Grenze International Journal of Engineering and Technology, June Issue

AI-based Office Assisting Robot

Mahendra Salunke[1], Pratik Walunj[2], Atharva Pailwan[3] and Shreya Ginivar[4]
[1-4]PCET's Pimpri Chinchwad College of Engineering & Research, Ravet/Department of Computer Engineering, Pune, India
Email: {mahendra.salunke, pratik.walunj_comp2018, atharva.pailwan_comp2018, shreya.ginivar_comp2018}@pccoer.in

Abstract*—Although we mostly rely on digital platforms and tools these days, we still depend on people for sharing various objects like a document and similar objects from one place to another between employees of an unstructured indoor office environment. For this purpose, we are proposing a mobile AI-based autonomous Office Assistant Robot, which can be assigned various tasks such as sharing a document or similar objects between two people sitting at a distance from each other in an indoor office environment. The robot uses sensors for detecting obstacles. It uses a differential drive mechanism for moving in the office. It has a single onboard computer. We have used free, open-source software and cost-effective sensors and materials to build a low-cost but precisely functioning robot. Our system comprises four units: one that processes sensor data, one which helps the robot in navigating in a dynamic environment, one that controls part of the office assistant robot system, and the last is the mobile application which is used for communication with the robot. We have used Robotic Operating System (ROS) Melodic as the ground operating system in the background of the Ubuntu Operating System. The robot can move autonomously in the office environment with the help of the Simultaneous Localization and Mapping (SLAM) algorithm. The results showed that the robot was able to reach the goal location 99% of the time in the dynamic environment. Finally, we came to the conclusion that the robot could work very efficiently in a dynamic environment.

***Index Terms*— Robotic Operating System, Cartographer, Autonomous navigation, Office assistant robot, Rviz.**

I. INTRODUCTION

The world is rapidly growing in the tech and automation sector. The sole motive of the automation industry is to reduce human intervention and help people lead a convenient life. This is achieved through control systems such as computers or robots. We also have various robots which autonomously move between places. This makes navigation a very crucial part of such robots. In our research, we have designed a low-cost, autonomously navigating robot which will be able to deliver small objects such as documents between two people sitting far away from each other in an indoor environment. We have used Lidar, ultrasonic sensors for mapping and detecting objects, and the NVIDIA Jetson Nano as a single onboard computer. The SLAM algorithm has been used for mapping and navigation. The GUI is another important aspect of the robot. New users can register themselves through the GUI which is displayed on the LCD touchscreen setup present on the robot's body. The user's name, location, email, and photo are fetched through the UI and stored in a CSV format in the memory of the robot. The user can communicate and send requests to the robot through the Mobile application. We can easily send objects using the robot. Once the robot reaches the destination, the onboard camera will be used for detecting the face of the receiver. The locking system will only open after recognizing the person to which the

Grenze ID: 01.GIJET.8.2.20

parcel should be delivered. This protects the privacy of users and even confidential papers can be sent without any hesitation. This paper comprises various modules like autonomous navigation which helps the robot to navigate, face recognition to recognize the faces of users, speech response to interact with the user. User interface to register the new user, and mobile application for communicating with the Robot.

II. LITERATURE SURVEY

In the research paper [1] the author used iRobot Create 2 Programmable robot base which is a low-cost robot base that makes the robot capable of moving from one place to another. In order to navigate in an environment, we need some kind of sensor so that the robot is able to interpret the environment and act according to it. In earlier research paper [1] Primesense 3D sensor as the vision sensor was used as obstacle detecting sensor which helps the robot to detect the obstacle, but as the cost of the sensor is high so it is not suitable for use in low-cost robots in contrast LIDAR's are comparatively cheap and lightweight. For a robot to map the environment we need some kind of mapping technique that maps the environment, this map will be further used for navigation purposes. The mapping algorithm is used in a research paper [2]. This algorithm uses LIDAR in order to measure the distance between the obstacle and the robot. It also uses wheel encoders to measure the traveled distance and velocity of the robot which in turn tells the robot's position in the environment, but it was observed that the wheels of the robot slip on the floor which creates noise in the encoder reading and which decrease the accuracy of the robot in accurately mapping the environment, also it increase the need of adding more sensor which can compensate this noise and provide a reliable data. In a research paper [3] the author used cartographer mapping technique which uses LIDAR odometry which has no effect of wheel slip on the environment mapping, also there is no need of other sensors for sensor fusion and perform an intense calculation. In the research paper [1] we could not find any compartment which securely locks the document in it while delivering the document so there may be a risk of someone stealing the document or parcel. To implement all these algorithms efficiently there is a need for a system that helps in getting the sensor data processing it and giving respective output. To do this ROS was introduced in the research paper [4]. ROS has capabilities to take in the sensor data through the nodes and send it to other nodes using the pipeline called topics it is widely used for research in the field of robotics.

