

**SC627**  
**Assignment-4**

Roll No. - 213234002

Name – Sourav Chatterjee

The aim is to balance 8 robots equidistant to each other. The first and last robots are fixed at (0,0) and (14,0) respectively. The maximum velocity is capped at 0.15m/s and maximum angle deflection in one command is  $\pi/18$ .

Here the balancing.launch file was already given which will behave as the main program calling the subprogram balancing.py for each of the 6 moving robots.

In balancing.py code, we capture the x, y coordinates & orientation of the robot itself as well as x, y coordinates of the robot to its right and robot to its left. This is simulated in gazebo environment via the odometry function. Angle of the robot is calculated by transforming the orientation via euler to quaternion function.

We know that as per balancing strategy, the velocity to be achieved by the local robot is given by

$$u_x = \kappa (\text{relative distance between robot and right robot}) \\ + (\text{relative distance between robot and right robot})$$

Here we consider  $\kappa=1$ .

With this we get x and y component of the velocity vector to be achieved in this step.

These components are then transformed to linear and angular velocities, capped to their maximum values, and provided to the gazebo environment for simulation

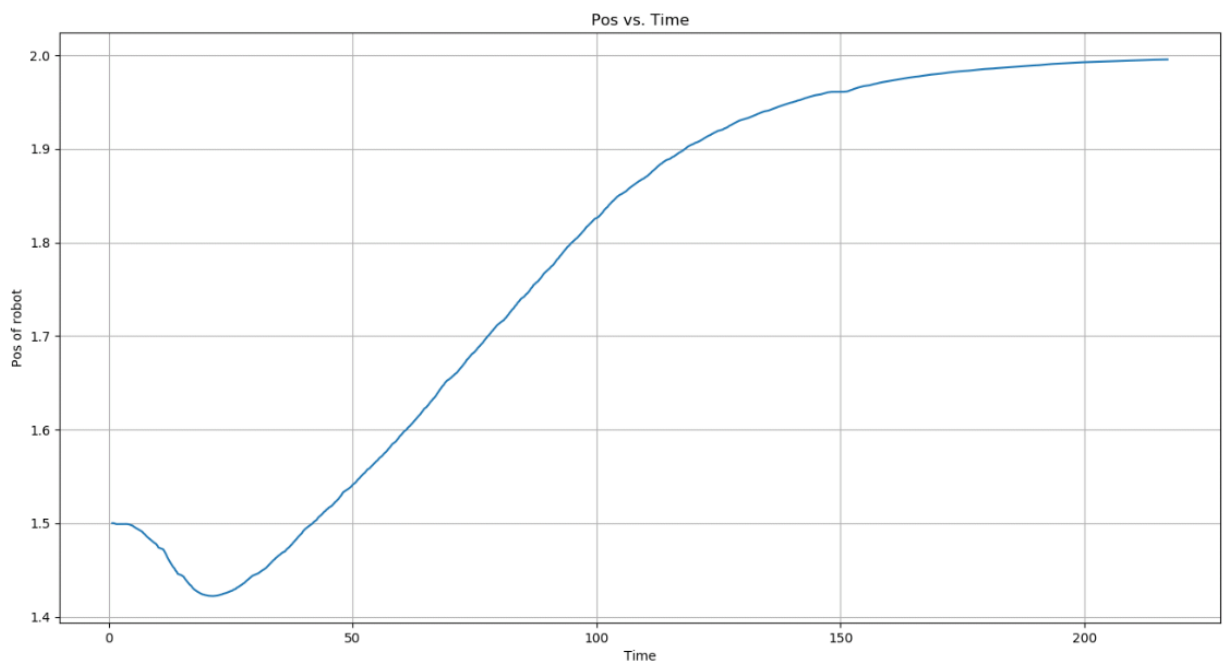
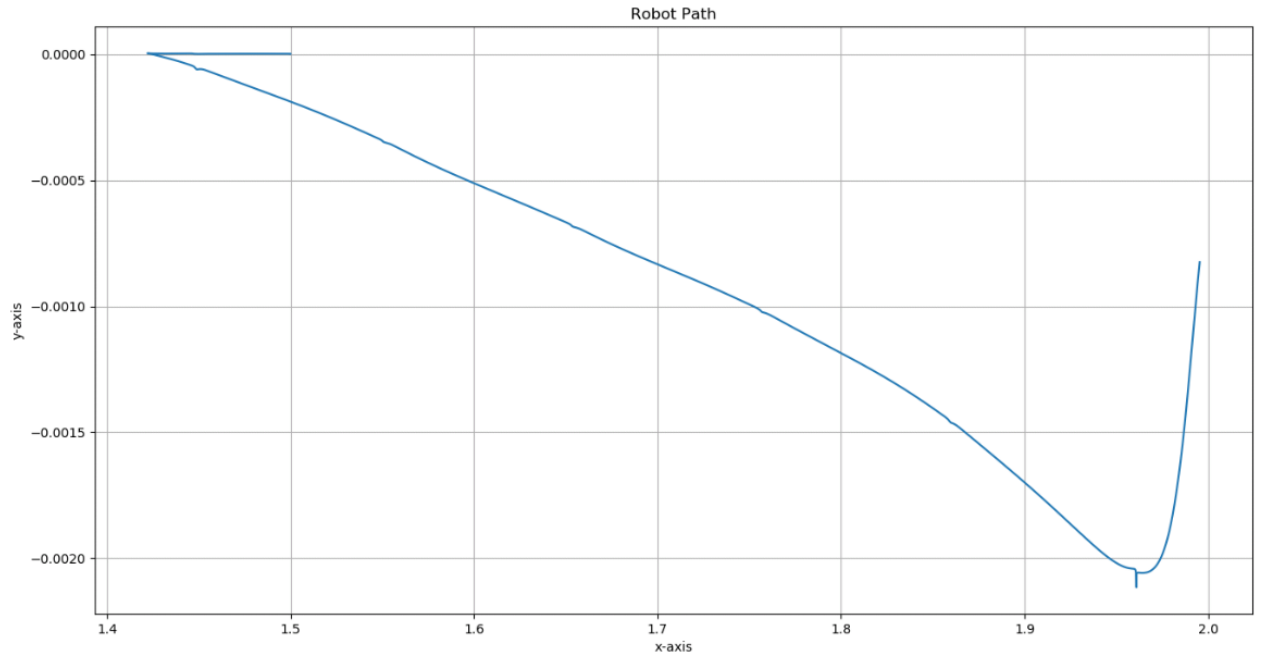
**SC627**  
**Assignment-4**

