CS365: Homework Assignment 3

Aayush Mudgal [12008]

Probabilistic Roadmap Algorithm

Code: will be uploaded on 22 March, 2015

Brief Outline of the implemented Algorithm

- A large number of training configuration are chosen uniformly and randomly (20,000)
- Similarity/ Difference metrics for the different configurations that were used are :
    - L1/ L2 norm over absolute angle configurations
    - L1/ L2 norm over relative angle configurations
    - L1/L2 norm over end vector points of different links

The first and third approach gave slightly better results as compared to the second one.

- Nearest Neighbor Algorithm from Scikit Learn was used to find the nearest neighbors of a configuration.
- Once the nearest neighbours are calculated, the weighted graph of connected configuration is obtained using shorted distance algorithm (implemented in NetworkX).
- Local Planning algorithm is implemented as follows:
    - Intermediate absolute angular configuration is obtained by taking the average of any two configurations in consideration.
    - If the intermediate configuration collides with a obstacle, then the two original configuration are not connected by a viable motion
    - This is recursively computed, till the two configuration are at a small distance apart (defined by the distance metric observed in earlier step)

A viable configurational changes from the start state to the goal state is obtained by traversing the obtained graph. If no such path exists and both the start and the goal states are valid, then either the sampling done is small to capture this motion. More sampling with an optimized algorithm may give the required path.

Results of the algorithm with Two Dof robot against the sample test case provided.

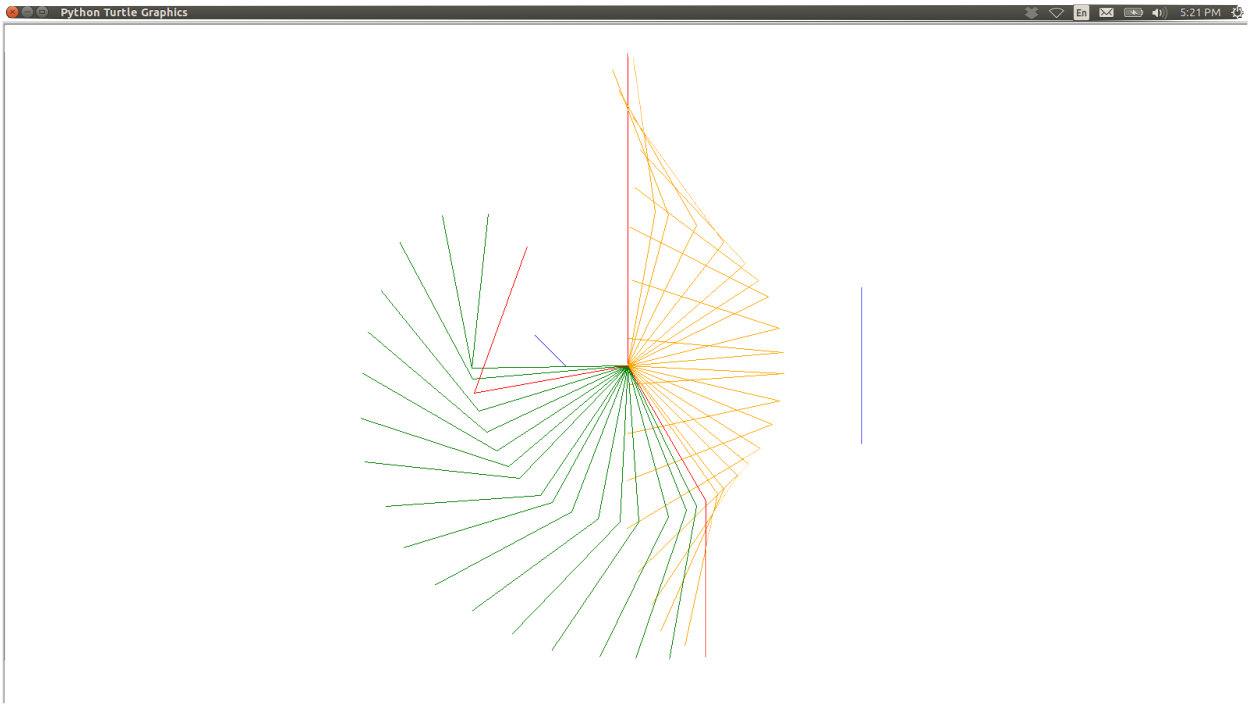
Red Color: The start and/or the goal configuration of the robot

Orange Color lines: Intermediate configurations of the robot from state 1 to state 2 as found by the implemented PRM algorithm.

