VOICE CONTROL AS A ROBOTIC COLLABORATION PLATFORM

Vázquez Limón, Luis Alberto[+]

Martinez Torres, Cesar[+]

[+] Universidad de las Américas Puebla

Corresponding Autor: luis.vazquezln@udlap.mx

Abstract

In this paper we exemplify how robotic collaboration can be applied using autonomous driving and speech recognition tools we can give users the opportunity to collaborate with a robotic system giving it objectives depending on the needs of the user. As the world evolves and industry 4.0 applications grow, the idea of making people able to collaborate rather than being replaced by robotic systems can modify the strategy and lead to a better and responsible industry.

For this work we applied different tools to show how the collaboration would work in a house environment. We used the python speech recognition tools as our human interaction portal with the robotic system and we combined ROS noetic and gazebo11 to simulate the robot for the human to give objective locations. The movement of the robot will be done using SLAM navigation integrated in Rviz connected to ROS for sending information and the simulation to receive location and scanner values. With the package correctly applied we show how the robot-human interaction is as clean as possible and how we can apply the same principle adding new tools or changing the autonomous movement algorithm if we keep the human interaction confined to voice commands.

Keywords: *Robotics, Autonomous Navigation, Collaboration, SLAM, Speech Recognition.*

1. Introduction

When talking about robotic systems, the way to control them has evolved depending on the application and objective of it. Lately the interest of making robotic collaborative has developed in many technological advancements in terms of kinematics, power input, degrees of freedom and obviously control parameters.

Currently, the understand of collaboration has changed a lot, the recent pandemic, and the better comprehension of robot teleoperation and home office made a breakthrough in the work environment were, automatization and dependency on robotic and similar components, are the most common work method. During this kind of change, is important to give tools to generate a real Human/Robotic collaboration, with the objective to make better application of the automatic systems and keep the human mind and abilities in every section of any process. With this we aim to prevent the replacement of humans in processes [1] [2].

One of the main parts of a real collaborative process is the communication, us humans have developed a language and different system to keep an efficient and effective communication between all the members of a community, team, or family. This communication was translated for communicating with the robotic devices, using programming languages and/or teach pendant to specify orders, configurations, and task to one or multiple Robots [3]. The main problem with the Human/Robot Communication comes from how specific the language can be, to learn to program the robots and manipulating them directly requires a good understanding of the programming language and the optimization parameters, this makes us need an expert in this to create a well-organized system that it cannot be modified in most of the cases which blocks completely the idea of collaboration [4].

In this project we are aiming to add a new communication tool to the Robotic Systems that give the opportunity to any kind of user to have a clear exchange of information and orders, this objective as simple as it seems requires two fundamental parts that are complicated on their own way. In the first place we must talk about the movement of the robot, the idea of moving the robot around in a 3D system using a simple communication seems interesting, when we talk about collaboration, we also aim for the robot to apply routines and also generate new one depending on the human necessities in that moment, in other words we want the robotic system to be autonomous and fulfill objectives with small or no human interaction like it can be seen on Smart Homes or Industries [5] [6].

Knowing the objective of having a real collaboration and understanding the abilities a robotic system must have for this purpose, we must now give an introduction about the theory concepts that are not only basic for the creation of our system, but that also are part of the tools we are going to use to implement the project. This theory part is going to be divided in 3 sections, first we will give an idea of what is robotics, and important information that are used in this kind of systems, next up we will give an explanation about the recent works in robotic collaboration. and the impact this project may have in this current situation. and finally we will talk about the tools we are using and how are they applied in this and other robotic projects.

2. Basic Concepts

2.1. Robotics and Collaboration

When we talk about robotics, we are referring to the science that oversees studying and comprehending the robots, a robot as described by Dr. Bob Williams from the Ohio University is: “an electromechanical device with multiple degrees of freedom that is programable to accomplish a variety of tasks” [7]. As described, a robot needs to have the capability to interact and move around the physical world, this movement is known as degrees of freedom (dof), a dof is a single movement the robot can make in our space, as an example the robot used in this project can move on a 2D plane which means it has 2 dof.

A typical industrial robot is built with 6 dof, these degrees of freedom are perpendicular and form what we call a robotic arm, which has a similar ability than a human arm in terms of being able to reach and orientate its end effector, there are also other multiple structures of the robotic systems which change depending on the dof and tasks that the system is built for. Different from the robots described before which are normally moving around in a fixed area, there are another kind of robots called “Mobile Robots”, these systems contain actuators that permits the movement around an area, like a house or a store and are not fixed, thus, they can move freely over the workspace.

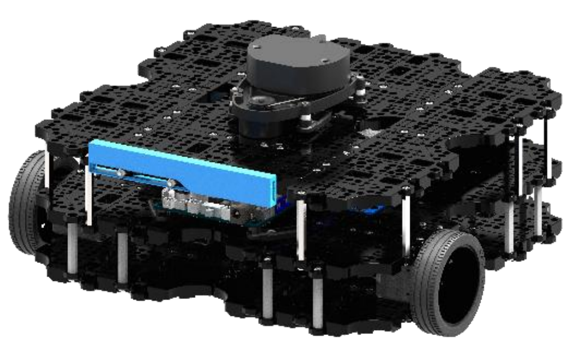


Figure 1 Mobile robot example Photo Obtained from Robot Advance webpage

Now that we understand the purpose and differences on structure of a robot and have an idea of what are the characteristics of the robot we are going to use in our word, we must go into understanding what do we mean with collaboration. As explained in the introduction of this work, the main idea of this project is to add an opportunity of real robot/human collaboration, exemplifying this in a simple robotic control system based on objectives and no routines, this collaboration requires certain characteristics that build an environment where is useful and reduce the effort made by the user and the wear of the robotic system, Arif Sirinterlikci and his team explains in their research: "Automation and Robotics in Processes", explains that adding robotic tools and automatization systems have as an objective to improve productivity, reduce costs, have safer conditions and making activities that are complicated for humans [8]. With this idea of the use of automation we can have an image of how the robotics systems are thought in the actual world which are seen as substitutes rather than improved tools which is the aim we look for in this work.