Roll No. - 213234002

Name – Sourav Chatterjee

**Result:**

**Bot\_2**

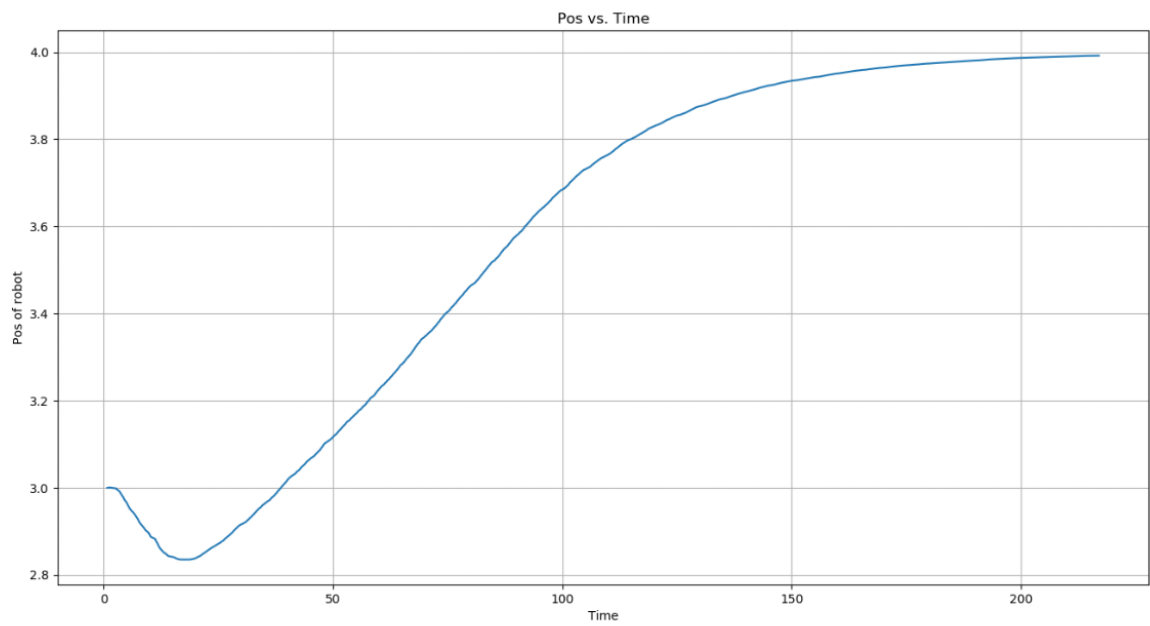
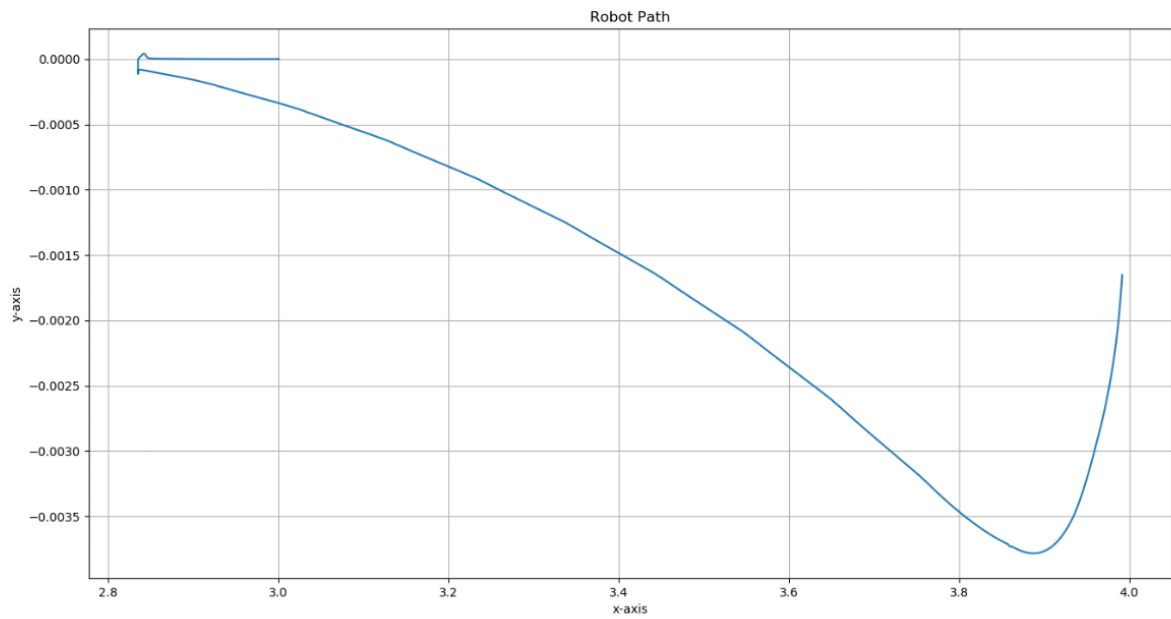