A. Gap Analysis

Considering the references Table 1 was created and it is observed that most of them uses Cameras, Lidar as a source of obstacle detecting sensor as well as for SLAM.

TABLE I. COMPARISON OF VARIOUS SENSORS USED

Sr. no.	Ultrasonic Sensor	IR sensor	Lidar	Camera
[1]				✔
[2]			✔	
[3]				✔

III. PROPOSED SYSTEM

We have made the robot frame such that it can easily maneuver around the indoor office environment. It has 4 wheels of which two of the side wheels have Dc motors integrated with it. Along with that, it has two caster wheels one at the front and other at the back which makes the robot balance. Fig 1 shows the actual picture of the robot and Fig 2 shows the custom designed base of the robot. We have used NVIDIA Jetson nano as a single onboard computer with teensy 4.0 attached to it through the serial interface. Lidar is used as a source of odometry and as a sensor that can detect obstacles. It can detect obstacles in a range of 0.5 to 8 meters in diameter. The robot has 2 ultrasonic sensors attached, one at the front and the other at the back. These ultrasonic sensors can detect obstacles that are smaller in size and cannot be detected by the lidar sensor. It also has an IR sensor mounted at the base facing downwards towards the floor to detect if there is no floor ahead in order to prevent the robot from falling in case the robot fails to follow its desired path. The motors are controlled by a motor driver who controls the speed and direction of the motors according to the signal from the teensy 4.0 microcontroller. Fig 3 shows the custom-designed PCB to connect all the peripherals and sensors in the manner shown in Fig 4. It has an LCD screen mounted at the top to show sensor parameters like battery percentage charging status along with that it is used to register new users, it has a 5mp camera which is used to recognize the face of the robot. It also has speakers to interact with the user with its speech capability, the robot has a 10000mAh 12-volt battery to provide power to the system, it has a digital voltage and current meter to show current consumption and battery voltage in real-time Fig 2, sometimes in case of emergency if we need to stop the robot, to do that it is provided with emergency stop button which can stop the robot and terminate all the robots operation. The software of the robot consists of the Linux operating system installed on NVIDIA jetson nano with ROS installed as a firmware to control all the sensor interfacing and other processing of the robot, in ROS the sensors are considered as nodes which broadcast sensor data to other nodes for processing. Ours system consist of nodes such as ultrasonic_front, ultrasonic_back, ir_front, ir_back, lock_on, battery_status, cartographer, transform, battery_charging, cmd_vel. Fig 5 and Fig 6 shows the diagram how various nodes communicate with each other.

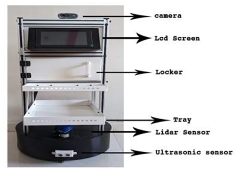

Figure 1. Robot structure

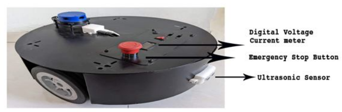

Figure 2. Robot base structure

Figure 3. Custom Designed PCB

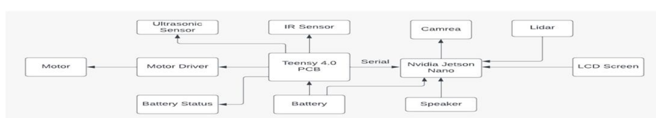

Figure 4. Block diagram of sensors and other peripheral interfaces

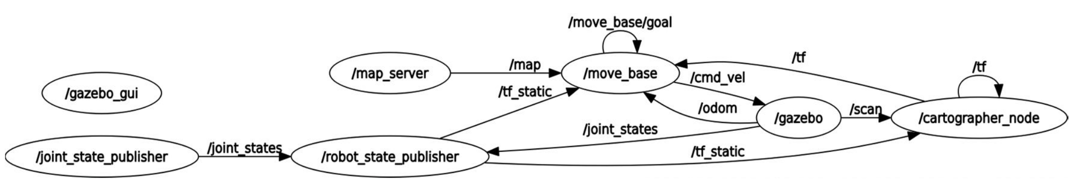

Figure 5. Nodes communication graph for simulation

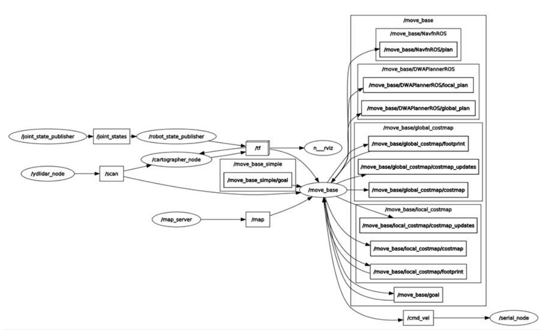

Figure 6. Nodes communication graph for real environment

V. METHODOLOGY

We have integrated various modules in our Robot which include face recognition, speech output, mobile application, ROS navigation, mapping the environment, all these modules communicate with each other to carry out the task of the robot.

