EECS 600, Algorithmic Robotics
Project 2 Proposal
Luc A. Bettaieb

A Goal-Subdividing Motion Planning System

For my second project, I will implement in simulation an algorithm that will able to take in the current position of the robot as well as the position of a global goal pose, subdivide the path into miniature separate goal poses for greater path accuracy, and incrementally reach the sub-goals such that the robot will eventually converge on the goal pose.

Velocity commands sent to the robot will be riddled with error so the sub-dividing of goals will be very useful in accurately controlling the path of the robot. Between these sub goals will be defined potential functions to impress upon the robot a desired path and sequence of velocity commands such that a smooth transition between sub-goal points will be attainable.

Error that drives the robot off-course from sub-goals will be handled intrinsically via the nature of the local potential functions; effectively creating a sink that drives the robot to the next local goal. Upon reaching the goal (within some threshold), the robot will then be given a new goal and follow the potential function to that one accordingly.

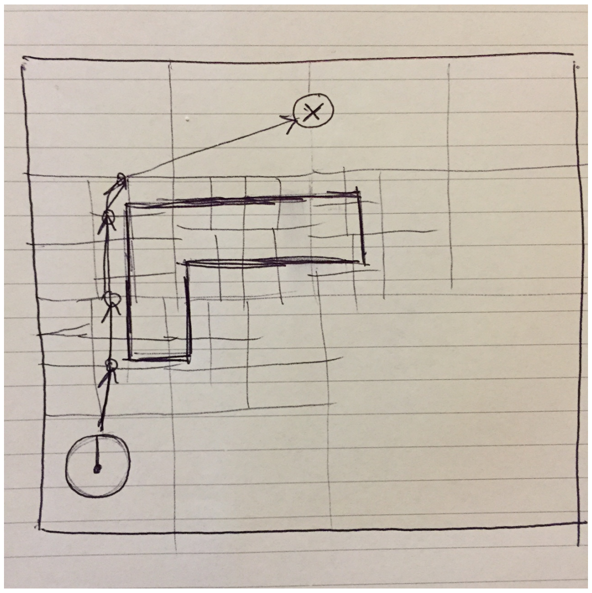
The path planning will use a occupancy grid map that will be populated using the costmap_2d package for local goal handling and an approximate cell-decomposition method for global goal handling (as shown in the illustration). Because of the 2D non-holonomic nature of the robot in simulation, the configuration space will be able to be satisfied simply be extending the obstacle collision ranges by the radius of the robot. The planning generation will take the closest path possible to the obstacle to maximize efficiency.

Mapping Node

- Use <u>approximate cell-decomposition method</u> to create an occupancy grid map of the environment.
    - Publish that map
- Use <u>costmap_2d to</u> create an additional map using noisey sensor data
    - Will assist in navigating to subgoals
    - Publish this

Main Controller Node

- Query user for a goal Pose2D
    - Publish it
    - When /robot0/odom says its at the goal within some tolerance, repeat.


Example Illustration

Global Planner Node

- Listen for a goal Pose2D message that is different from the last one sent (if that's a problem)
- Use map from the mapping node to traverse the data structure and generate a global path
    - Djikstra's shortest path (conditional, no obstacles)

Local Planner and Controller

- Create queue of subgoal Pose2D's that will be along the path and have a heading along the path.
- Use costmap_2d data to keep heading toward subgoal and drive robot forward.
    - costmap_2d creates a costmap which essentially is an occupancy grid map with values from 0 – 255. Using these traits and the knowledge of the cells around the current one, a potential function can be emulated.
- Upon /robot0/odom arriving at subgoal within some tolerance, pop queue to the next position and repeat until robot has arrived at global goal.
    - At this point, the main controller node will be ready for a new goal.

The motivation for this project is to provide base functionality for the planning nodes of the CWRU SnowMower which will be competing at the ION Autonomous Snowplow Competition in January. Of course, the path planning will need to be altered to solve the coverage problem that is presented when having to move snow off of a set area; but the general algorithm should still be retained.