41068 ROBOTICS STUDIO 1

Search and Rescue

SPRINT 1, 2, 3 & 4

DOCUMENT VERSION 0.1

DATE 20/08/23

Team Details

Student Number	Full Name	Email
14251971	Aidan Korpar	aidan.j.korpar@student.uts.edu.au
14241806	Callum Power	callum.power@student.uts.edu.au
14172391	Johnson Ly	johnson.ly@student.uts.edu.au
14277711	Jai Bidduplh	kevin.j.biddulph@student.uts.edu.au
13518598	Ryan Kim	ryan.j.kim@student.uts.edu.au

Table of Contents

Sprint 1

Project Description

This project aims to equip Spot, a **quadrupedal robot dog** designed for rough terrain navigation, with the ability to map, localise, and autonomously navigate its environment.

Key tasks:

- Integrating sensor data
- Implementing SLAM algorithms
- Path planning

This project presents an excellent opportunity for a student interested in mobile robots and autonomous systems to work with a real robot and develop its autonomous capabilities.

Skills required:

- Knowledge of robotics
- Computer vision
- Control systems
- Python
- C++

Familiarity with robotic platforms like Spot is advantageous.

SLO 1.1: Communicating with the stakeholders

Meeting Date:

18/08/23

Stakeholder Name:

Raphael Guenot-Falque

Summary of Notes
The core objective of the project is to develop a fully autonomous robot that can effectively map and search an area in a minimal amount of time. The focus can be office environments as the final demo will be held within a UTS lab, although the robot should be tested in multiple environments to test its reliability. With thorough testing, a variety of simulated environments will be used to benchmark different path planners and mapping algorithms to minimise the time it takes to complete each task. The project can take inspiration from existing research and ROS packages. Research about occupancy mapping and related packages. The technical report can delve into the node system of ROS to explain the architecture of the robotic system. Recommendations include getting Spot set up as early as possible. Planning and time management is important in achieving milestones and completing the project. Split and delegate tasks across team members. Take inspiration from what's already out there and pull that into a suitable system that provides a solution to the problem statement.
Objective: - The main objective of the robot is to map out an area and identify the multiple targets in as little time as possible. - The robot will not need to help the targets out of the environment only identify and mark on the map. - The robot should be fully autonomous, no manual controller required in final system. - The robot will most likely be in an office environment as the final demo will be in a UTS lab, but test with a variety of simulated environments to ensure reliability. - The target object doesn't matter, there will be specific recognition/object detection algorithms. Mapping: - When mapping an environment Spot should pick the path that provides the most information and reduces uncertainty. - Mapping should also consider context and continuity to reduce back tracking. - Research occupancy maps – ROS Cartographer/Hector_SLAM. Testing - Testing different algorithms should be about finding which one minimises the time to complete a task i.e., Path Planning, Mapping. ROS and Packages - There are lots of packages and research papers on this type of topic, have a browse and pick packages related to the project. - Technical Report can describe the architecture of the robotic system through the node system in ROS. Recommendations: - Get Spot set up as soon as possible. - Plan your time effectively to prevent crunch at the end. - Split and delegate tasks across team members - Take inspiration from what's already out there and pull that into a suitable system that provides a solution to the problem statement.

SLO 1.2: Identifying and agreeing on the priorities, goals and system requirements

Priorities

1. Talk to Stake Holder
2. Setting Up simulation environment
3. Set up Spot API

Goals:

1. Set up ROS/Gazebo Simulation environment.
2. Implementing Sensor Data and Spot AP
3. Coding and Developing **SLAM** Algorithms and Pathfinding methods for Mapping and Localization.
4. Testing performance and efficiency of Search & Rescue method.
5. Live demonstration of Search & Rescue mission.

System Requirements:

1. Linux OS: Ubuntu min: 18.04 | recommended: 20.04
2. RAM: 8 GB
3. CPU cores: min 1 | recommended: 4
4. ROS version: ROS Noetic Ninjemys (for Ubuntu 20.04)

SLO 2.1 Problem Statement

Search and rescue are a dangerous and costly task where emergency services need to quickly map out environments and localize rescue targets for a safe evacuation pathfinding. This means that rescue operations come with associated risks that need mitigation. Removing the need for humans to risk their lives would improve the overall process of search and rescue and minimise costs.

SLO 2.3 Functional requirements and design parameters of project

A list of functional requirements

- Use sensors to localise robot position and orientation
- Use sensors to map out surrounding environment
- Path planning using mapping data
- Identify target objective within environment using sensor
- Mark targets on map

A list of design parameters

- The SPOT robot must not interfere with obstacles or boundaries
- The SPOT robot must not mark anything that is not the target
- The SPOT robot must be completed by the end of the 12 week period
- The SPOT robot must not map an area that has already been mapped in the same run
- The SPOT robot must use a range of sensors for mapping and locating

SLO 2.4 Technical statement

Reasons for simulations

Simulations are vital for the development and deployment of robotic systems. They provide several benefits compared to testing and developing on the real hardware.

Cost-efficient

By simulating the robotics platform in a virtual environment allows for immediate access to the robot for development. Testing on the physical robot can be expensive, particularly with unproven algorithms and software that could damage the robot in unexpected ways. Simulations save on maintenance of the robot's hardware.

Safety

Testing the robot within a simulated environment will much safer during development than on the real hardware. This can prevent damage to the robot which can be expensive as well as inhibit any dangerous situations that could induce injury/harm from the robot.

Rapid Iteration

Having the robot as a simulation increases the productivity of development as different algorithms or sensors can be tested immediately inside a virtual environment without having to wait for the physical hardware.

Testing and Algorithm Validation

Testing inside a virtual environment can be done easily. The robot can be placed inside numerous environments that test different components of the system to see how suitable such a system can be for a particular environment. Additionally, different algorithms can be easily swapped out and benchmarked inside these virtual environments enabling for efficient evaluation of these algorithms.

Justify choice of sensors

The Boston Dynamics Spot robot already possesses a range of sensors to assist in the function of the robot. The sensors we will be using for this project allow the SPOT to know where it is in an environment and to map out the area. We will be utilising:

Add-on attachment LIDAR

LIDAR allows SPOT to map its surrounding environment with high accuracy, providing live information on obstacles in its path. The LIDAR has a short-range meaning SPOT will walk around the environment to create a map as the search area increases, providing important data on the surrounds to allow decisions to be made on pathing for the search and rescue operation.

Integrated SPOT RGB camera

An RGB camera will allow SPOT to identify objects and surfaces based on optical feedback, this can allow adaptability to environmental challenges but most importantly allows the target to be identified through visual identification. The RGB camera will allow SPOT to identify colour and shape to interpret when a target is found, enabling a successful search mission.

Integrated SPOT Inertial Measurement Unit (IMU)

An IMU allows SPOT to know where it is within an environment by providing a range of data, which is then used to identify the location of SPOT and prevent interference with obstacles and boundaries. An IMU will allow spot to accurately predict location and adapt to environmental obstacles as they arise, to safely navigate without risk to the robot or target.

Justify choice of platforms

Boston Dynamics, Spot the dog will be used as the robotics platform of choice for its mobility and variety of sensor attachments.

Compared to other robotics platforms such as TurtleBot which moves on wheels, Spot has motorised legs that allow for easy traversal of rough terrain making it perfect in a search and rescue situation where debris could get in the way of the robot. Along with its array of sensors, Spot will be able to map the search area with ease.

SLO 2.5 Design objectives.

Pathfinding Algorithm

This aspect is essential in a search and rescue setting. It is essential the robot can autonomously navigate an area in the most time effective way.

Object detection Algorithm

Object detection is vital in search and rescue. An effective object detection algorithm will allow the specified objects/subjects to be accurately marked on a map.

Mapping Algorithm

The mapping algorithm is necessary for both the pathfinding and object detection as the pathfinding will use the data from the mapping algorithm and the object detection will need to place markers on an accurate map to help responders extract the targets quickly.

SLO 2.6 Evaluation criteria

Design Objective	Testing Procedure
Pathfinding Algorithm	To test the pathfinding capabilities of the robot, multiple algorithms will be run on the robot and timed. The criteria for picking which algorithm will be the one that takes the least amount of time to plan a path and search the area.
Object detection Algorithm	To test the object detection capabilities of the system, a multitude of sensors will be tested to see which one is most suitable for detection of the target object. This will include picking a reliable algorithm for processing the sensor data. The criteria for selection will be sensors that give reliable and consistent detection of a chosen object.
Mapping Algorithm	To test the mapping capabilities of the platform, the mapped data will be overlayed and compared with the ground truth of the simulated environment to check its reliability and consistency.

SLO 2.7 Timeline for the Sprint 1

Date	Activity	Notes
7/8/23	First Class	Given Sprint 1 and Stakeholder
14/8/23	Week 2 Class	Contacted Stakeholder and set up meeting. Start Project scope document
18/8/23	Meeting With Stakeholder	Had meeting with stake holder Had group meeting to define design objectives
21/8/23	Sprint 1 Due	

SLO 3.1 Configuring the system

In setting up my Linux and ROS I had one major issue. When trying to launch ros I would get an error related to using the VM. The error being VMware: vmw_ioctl_command error Invalid argument. To get around this issue I went to my vm that I used for pfms. This system was running Ubuntu 20.04 and when following the steps ros was able to launch without any issues. Ubuntu 20.04 on the VM also allowed me to copy and paste between my host machine and the VM which 18.04 that was supplied did not work.

In the lab I was able to launch the both the turtle bot house and the turtle bot world using roslaunch turtlebot3_gazebo turtlebot3_world.launch and roslaunch turtlebot3_gazebo turtlebot3_house.launch. I had no issues with this process as the steps were easy to follow.

SLO 3.2 Set up a simple indoor office environment in Gazebo

In the week 2 lab we had to launch the Turtlebot in the house environment and use the laser sensor data to visualise what the robot is seeing in Rviz and then plot the odometry data as we controlled the Turtlebot and drove it around the house. We were also able to access the on board camera.

I had one major issue with this that stumped both myself, my peers and the tutors. In this case when in Rviz the laser data nor the Turtlebot was being displayed and an error regarding the wrong base frame was being given. We went through uninstalling and reinstalling Turtlebot but this did not help. After using the echo command we were able to see that there was data there. In the end we realised that we were using the wrong command to launch rviz. We were using the one to try and connect as if the robot was connected via ethernet to my computer and not the one to launch the simulation. After using the correct command the rest of the lab flowed smoothly.

