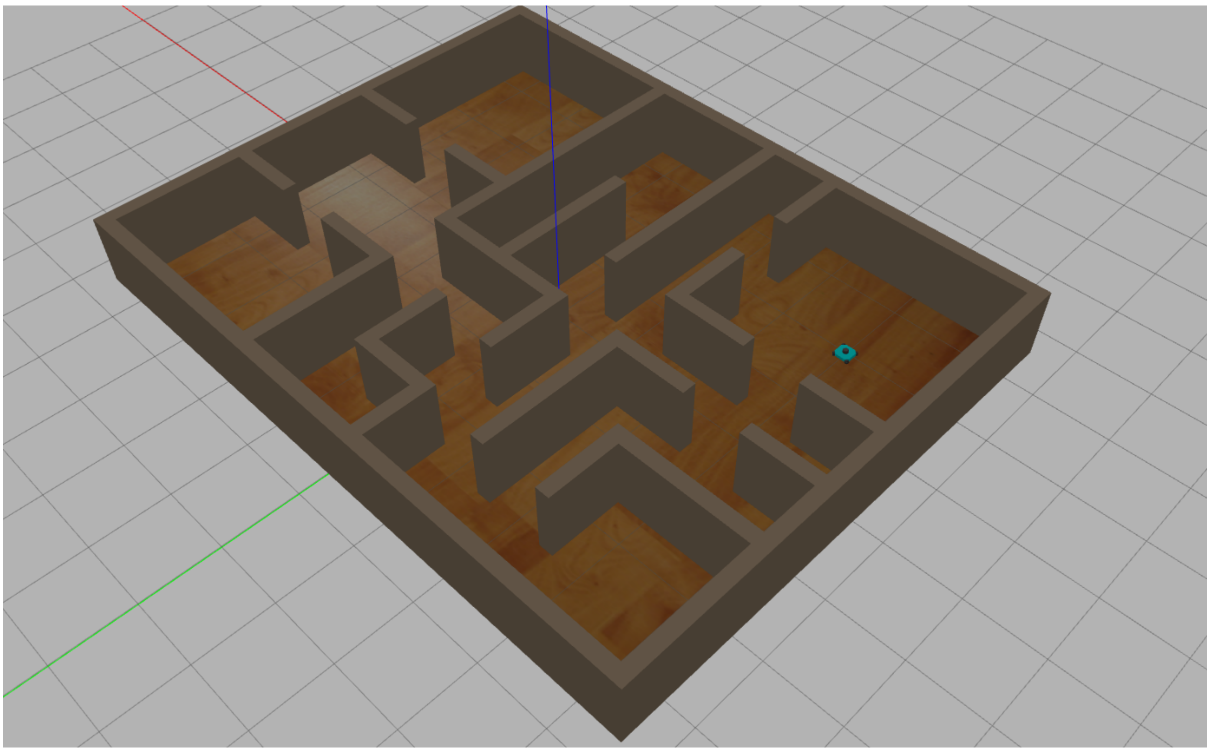
Project APR - Autonomous Path Finding Robot

Arthur Gomes
Dept of Mechatronics Engineering
MIT Manipal
Reg number 190929090
Contact: arthurgomes2727@gmail.com
Faculty Guide: Dr Jeane Marina Dsouza

Introduction

Robot path planning is an area of ongoing research and development. The industry and by extension the consumer market always presents new challenges for engineers. A novel algorithm for autonomous point to point motion in an obstacle filled environment is the main point of focus of this project as well as the holonomic robot base designed to implement it. The entire project is simulated in a Gazebo environment with ROS (Robot operating system) framework supporting the path planning, velocity controllers and sensor output channels.