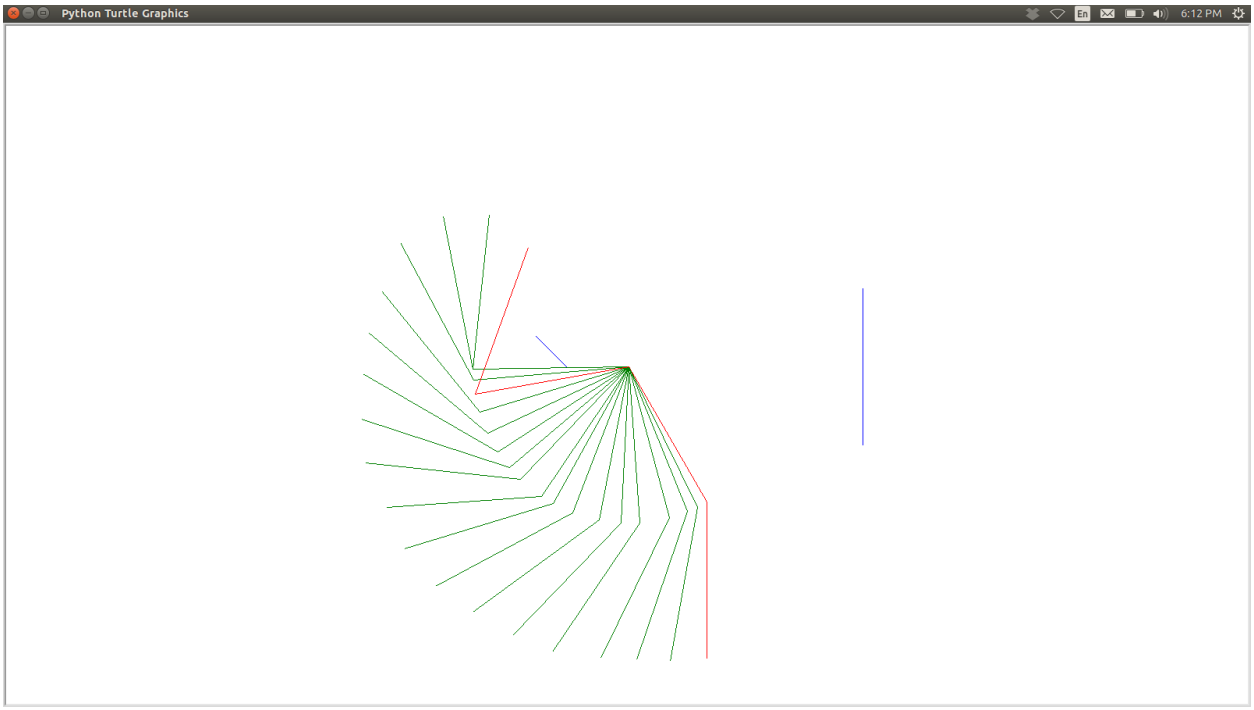
Green Color lines: Intermediate configurations of the robot from the state 2 to state 3 as found by the implemented algorithm