SLO 4.1 Time management

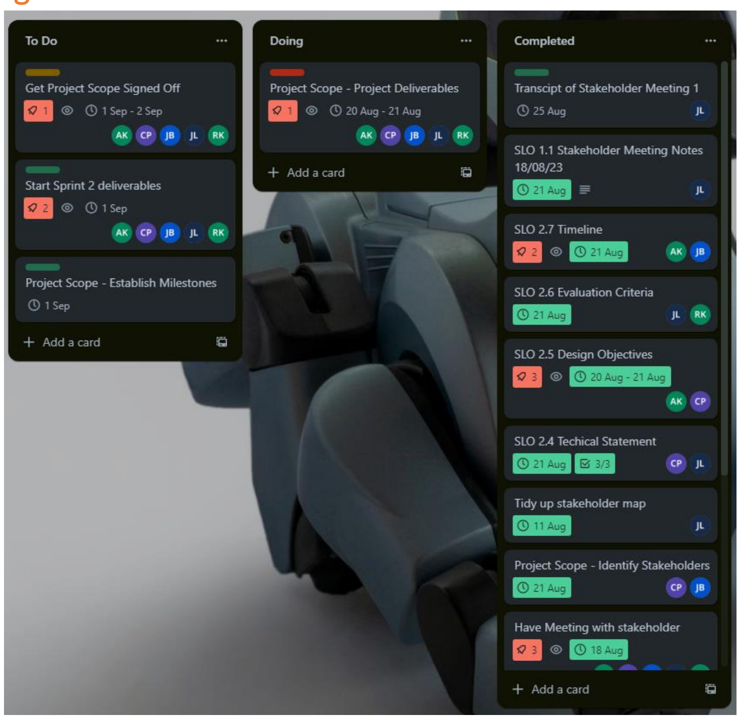


Figure 1 Team Trello Showing Work Completed On Time

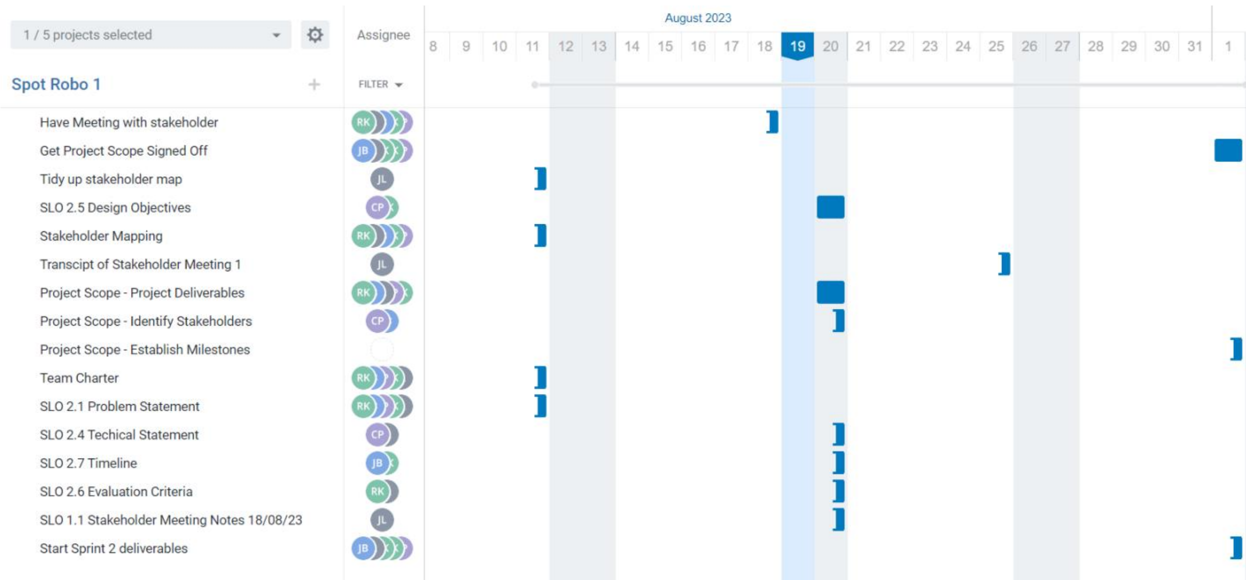


Figure 2 Team Gannt Chart Based on Trello

SLO 4.2 Communication skills

One instance where I demonstrated effective communication within the group was in our group meeting following our stakeholder meeting. In this meeting, I effectively allocated tasks and sections to individuals to increase our workflow and speed. At the conclusion of this meeting not all tasks were completed so team members volunteered to finish off each section. The following day I checked in with all members and had a quick meeting to finalise the team components of sprint 1.

Sprint 2

SLO 1.1 Identifying constraints and risks

List of constraints

- The SPOT robot must not interfere with obstacles or boundaries
- The SPOT robot must not mark anything that is not the target
- The SPOT robot must mark the target with a 95% certainty
- The SPOT robot must be completed by the end of the 12-week period
- The SPOT robot must not map an area that has already been mapped in the same run

List of risks and mitigation strategies

Hazards	Risks	Mitigations	Residual Hazard Level
Robot loses connection	Robot damage Human injury	Ensure steady Wi-Fi. Connect via cable. E-stop.	L
Robot goes out of bounds	Potential damage to Robot Potential Injury to People Potential damage to surrounding environments	E-stop. Simulation testing.	L
Robot collision with obstacles	Direct damage to robot Direct hazard to people Damage to surround environments	E-stop. Simulation testing.	L
Unencrypted Data	Vulnerability of sensitive data Leak of confidential/private data	If stored, encrypt. Don't store data.	L
Erroneous code/algorithms	Robot damage Human injury	Double check algorithms. Simulation testing.	L
Loss of Light	Robot damage Loss of position and orientation Unable to identify targets	Add flashlight to robot platform.	L

SLO 1.2 – Identify any cultural, ethical, environmental, legislative and economic perspectives

Ethical

- → Privacy Concerns
  - The SPOT robot collects and use location data to navigate and perform tasks. This raises concerns about the tracking and storing of individuals' movements, potentially infringing upon their privacy.
- → Job Displacement
  - The SPOT robot will have a positive influence on job displacement, taking away the need for humans to be put at risk during search and rescue operations. SPOT still requires a team to rescue the intended target, so will greatly improve the efficiency of rescues.
- → Decision-Making Algorithms
  - The Spot robot may prioritize efficiency over safety, leading to risky decisions in certain situations that could compromise the well-being of a rescue target or the integrity of the robot itself.

Legislative

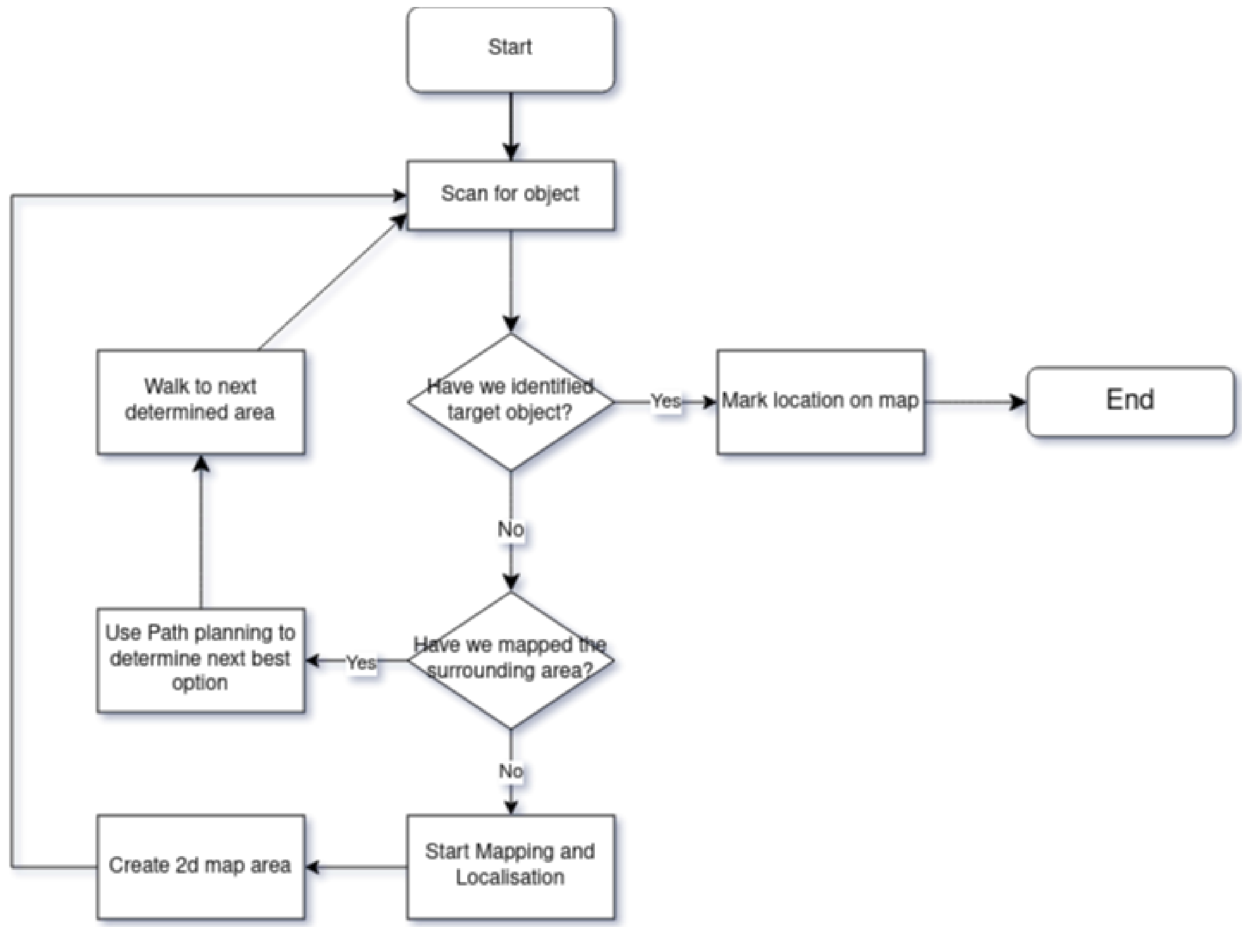
- → Regulation and Safety
  - The robot would have to get regulatory approval before being deployed and used. This would be to ensure it abided by all set safety standards in the industry. Especially with interactions between the robot and humans to ensure no damage or injury can be done.
- → Liability and Accountability
  - Guide lines would need to be set as to who would be responsible for any accidents or harm that could take place. Such as an item braking during the search procedure.
- → Data handling
  - Search and rescue robots have to collect large amounts of data to complete their tasks however a his data needs to be handled in regards to the regions specific laws. This can relate to privacy concerns for any data collected such as images or faces of people the robots sensors pick up. It would need to abide by Australian privacy laws such as the APP.