Before starting the explanation of collaboration and how can it be applied in robotics and human relationship, we must first understand what the part human and robotics system is play nowadays in our society. As explained, the use of robotics systems in daily tasks and works has become a popular solution to prevent human safety risk and improve the speed in repetitive processes, this introduction of robotics in what once was a human work, have opened a debate about the possibility of works being taken by robots leaving humans unemployed. [9].

The main reason people are afraid of robots controlling the labor market, comes from the way that we are managing human and robot interaction, robotic systems have been built with the purpose of act in a specific process, this makes it that the system is applied to the assembling line in the specific process and leaves human interaction apart making what we call a cooperation.

If we want to change the idea of robots being a substitute of human effort, we must aim to change the philosophy of human/robot cooperation and turn it into a collaboration. A collaboration requires for both parts to work at the same time in the same process, different from the cooperation, both parts not only share the resources nor make a division of the work but using both abilities and creativity to not only take part of a task, but also find better solutions and functions that can improve productivity [10]. This collaboration requires not only for both parts to share information and resources, also needs a retro alimentation of both parts, where one can help the other make a better work, there are multiple tools that need to be used to make robotic collaboration, but one of the most useful and the ones used in this work are ROS, Rviz and Gazebo.

2.2. ROS, Rviz and other tools

In this research we are aiming to give an example of how speech control for robotic collaboration can be applied, this means that instead of calling previously programmed routines we are going to use voice commands to give objectives to be reached by our system without any other human interaction. For making this example we are going to reach out for different tools, that alone can-do specific tasks, like human interaction and the autonomous drive system. The tools we are going to use are divided in the robotic tools and the interaction tools, as robotic platform we will use the next tools: ROS for robotic control, Rviz as an autonomous navigation and sensor visualization and Gazebo as a simulation system, before going into how and why are this tools useful, we have to give an explanation of each one of them and their applications as this will give us an idea on what use are we going to give them.

In this project we selected ROS or Robot Operating Systems (ROS) as our robotic control platform, as its names specifies, ROS is a middleware that simplifies the control of robotic devices by letting us create different nodes and topics to give direct orders to robotic systems from a host computer. There are multiple advantages in using ROS as its official website specifies, one of its main characteristics is the fact that all ROS system is open source, this mean that the use and modification of the system is completely free and possible almost without obstacles [11]. With ROS we can generate different scripts in c++ or python to use information from the robot sensors and send orders to change speeds in the motors or other movements, we also can call services and subscribe to the data management they do like Rviz and others [12]. In the end what we do is build a full launch file that calls the nodes, services and connections for our full robotic system.
During our work we used Rviz and Gazebo11 as our main tools, we choose them because they are compatible with ROS, Gazebo is a simulation tool where we can add objects and of course add robots with sensors and actuators that keeps sending and receiving information like a real-life robot would. Rviz in the other hand works as a data processing system, it uses the data from the robot sensors to identify objects and locate them using the current position of the robot giving the opportunity to do navigation.

Now that we have a basic understanding on what is the objective of this project and what are we aiming for, and most importantly we have and understanding of the tools we are going to use and some examples of how they can be applied, we must move on to the main part of this article which is the development of our project. As explained, we are aiming to demonstrate how the robotic collaboration can be applied using an example of navigation inside an environment by calling different objectives using voice control.

3. Methodology

When making the development of this project, we had to divide the task into multiple and smaller ones that permit a more focused effort and comprehension of the tools and objectives. We defined our first task as the understanding of the capabilities of ROS as a controller, we tested different nodes and built our own teleoperating system using voice commands, then we can use Gazebo as a simulation environment to test what we learned in the first step and see the reaction of the robot, the third task will be to combine Rviz and use the already known navigation algorithms, to test the combination of ROS with both tools, in the end we will combine what we learnt to build our voice controlled navigation to show in simulation in gazebo how and why does it works.

During our work in this project, we generated a solid base in the comprehension of ROS and the other tools we mentioned, this comprehension started from building a simple voice control, using speech recognition tools and Google API. These two small projects were fundamental for the first part of the project, as we also used Gazebo to show how the simulation sends and receives information and target the difference with the real world for when the project is applied physically.

When we understand finally the information sent by the robot in the simulated world, and how it reacts to different kind of inputs, we can use this information to build our SLAM navigation model using the scanner data and the robot position for making the navigation after we build the map of the area. As said, we need a previous built map, this map is generated moving the robot around the area and saving the reads of the Laser built on top, then we can navigate knowing the starting position of our robot, and a given objective position, the robot can create a route to reach it while also using the scanner to see its surroundings and obtain a better route as it is needed. Is important to point that even when this document dive in the navigation using SLAM, the system is supposed to work with any kind of navigation if objective input can be defined by the user.

The last step of this project is developed when we understand finally how a navigation node is built, and the importance of each of the services that the node has to be subscribed when using ROS, we can add the interaction between our user and our system, for this as explained before we want to use the user voice to give objectives to our robot automatic system, as it can be applied in different environments, the voice control needs us to recure to a Speech to Text system, that is able to turn any audio signal received to the computer into a speech that is closer to the audio received, for this purpose we have to access to a database and try to find the better match to the audio received. This step can be divided in two parts, first we will apply our system to the first node we made, which is a teleoperation node, and see how this will work in the simulated robot, after this we will apply our speech to text system to send objectives to our navigation node as intended from the beginning of the project.

