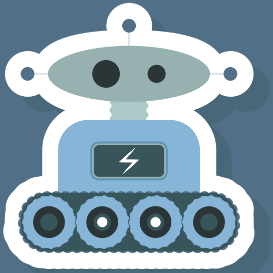
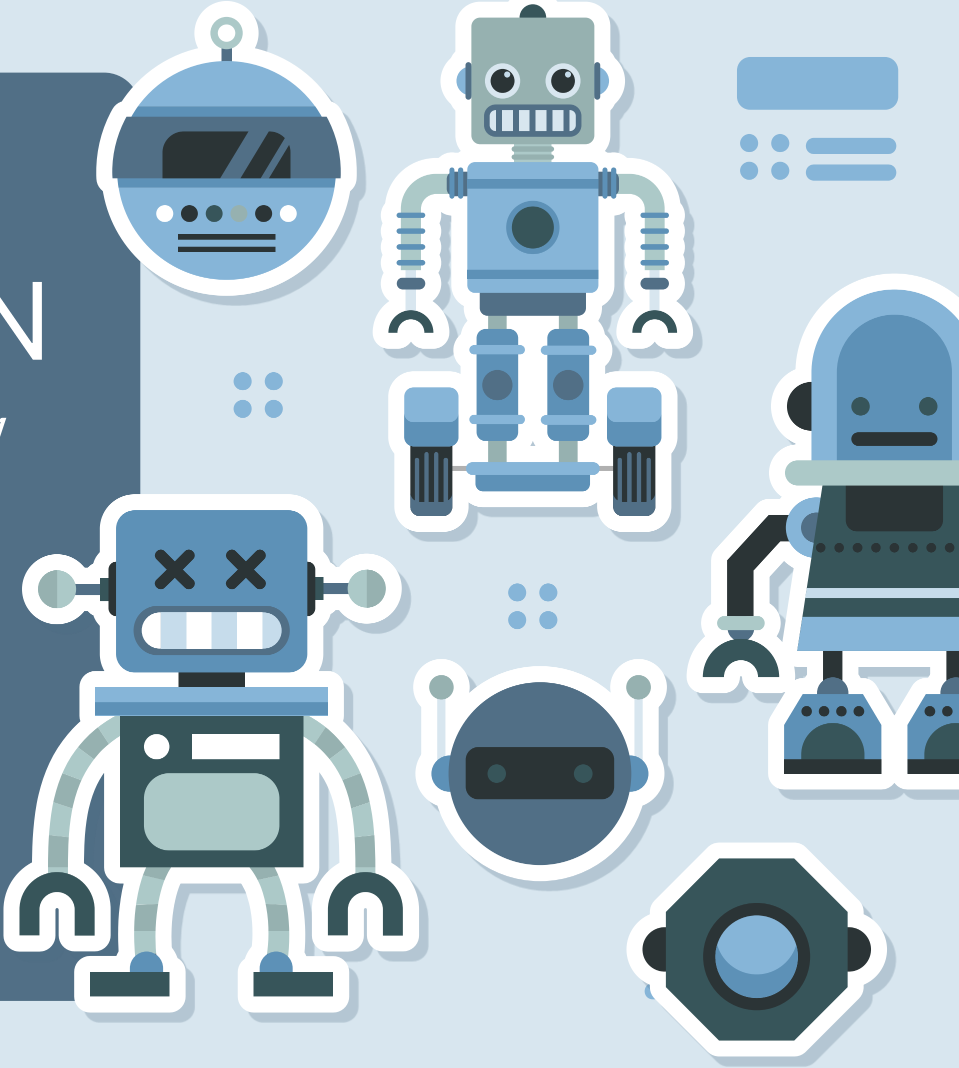
FINAL PRESENTATION

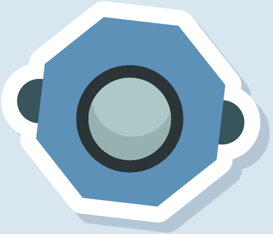
Introduction to Robotics - Group 01

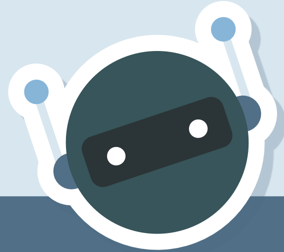
Miguel Miranda - ist1113191
Raquel Cardoso - ist199314
Tomás Pereira - ist1112273