Economic

- → Cost-Benefit Analysis – Search and rescue operations are costly. By using an autonomous solution such as Spot the dog, the need for humans to enter hazardous situations is reduced. This in turn reduces the operational costs associated with hazard pay and additional life-preserving and search equipment.
- → Job Market Impact - Implementing an autonomous solution will see the need for trained engineers to program and monitor the SPOT, to ensure it can adapt to any situation and continuously improve functionality. This will see jobs not only in SPOT control but research and development of improved sensor and detection algorithms for search operations.
- → Market Demand – There is a recognised market demand for technology to assist in search and rescue operations as currently it takes lots of time and resources and puts humans at substantial risk. The Search and Rescue market would be the main demand for an alternative such as the SPOT to assist in rescue operations.

SLO 2.1 - Present the overall project idea

Flow Chart



Brief Description

The Search and Rescue project is designed to utilise object detection and SLAM algorithm data that will interface with the Boston Dynamics SPOT robot, providing an autonomous robotic solution to replace the need for humans being exposed to hazardous environments. The algorithms used for both object detection and SLAM will be researched and tested to establish the most suitable and accurate results when undertaking a search to locate and mark a specified object on a map generated by the robot whilst navigating the adverse environments. Using the Spot Robot, we are going to attempt to overcome the obstacles that are currently found in the situations described. As a result of this project we hope to deepen our understanding of how these autonomous robots can be integrated into everyday life.

Pseudo Code

Turn on Spot Define intended target start automated search and rescue mission While constantly searching for intended target Mark searched targets to prevent re-detection Use IMU to establish global location Scan current environment to calibrate surroundings Map search area Make a decision if an area is unsearched - If path is not obstructed, search first available - If path is obstructed, search for next available option If target is still not found walk to unsearched area whilst scanning 'Look' around unsearched area to fill out map If target still not found Continue searching unsearched area If target found Mark on map Mission complete	Start robot launch simulation environment export spot robot model inital scan for target object using Mapping and Localising WHILE target not found SWITCH have we identified target object CASE target not in sight Start Mapping and Localising Start Pathfinding CASE target found target found END END END Mapping and Localising create 2d mapping area scan area using sensors transform sensor data to coordinates transform local reference frame of SPOT to global frame interpret sensor to detect target check if target found update 2d mapping area END Pathfinding detect available pathways based on 2d mapped area WHILE Map is not fully unexplored IF there an unexplored pathway path through the unexplored space perform Mapping and Localising IF the robot is in a dead end walk back to initial position THEN THEN END

SLO 2.2 Robot navigating in the environment while displaying sensor data

Setting up the turtle bot in the environment is very simple and easy especially as we were provided the commands to do so. I then proceeded to launch Rviz to visualise the robot and laser scan. When launching rviz I just used the rviz command and tried adding the spot robot and laser scan topic however there was errors due to frame errors. To fix this I launched rviz using "roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch". This fixed my problem and I could add the laser scan data and visualise it as I drove the turtle bot around. As the rest of the commands were provided there was not much difficulty with this task.

Seeing the visualisation while driving around the turtle bot was pretty cool.

SLO 3.1 Provide a short description of a mapping task

Mapping is the process of creating a representation of the environment using data from sensors. There is both 2D and 3D representation and are both essential for navigating and autonomous decision making. This mapping can then be used for obstacle avoidance and localisation.

Three sensors used for robotic mapping includes; RBGD (Red, Blue, Green, Depth), LiDAR and IMU

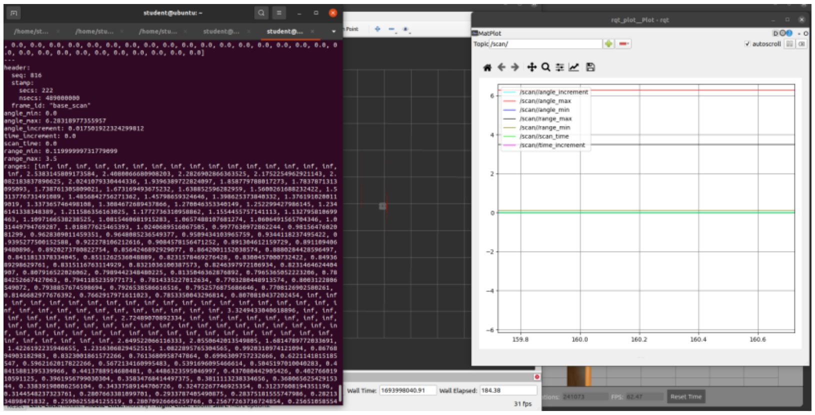
There are many challenges that come along with mapping. The more common issues relate to sensor limitations, where the correct data is not interpreted such as a clear/see through object. Cheap sensors also have limitations relating to distance and resolution. Localisation errors which is spoken about in 3.4 can affect mapping as inaccuracies in the robots position relative to the map will lead to wrong representations of the environment. Mapping is also a very resource heavy process that can take a lot of computing power and without the right processors can delay and lag which can lead to miss aligned and wrong map representations.

SLO 3.2 Demonstrating the mapping task in Gazebo environment.

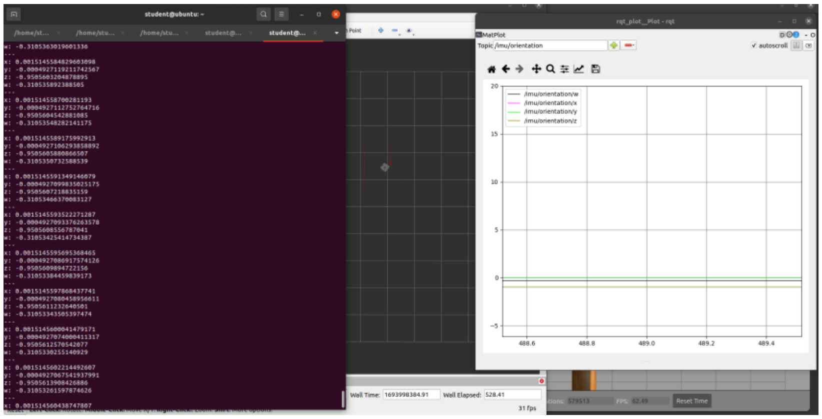
There were not many challenges in setting up the mapping with turtle bot. We just ran the slam package that was included and could visualise these results using rviz. Opening rviz and adding each topic was tedious and annoying but we were taught how to save presets and can just open this. The only challenge with the mapping of the spot robot was finding and installing the required packages luckily a team member in the group had combined these into a git repository so I could just git clone and then call the hector_slam package which worked very well. I then walked the robot around using the telop command and watched the mapping update in rviz as more data was collected.

SLO 3.3 Demonstrate displaying sensors

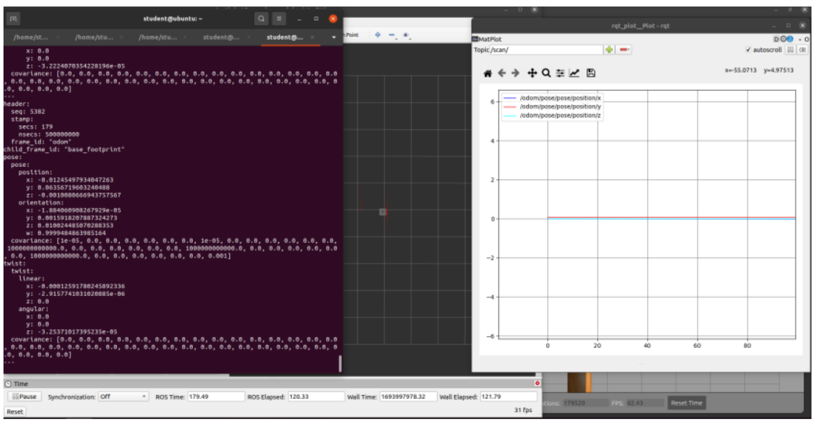
Laser Readings



IMU



Odometer



SLO 3.4 Provide a short description of the localisation task

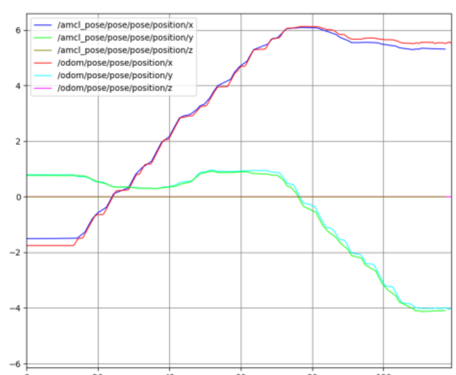
Localisation is the process of a robot determining its own position and orientation known as its pose within its surrounding environment. This is essential for all robots as it allows for interacting with their environment as well as mapping and path planning. There are two primary methods for localisation these are using odometry and using sensors. Odometry uses the values from the motors (motor encoders) to estimate how far and in what direction the robot has travelled. Using sensors takes in information from the surrounding environment to estimate where the robot is.

Three sensors we will use for robotic localisation are LiDAR, IMU and depth sensors.

Challenges related to localisation include accumulation errors seen when using the odometry method. Sensor noise can be induced in challenging lighting conditions and such accuracies can lead to uninterpretable data. Filters such as debounces can be used to mitigate this noise. Accurate sensor calibration is essential for precise localisation. Real-time computing is another concern, as localization must occur quickly and efficiently to support robot navigation and decision-making. Addressing these issues requires a holistic approach, combining advanced sensor technologies, robust algorithms, and efficient real-time processing to ensure reliable and accurate robot localization in dynamic environments.