Problem Statement

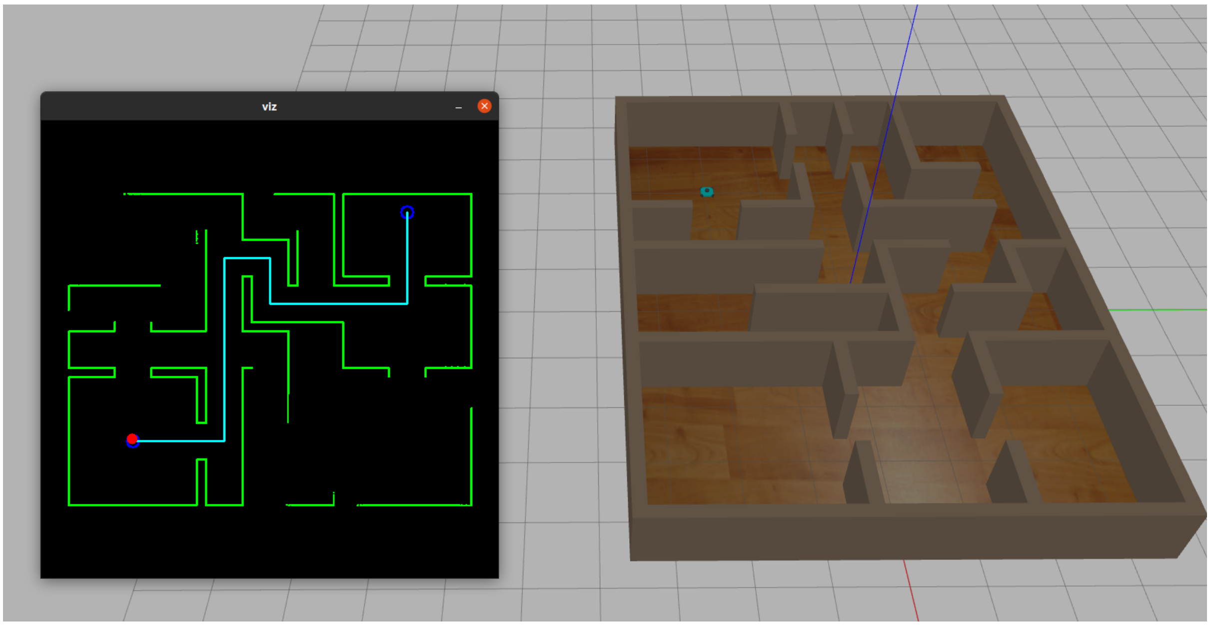


An indoor environment of rooms and corridors has to be traversed by the robot in order to calculate a path from its starting position to a given location in the floor layout. Thereby finding a path to wherever the desired location is. The robot must perform a mapping function as well to record explored areas for future reference. The robot must perform these functions as fast as possible with minimal error.

Real World Application

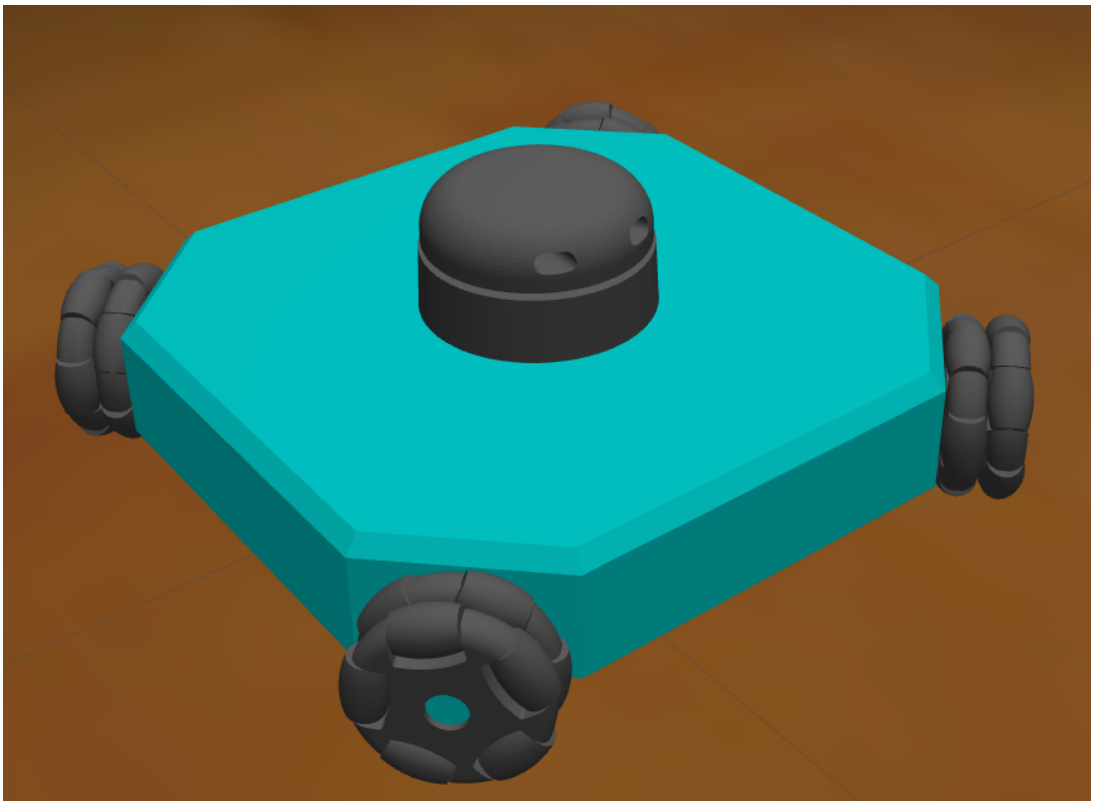
This problem statement is inspired by real-life scenarios where law-enforcement or fire rescue teams are faced with dangers and risks posed by unknown indoor environments. The robot is designed to assist in these kinds of situations by providing an expendable asset for reconnaissance.