4. Results and Analysis

4.1. ROS voice control package

For testing this package creation, we are going to build our own simple control package, this control package requires us to program a simple node, connecting to the robot sending a topic, specifically we want to send a topic that is named as *"/cmd_vel",* this topic is an input channel for the robot that let us send a change in the speed of the servos of the robot. In case of our robot, we have 2 servos which generate a movement in a 2D plane, if both servos move at the same positive speed they will move forward, if the speed is negative the movement will go backwards but if the speed is different, it will go left or right.

To give the opportunity to users to send this speed changes to the robot, we are going to write our own script in python where we will expect an audio input, using the speech recognition tool of python connected to google speech to text API, this will return a string value of the word said into the microphone. The speech recognition system takes our voice input and send it to google speech to text API, here the neural network defines the word is heard in the audio and returns a string variable with the word.

With the word now in a string value, we can use a simple switch case to compare with the different orders we expect and send the "/cmd_vel" topic with the speech change that user specified, like going forward or backwards. As part of the system, we give instructions to user that the microphone is ready to receive the next input, to give a visual confirmation to the user when it can give next order, also we repeat the last word user said in text, so the user can see it was heard correctly.

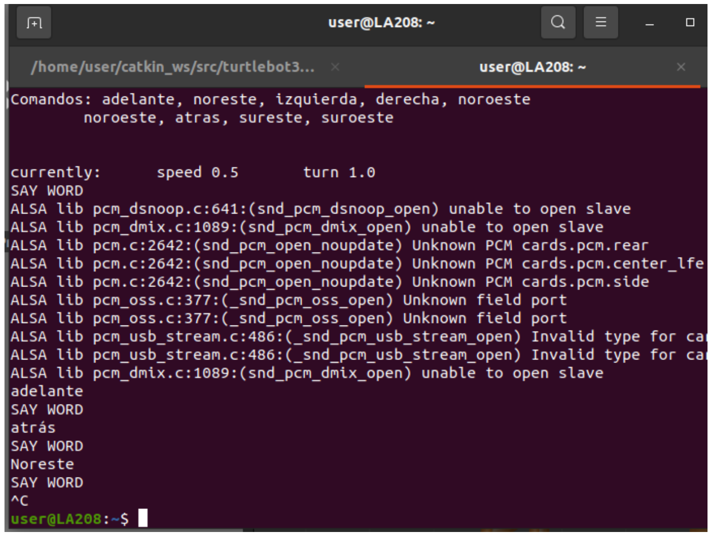


Figure 2 Command Window Speech Request

The instructions sent with voice in Figure 1, must be received by the robot whether is physical or simulated, in this work we focused on the simulation using Gazebo latest version (11.0). In gazebo we can generate a simulated world with various objects positioned where we need them, we can also retrieve different robot models to simulate inside gazebo, in this case we will use the waffle pi robot created by Robotis. The waffle pi robot is a planar robot that has 2 main sensors attached to it, a frontal camera, and a LiDAR on top, both sensors are simulated in gazebo, and we can access to their information using ROS. When the robot and objects are correctly declared in the gazebo launch file, we can execute it and will have the window shown in Figure 2.

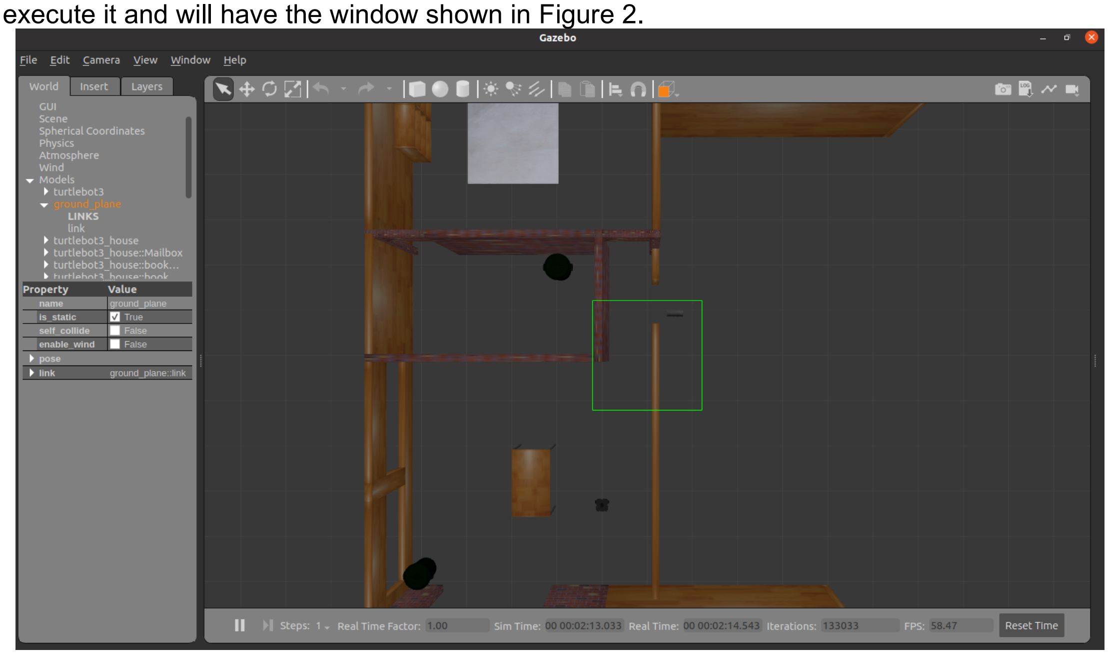
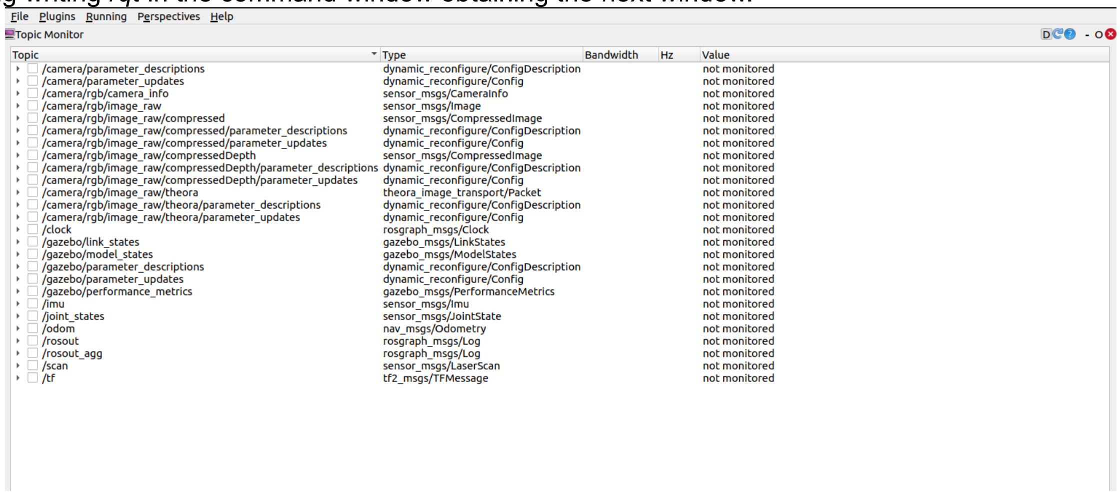