RESULTS FOR DEGREE OF FREEDOM=2

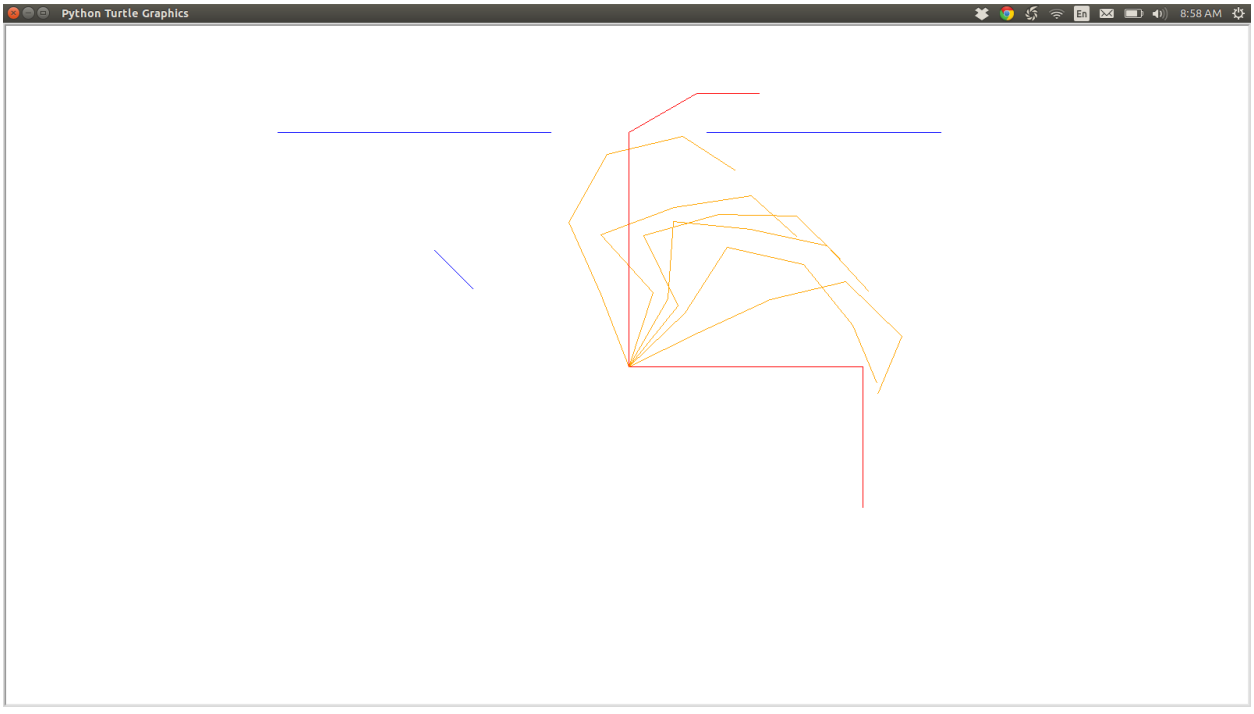
Robots.dat and goals.dat file are those provided with the question