WORK DEVELOPED





01

MINI-PROJECT 1

Mobile Robot Localization

02

MINI-PROJECT 2

Mobile Robot Path Planning

MINI-PROJECT 1

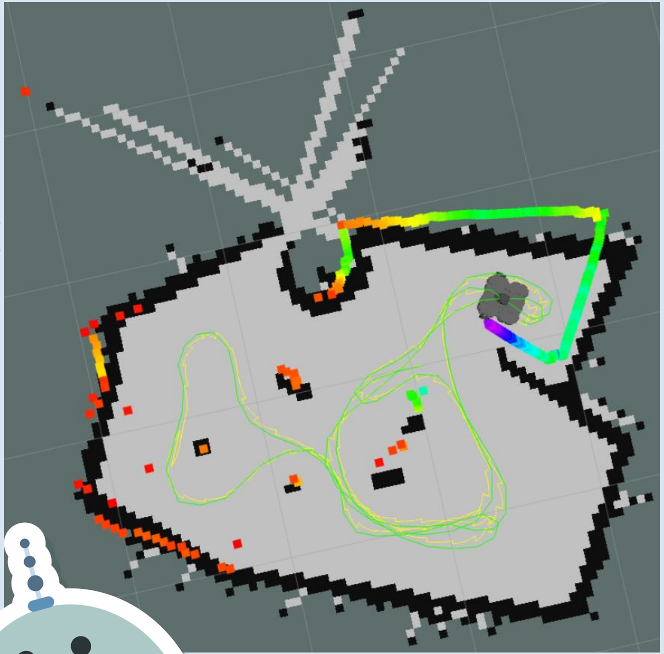
EKF

Uses probabilistic methods to merge data from multiple sensors, refining the robot's position and orientation estimates.

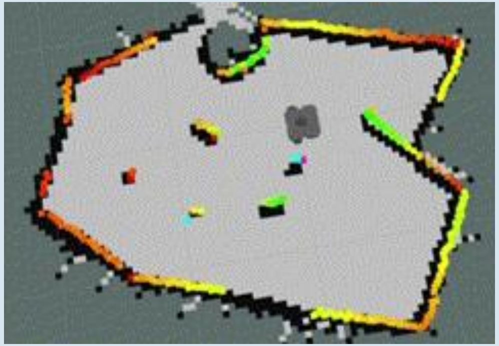
AMCL

Particle filter approach that allows flexible, probabilistic localization on a pre-recorded map, with a focus on dynamic environments.

EXTENDED KALMAN FILTER (EKF)



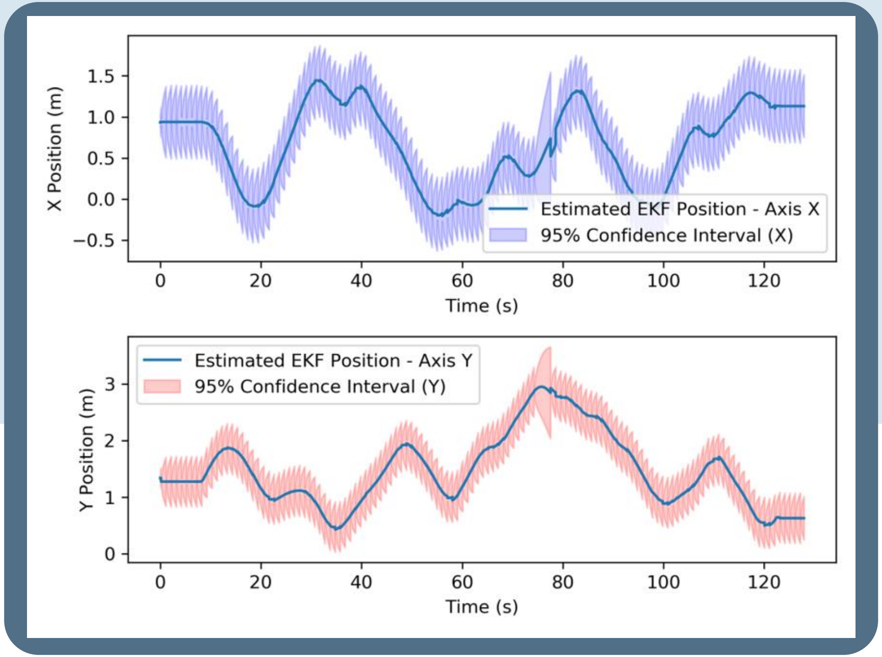
- data from odometry, IMU and a high-precision sensor:
  a. odometry tracks movement speed and direction
  b. IMU measures rotational movement
  c. high-precision sensor provides periodic absolute position updates to correct for drift in other sensors