Figure 3 Gazebo World House Environment

As we can see the robot and the elements declared to build the house appear in the 3D view of the window, also in the right-hand side, we can see a list of the elements that conform the simulation that include the models of the walls and furniture, this also includes information of the robot and the links and position of all the parts that conforms the robot. As mentioned in the methodology, gazebo's robot simulation is also capable of sending simulated information from the robot, we can see this information using writing *rqt* in the command window obtaining the next window.



Topic	Type	Bandwidth	Hz	Value
/camera/parameter_descriptions	dynamic_reconfigure/ConfigDescription			not monitored
/camera/parameter_updates	dynamic_reconfigure/Config			not monitored
/camera/rgb/camera_info	sensor_msgs/CameraInfo			not monitored
/camera/rgb/image_raw	sensor_msgs/Image			not monitored
/camera/rgb/image_raw/compressed	sensor_msgs/CompressedImage			not monitored
/camera/rgb/image_raw/compressed/parameter_descriptions	dynamic_reconfigure/ConfigDescription			not monitored
/camera/rgb/image_raw/compressed/parameter_updates	dynamic_reconfigure/Config			not monitored
/camera/rgb/image_raw/compressedDepth	sensor_msgs/CompressedImage			not monitored
/camera/rgb/image_raw/compressedDepth/parameter_descriptions	dynamic_reconfigure/ConfigDescription			not monitored
/camera/rgb/image_raw/compressedDepth/parameter_updates	dynamic_reconfigure/Config			not monitored
/camera/rgb/image_raw/theora	theora_image_transport/Packet			not monitored
/camera/rgb/image_raw/theora/parameter_descriptions	dynamic_reconfigure/ConfigDescription			not monitored
/camera/rgb/image_raw/theora/parameter_updates	dynamic_reconfigure/Config			not monitored
/clock	rosgraph_msgs/Clock			not monitored
/gazebo/link_states	gazebo_msgs/LinkStates			not monitored
/gazebo/model_states	gazebo_msgs/ModelStates			not monitored
/gazebo/parameter_descriptions	dynamic_reconfigure/ConfigDescription			not monitored
/gazebo/parameter_updates	dynamic_reconfigure/Config			not monitored
/gazebo/performance_metrics	gazebo_msgs/PerformanceMetrics			not monitored
/imu	sensor_msgs/Imu			not monitored
/joint_states	sensor_msgs/JointState			not monitored
/odom	nav_msgs/Odometry			not monitored
/rosout	rosgraph_msgs/Log			not monitored
/rosout_agg	rosgraph_msgs/Log			not monitored
/scan	sensor_msgs/LaserScan			not monitored
/tf	tf2_msgs/TFMessage			not monitored

Figure 4 Topics sent by the simulation

As shown in Figure 3, the information sent by the simulated robot consist of different topics that contain the robot data, these data can be the servo speed, the odometry of the robot, marked as *"/odom",* and the information retrieved by sensor, named as *"/scan",* that contains a matrix with the values of the LiDAR scanner in top of the robot, this data represents the distance that from the robot to any close object and can be used to understand the robot surroundings.

When the simulation in gazebo is active, we can call the speech teleoperation node we previously created, and the topics sent by this node (/cmd_vel) will modify the speed of the simulated robot depending on the user input. This is an example of how powerful ROS as a tool is, letting us make a cleaner human/robot connection that other communications cannot give us, like teach pendants or other control units like PLC´s. Now that we understand how topics are sent using user inputs, we must finish with the autonomous part of the collaboration, for this we will reach out to the SLAM navigation algorithm and apply it to navigate around our simulated house using speech inputs for retrieving objectives.