SC627  
Assignment-4

Roll No. - 213234002

Name – Sourav Chatterjee

Bot\_3

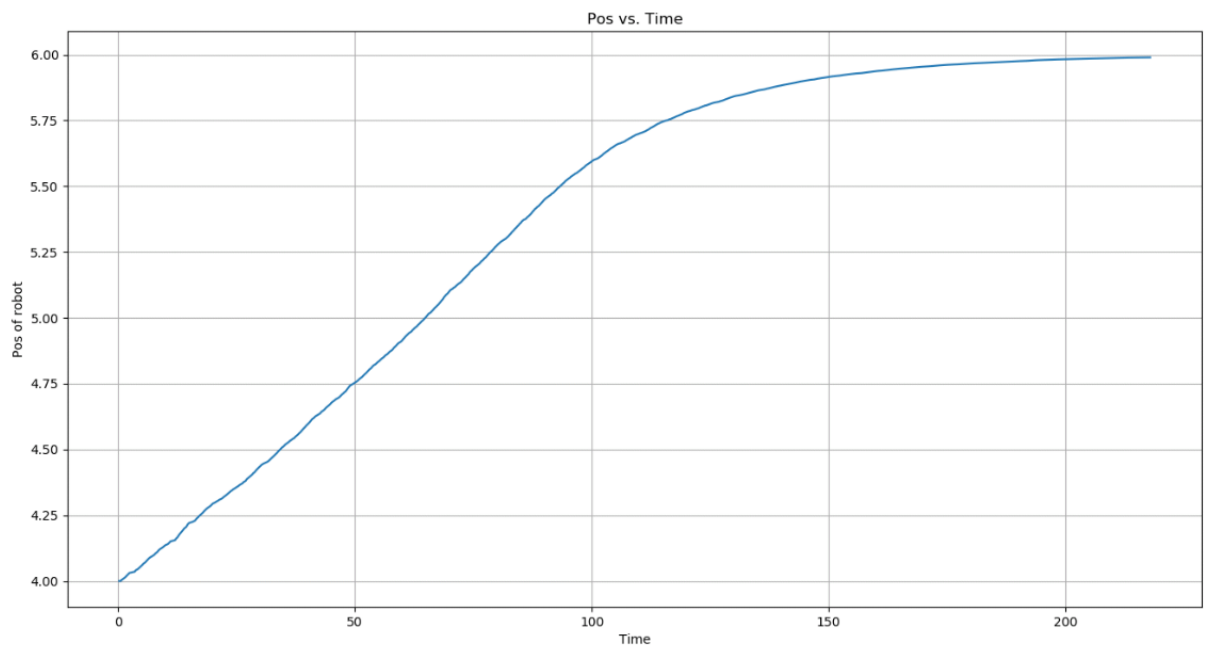
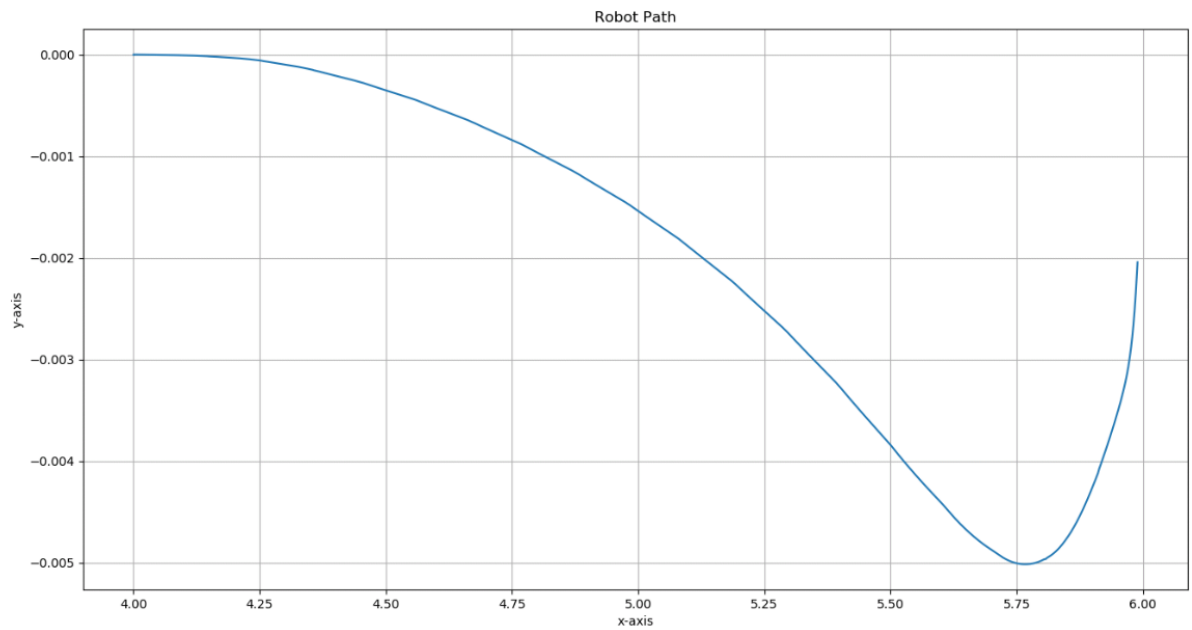