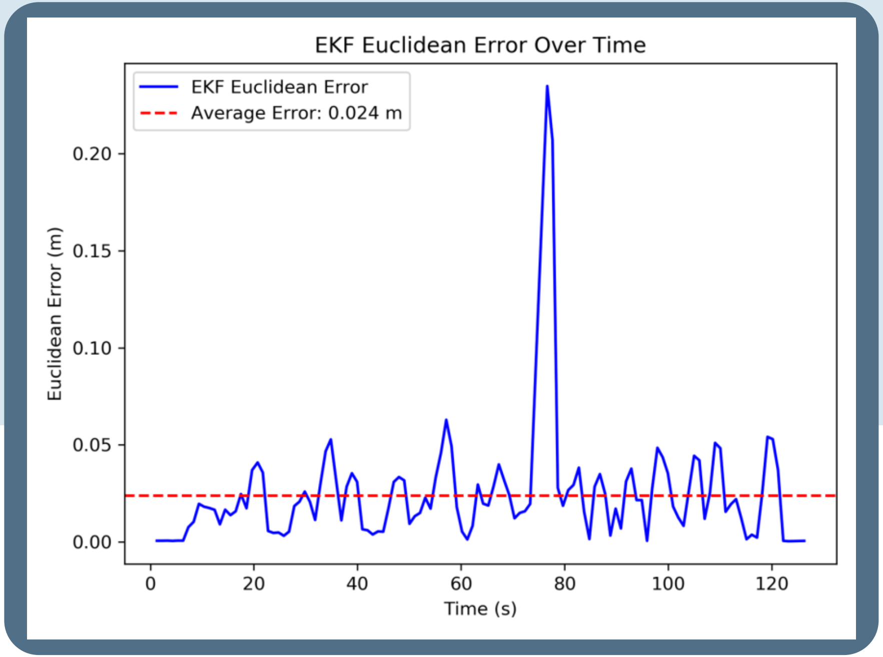


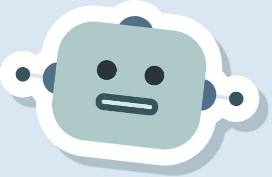
🟩 EKF estimated path

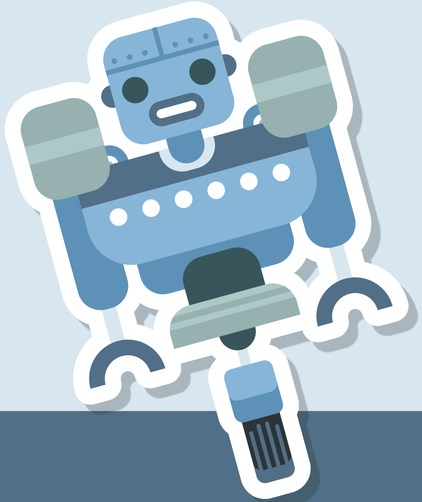
🟨 High-precision sensor path

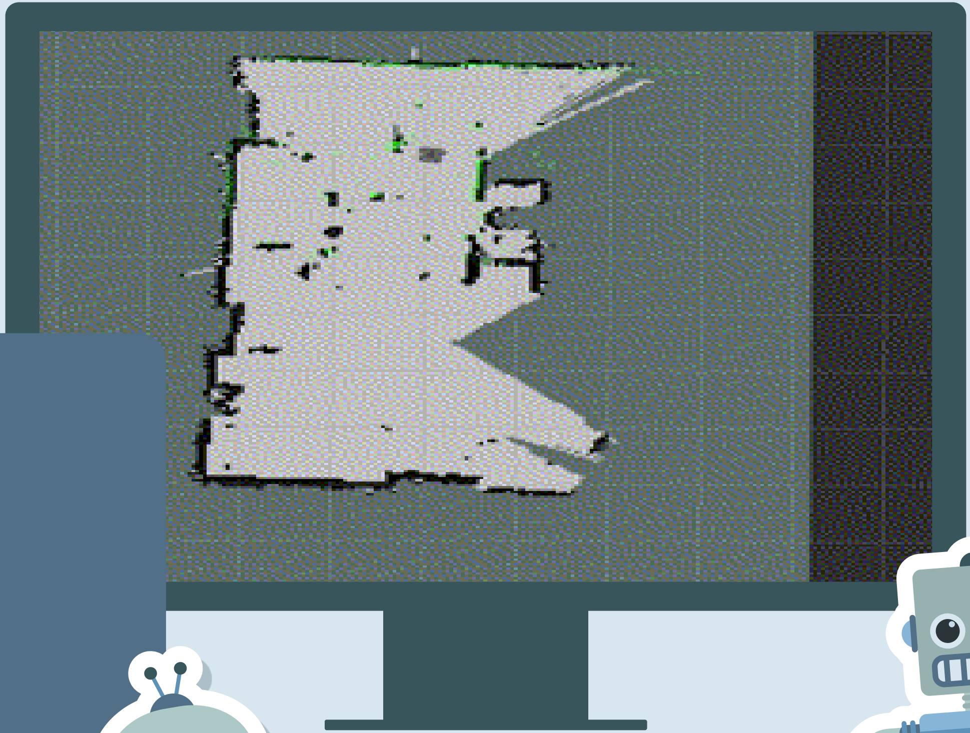
EKF ERROR ANALYSIS







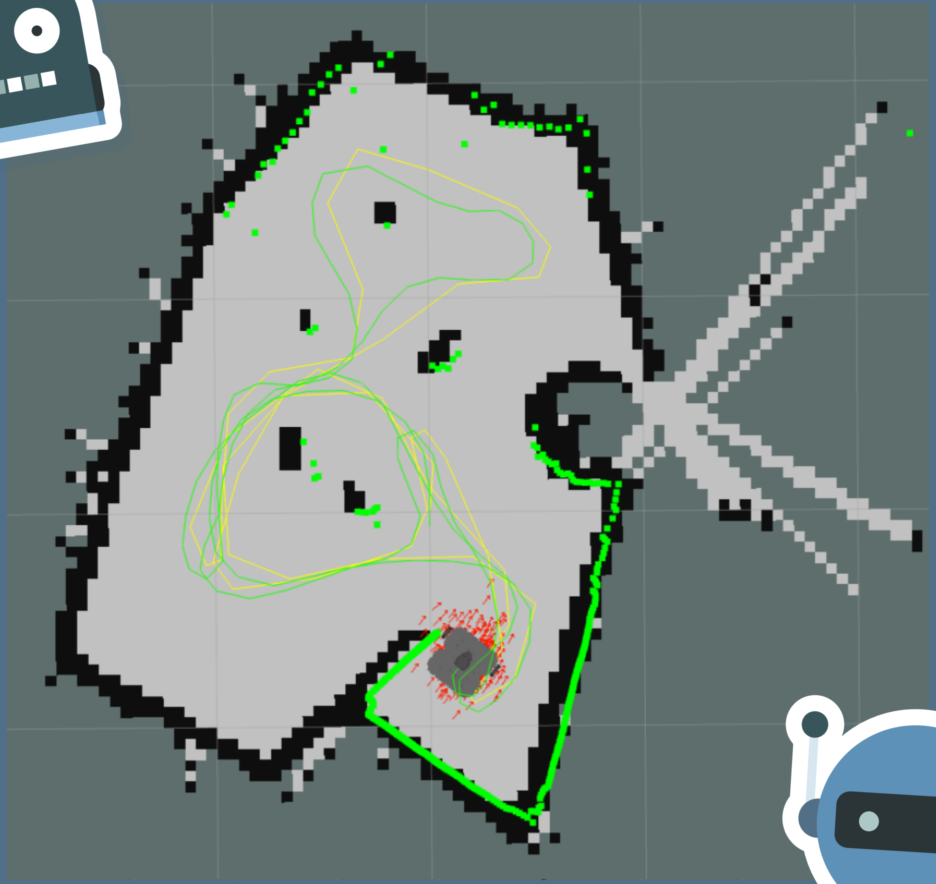




GMapping

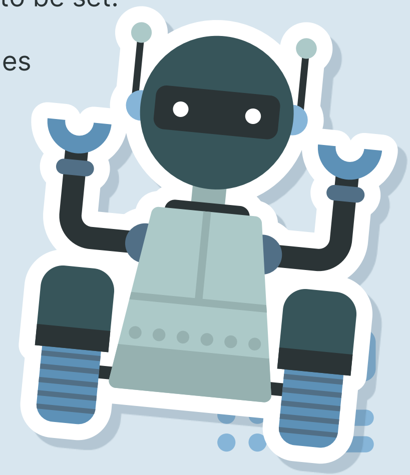
SLAM algorithm that uses laser scan and odometry to create a visual representation of the environment