4.2. SLAM Navigation and Mapping

The navigation system we are going to apply is SLAM, this requires us to access the *"/scan"* and *"/odom"* topics to retrieve information of the area. The SLAM method works in two parts, the first one is a mapping of the area, moving the robot around retrieving as much information of the place as possible, for this purpose we use the Rviz tool, which is a tool that let us visualize the position of the robot and the objects the sensor, in our case we will use the LiDAR info saved in the *"/scan"* topic. The mapping section of the SLAM method needs us to move around the robot slowly around the environment, to find all the obstacles and walls around, when the map is done, we can save it as a yaml file to be access by the navigation node.

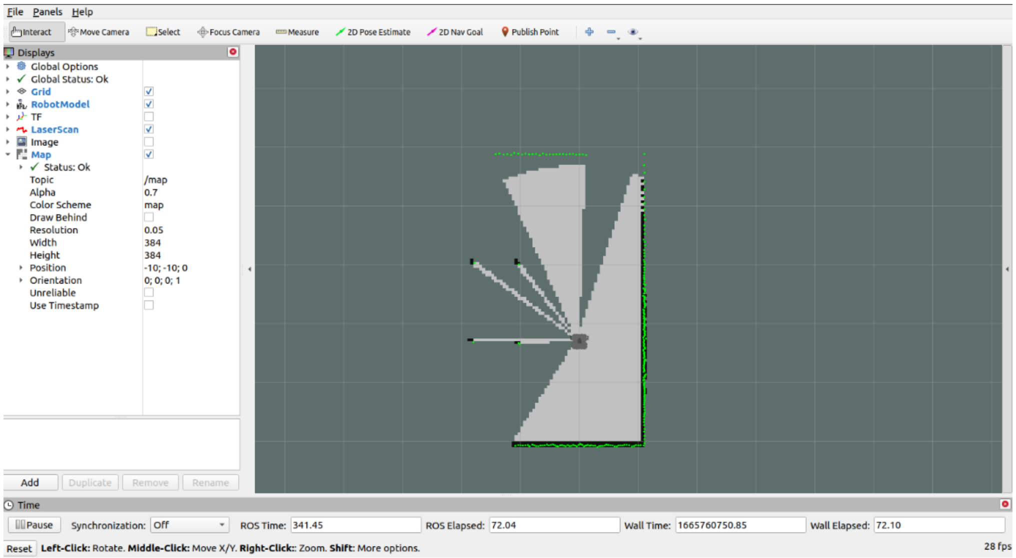


Figure 5 Map Generation using SLAM

With the map built, we can again access to rviz opening this map and calling the robot info, the Rviz window we are going to call in the navigation section includes two tools, the estimated pose and the Nav Point markers, according to the research of Huishen Zhu in SLAM navigation, this is done by following 3 steps:

Localization: This step needs us to use the odometry and sensors of our robot to identify not only a general position but a localization at the already generated map

Planning: The algorithm used for navigation permit us to replan the route all the time while it is being applied. This path generated is done by using the map information and with feedback of the LiDAR and odometry values of the robot, this gives us the chance to replan the route when an unexpected obstacle is found [13].

Route Following: When we have the path already generated, we must translate it into servo velocities that generate the estimated movement, there is a problem with this when applied in the real world as the system can have error for uncontrollable variables like the smoothness of the terrain, the slope and other changes that can vary the speed that the route needs [13].

Normally to give objectives to the SLAM algorithm in ROS, we use the Rviz defined by user marker Nav Point, this marker sends the order to generate a route from the current position of the robot to selected point, in our case we want to prevent this screen interaction to make it more direct with the user, this needs us to define previously known objectives that we can call, this objectives have to give the x, y, z position and the orientation we want the robot to have, in our case we grab the map in gazebo and marked 4 possible objectives that can be useful for user, we can see them in Figure 5

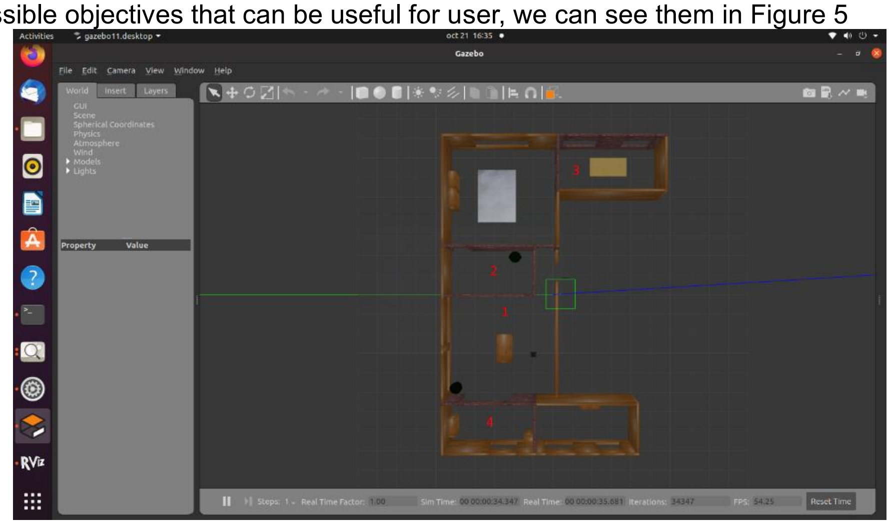


Figure 6 Map with marked objectives

In Figure 5, we can see marked as number 1 the kitchen of the house or what we can call our base, in number 2 we decided to call it bathroom and number 3 is a guest room finally number 4 can be seen as the living room. Each of these objectives have specific coordinates that we had to define, this also means we can create, modify, or eliminate points as we need it depending user needs and tasks to take part on. In this case we selected these points because they can give us different challenges in terms of route planning as it can have sharp turns or long distance to travel. Each one of these points need to be declared with a single key word, to make it easier for the user to call them when needed also, because of the navigation algorithm, all the declared points need to be inside our previously Rviz generated map, this is because the system will never be able to reach any point outside the map.