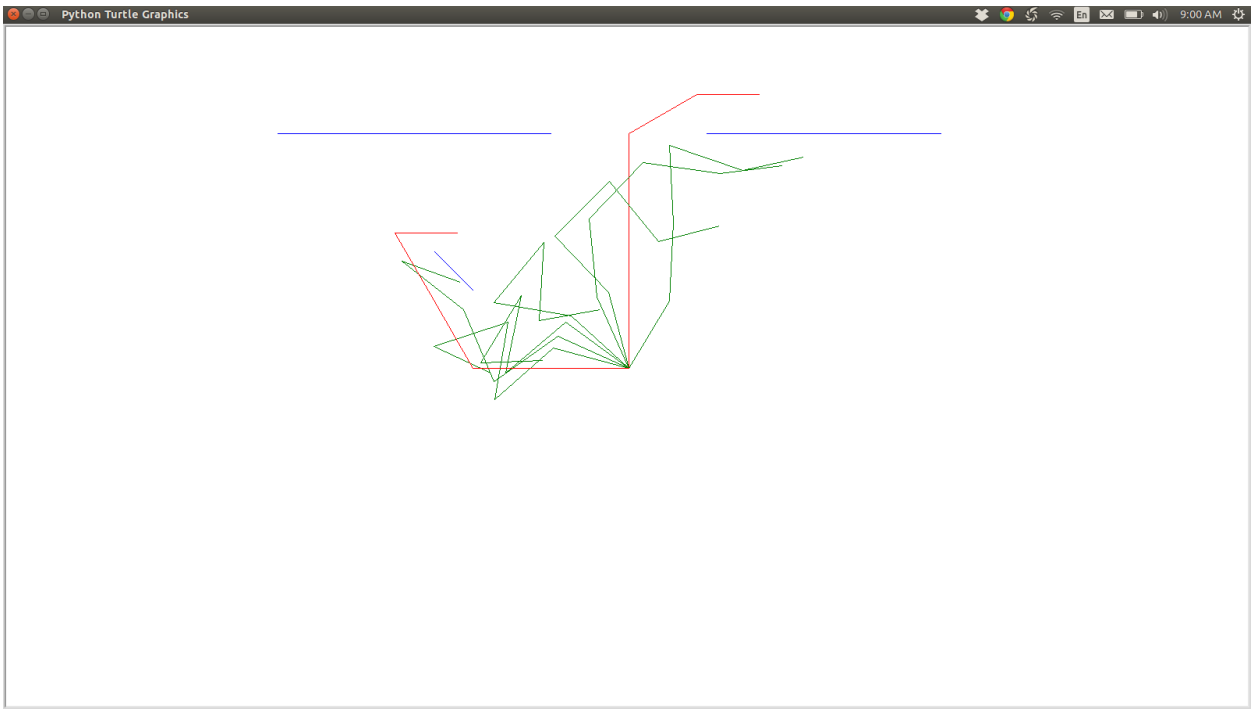


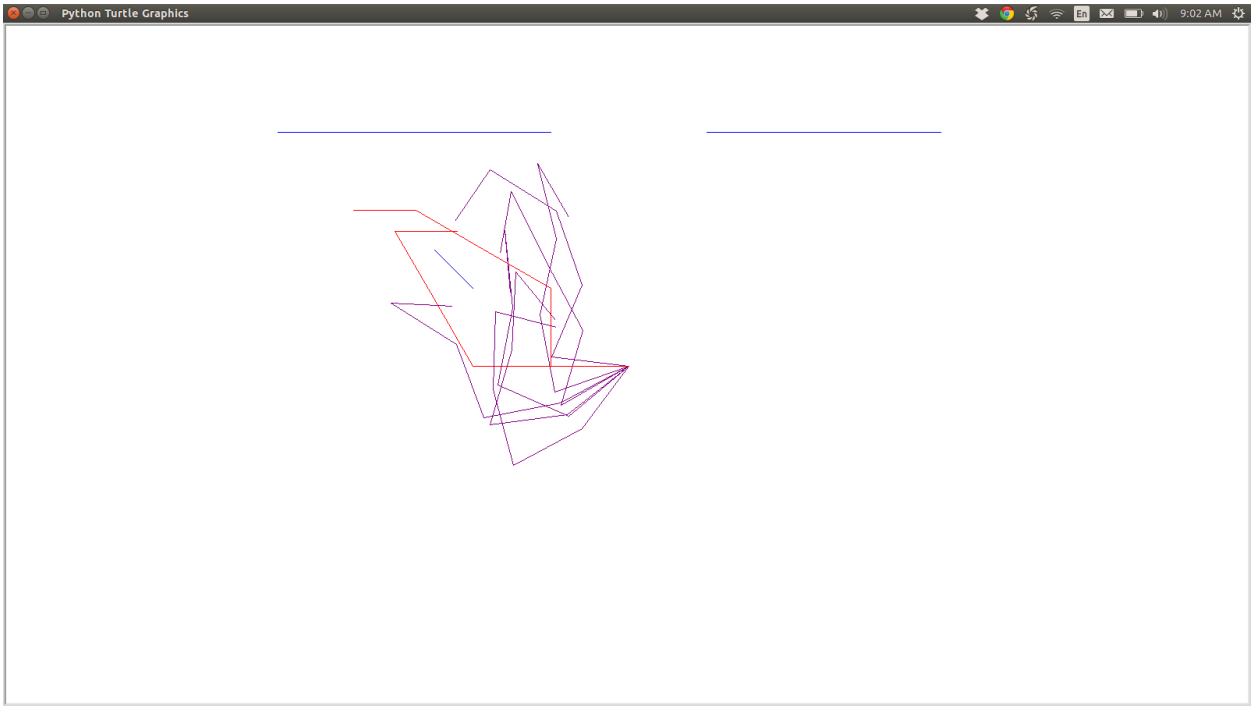
.