SLO 3.5 Demonstrating the localisation task in the Gazebo environment

I focused on the particle filter method for localisation. There were a few challenges in setting this up. The first was needing to map out a whole area and save the map file. Without fully knowing the commands I got quite lost. The harder part about this was after mapping trying to display the ground truth location of the robot. It was easy to see this in gazebo when clicking on the robot but there was no topic to display it meaning it could not be mapped. The below graph shows my output of the particle filter method compared to the odometry method. As you can see the values are slightly off as the particle method follows slightly behind the odometry as it does not update as often.



There were many errors that occurred when trying to do localisation through the particle filter method in the spot robot. This included errors launching out predefined map, errors when playing back the localisation and problems trying to map the topics as they could not be put in the rqt_plot.

SLO 3.6 Compare the localisation accuracies against the ground truth

Using the RMSE formula the RMSE value was calculated

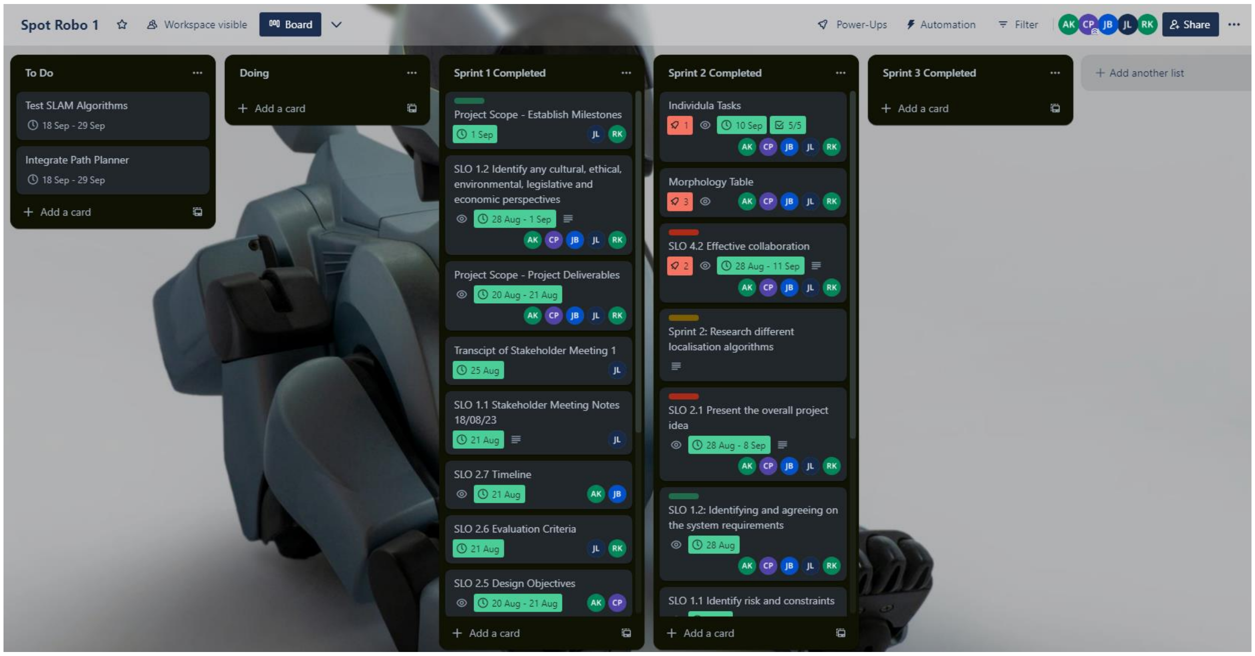
RMSE = $\sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}.$

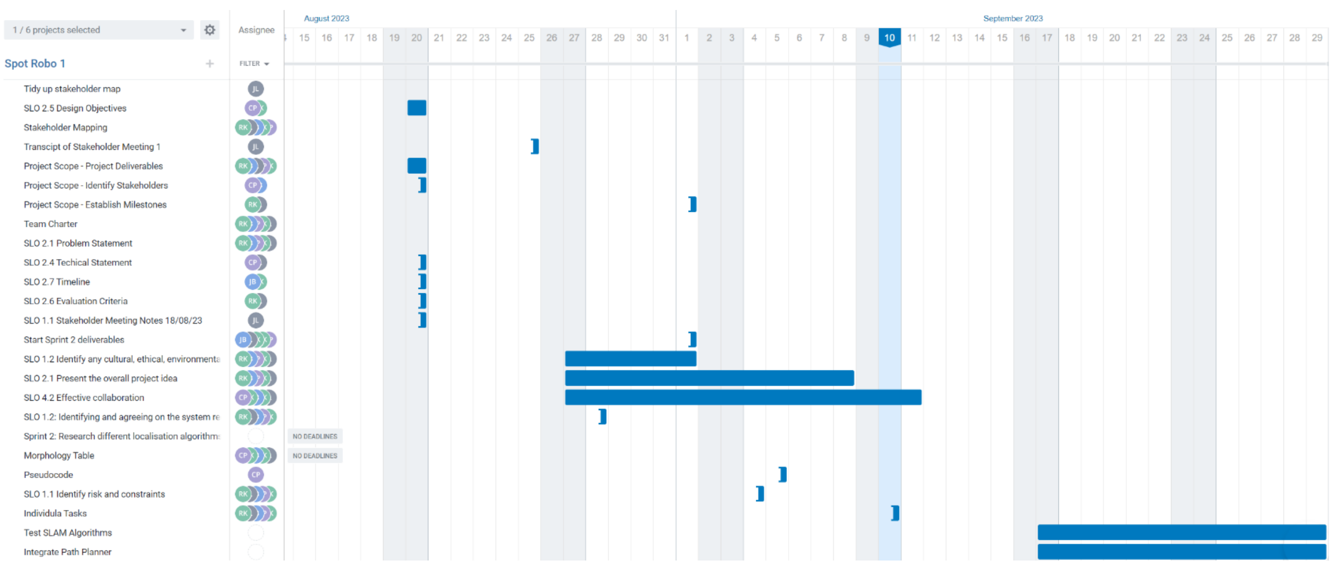
RMSE = 1.509248766

The RMSE shows a value greater than 1 meaning that the model was not well optimised. This is further seen by the graph above. This result was due to the spot robot using the odometer data and as we have legs and not wheels this data is severely unreliable.

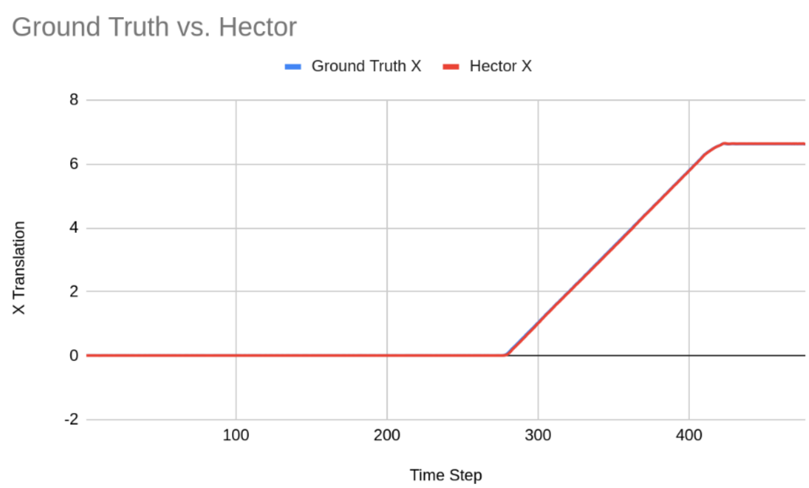
You can see that the particle mapping reorientating itself at some points however it gets lost very quickly again.

SLO 4.1 Time management





SLO 4.2 Effective Collaboration

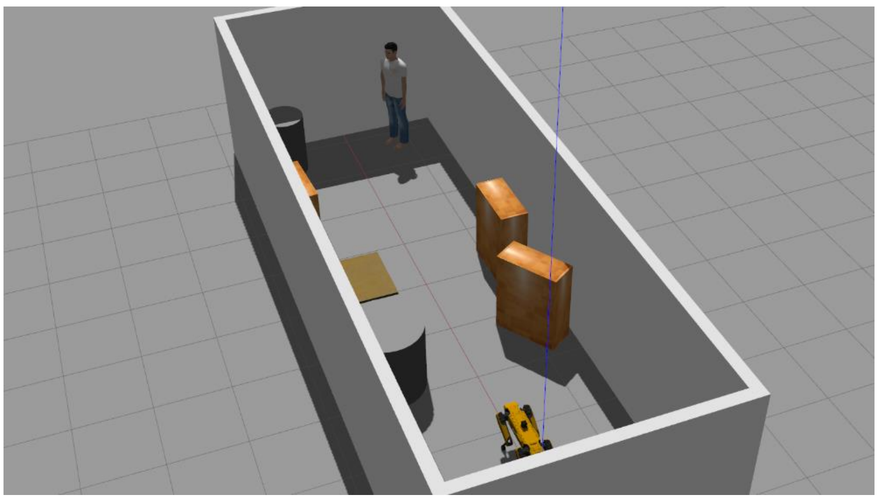


Results

Algorithm	RSME
AMCL (Particle Filter)	1.509248766
Hector (Scan Matching)	0.01339841487
EKF (Laser + IMU)	4.482819302

From the above we can see that the localisation associated with hector mapping was the best result with a RSME value of 0.01339841487. This algorithm used scan matching of the laser scan data.

The AMCL had a very high RSME value meaning it is poorly optimised. This is further seen by the AMCL graph. This result was due to the spot robot using the odometer data and as we have legs and not wheels this data is severely unreliable.



The EKF method utilized a combination of laser scan data from HECTOR slam and IMU data in the efforts to create a more accurate localization estimate. However, due to complexity of the configuration setting with leveraging different data sets, a lot of issues was encountered despite successfully implementing both sensors. We suspect that the unfiltered IMU data is causing a large bias and making the resulting body pose build up error over time as shown in the EKF vs Ground Truth graph.

From this, the method we will be using for our assignment is the localisation associated with the hector package.