A. Robot frame design

The robot frame is built using 1.2 mm sheet metal and the upper body is built with aluminum extrusion along with some 3D printed parts. The wheels are mounted to the chassis. The robot is 450 mm in diameter and 700mm in height. It has a secure compartment where confidential documents can be kept, this compartment will only open when the user is authorized user to take the document or parcel. It has addition two trays to keep more documents. The robot can carry a payload of maximum 10kg.

B. Building of a map

Initially, we have to launch certain roslaunch files to start building the map such as serverbringup, bringup, slam, and rviz to visualize the data. Here the robot is moved manually to cover the whole area of the floor. The cartographer is used as a mapping technique to build the map of this floor. Fig 7 shows the process of mapping in simulated environment.

C. SLAM

SLAM stands for simultaneous localization and mapping. Here, the robot's objective is to construct a map of an environment without prior knowledge of the environment. In our robot we have used cartographer as a mapping technique which uses lidar odometry to map the environment

D. Autonomous navigation

The serverbringup, rviz, navigation, and bringup node are used to start the navigation.
We need to pass the map to the navigation node so that the robot would consider that map a global cost map and perform path planning considering the map. Fig 8 shows autonomous navigation in a simulated environment.

E. Move base

The move base nodes perform the task of interacting with the ros navigation stack. The move_base node combines the global and local planners to perform the navigation. To move the robot to the desired location on the map, the user has to send the caller and receiver names to the robot through the mobile application. The robot matches these names with the names in the CSV file and gets the X and Y coordinates from it. Then the robot plans the path to the desired location and follows the path to reach the location.

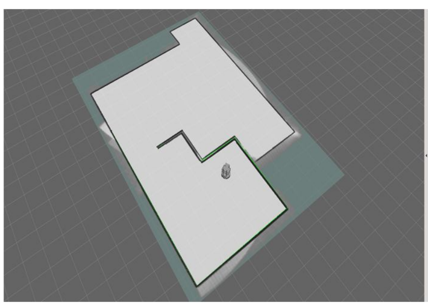

Figure 7. Making a map of simulated world

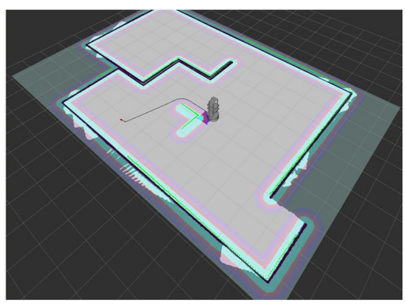

Figure 8. Autonomous navigation in the simulated world

F. Face Recognition

We have implemented the face recognition system in our robot for security purposes, basically, the robot will detect the face of the person and if the face is matched with the name only then the parcel will be delivered. For this purpose, we have used the ‘face_recognition’ library in python. ‘face_recognition’ is one of the powerful libraries for face detection. You can import the face_recognition module and then easily manipulate them. This library is built using dlib’s powerful and agile deep learning algorithms. It has an accuracy of 99.38%. ‘face_recognition’ library is easy to implement and takes care of automatically detecting and recognizing the facial features in the image. It recognizes faces in images and identifies who they are. Also, this library can detect a person by using a single image of that person.

G. Voice Interaction

As the prime way to interact with the robot is through voice, we have used the pyttsx3 library for communication. Once the robot reaches its destination it will greet the user according to the time. Pyttsx3 is an easy-to-use Python library. We have used this library to convert text to speech for interaction. Unlike other libraries, pyttsx3 works offline and is compatible with python2 and python 3 as well. Pyttsx3 supports various engines like sapi5, nsss, espeak.

H. Flutter

Flutter is a Google-developed toolkit for developing easy yet effective mobile applications on both iOS and Android operating systems. ‘Flutter’ is the framework and ‘dart’ is the programming language to create applications. To build any application one has to start with widgets. Widgets are the building blocks of flutter applications. The view in an application is described by the widgets given their current configuration and state.
Widgets are broadly categorized into two types Stateless Widget, Stateful Widget Fig 19 shows the custom-designed application.

I. Stateless Widget

While building a mobile application, one should know when to update the state of the application and scenario when it shouldn’t. Stateless widgets as the name suggests are used when the state of the application is not going to change. These widgets are immutable and any changes in variables won’t change the state of the app. When the displayed elements are static or are not going to change then stateless widgets are used. For example, when a UI is static then a stateless widget is used to create the UI.

J. Stateful Widgets

A stateful widget has a mutable state; that means it can change state according to user inputs or to any events. These widgets are used when there is any user interaction. For example, using a slider to take input from the user. A stateful widget is useful when the part of the user interface you are describing can change dynamically.