ADAPTIVE MONTE CARLO LOCALIZATION (AMCL)

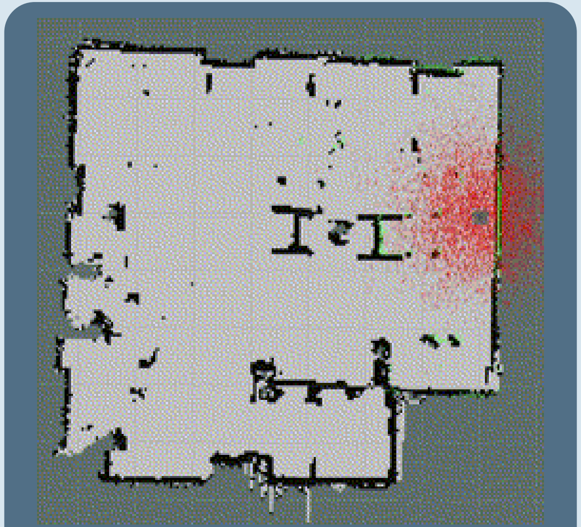


🟩 AMCL estimated path 🟨 High-precision sensor path

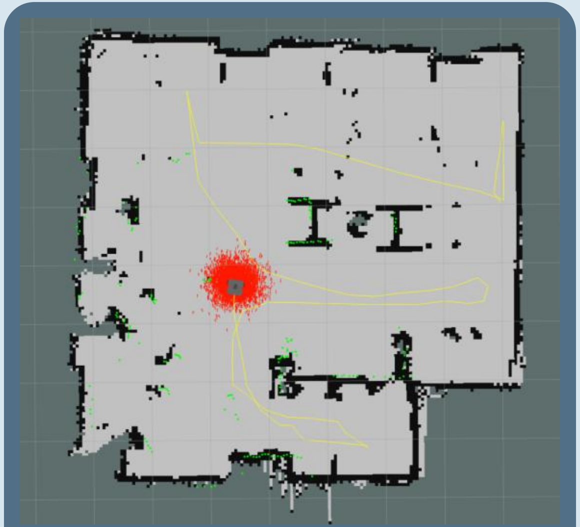
- AMCL gives us robot localization and orientation with the information from:
  - Map
  - Laser Scanner
  - Odometry
- Parameters that needed to be set:
  - Initial Covariance
  - Number of Particles
  - Initial Pose



