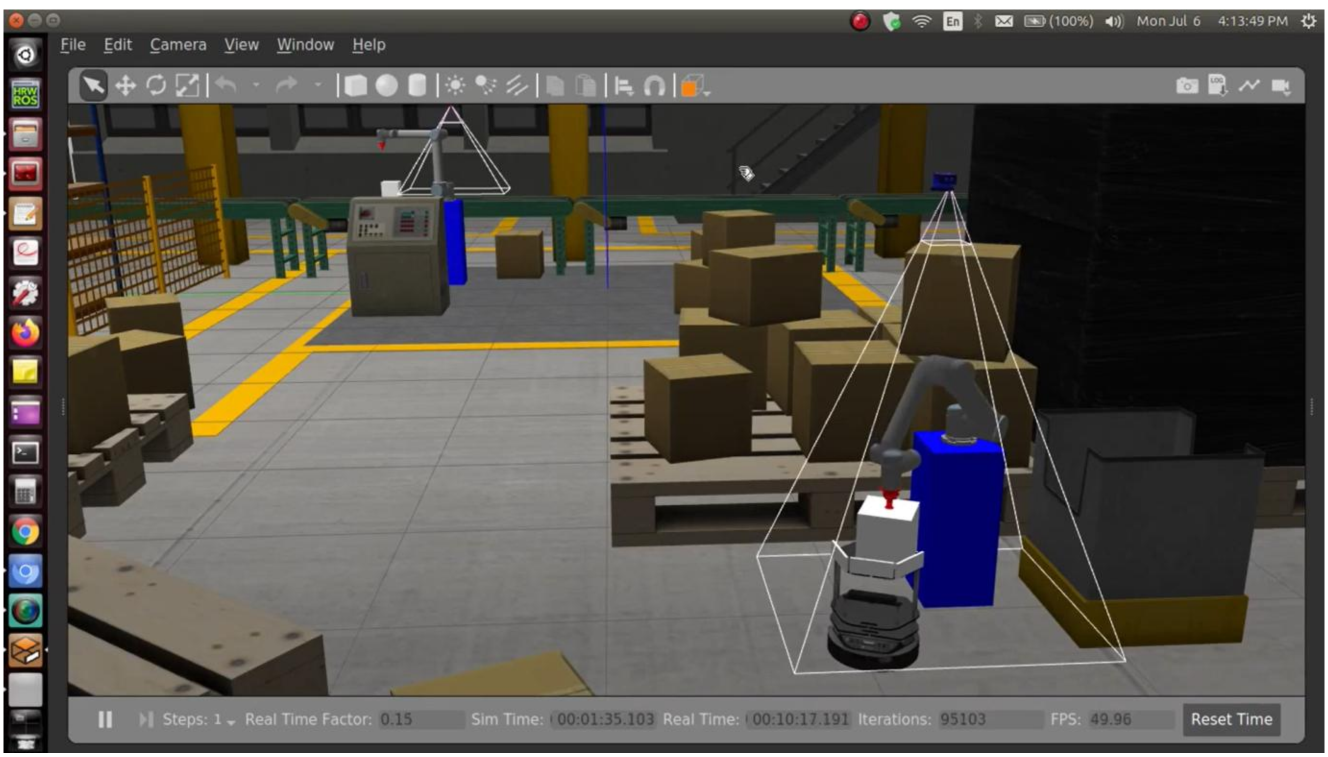
The robot is able to detect the landmarks in each frame that it captures using the camera. It then is able to measure and make note of the displacement in the area that it moves in, using this it plots a map of the area that it moves in, this has been made possible by the OpenVSLAM framework.

It is also able to collect information about the objects in the room and detect the desired object based on the database provided. This object detection is made possible by using OpenCV. Once the object is recognized, the gripper picks up the object, then moves it towards the desired location and then carefully drops it.





(100%) Mon Jul 6 4:09:36 PM
File Edit Camera View Window Help
Steps: 1 Real Time Factor: 0.15 Sim Time: 00:01:01.118 Real Time: 00:06:03.912 Iterations: 61118 FPS: 59.68
Reset Time



(100%) Mon Jul 6 4:13:49 PM
File Edit Camera View Window Help
Steps: 1 Real Time Factor: 0.15 Sim Time: 00:01:35.103 Real Time: 00:10:17.191 Iterations: 95103 FPS: 49.96
Reset Time