Solution



The solution is presented in the form of a ROS(Robot Operating system) package that contains the necessary scripts and model data for simulating the world and the robot. The solution can be divided into two parts: the algorithm for navigation and the robot base.

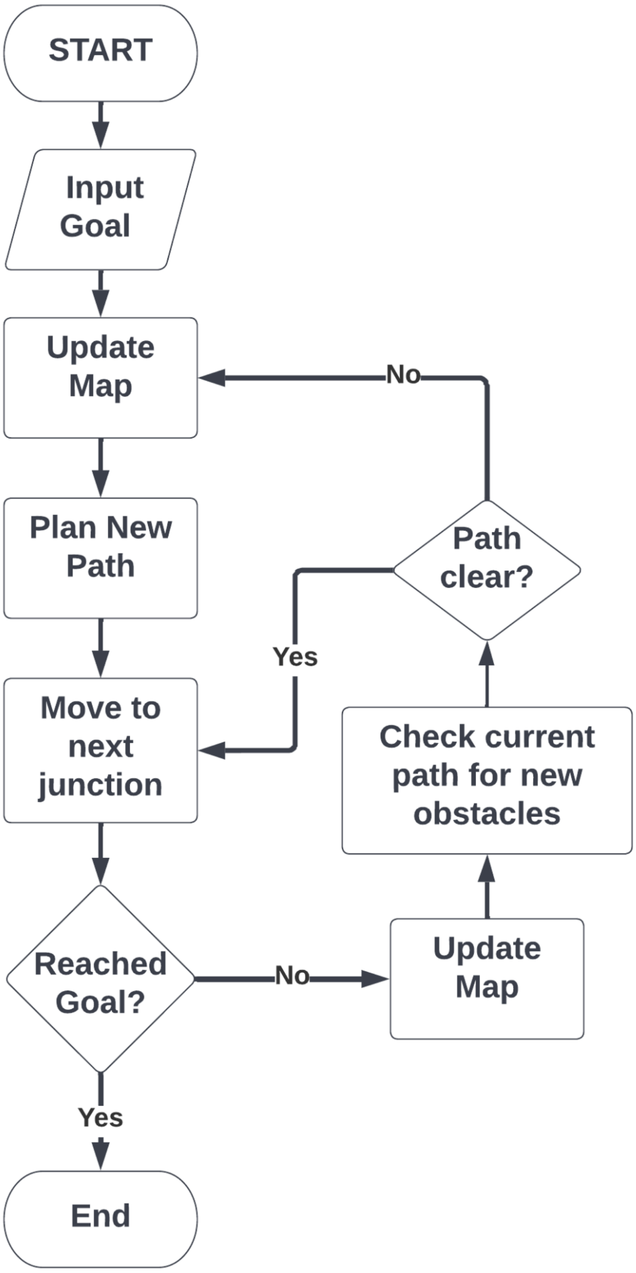
Robot



The proposed robot is a four wheel holonomic mobile base. It consists of a main body with four 58mm omni-wheels and a mounted Lidar sensor. The omni-wheels are used for their ability to rotate in a driving direction and translate in a sliding direction simultaneously. The Lidar Sensor is the Slam Tec A2M8. Readings from the lidar are used to update the map of the environment. Robot localisation is provided by wheel odometry, wherein the speed of each wheel is integrated over time to track robot position and orientation.