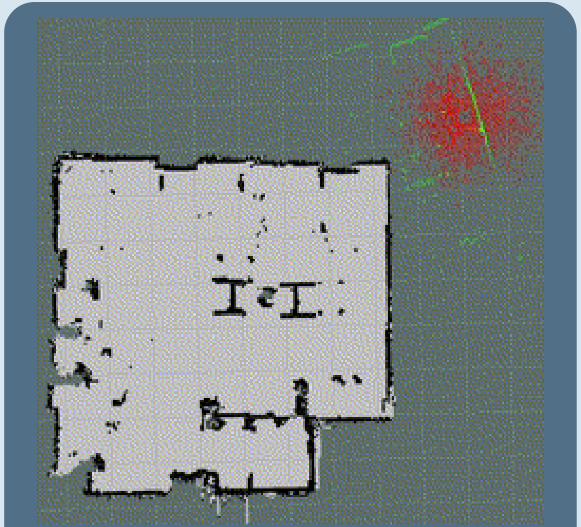
ADAPTIVE MONTE CARLO LOCALIZATION (AMCL)



AMCL behaviour with normal starting position

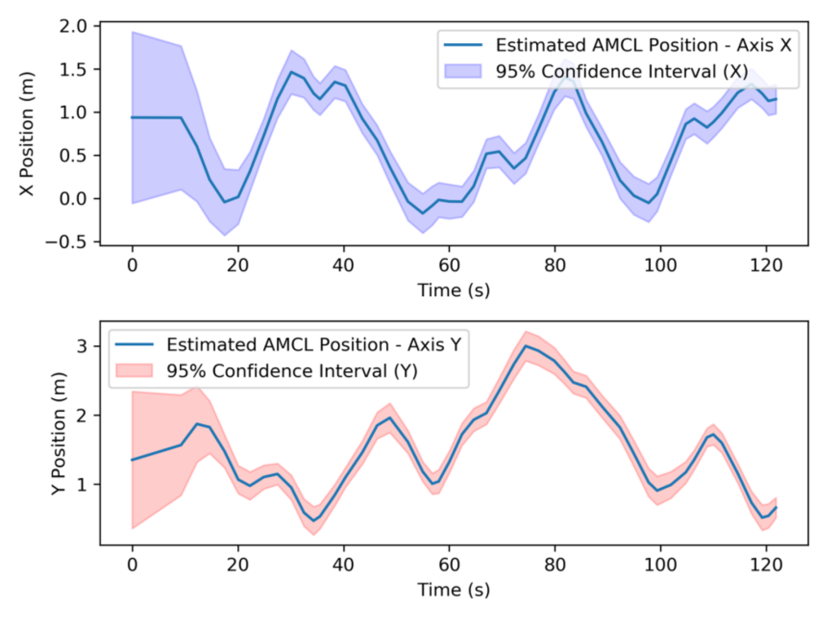


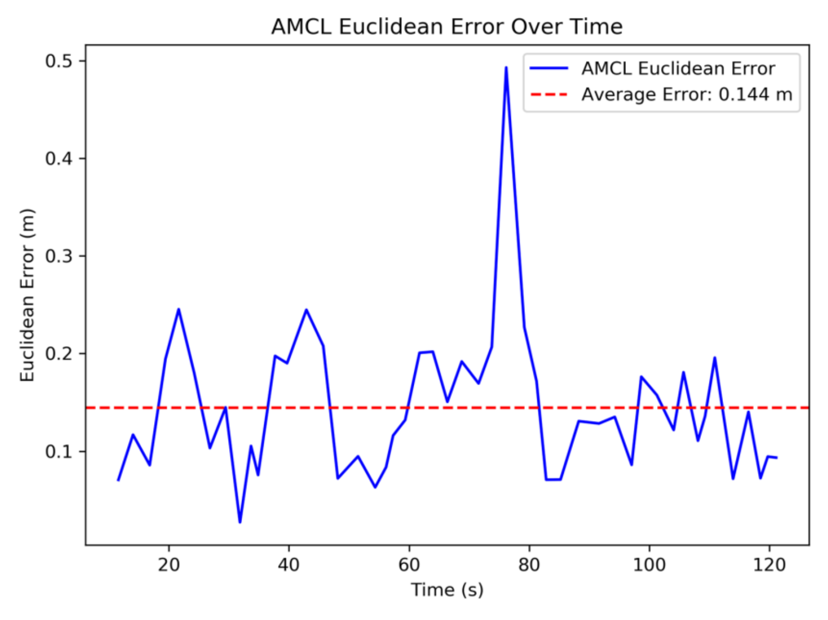
Final Path described by the AMCL algorithm

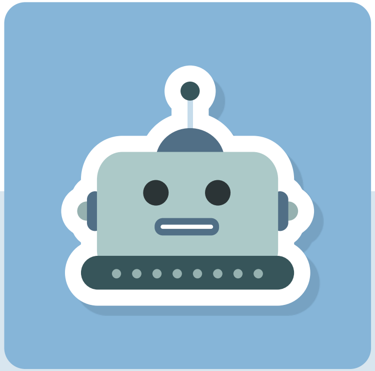


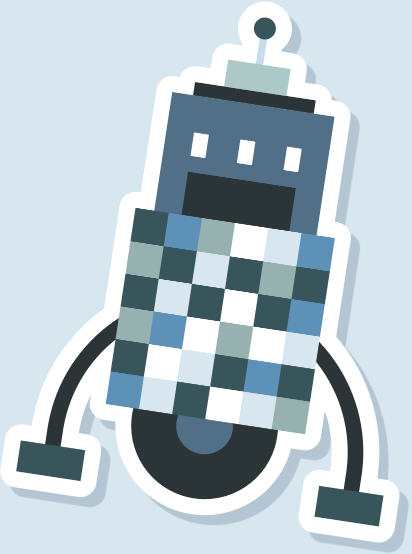
AMCL behaviour with wrong starting position