To understand better our system, we built a diagram of how all the tools and data are connected between each other. This diagram explains the full process, starting with the speech order of the user, giving as a result the robot movement that aims to achive the given objective. The tools we connected between each other are: The speech recognition python's tool, the Google speech recognition API and the SLAM navigation algorithm.

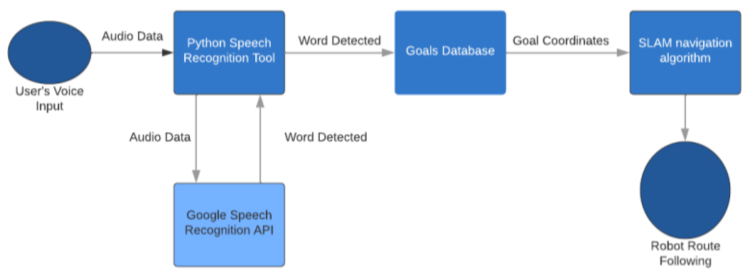


Figure 7 Full system Diagram

In Figure 7 we can see marked as circles the physical actions of the robot, the dark blue squares represent host computer processing and the light blue square represents external processing. The system is fairly simple, we start with the user sending audio data to the speech recognition tool, this data is then sent to the Google API and it returns the detected word, the detected word is compared to our goal database looking for a match, when we find a match we extract the coordinates of the objective, the SLAM algorithm then uses this info to create a route, in the end the robot follows this route until we reach our objective.

The SLAM navigation algorithm has its own work diagram that connect the info received by the robot, and the route generation techniques for the system. If we want to see how this diagram looks like, ROS provide us a powerful tool that shows connections between the topics and messages that are applied in the robot, at the time, we can call it using the *rqt_graph* command, in our system we will have the next connections.

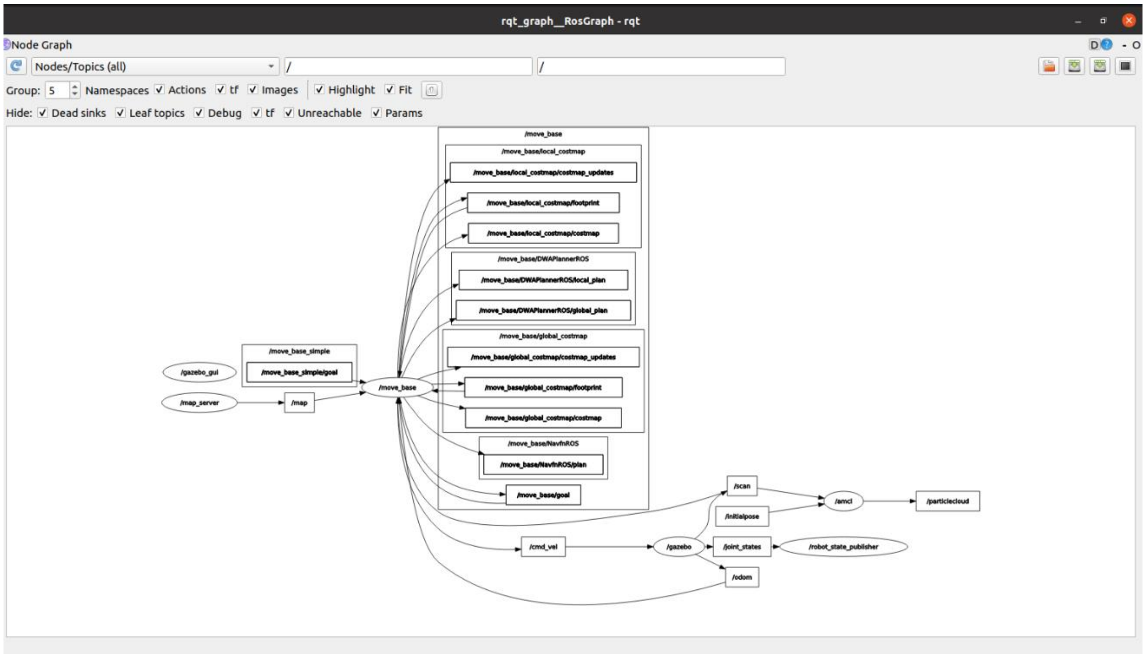


Figure 8 Connection for the Navigation System

In figure 6 we can see how gazebo has all the topics we use for the navigation, like */scan, /joint states* and */odom,* the */scan* topic is then combined with the initial pose topic created in Rviz to build the */particle_cloud* topic which is the position of each of the points the LiDAR laser crashes with an obstacle. The navigation algorithm is seen in the */move_base* topic, this topic calculates the cost of the map to check how "expensive" in terms of distance, and other characteristics, is the planned route in our map and generate different routes and choose the best one, when the route is defined, a change in the coordinates of the robot is send and the */cmd_vel* topic changes the speed of the robot joints in gazebo matching the new coordinate system, this is repeated until we reach the objective which is declared by the subtopic */move_base/goal*.

To show how the robot moves around the world, we will use Plot Juggler to subscribe to the x, y position of the robot, these two coordinates will give us the current position of the robot in the plane, and we can use them to effectively track the robot movement around the map, reviewing the difference between the planned route of the robot in Rviz and the real movement. Plot Juggles gives us the opportunity to plot any data in a simple graph, for this we will plot 2 things, the x, y values of the robot odometry separately, and the full position of the robot filmed real life to see the route it took, to test it we are going to do separate routes with the robot starting in an specific non-tagged position and asking to move to the 4 objectives we have.