K. Firebase

Firebase is a backend-as-a-service platform that provides services like real-time database, authentication, and cloud storage (Firestore). Firebase is a NoSQL database that can be easily integrated with flutter applications.

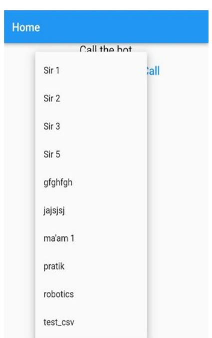

Figure 9. Mobile Application

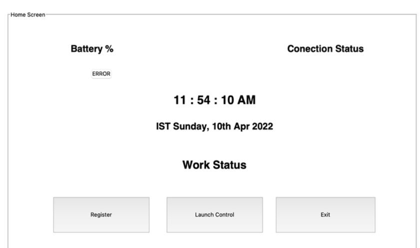

Figure 10. User Interface for Robot

L. User Interface

We have mounted an LCD touchscreen display on the robot’s body. A UI has been designed which will be showcased on this display. We have used the ’tkinter’ library in python for designing the UI.
It will help users to launch the robotic operations and new users will be able to get their registration done. Fig 10 shows the user interface.

VI. EXPERIMENTS

At first, we tried to implement face recognition using the YOLO (you only look once) algorithm. The YOLO algorithm works well when there is a large amount of data. But as we had to add this module to the robot system we faced an issue of the limited amount of storage and we opted for another algorithm 'face_recognition'. As the

face recognition algorithm works with minimal data input (a single image) it was inlined with our needs. We performed a test by giving the faces of 20 people out of which the robot was able to recognize all the 20 people.
We further experimented with the mapping technique. At first we used gmapping in which we used wheel encoders as a source of odometry and noted results. Then we experimented with the cartographer, in which we used lidar odometry and noted the results.

VII. Results and Outcomes

We selected an 800 sq ft room to perform autonomous navigation in which we carried out navigation 130 times of which the robot was able to perform autonomous navigation 128 times with an average of 0.5-meter error between the goal and the robot. This shows that the robot is able to perform autonomous navigation precisely with 99% accuracy by using the cartographer as a SLAM technique.
For face recognition as our robot was able to recognize all the 20 people, we got an accuracy of 100 % in it.

VIII. Conclusions

In this paper, we are introducing an AI-based Document sharing robot that uses the minimum and low-cost sensors to perform autonomous navigation. We also concluded that the gmapping technique with only a wheel encoder is not suitable on slippery floors and hence, the cartographer mapping technique is the best mapping technique for performing the simultaneous navigation and mapping. Our results to date show that simultaneous navigation and mapping can be done with high precision by using only lidar odometry.

Future Scope

As an extension to our work, in the future, the following features can be added:

- The robot can be controlled by using voice commands,
- After further programming upgrades and intensive mechanical integration, we could use the robot on multiple floors and throughout the campus.
- The robot can be used in hospitals as well, for giving medicines to patients on time, and also patients can call the robot anytime. Also, the medical reports can be shared via the robot.

Acknowledgment

The authors wish to thank Dr. Harish Tiwari (Principal) and Dr. Archana Chaugule (HoD) for their support and guidance. This work was supported in part by a grant from Pimpri Chinchwad Education Trust's Pimpri Chinchwad College of Engineering and Research, Pune.

References

[1] Isanka Diddeniya, W. K. I. L. Wanniarachchi, Naleen Chaminda Ganegoda. June 2019. Efficient Office Assistant Robot System: Autonomous Navigation and Controlling based on ROS. In *International Journal of Multidisciplinary Studies*. DOI: 10.4038/ijms.v6i1.p3

[2] Rajesh Kannan Megalingam, Chinta Ravi Teja, Sarath Sreekanth, Akhil Raj Department of Electronics and Communication Engineering, Amrita Vishwa Vidaypeetham, Amritapuri, Kerala, India. ROS based Autonomous Indoor Navigation Simulation Using SLAM Algorithm

[3] Cui Zhi, Shi Xiumin. October 2019. Research on Cartographer Algorithm based on Low-Cost Lidar. In *International Journal of Engineering Research & Technology(IJERT)*. ISSN: 2278-0181

[4] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng Computer Science Department, Stanford University, Stanford, CA Willow Garage, Menlo Park, CA Computer Science Department, University of Southern California. ROS: an open-source Robot Operating System.

[5] Yang Qian. September 2013. Design and Control of a Personal Assistant Robot. In *Ecole Centrale de Lille*. NNT: 2013ECLI0005

[6] Udo Frese, René Wagner, Thomas Röfer. September 2010. A SLAM overview from a users perspective. In *KI - Ku_nstliche intelligenz*. DOI: 10.1007/s13218-010-0040-4.