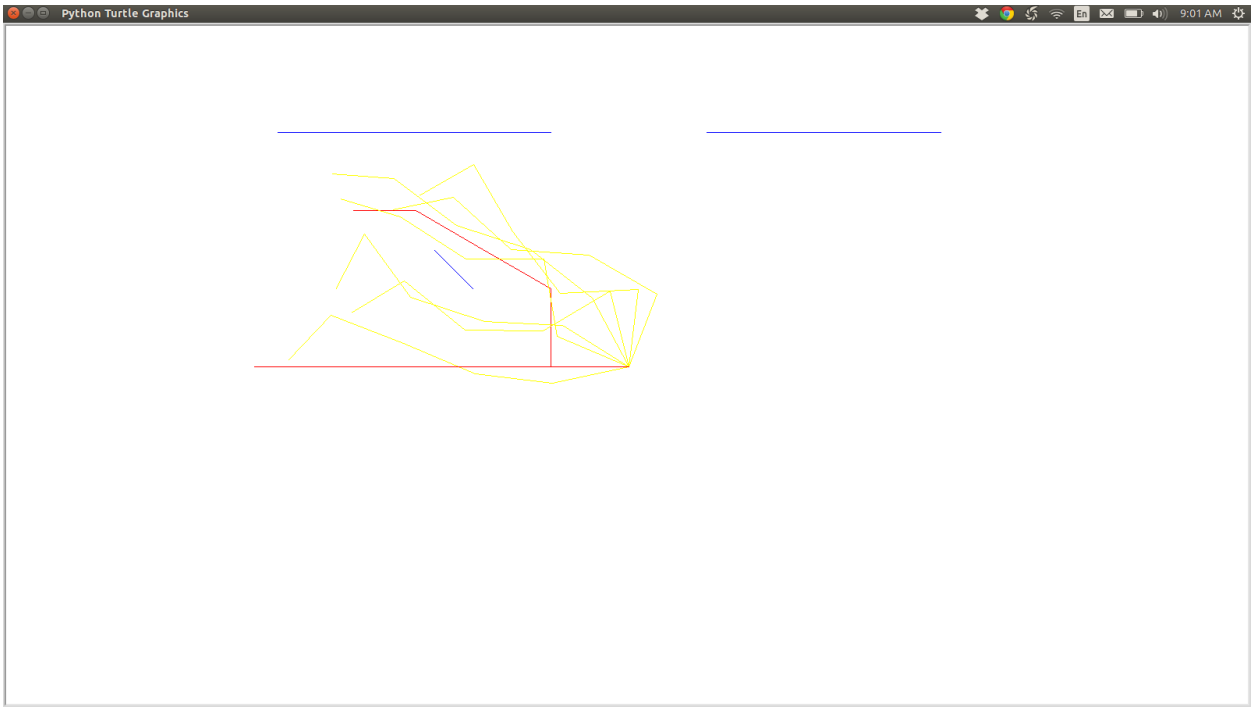


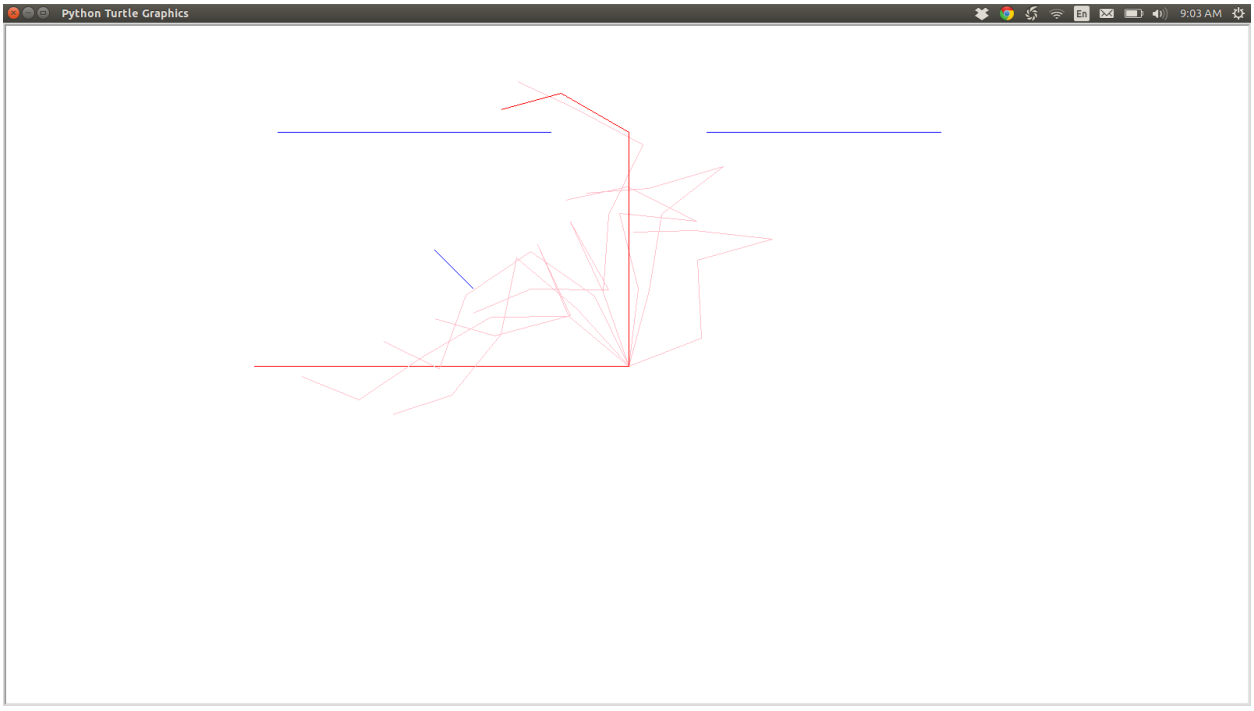
RESULTS FOR DEGREE OF FREEDOM=5



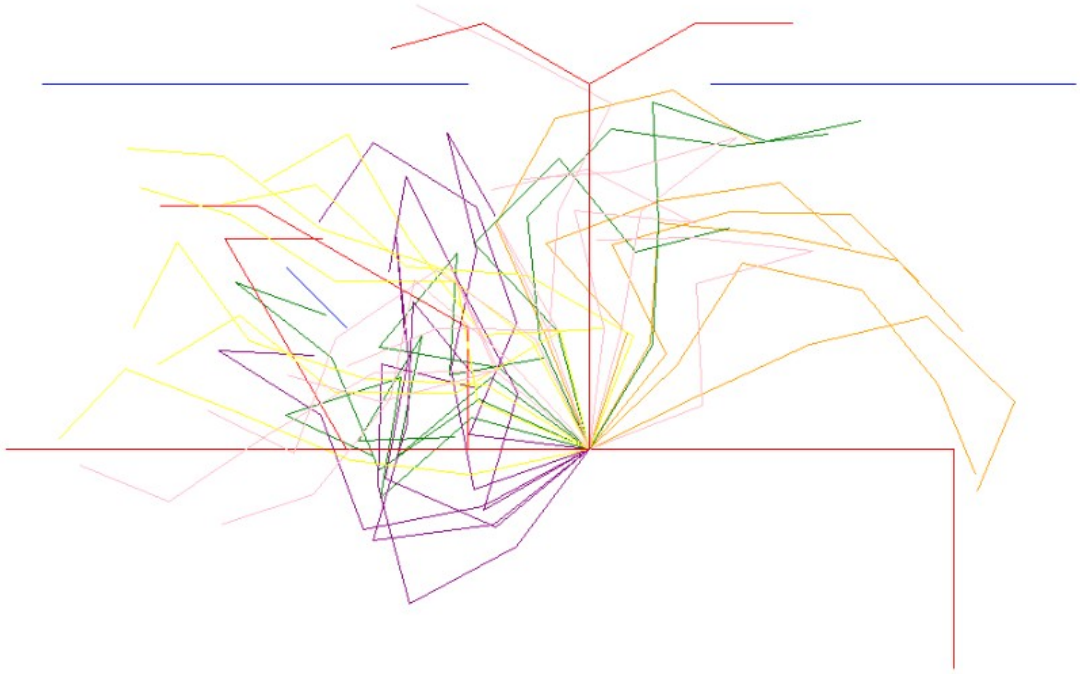

Python Turtle Graphics
9:00 AM


Python Turtle Graphics
9:02 AM

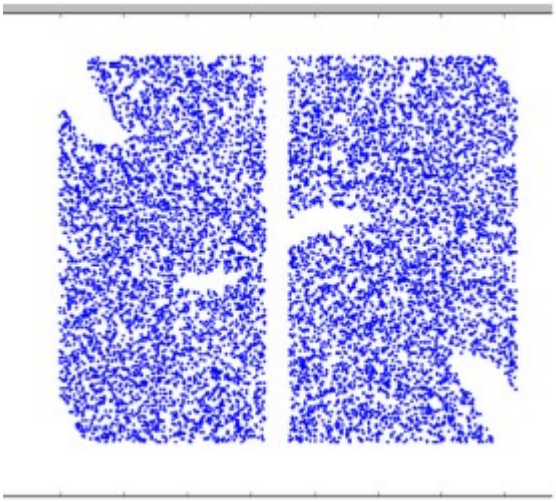

Python Turtle Graphics
9:01 AM


Python Turtle Graphics
9:03 AM

Question 1



Are the line obstacles lines or regions in the graph? Why?
Answer 1:
The line obstacles are regions in the graph. This becomes clear if we draw a map of the good and bad positions of the robot arm as theta1 vs theta2 for the space. We clearly get regions. This idea can also be obtained from a voronoi diagram. There exists a mapping between the 2-d space and the corresponding c-space which is non-linear in nature , so it can never be a line. Moreover the regions would be well-defined. In the c-space whenever the robot arm intersects the obstacle or even if the end-effector just touches the obstacle, the pose is not allowed. The C-space for 2 degrees of freedom has the pattern given below proving existence of regions :



Question 2:

What is the minimum that is redundant?

Answer 2:

Four is the minimum Degree of freedom that is redundant, this is because the work-space has three degree of freedom, and thus anything greater than three will be redundant.