4.3. Speech Navigation Test.

As explained, the test is done to show how the system can reach the objectives defined by the user, saying the objective it wants to reach saying *"Baño", "Cuarto", "Cocina"* or *"Comedor".* We choose to start at a specific location outside our already declared objective, that way we can have a similar path to take to reach the most part of the areas and compare not only the specific route on two points, but the difference caused trying to reach different points from the same starting point.

We are going to start with the farthest point, which is the Guest Room tagged as *"Cuarto",* to reach this point system must create a route that cross the whole house which can be seen as the most complicated path in terms of distance but also easy as it mostly is a straight line as we will see.

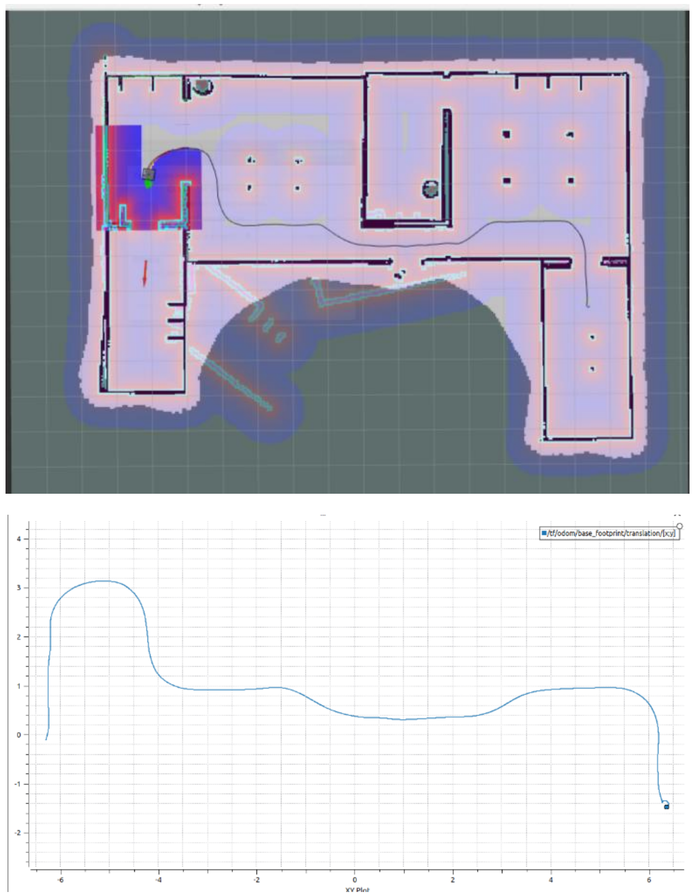


Figure 9 Route comparison Guest Room

In Figure 7 we can identify how the route created by the SLAM algorithm, in the beginning can have a lot of distortions represented by some sharp changes in the top image in Figure 7, we can write down these distortions are not seen in the bottom image, as the turns are cleaner as so are the straight lines, this is caused by two phenomena, first the constant reroute of the SLAM algorithm, that depends on the current position of the robot and the new data scanned by the LiDAR, in the other hand the robot accuracy and movement freedom can change also the final path, as if the wheel gets stuck or moves a bit faster can change the path. Figure 7 show us the response after user ask robot to move to Guest Room. The movement to objective even with a far distance is completely autonomous and, as expected, user only had to give the needed objective.

One of the most complex points we declared was the Bathroom, not only because it is a very far from the start position, but also because it is in a confined space, which makes it harder for our navigation algorithm to reach it, the sharp turn to enter the room, normally is easier to enter by stopping and turning slowly, but SLAM will always try to make a continue movement to reach the objective as we will see in the next Figure.

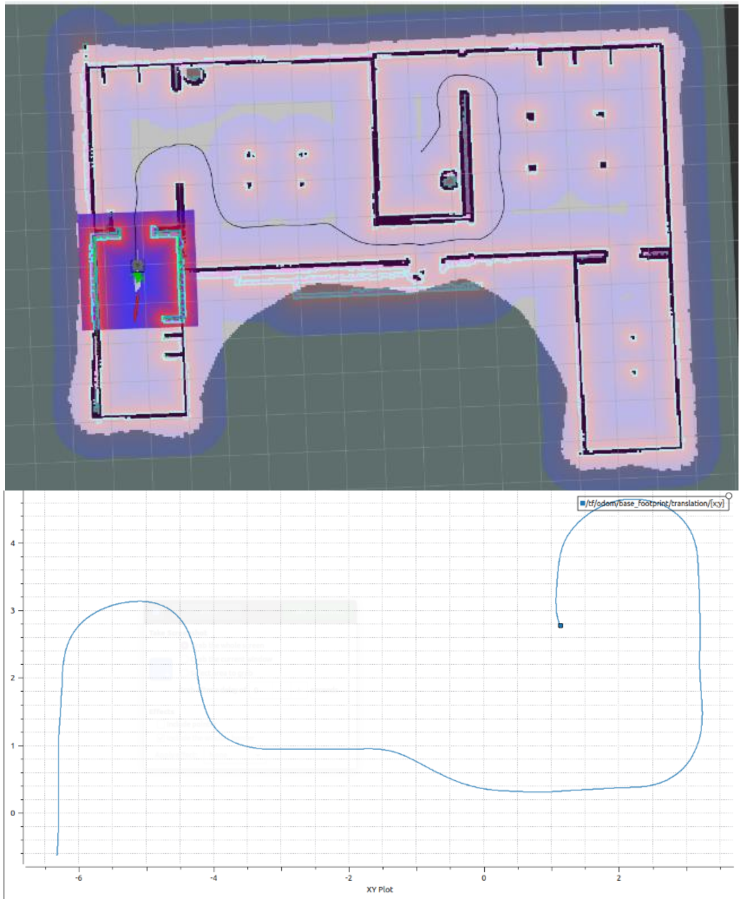


Figure 10 Route Comparison Bathroom

The most important thing to write down in Figure 10, is the fact that when reaching the bathroom entrance in the top image, although at first for SLAM planning is hard to know how will it reach that point, we can see that in the bottom image the movement is clean when reaching the bathroom as the robot gets closer to the entrance of the bathroom, the information from the scan gives the algorithm the opportunity to make a better route to enter the room and reach the objective, as we can see, the system is also able to solve complicated routes without the help of the user as expected, like in the last example, the objective was reached with only the name of the tag as input.

