**Instructions**

This code to find the final path from start point to end point uses a combination of two algorithms, **A\*** and **RRT**. A\* is in use when there are few obstacles between the start and end points. The algorithm switches to RRT when too many obstacles are between the start and end points.

**Dependencies**

The code is programmed in Python, specifically python3.

The dependencies are as follows:

Matplotlib – pip install matplotlib

Tkinter- apt-get **install** python-tk

**Name of the Source Code Files and Directory Structure:**

**Directory Structure:**

* Project 3
  + - VREP
      * vrep\_code.py
      * vrep\_code\_2.py
      * FinalPath.txt
      * FinalPathv.txt
      * Other files related to VREP Remote API dependencies
      * map
        + mine.ttt
    - astar\_rrt.py
    - diffConstraints.py
    - getActions.py
    - V-REP Video.mp4
    - V-REP Video 2.mp4

**Details:**

1. To run code, execute “astar\_rrt.py”. This has call to functions in files “diffConstraints.py” (to compute differential constraints) and “getActions.py” (to get the 8-action connected space).
2. Simulation Videos:

Both the videos are based on the Start Point (-4.55, -4.05) and End Point (-1.55, 0.55)

* “V-REP Video.mp4” is associated with “vrep\_code.py” in VREP Folder
* “V-REP Video 2.mp4” is associated with “vrep\_code\_2.py” in VREP Folder

1. VREP Folder

* Inside this folder are all dependencies needed to connect to VREP Remote API.
* VREP Simulation codes:
* “vrep\_code.py”, uses the “FinalPathv.txt” generated on executing “astar\_rrt.py”. This code simply gives left and right wheel velocities associated with the nodes of the final path to the “wheel\_left\_joint” and “wheel\_right\_joint” of Turtlebot2. The video associated with this code is “V-REP Video.mp4” which is in the root folder.
* “vrep\_code\_2.py”, uses the “FinalPath.txt” generated on executing “astar\_rrt.py”. This code gives left and right wheel velocities associated with the nodes of the final path to the “wheel\_left\_joint” and “wheel\_right\_joint” of Turtlebot2, as well as corrects any orientation related errors during simulation. It adjusts orientation by rotating either left or right to match that of theta associated with each node in the final path. This, however, takes a bit longer than the previous approach but is more accurate. The video associated with this code is “V-REP Video 2.mp4” which is in the root folder.
* The path generated on executing “astar\_rrt.py” generates 2 text files,

- “FinalPath.txt”, which stores x,y,theta,ul,ur of each of the nodes associated with the final path. This text file is used by “vrep\_code\_2.py” for simulation of the path on Vrep

- “FinalPathv.txt”, which stores ul,ur (left and right wheel velocities) of each of the nodes associated with the final path. This text file is used by “vrep\_code.py” for simulation of the path on Vrep.

* Map that is used for “vrep\_code\_2.py” is in the folder “map” and the name of the map is “mine.ttt”. This map has a “Dummy” at the left bottom corner, which is used as a reference frame to find the orientation of the robot. This map can be used for “vrep\_code.py” as well.

**If running python files on Ubuntu Terminal use:**  
python3 <<nameOfFile>

**On Running the code, GUI will be generated showing:**

* Nodes explored
* Optimal Path

To skip animation, close window.