Sprint 3

SLO 1.1 Progress update with stakeholder engagement

Meeting date: 22/9/23

Stakeholder name: Raphael Guenot-Falque

Talked about using Voxel blocks for mapping and the use in industry for this method as it reduces required processing power The issue with using SLAM is local detection requires measures to deal with moving objects such as people in the environment, this can cause the robot to go off course or clash with the moving objects, need to be aware of how to address this 2D mapping cannot detect features very well, in an office environment it can't tell what is a table or chair and may cause damage to the robot as the gap between the legs can be seen as a vacant space able to be navigated. 3D can avoid these issues, however it comes at an increase in computational power. Computational power is the most important aspect for these projects and will govern how large the robot has to be, LIDARS and sensors will use computational power as well as the algorithms running so be mindful to have enough computational power to drive all the required processes. Can use a 'terminator' program to split the map into quadrants to reduce the amount of computation needed for heavy resource use tasks. Target objective can change and can be a range of different objects to detect, our one will be a basic object to test our project for this assignment
Johnson NOTE - ROS Packages: Dynablox, Cartographer Test for computational resources, between SLAM methods Look at 2D vs 3D benefits for mapping Sensors on SPOT: - Infrared Depth Sensors - 2D LiDAR What is our target object: - Ask them when we get up

SLO 2.1 Present the idea of SLAM and path planning in the context of your project

Describe how SLAM can aid your overall project

SLAM or simultaneous localisation and mapping can aid our project by assisting in the mapping of unknown environment and data collection. Passing this data into our path planning algorithm will allow our SPOT robot to autonomously navigate these unknown environments. It will also allow us to build an accurate map while navigating the environment. This capability ensures our SPOT robot can explore unknown areas effectively whilst collecting data that can be used for map creation or later on in our project object detection.

Describe how path planning can aid your overall project

Path planning can aid our project by providing an optimal route for SPOT whilst undertaking precise mapping and localisation of the environment. The path planning function will utilise integrated sensors for SPOT to explore an environment, choosing the most suitable paths by feeding data to our SLAM algorithm to map and undertake object detection to locate the intended target. Path planning will prevent SPOT from exploring areas that could cause damage to environment or the robot itself, which is crucial when undertaking a time-sensitive search and rescue operation in environments that can potentially be unstable or dangerous for human interaction.

SLO 3.1 Provide a description of SLAM

What is SLAM and why it is important for robots? (brief paragraph).

Simultaneous Localisation and Mapping (SLAM) is very important in robotics and autonomous navigation. It allows robots to understand their environment and navigate in it. The two aspects of SLAM are localisation and mapping and when put together allows robots to autonomously explore an unknown environment whilst creating a map and determining its own position.

List of three sensors those can be used for a robotic SLAM task.

Three sensors that can be used for slam are Lidar, IMU and RGBD or in our case just depth.
What are the challenges in SLAM? List and briefly describe two popular SLAM techniques in ROS together the type of sensors suitable for these algorithms.

As you can imagine, SLAM is a challenging problem and as such there a number of challenges that need to be delt with. Regarding our project one of these problems would be dynamic obstacles. Moving obstacles such as moving people, vehicles or objects poses an issue has the SLAM algorithm must be able to detect and keep track of these objects and update the map accordingly. Luckily for our project this is a minor issue. Our robot also has kinematic constraints. For the robot to turn around it has to turn, it can't easily move in any direction. This needs to be accounted for when moving. SLAM algorithms can also be computationally expensive as it is mapping large areas in high resolution. Any lag or delay can introduce incorrect measurements into the mapping. These incorrect measurements or data can also come from sensor failure or incorrect calibration. This can lead to incorrect mapping and localisation meaning the robot no longer knows where it is in its space which can lead to collisions and navigation issues.

Two popular SLAM methods are Gmapping and Cartographer

Gmapping is short for grid-based mapping. It creates a 2D occupancy grid that represents the surrounding environment. Gmapping is very suitable for indoor environment and uses the following sensors 2D lidar and odometry.

Cartographer supports both 2D and 3D SLAM. It uses point clouds to create a map of the environment. Cartographer is suitable for indoor and outdoor use where highly detailed maps are required. Cartographer uses the following sensors 2D or 3D lidar and IMU.

SLO 3.2 Demonstrating the SLAM task in Gazebo environment.

Setting up SLAM in our robot was quite easy.

The difficult part was figuring out a way to overlay our generated map with the one in gazebo. The two possible solutions I found was using the turtle bot to go explore our map and record it and then save the map as a YAML file so that it could be overlayed in gazebo to compare against our SLAM method. The other option was to take a screen shot of each one at the same scale rate and manually over lay the files by reducing the opacity of one.

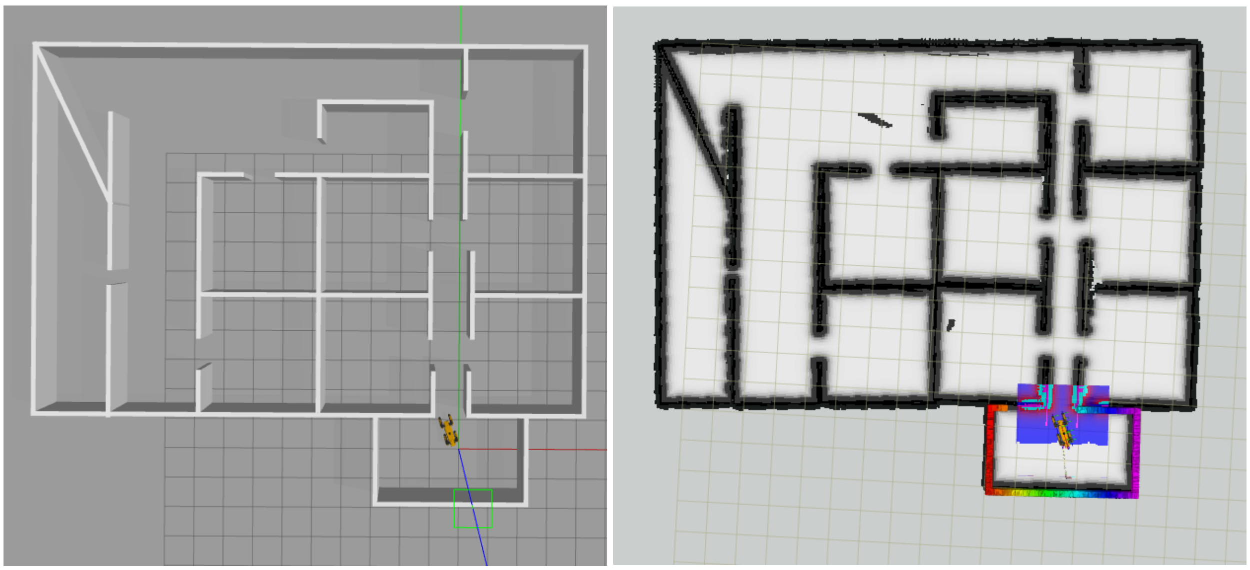


Figure 3 Gazebo Map (left) Rviz SLAM Output (right)

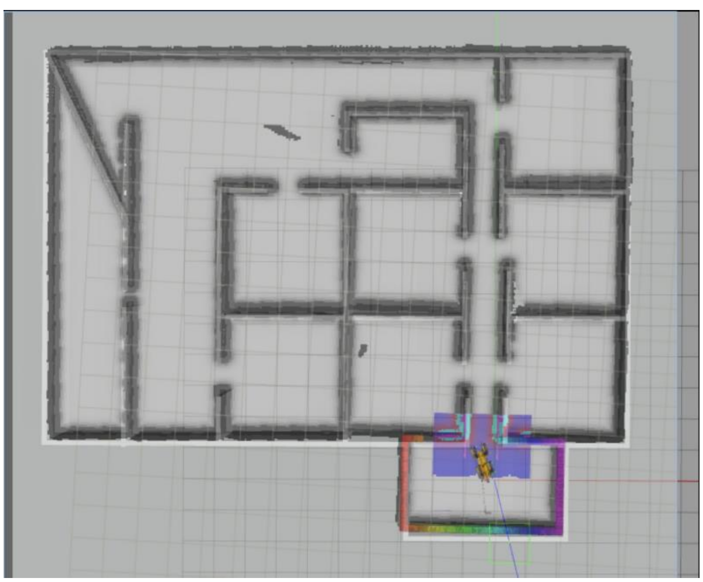
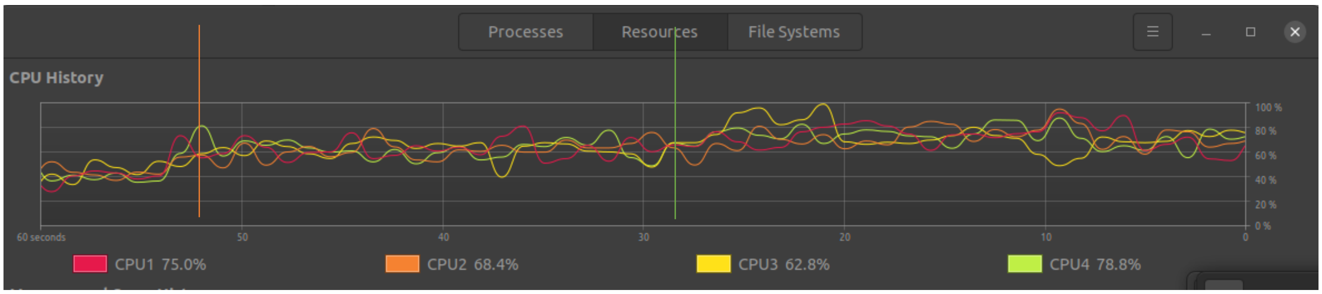


Figure 4 Overlay of Gazebo Map and RViz SLAM Output

SLO 3.3 Analyse computational aspects in simulations

The below graph shows three distinct sections. The first below the orange line is before launching slam and we can see its sitting around 50% CPU usage. After Launching SLAM, between orange and green line, we can see it spikes and increases to sit around 70-80% CPU usage. After the green line I launched our path planning and exploring algorithm which we can see as the spot robot moves around and is detecting un scanned areas the computation power increases and sits closer to 90% even spiking to 100% a few times. As I have only allocated 4 processors to my VM its using all 4.



SLO 3.4 Provide a short description of the Robot Path Planning task.

What is path planning?

Robot Path Planning is the task of determining a trajectory for a robot to navigate from its current position to a desired destination without collisions. It involves finding continuous waypoints at small increments that guide the robot to the desired destination. These way points ensure that the robot can navigate around obstacles along the way.

List of three types of path planning algorithms with a brief description

Three types of path planning algorithms are RRT, Dijkstra's and A*

RRT stands for Rapidly-exploring random trees and are highly effective in complex environments. They work by randomly generating points and connecting them to the closest node available. Each point that is created must check that it lies outside of an obstacle and that when chaining it must also avoid an obstacle. The algorithm ends when a node is generate within range of the goal or a limit is hit (Chinenov ,2019). RRT* is a refined algorithm that refines the RRT output and creates the shortest route to the goal.