AMCL ERROR ANALYSIS



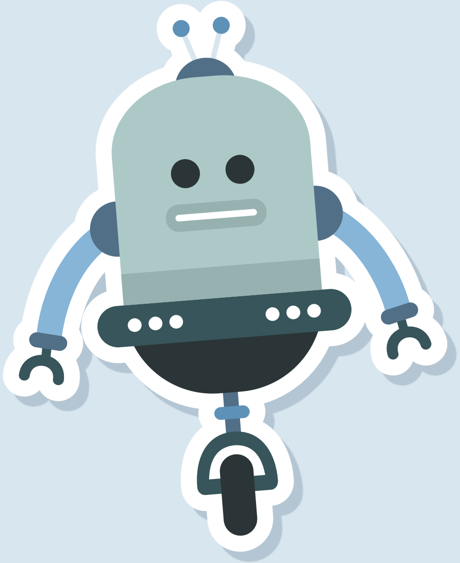




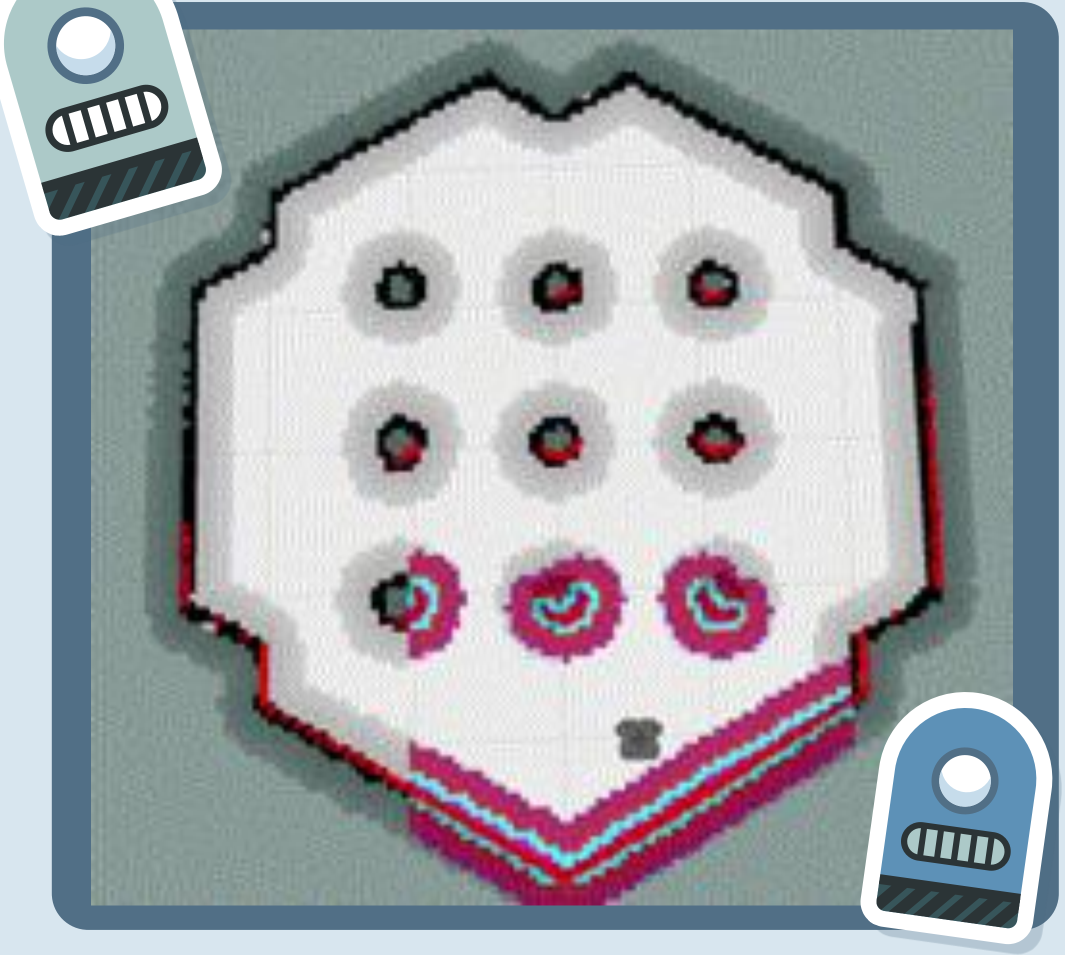


MINI-PROJECT 2

Pathing algorithm with RRT and Cost2D mapping.



