Advanced Software Development for Autonomous Mobile Robots

Team : PHOENIX

Team Members :
Elton Wilroy Braggs 32954
Rishabh Rana 32977

Introduction

- Reaching given goal points in arena and collecting reward points, avoiding obstacles in the path.

- Given four robots competing same time, our robot need to avoid also the collision with other robots and reaching early to goal points.

move_base package

- In this project, we used an action client that communicates with action server /move_base that uses a message MoveBaseAction.

  {client = actionlib.SimpleActionClient("move_base", MoveBaseAction)}

- It provides an implementation of an action that, given a goal in the world, will attempt to reach it with a mobile base.
- The move_base action node links together a global and local planner to accomplish its global navigation task.

How robot will choose next goal

- Goal points list is sorted on the basis of minimum distance between goal and robot.
- Robot will move towards nearest Goal and collect reward.
- The subscribed goal topic gives the goal list. The distance between the current position of each goal is calculated.
- The sorting is done in the minimum distance in the ascending order.
- The move_base action client is started which is given the sorted goal.

Topics to which we subscribed:

- '/goals' – to get the goals
- 'amcl_pose' – to localize tutrtlebot in given map.
  [geometry_msgs/PoseWithCovarianceStamped]

- The feedback server is required to provide the following information:

If the goal status is active, it creates a local path from the current position to the goal position. Then it subscribes to the cmd_vel topic for the turtlebot to move.

If the goal status from the feedback server is 4 , 5 or 6 , it skips the goal and moves to the next goal.

Parameter values changed from the default move base parameters:

Path_distance_bias = 8.0

Inflation_radius:

global_costmap = 0.2

local_costmap = 0.1

Velocity and the width values were increased.