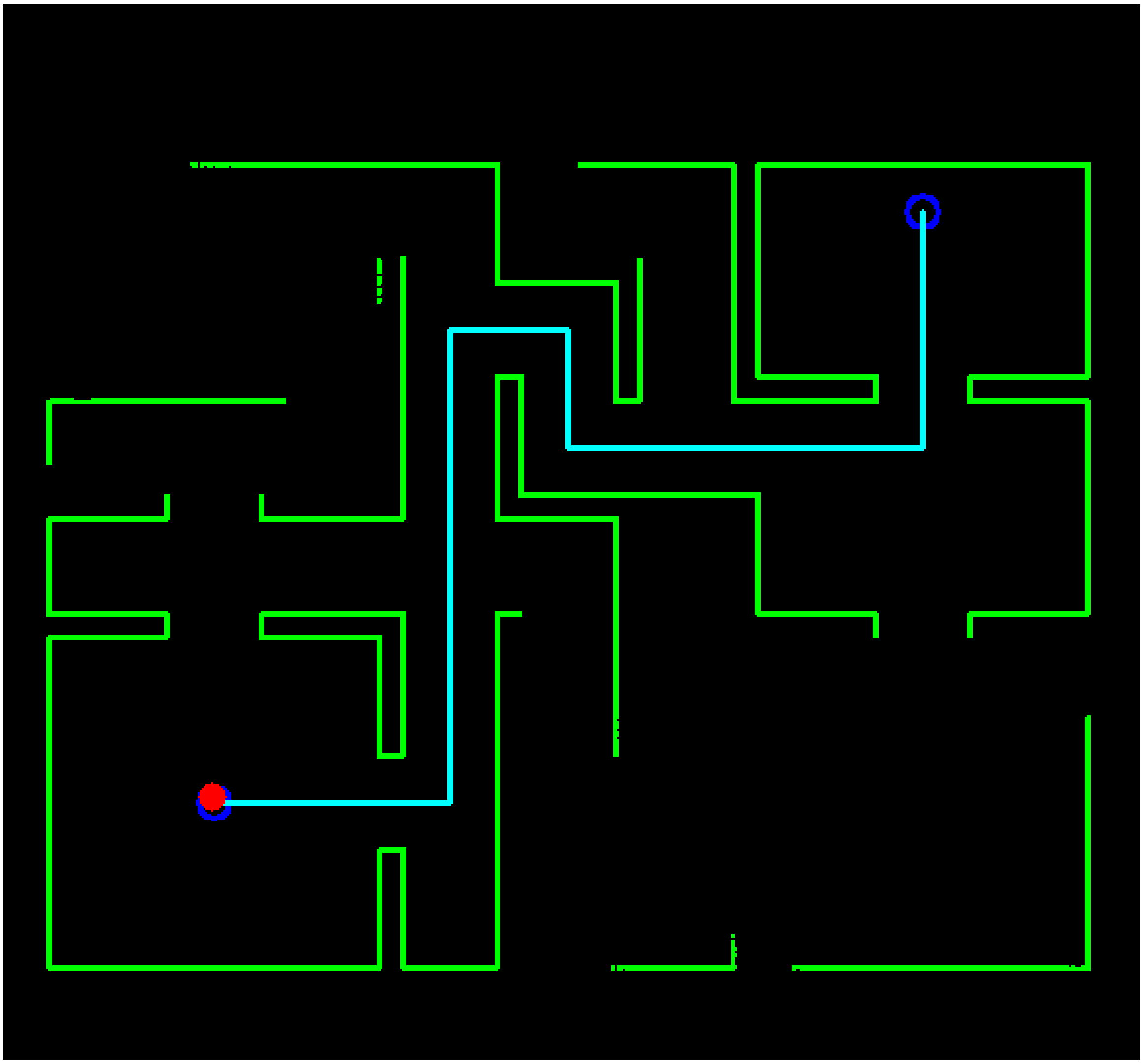
Navigation algorithm

The Navigation algorithm consists of two main components: a mapping algorithm and a search algorithm.