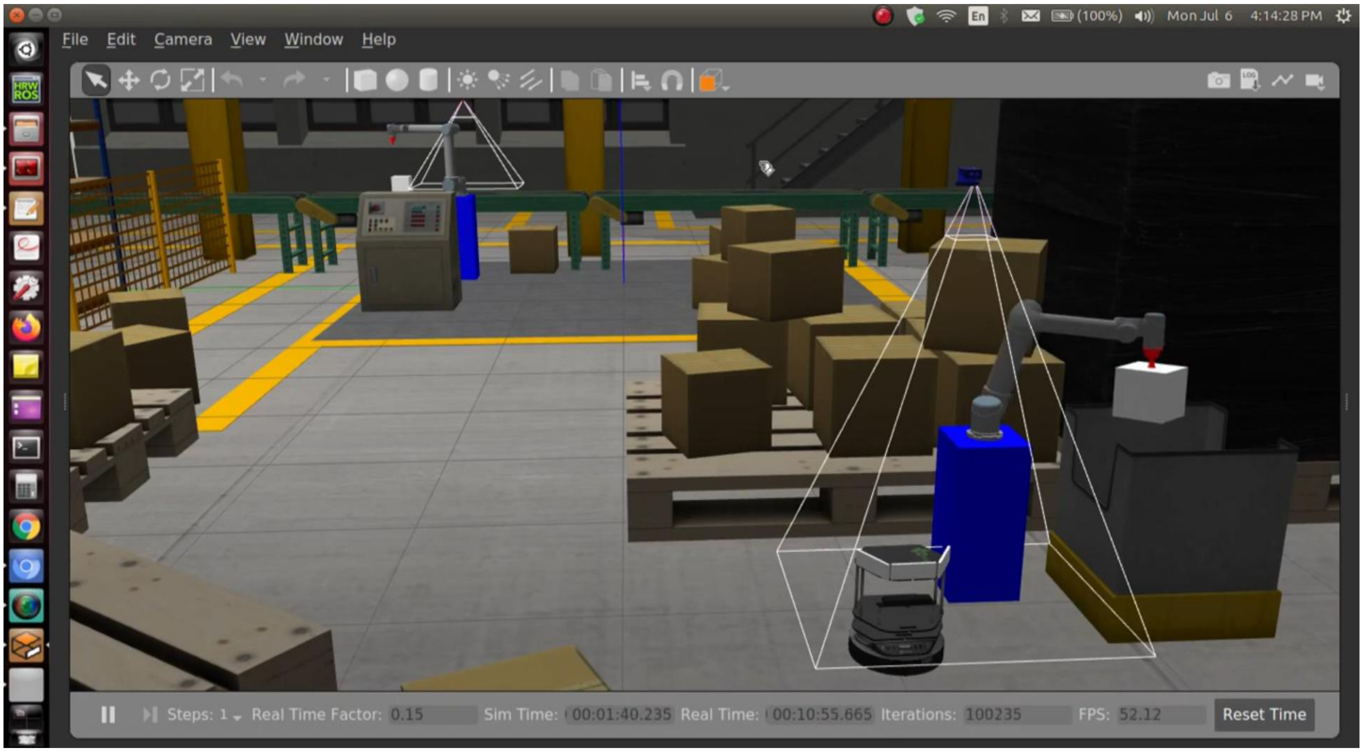


Fig.12 Result

Chapter 6
APPLICATIONS AND ADVANTAGES

This project has multiple and varied applications, but at the moment we are focusing mainly on industrial use.

- It can be used to pick up objects based on the need and the nature of the industry.
- It can move objects and various industrial equipments and parts from one place to another in the workplace.
- This robot will minimize the tedious and hazardous work and can help workers to maximize their productivity in other important work rather than picking and moving objects.
- It will minimize the human interaction which is the new norm these days.
- Without traditional guidance, these robots can navigate the industrial rooms, hallways, and can even find alternate routes when another is blocked. They will stop at pick-up points, and move the object to the desired place inside the workplace.

Chapter 7

CONCLUSION AND SCOPE FOR FUTURE WORK

8.1 CONCLUSION

The main aim of this project is to make the lives of industrial workers easy. Using a combination of mechanical, electronics and computer engineering we have come up with a robot that can be used in multiple domains, but in this particular scenario it is being used in manufacturing industries. It helps in maximizing the productivity of workers and helps the industry to grow more efficiently. It also minimizes the human interaction which is the new norm.

8.2 FUTURE WORK

The AGV (Autonomous Guided Vehicle) has the potential to be used in the medical industry to be used in nursing homes and in old age homes to be used by nurses and patients to move objects like medicines or water bottles etc. It can also be used to transport objects to people that are in quarantine and reduce the chance of infection spreading. There are chances that this robot can be used for military applications and even by our police for remote bomb deactivation. This can also be used as a personal robot and can be trained to do simple tasks such as getting the newspaper in the morning or even giving medicines to the family members at a specific time in the day.

There is scope for the AGV to be able to detect multiple objects and map multiple areas. The gripper arm can be modified according to the usage of the AGV and the size of the robot can be changed according to the load of the object it has to carry. A voice recognition feature can also be attached to the AGV, making it easy for individuals who find it difficult to operate mobile devices or disabled people who do not have the capability to operate the AGV manually.

Chapter 8

REFERENCES

[1] Design and Implementation of Robot Arm Control Based on MATLAB with Arduino Interface T.Rajesh, M. Karthik Reddy, Afreen Begum, D.Venkatesh. International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE) Vol 4, Issue 2, February 2018

[2] Design of mobile robot with robotic arm utilizing microcontroller and wireless communication - I.B.Alit Swamardika International Journal of Engineering and Technology (Volume 9 Issue 2, April 2017)

[3] Pick and Place Robot with Wireless Charging Application by N. Firthous Begum, P. Vignesh International Journal of Science and Research (IJSR) Volume 4 Issue 2, February 2015

[4] P. Velrajkumar, S. Solai Manohar, C. Aravind, A. Darwin Jose Raju and R. Arshad, "**Development of real-time tracking and control mobile robot using video capturing feature for unmanned applications**," 2010 INTERNATIONAL CONFERENCE ON COMMUNICATION CONTROL AND COMPUTING TECHNOLOGIES, Ramanathapuram, 2010, pp. 90-92