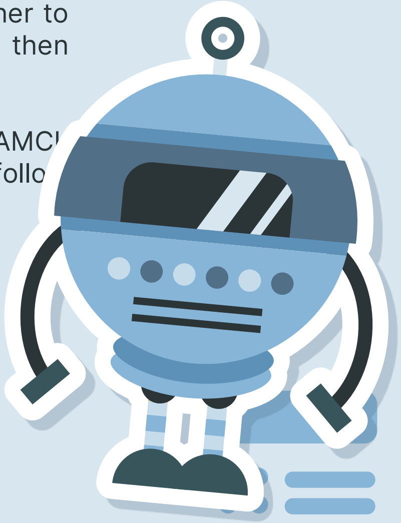
RAPIDLY EXPLORING RANDOM TREE (RRT)



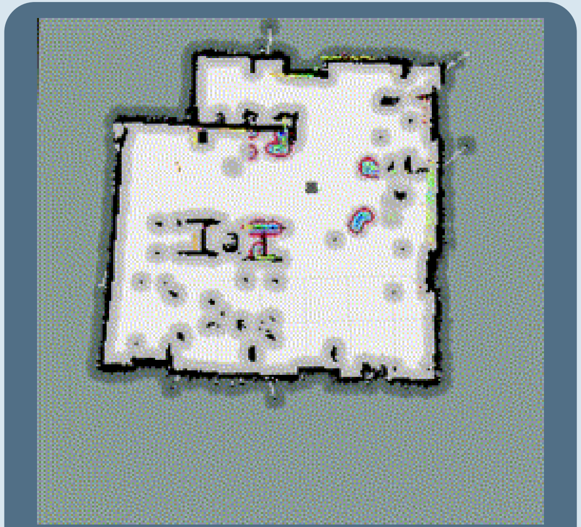
⬛ RRT planned path 🟩 AMCL estimated path

RRT is used as a global planner to calculate the path which is then shown in RVIZ.

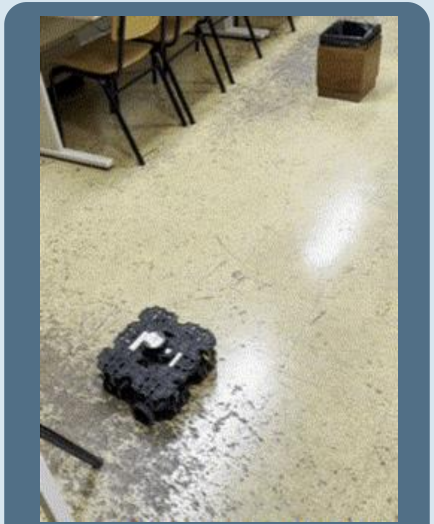
DWA local planner and AMCL are then used to try and follo the calculated path.



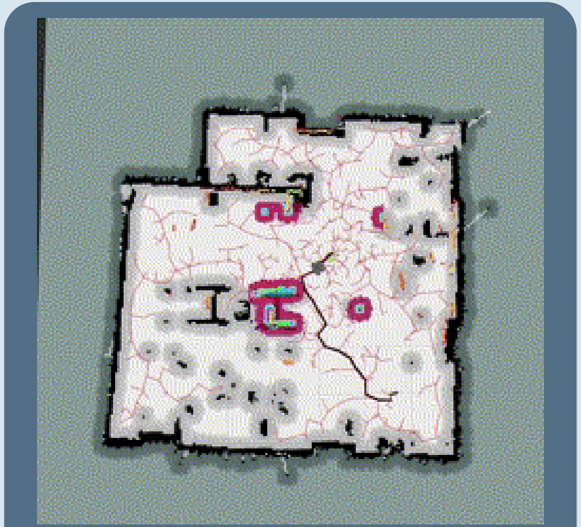
LOCAL AND GLOBAL COSTMAPS



RRT behaviour without unpredicted obstacles - RVIZ



RRT behaviour with unpredicted obstacles - Real life



RRT behaviour with unpredicted obstacles - RVIZ

STEP BASED METRICS COMPARISON

Step Size	0.1	0.3	0.5
Best Case	0.121	0.133	0.144
Average	0.296	0.265	0.318
Worst Case	1.234	0.553	0.568

Execution time per step size (seconds)

Step Size	0.1	0.3	0.5
Best Case	2217	3282	3648
Average	4352	4710	5318
Worst Case	12678	7688	9086

Number of attempted nodes per step size

STEP BASED METRICS COMPARISON

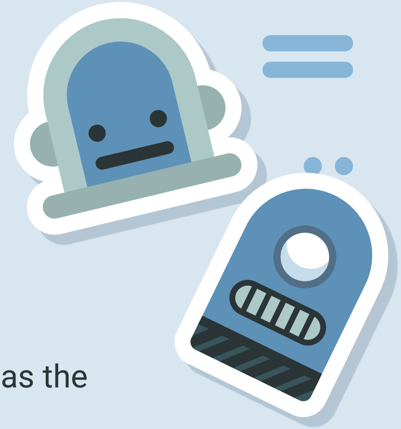
Step Size	0.1	0.3	0.5
Best Case	76.95	84.64	85.80
Average	83.25	86.96	87.32
Worst Case	86.39	87.65	87.80