Dijkstra's algorithm is used to find the shortest path between two points in a graph. By modelling the environment as a graph the algorithm can be used in robotics. The algorithm uses the weights (distances) of the edges to minimise the total distance between the source node (starting point) and other nodes till the goal is reached (Navone, 2023). Dijkstra's algorithm is computationally very heavy but will guarantee the most optimal path.

A* combines the Dijkstra's algorithm with a heuristic function to create a far more efficient search algorithm. Instead of using priority queue to store all the nodes A* uses binary tress. By using a heuristic function to get the distance from a node to the goal a far more efficient method is created (S, 2023).

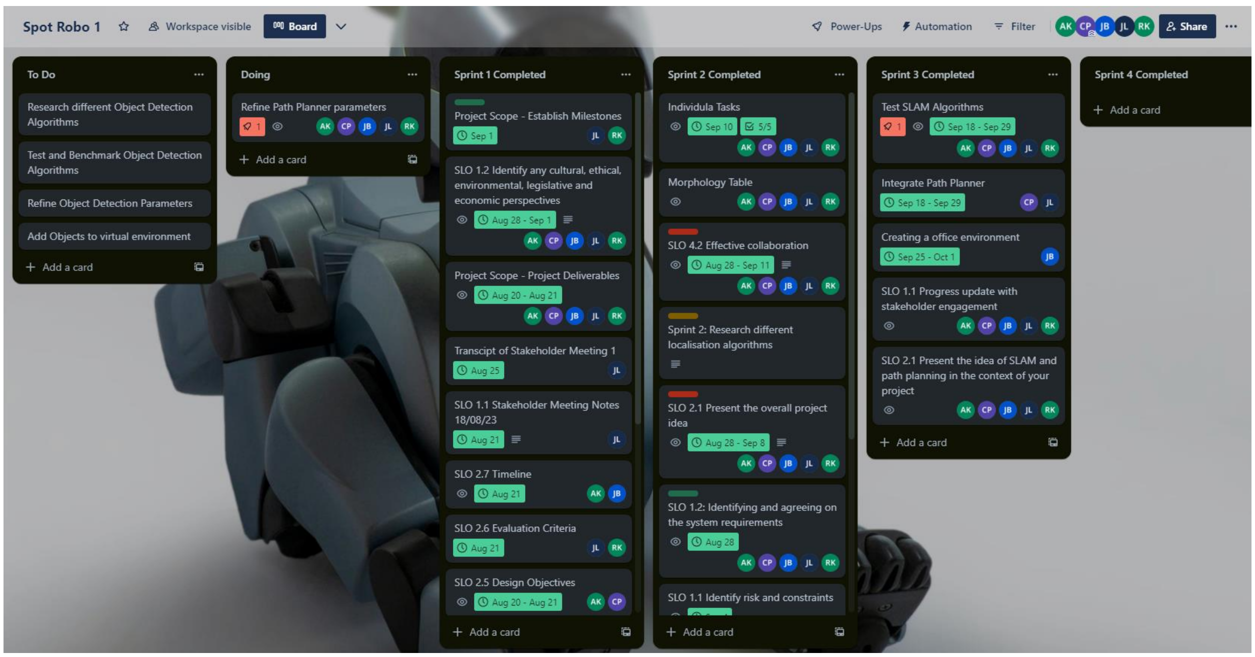
What are the challenges in path planning?

The challenges related to path planning include adapting to dynamic obstacles like moving people or vehicles, accounting for kinematic constraints that limit a robot's motion. Path planning algorithms are also highly computationally expensive, choosing a less good algorithm that is less computationally expensive might be necessary in some situations. The algorithm secondary job of the algorithm is to avoid obstacles on the way to the destination. Sensors can introduce noise and uncertainty into the SLAM algorithm as mentioned above that can hinder the efficiency of the path planning algorithm. SLAM and path planning must work hand in hand.

SLO 3.5 Demonstrating the path planning task in the Gazebo environment.

Setting up the path planning was very easy. Once the area was mapped out from SLAM, I was able to use the 2D nav goal anywhere in the scanned area and the spot robot would automatically create the path planner for the goal. It uses the Move Base Package that is built into our SLAM package. The default path planner used is Dijkstra. It even worked if I placed the goal in an unexplored area and as it got closer to the goal and saw an obstacle it would alter the path to the goal. This was not very useful for our application however as we need to explore an area using the 2D nav goal is not an effective way to search an area. To solve this I used a package called explore lite that when run automatically starts searching unexplored areas.

SLO 4.1 Time management



SLO 4.2 Create a GitHub repository.

This link for my git repository is below:

https://github.com/Korpie15/RoboStudio1.git

SLO 4.3 Use Doxygen to produce documentation of your program.

No report component

Sprint 4

1.1 Stakeholder engagement on project completion

Meeting date: 22/9/23

Stakeholder name: Raphael Guenot-Falque

Summary Notes
reducing the range finder is on way of forcing the robot go into the room . (we have tried this and was faulty, robot loses itself) perception is too slow? detection frame of the human is too slow compared to how fast the robot is scanning and pathfinding the rooms. ONE WAY, using a 360 view camera to not miss human beings when 'leaving' rooms after robot full scans a room. ANOTHER, make the robot do a 360 on the spot to make a full view, more downtime on SLAM and pathfinding systems and give more chances for human detection USE the simplest version of fixing the issue for the above "hack it a bit" , rig it to always end up facing a human being ("put it at a dead end"), do this if all else fails and if we can't make it detect humans on random positions. EVALUATION CRITERIA: instead of time efficiency, maybe test consistency, same position condition, see how consistent i runs from start to end over 10 different runs. perception system > look at lesser frame/parameter systems? to run faster without GPU (MINI NET is an alternative to yolo v3)

SLO 2.1 Develop an evaluation criteria

Evaluation Criteria

Evaluation Criteria	Value	Weight	Total
Mapping			
Map Accuracy	1.00	0.3	0.3
Path Planning			
Obstacle Avoidance	0.95	0.2	0.19
Time Efficiency	0.75	0.1	0.075
Object Detection			
Identification	1.00	0.3	0.3
Computational Power			
Computational Cost	1.00	0.1	0.1

Total = 96.5%

Description of Criteria

Map accuracy - is how accurate the map is to the reference model. Calculated by taking an area of the ground truth and compare it mathematically with the explore ground truth.
5 - Every details is mapped 100%
4 - most details are mapped 90%
3 - some details are mapped 75%
2 - few details are mapped 50%
1 - basic details are mapped
0 - no details are mapped 0%

Obstacle Avoidance - is how well SPOT can navigate around the environment without touching walls or other obstacles. It will be marked out of 20 and every collision will subtract 1 point.

Consistency In Avoidance - is measured by repeating the same starting position and end goal post sent to SPOT over 5 repetitions. A mean score of the above test will be calculated to measure consistency rating.

Identification - is specifically marked based on is precision to correctly detect and classify Human objects out of all detected objects within the ground truth. 10 sample objects with 5 of them being human objects, will be placed within a testing environment. Each human object that is correctly detected (0.5 points) and classified (0.5 points) is added to a total mark of 5.

Computational Cost - is how much CPU usage is used when running the program.

1 = uses 80% total load , 0.5 = uses 100% CPU load but still works as intended, 0 = uses too much CPU load and doesn’t work

SLO 3.1 Project evaluation and testing: Overall

Overall achievements

The overall achievements can be broken down to topics that were covered through out the previous sprints. With the final topic being Object detection which is the major task for our group for this sprint. These major achievements are listed below:

Localisation and Mapping

The localisation and mapping tasks were completed early on in sprint 1 and 2 and are the basis for SLAM. Surprisingly SLAM was easier than just localisation or mapping as there are so many packages that can be used for SLAM but separating them for just the mapping or localisation proved to be a little difficult. The achievements of this were depicted in sprint 2 when I tested the AMCL localisation method against the ground truth which wasn't that great but another group members using the localisation method in the hector SLAM package worked perfectly.

Figure 5 AMCL vs Ground Truth From Sprint 2

SLAM

The team tested a bunch of different SLAM options in sprint 2 and settled on gmapping as the one we will be using. The champ package provided us with many pre configured launch files for different SLAM algorithms that we were able to further configure for our needs. Such as increasing the max range the SPOT robot was allowed to see and changing the update frequency. Gmapping was an effective SLAM method as spoken about in sprint 3. The figure below shows the comparison of the SLAM map compared to the gazebo map which proves how well the SLAM works.

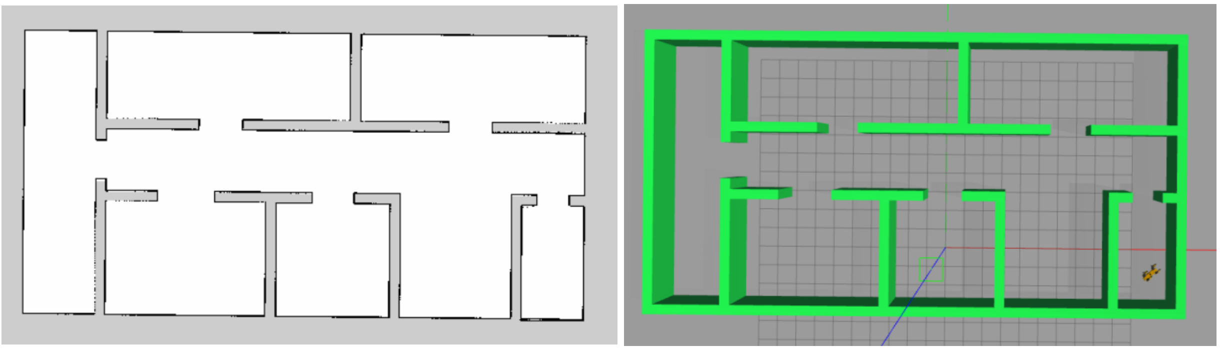


Figure 6 SLAM vs Gazebo World

Path planning

Our robots default path planning method is Dijkstra's which is explained in SLO 3.4 of Sprint 3. This is something that was built into our robot package. We do have the option to change it to A* but there is minimal difference for the added computational power we found it to not be useful. The path planning can be visualised in Rviz by the red line that can be seen in the image in the exploration image below.