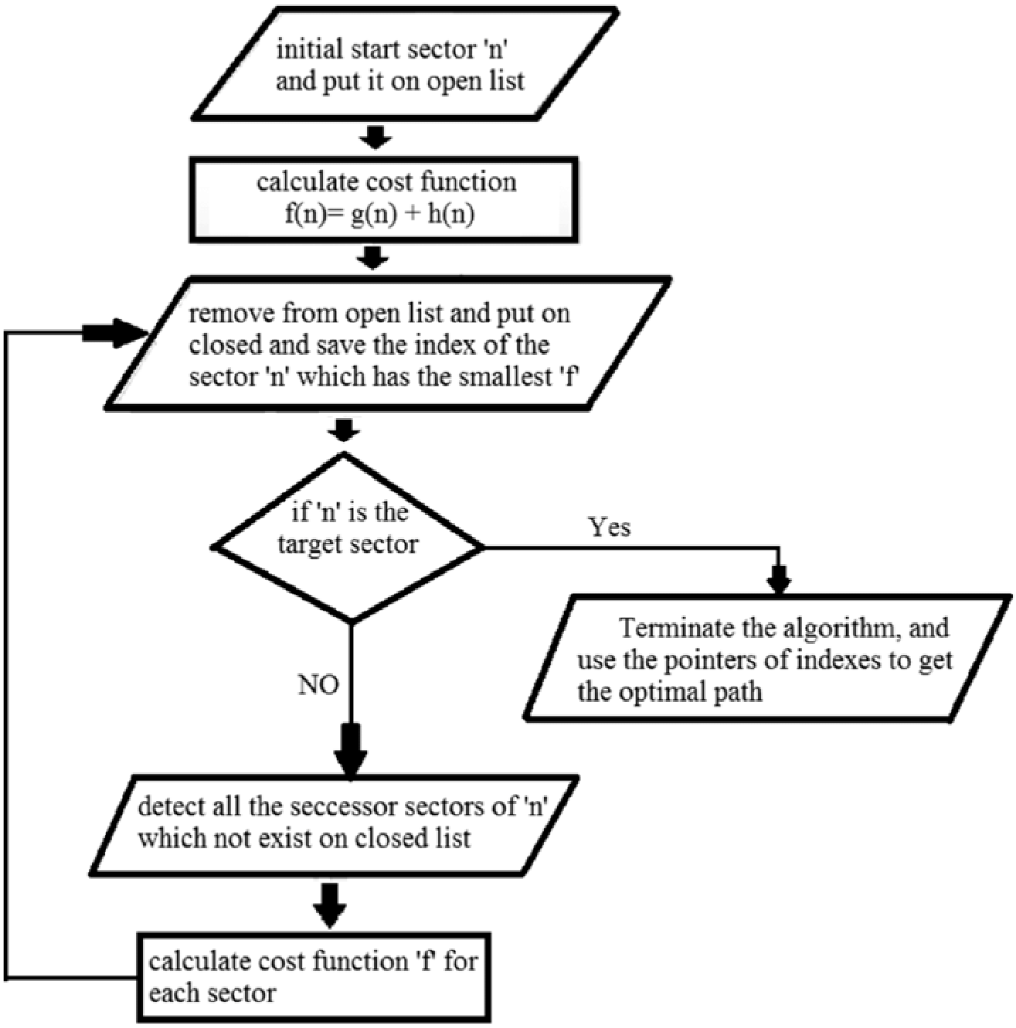
Mapping Algorithm

To enable environment mapping, Lidar data is converted into a 2D point cloud which is then used to fill an occupancy matrix which can be tuned for environments with different dimensions. This allows for maximum computational efficiency. The map refresh rate is only limited by the physical capabilities of the Lidar Sensor itself.



Search algorithm

The A* graph traversal and search algorithm has been used to great effect in this project due to its completeness and optimal efficiency. The elements of the occupancy matrix are considered to be nodes and the heuristic function is the straight line distance of the node from the target location.



Velocity Controller

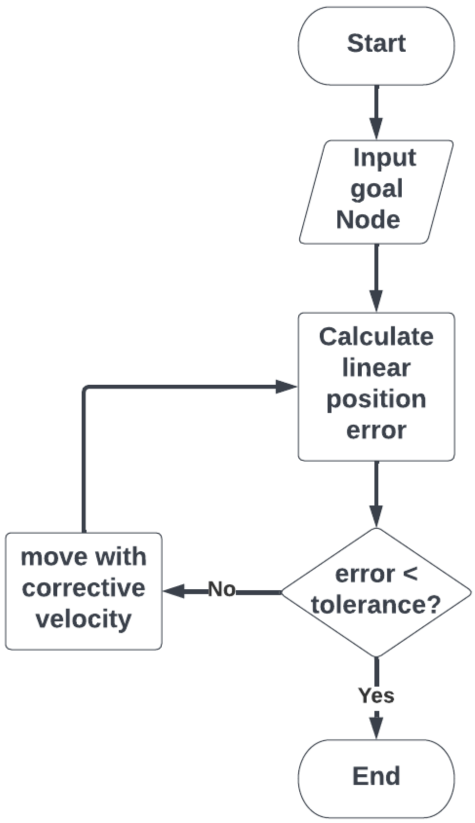
The velocity controller is used to actuate different wheels at different speeds depending on the desired velocity vector. The velocity controller used for this robotic base is governed by the equation.

