RBE 3002 PDR

Team 13

Functionality

Phase 1:

Gmapping, Centroid finder, pathplanner, pure pursuit

Phase 2:

Pure pursuit

Phase 3:

Amcl, pathplanner, pure pursuit

Centroids

Identify walkable areas that are next to unknown areas to find the frontier cells

Group the frontier cells based on the euclidean distance between them

Calculate the centroid of each grouped frontier based on the mean of the cell locations

Weight each centroid based on size of and distance from the robot

Send the best one one as the goal to pathplanner

We are going to add code to ensure that a centroid always appears in a walkable cell so a path to the centroid can be generated

Used: Phase 1

Pathplanner

We use the path planning node that we developed in lab3.

The pathplanner node uses A* to find the path

- Special Heuristics
  - If a cell in the path is close to the cspace, extra cost is added to ensure the path tends toward the middle of the aisle

Used: Phase 1, Phase 2, Phase 3

Pure Pursuit

Set a variable to the optimal distance for the robot to “follow the carrot” from–this is done through tuning via a trial and error approach.

Iterate through the cells of the path, calculating the Euclidean distance from the robot to the path, and subtract that value from the distance variable. Once the difference is within an epsilon (a small value of about 0.001, because of the way numbers are stored in binary), remove set that cell as the new target. Remove all cells prior to this in the array and repeat.

Add error handling for the case when the robot loses the path: increase the distance variable until the value is within epsilon, set that goal, once it’s reached, set the distance variable back and repeat the process.

Add the final conditional for when the goal is reached. Through trial and error, it was discovered that the optimal way to determine if the goal was reached was by using the distance variable multiplied by a coefficient, and calculating the difference between that value and the final cell in the path list.

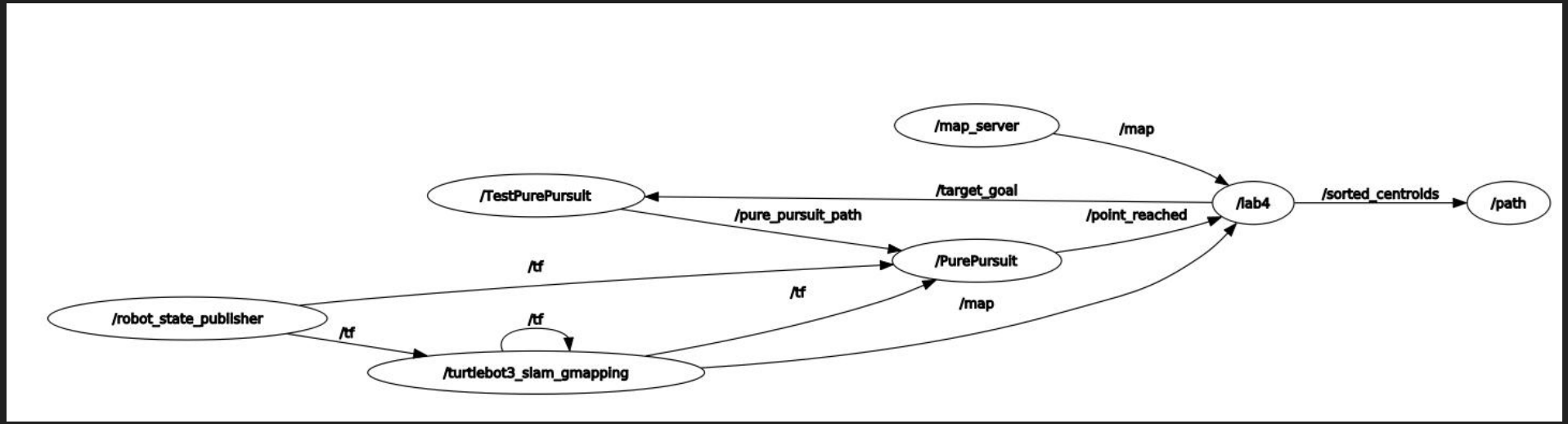
Once the goal-reached message is received, the next centroid in the list of frontier centroids will be used to generate a new path and repeat pure pursuit.

If there are no more centroids, there is a new message published which uses the (0,0) odom coordinate to send the robot to its starting position.

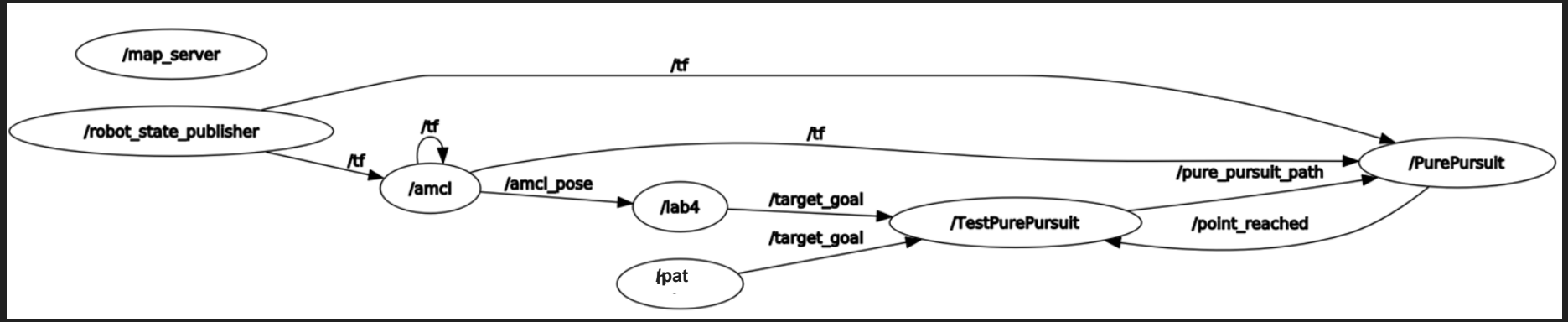
We plan to add code to sense when the robot is too close to the wall, move away slightly and call pathplanner to create a new path.

Used: Phase 1, Phase 2, and Phase 3

Phase 1/2 - Node Graph



Phase 3 - Node Graph



Challenges and Risks

- Challenges
  - Tuning pure pursuit and amcl
  - Transformations

- Edge Cases
  - Weird robot errors that may require a restart
  - Battery dying