Exploration

For exploring an unknown area the team is using a package called explore_lite which is a greedy frontier-based explorer. This means it will keep exploring its environment until no more frontiers can be found. We had great success implementing this package as it was as simple as downloading and running the explore.launch file.

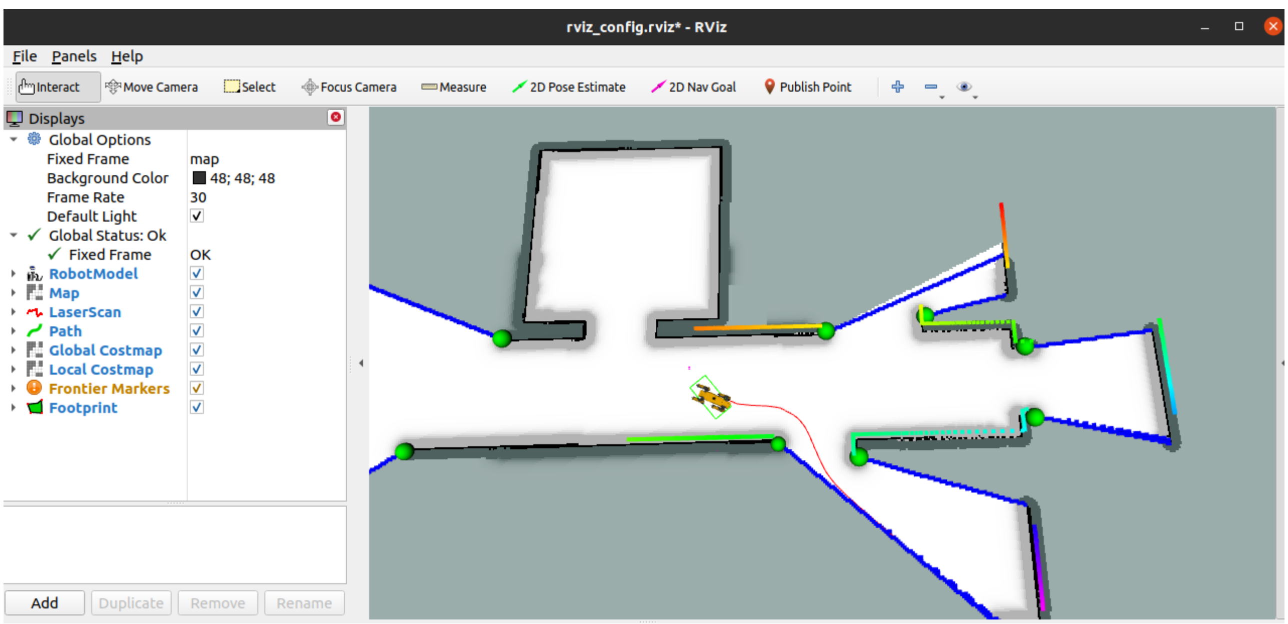


Figure 7 SPOT While Autonomous Exploring

Object Detection

The object we are detecting in this search and rescue scenario is people. As such there are many different algorithms that can be used to detect humans. The algorithm that I tried testing with was yolo which stands for you only look once. This package worked great and simply running the launch file would start detecting objects. As we were only detecting people, I could remove every other object from the parameters so it would only try and Identify a person. I did also try to run yoloV3 however this was very computationally heavy and would not load at all as it was trying to run on my CPU. There is an option to use the GPU however as this will be running on spot which does not have a dedicated GPU I thought best to not attempt this option. The below image shows initial testing of running the yoloV2 where it is detecting a person. In this exact frame the algorithm is 68% sure that it is detecting a person.

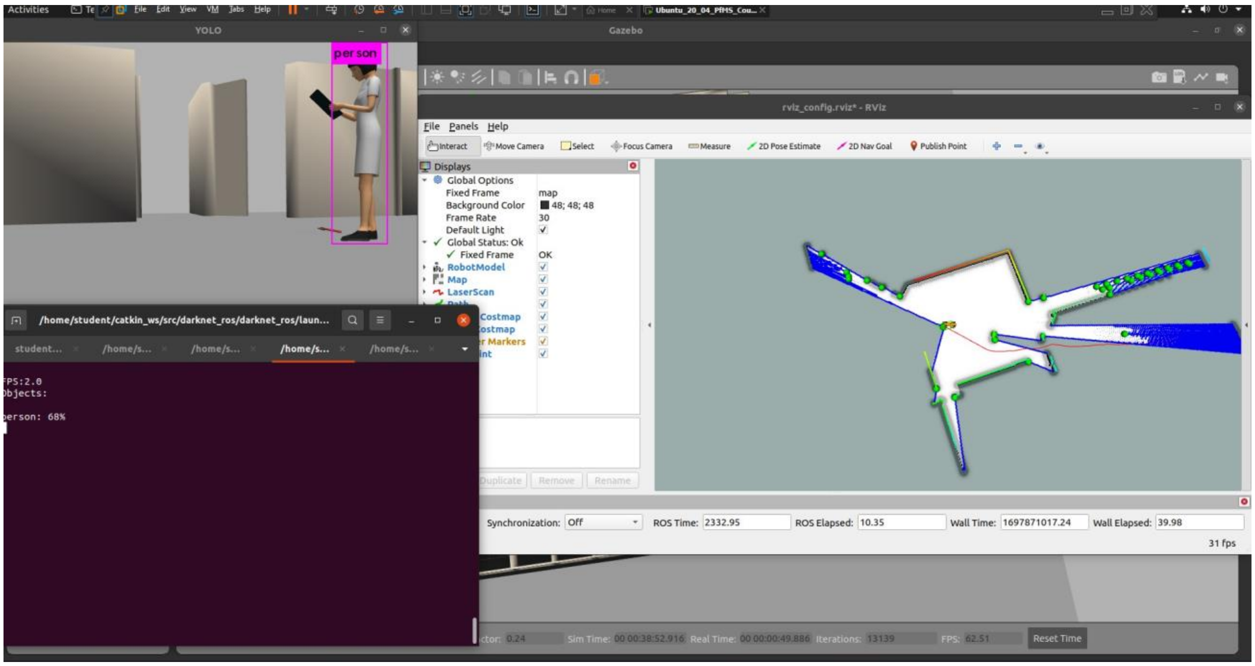


Figure 8 Object Detection With Yolo V2

You can see that it is running at around 2 frames per second which is quite slow and when combining all aspects this proves to become a problem. I managed to get the frame rate higher by lowering the camera size from 416x416 to 100x100. At this size the frame rate was around 20 however object detection proved to be harder due to less pixels as well as the bounding box that defines a person being offset from the actual person as seen in the figure below.

Figure 9 Object Detection With Lower Camera Resolution

Another group member went down the path of using OpenCV's trained models of people. It was found that this method worked very well as it was less computationally heavy than yoloV2 that I was running. When running this method the average frames were around 30 and used less processing power which will be ideal for when everything has to run at once.

Combining all aspects

The final step was to combine all these previous aspects of SLAM, Path planning, exploration and object detection together. The hardest of these being the object detection to leave a marker on the rviz map where it detected a person. Combining the rest of the steps was as simple as creating a launch file that launches the separate launch files for all the previous steps. The figure below shows initial testing of placing a marker when a person is being detected. To start with we just plotted a marker every time spot detected a person with over 70% certainty.

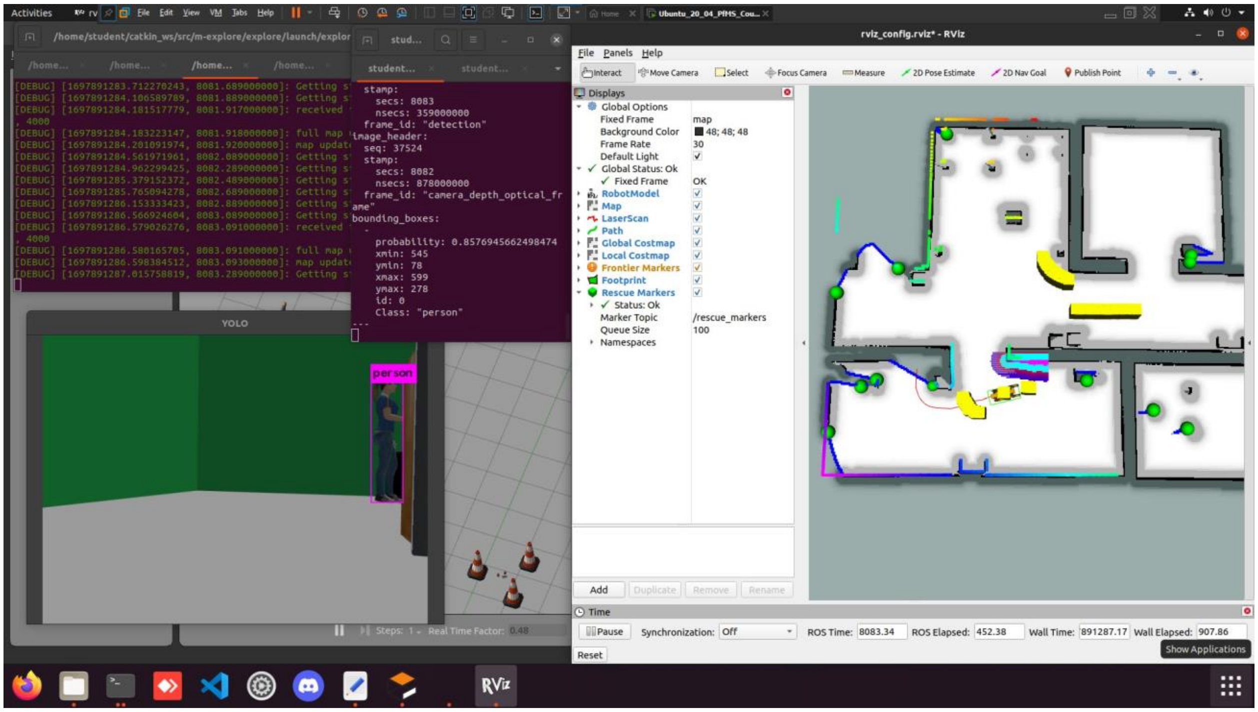
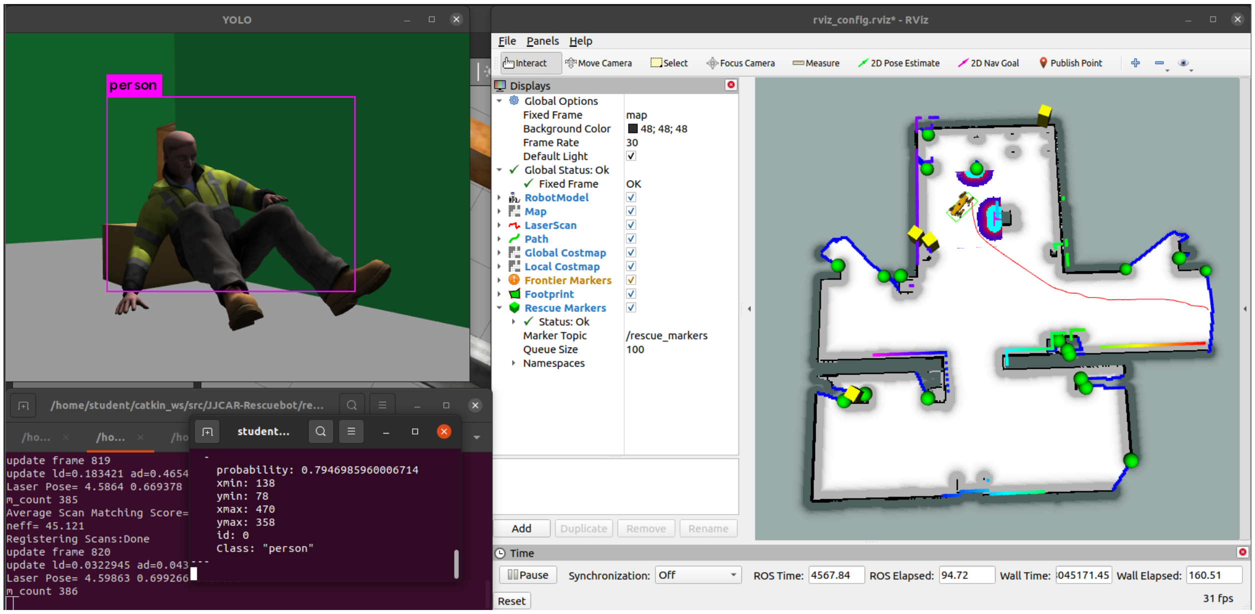