The system was also tested using easier objectives, like the kitchen and the living room, both closer to the starting point, and we found that the system could also reach these objectives without problems, this was also tested navigated around the points of the system (Bathroom-Kitchen, Guest Room-Bathroom etc.) With these tests we gave a good example of how the system will work for collaboration purpose, as user is only providing the objectives, he wants the robot to reach, and the robot will try to reach the objective.

Even when the tests proved successful, we found some complications in the microphone use, the system of data collection with the microphone, opens the input for as long as it identifies there is data coming in, this can give us latency values which can make it uncomfortable for users, but we also found that this error can be countered with the use of a headset with a microphone that is less sensitive of the ambience noise and can also be turned on and off depending what we need.

In the other hand, we realized the navigation problem in the physical world when making tests of the system in a maze, this maze had some irregular areas and some sharp turns so for the robot it was hard to generate a path and follow it. We didn't face this problem in the simulation as the ground and other variables are considered as ideal, also there is no error considered for different speed in the motors or battery factors that can change the result.

5. Conclusions

As explained, the main goal of this research was to exemplify the use of robotic systems as a real-time collaborator in humans' activities, the project we used for this was a 2D robot moving around using

autonomous navigation. This example shows us a way a robot can complete tasks with a quick input of the user, this can be used as a house companion where we can ask a robot to do certain activities in an area with the right tools.
An example of the use of a system like the one built here can be seen in some assembly lines, as we know there are areas of the assembly line that require for humans to place and screw small parts, normally these persons have to go for the parts when needed and this can be an extra effort that can be reduced with a collaboration, if the human can request the parts using an audio input and the robot can navigate where they need them and make a faster and effortless work for both parts.
If we want to make our system better in terms of commodity and usefulness, we need to solve the microphone misreads that we explained, we found a temporal solution with the headsets and other tools that cannot be applied in all cases, if we aim to have the system working every time and situation we have to improve the speech recognition system so it can only focus on words say by the user and no other noise or even be able to discern between other people voice and user voice.
The main advantage of the system we built, and the use of ROS is that the navigation algorithm can be changed depending on the requirements of the project it wants to be applied in. An example of a different navigation system that can be applied here is a camera navigation system, this works using cameras around the area to identify the location of the robot and objectives are given with an already planned route using mathematic equations that can be also called by using an audio input like we proposed.
There is more advancements needed to truly reach an useful human/machine collaboration, in this article we look a simple use of a robot capable of moving around but as more tools are added and functionalities are programmed not only we give the robot the opportunity to have different and more specific applications, but we are also giving the human collaborator the opportunity to better use the robot companion and understand better the importance of both parts in a better and more organized world.

References

[1] Sedil, T.S, Yoo, J. *Pandemics and Automation: Will the Lost Jobes Come Back?* International Monetary Fund, 2021
[2] Acemoglu D. *The Post-Covid World* Remarking: Finance and Development, pp:4-9, 2021
[3] Floyd R. E. *Robotic Languages* – Written for Engineer or Programmer? Proceedings 14 Annual Conference of Industrial Electronics Society, pp 232-236. 1988
[4] Sheridan, T. *Eight ultimate challenges of human-robot communication.* IEEE International Workshop on Robot and Human Communication, pp: 9-14. 1997
[5] Vasieck D. Jalowiczor J Sevcik L. Voznak M. *IoT Smart Home Concept* 26^{th} Telecommunications Forum (TELFOR), pp 1-4 2018
[6] Jazdi N. *Dynamic calculation of the reliability of factory automation applications: industry 4.0 applications* IEEE International Conference on Automation, Quality and Testing Robotics (AQTR), pp 1-6. 2016
[7] Williamson, B. An Introduction to Robotics, Dr Bob Productions, pp: 3-23, 2021
[8] Sirinterlikci, Arif & Karaman Akgul, Arzu & Imamoglu, Oksan. *Automation and Robotics in Processes.* Béla Bilták. pp: 158-163, 2011
[9] Oppenheimer, A. *Salvese quien pueda: El futuro del trabajo en la era de la automatización,* Penguin Random House, 1st ed., 2019
[10] Camarinha-Matos, L. Afsarmanesh, H. *Concept of collaboration,* Information Science Reference. Volume 1, 2008
[11] ROS Official Webpage: *Why ROS: It´s the fastest way to build a robot,* https://www.ros.org/blog/why-ros/, 10/07/2022
[12] ROS Documentations: *Ros Tutorialsl, http://wiki.ros.org/ROS/Tutorials, 10/07/2022*
[13] Huishen, Z. Peng, Z. Songtao L. Xijun K. *A SLAM Based Navigation System for the Indoor Embedded Control Mobile Robot.* 4th International Conference on Robotics and Automation Sciences (ICRAS) 4th International Conference on Robotics and Automation Sciences (ICRAS) pp 22-27 2020