Percentage of nodes blocked by obstacles

Step Size	0.1	0.3	0.5
Best Case	20	7	5
Average	48	17	12
Worst Case	79	34	24

Number of nodes per path

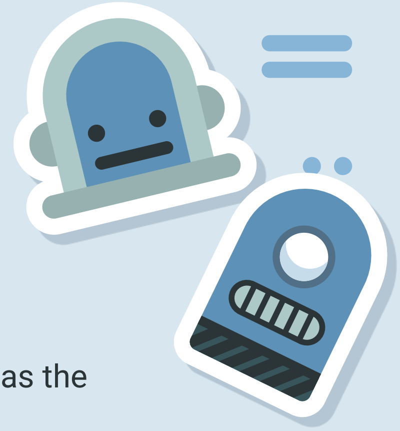
RRT PLANNER PARAMETERS



These were the **final parameters** used in the RRT planner after **various experiments** to find what was the best setup while maintaining both **computational efficiency** and **precision in navigation**

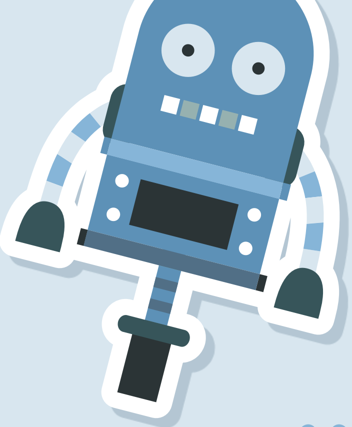
goal_tolerance = 0.2	How close the planner needs to be to the goal for the plan to be successful
step = 0.3	Distance between consecutive points in the path generation process
min_num_nodes = 500	Minimum number of nodes the planner will generate in the process
max_num_nodes = 5000	Maximum number of nodes the planner will generate in the process

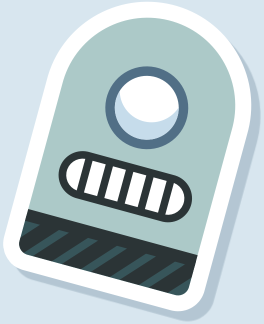
MOVE BASE PARAMETERS



These were the **final parameters** used for the Move Base after **various experiments** to find what was the best setup while maintaining both **computational efficiency** and **precision in navigation**

conservative_reset_distance = 0.3	The distance around the robot within the costmap will be reset if a path is unable to be found
controller_patience = 15	Defines the time (in seconds) the controller will wait for the robot to make progress along the path before considering it stuck and initiating recovery
planner_patience = 5	Maximum amount of time (in seconds) the system will wait for a plan to be generated before giving up and potentially initiating recovery
controller_frequency = 10	Specifies the rate (in Hz) at which the controller will update the robot’s commands





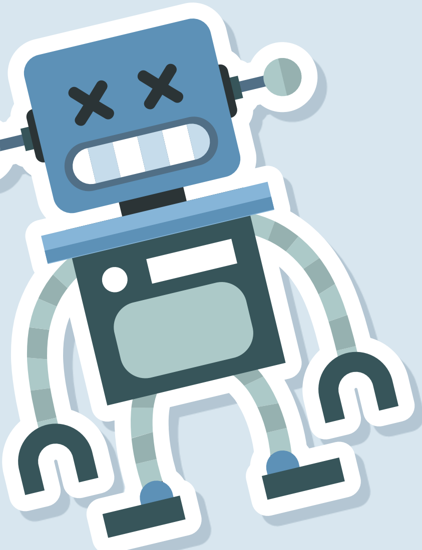
CONCLUSION

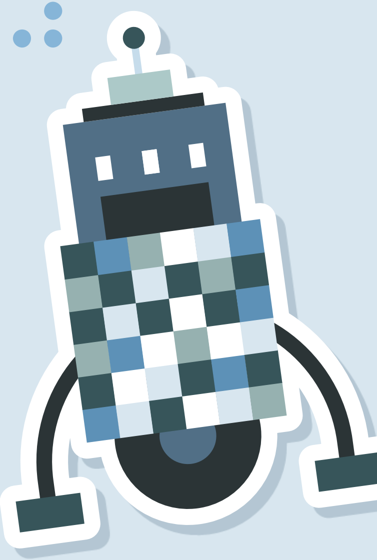
Localization:

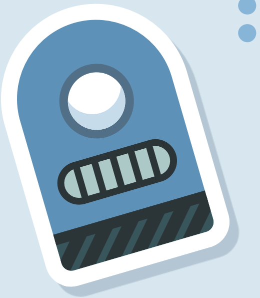
- EKF: highly accurate, especially with frequent sensor updates.
- AMCL: extremely valuable and adaptable, even with a extremely incorrect initial position.

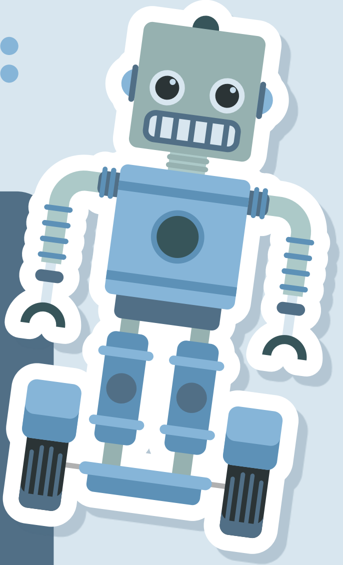
Path Planning:

- RRT combined with DWA enabled robust navigation in complex and unpredictable spaces.









THANK YOU