Figure 10 Testing Marker When Detecting Person

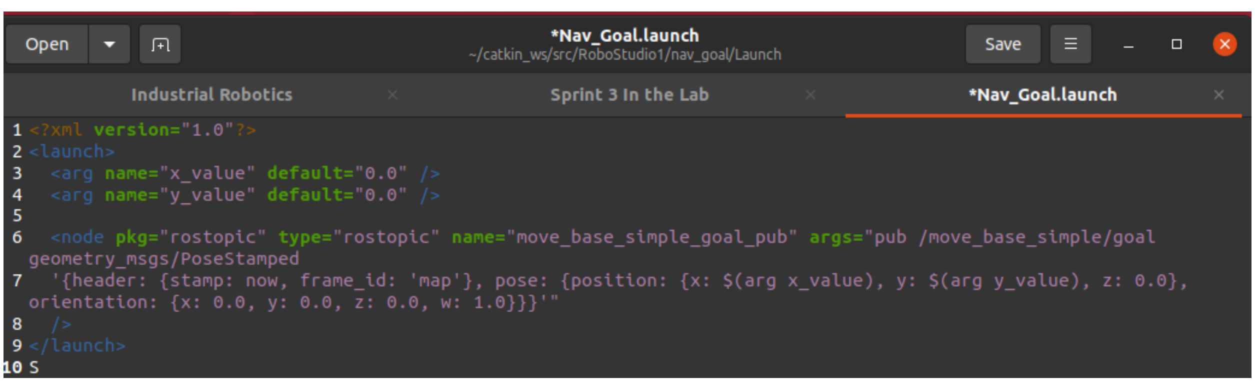
Some simple depth calculations can be done to get the position of the person in reference to the robot and then transform that into the global position we are able to place the markers where the person is. This is where we discovered the issue of the low frame rate with yolo. Because our frame rate is so low when placing a marker the point it chooses is based on the camera but as it is so far behind the spot robot the markers are in the wrong place. This issue was not encountered when using OpenCV's object detection as the frame rate was high enough. Below is an image of our program detecting humans and placing a marker on them.



Due to the uncertainty with OpenCV's trained models we did end up using yolo V2 with a lower camera size to increase our frame rate. This we found to be the optimal and most reliable.

Creating Launch files

The below figure is of a simple launch file I created that publishes an x,y goal to the move_base_simple/goal topic of our spot robot. This was created for the sprint 3 path planning. The user can send a x y coordinates to the robot that it will navigate to. The command used with this launch file is "roslaunch nav_goal Nav_Goal.launch x_value:=-6.0 y_value:=12"




```
<?xml version="1.0"?>
<launch>
  <arg name="x_value" default="0.0" />
  <arg name="y_value" default="0.0" />

  <node pkg="rostopic" type="rostopic" name="move_base_simple_goal_pub" args="pub /move_base_simple/goal geometry_msgs/PoseStamped
  '{header: {stamp: now, frame_id: 'map'}, pose: {position: {x: $(arg x_value), y: $(arg y_value), z: 0.0}, orientation: {x: 0.0, y: 0.0, z: 0.0, w: 1.0}}}'"
  />
</launch>
S
```

Figure 11 Nav_Goal Launch File

I was very proud of my self when I was able to get it working as it is not something I had done before not did I know how to do.

Overall

Overall our project is working as intended and successfully marking the detected object. By using the evaluation criteria from SLO 2.1 we have been able to modify parameters of all aspects through trial and error to work out what the best possible combination is. The below table is of our group's final parameters.

Evaluation Criteria	Value	Weight	Total
Mapping			
Map Accuracy	1.00	0.3	0.3
Path Planning			
Obstacle Avoidance	0.95	0.2	0.19
Time Efficiency	0.75	0.1	0.075
Object Detection			
Identification	1.00	0.3	0.3
Computational Power			
Computational Cost	1.00	0.1	0.1

Our group is very happy with these results as we are completing every criteria to a very high degree. A total score of 96.5% was calculated which shows there is still room for improvement in many areas however we do cover all deliverables set out by our stakeholder.

SLO 5.1 Critical reflection on the subject learning

I quite enjoyed the subject as a whole. I found it quite easy to achieve the subject learning objectives as they are laid out very well with great descriptions, so it was easy for me to understand what it is I need to complete to achieve each learning objective.

The only challenges related to the learning objectives was in the first two weeks of the semester when the SLO's were not sprint based but subject based and we would have had to split it up and sort it out ourselves. This was quickly solved when the subject coordinator split up the SLO's for us based on sprints.

For the remainder of the subject the learning was both interesting and intriguing. I thoroughly enjoyed the content of SLAM and path planning. There were some difficulties with downloading some git packages such as cartographer however, these were overcome quickly by discussing with peers and tutors or using another package that does a similar thing that didn't have install errors. For this case the team decided to use gmapping instead for our SLAM algorithm.

Overall I really enjoyed the subject especially the content and the way it was run.

SLO 5.2 Statement of what you will you do differently.

There is not much I would change given the chance to do the subject again. I have achieved marks I am very happy of so far and hopefully will continue this for the final sprint submission. After doing other sprint format subjects I knew the key to success was to start early and never leave things to the last minute as things will always take longer then expected. As such getting majority of the work done over the first few days of receiving the sprint put me in a really good position with my other subjects and I was able to maintain a low stress level. The only thing I would do differently is spend more time furthering my understanding of topics and playing around with the packages I installed. While this wouldn't change my mark it would give me a better understanding of the topics and how things works which will benefit me in my other subjects and in my future working career.

SLO 5.3 Peer assessment

Group 5 demonstrated a deep understanding of the challenges associated with the utilization of turtle bots in adverse conditions and detecting hazards. They excelled in the thorough comparison of various mapping algorithms, showcasing a strong grasp of the underlying concepts and their decision process. Their ability to articulate comprehensive responses to the questions posed during the presentation was notable, reflecting their in-depth knowledge of the subject matter.

Furthermore, group 5 displayed an insightful discussion on potential avenues for future research, demonstrating a forward-thinking approach to the topic. However, it was observed that certain members exhibited a greater level of confidence and proficiency in communicating the content, leading to a more comprehensive understanding of the material. Encouraging greater participation from all members could further enhance the overall group dynamic and promote a more inclusive learning environment.

Overall, Group 5's presentation was informative and well-researched, showcasing their dedication to exploring the complexities of navigating turtle bots in challenging conditions and emphasizing the importance of hazard detection. I really enjoyed the heat map style plotting for hazards.

SLO 5.4 Concluding statement.

In the project, the team accomplished the creation of a software package that enabled a quadruped robot, specifically SPOT, to navigate through an office-like environment for search and rescue purposes. This software likely involved the integration of various technologies such as ROS (Robot Operating System) and Gazebo, indicating a sophisticated understanding of robotics and simulation environments.

Personally, I gained a significant proficiency in ROS and Gazebo, along with the ability to create, modify, and utilize launch files. Moreover, I improved my skills in cloning git packages and comprehending their functionality, as well as integrating them cohesively.

Looking ahead, I anticipate that these newly acquired skills will be highly beneficial as I complete my degree. Moreover, I believe these skills will be useful in my professional career. The ability to navigate complex software packages, understand integration processes, and effectively utilize various tools are essential competencies in the field of robotics and software development. Additionally, I recognize that these enhanced skills can serve as valuable assets, contributing to the enrichment of my resume and professional networking profile, such as LinkedIn. I am confident that this could potentially broaden my career opportunities and help me stand out in the competitive job market.

References

Chinenov, T. (2019). *Robotic Path Planning: RRT and RRT**. Retrieved from The Classy Tim: https://theclassytim.medium.com/robotic-path-planning-rrt-and-rrt-212319121378

Navone, E. C. (2023). *Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction*. Retrieved from Free Code Camp: https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/#:~:text=Dijkstra's%20Algorithm%20finds%20the%20shortest,node%20and%20all%20other%20nodes.

S, R. A. (2023). *A* Algorithm Concepts and Implementation*. Retrieved from Simple Learn: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm#:~:text=It%20is%20a%20searching%20algorithm,can%20find%20its%20own%20course.