$$u_i = h_i(\phi)\dot{q} =$$

$$\begin{bmatrix} \frac{1}{r_i} & \frac{\tan\gamma_i}{r_i} \end{bmatrix} \begin{bmatrix} \cos\beta_i & \sin\beta_i \\ -\sin\beta_i & \cos\beta_i \end{bmatrix} \begin{bmatrix} -y_i & 1 & 0 \\ x_i & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix}. \tag{13.5}$$

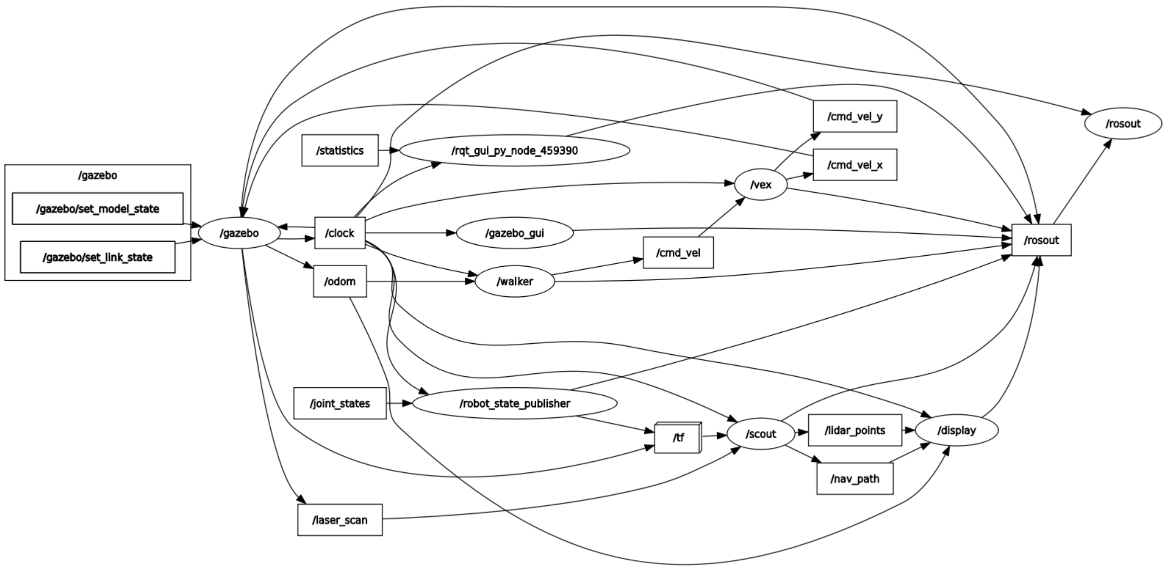
Position Controller

The position controller implemented is a linear error PID closed loop controller that uses odometry feedback and commands velocity twists to be executed by the velocity controller.



Overall ROS Network

The overall ROS network of processes (nodes) can be visualised by the following graph. The graph was generated during runtime of the package. It shows the relationship/flow of information between nodes as messages in topics.



Tools/Software Used

- **ROS noetic**
- **Ubuntu 20.04**
- **Gazebo 11.0.0**
- **Python3**

Coursera Courses Referred

- Modern Robotics, Course 1: Foundations of Robot Motion
  https://coursera.org/share/94815caa856013ea1261219a22ff72b3

- Modern Robotics, Course 4: Robot Motion Planning and Control
  https://coursera.org/share/a9be0d6df2cf184ba4d4abc160259c5e

- Modern Robotics, Course 5: Robot Manipulation and Wheeled Mobile Robots
https://coursera.org/share/bff771cac21da7397de584ed45ab118f

- Python Basics
https://coursera.org/share/4c16b47b49545913bf430248b81a8fc8

- Python Data Structures
https://coursera.org/share/b03a8e71c7e448bfa7b30a753642b7d2