

Morgan Garbett

Ryan Wiesenberg

Lab 3

This implementation of A* is a class-based derivative with four main classes: Robot, Node, Map, and Astar. The Robot class is the carryover from lab 2 where we implemented a robot that takes in waypoints and then drives to that location with an open-loop controller. To adapt this, the robot now has a Map object composed of several GridCells of different types. These Grids are created based upon the map data and broken down into empty cells [threat = 0], obstacles [threat = 100], and unknown cells [threat = -1]. This grid is used to help with Astar. Every time the user inputs a PoseStamped via Rviz the robot creates a new adaptation of Astar for the new goal node based upon the current given map. The Astar path is broken down into Nodes which have a location, a parent, and a cost.

The Robot's position acts as the initial Node with the position being the nearest GridCell, no parent, and a cost of zero. The Astar algorithm then expands each node that is 8-connected to the previous node based on the map's GridCells increasing each nodes cost by the distance traveled center to center from the parent to the expanded node. These Nodes are arranged in a toVisit list in descending order based upon their cost and addition to the Euclidean distance to the target node. The beginning of the list is then repeatedly popped and expanded until it reaches the end. These nodes are also checked for validity against the list of obstacle nodes; if they are within the robot's diameter added to the location of the obstacle then they are considered invalid for Astar. The Astar algorithm then returns a node with all of its parents as the optimal path.

The waypoints are extracted from this path by repeatedly checking if the path required a change of orientation of the robot to get from Node to Node. If so, the Astar algorithm marks this Node as a waypoint and then will print out all the waypoints on the path if requested. Because of how the validity of points was assessed the actual robot could run between them with no problem.

Rviz interface during planning