SC627  
Assignment-4

Roll No. - 213234002

Name – Sourav Chatterjee

Bot\_4

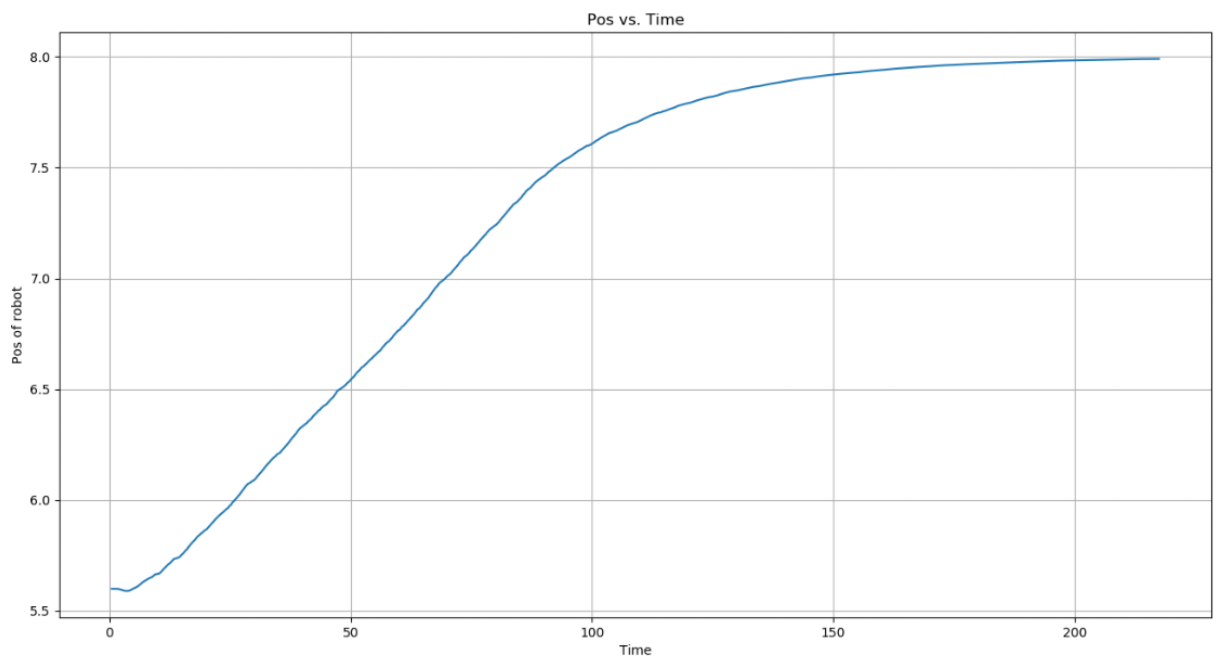
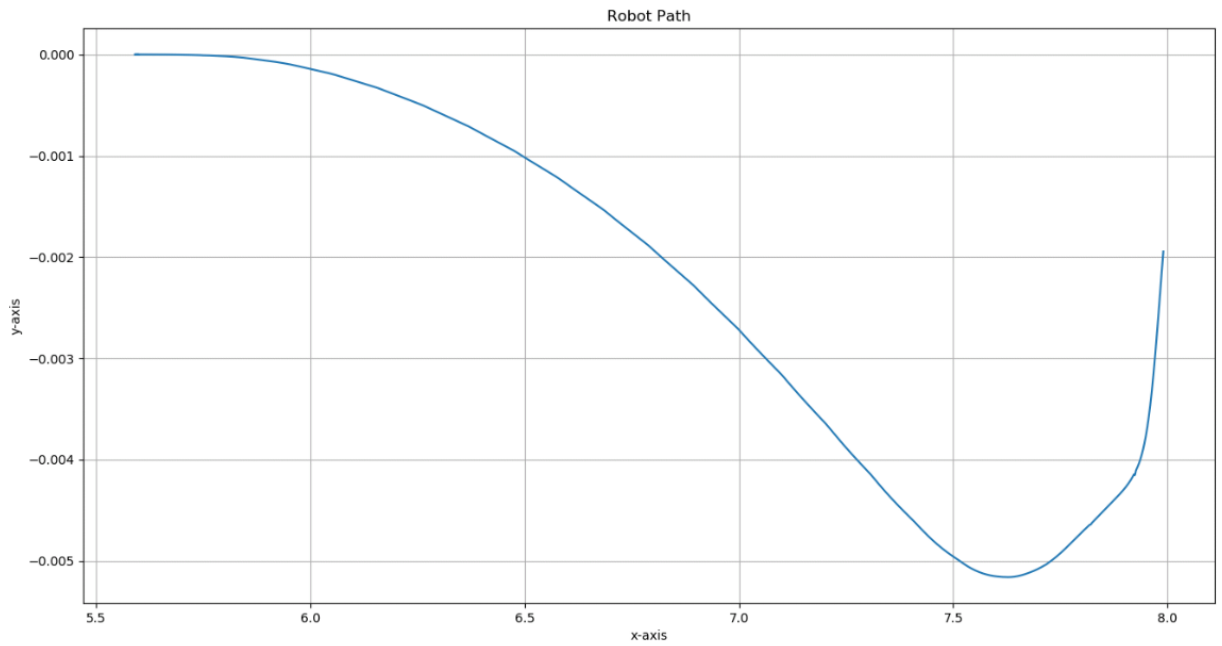


SC627  
Assignment-4

Roll No. - 213234002

Name – Sourav Chatterjee

Bot\_5

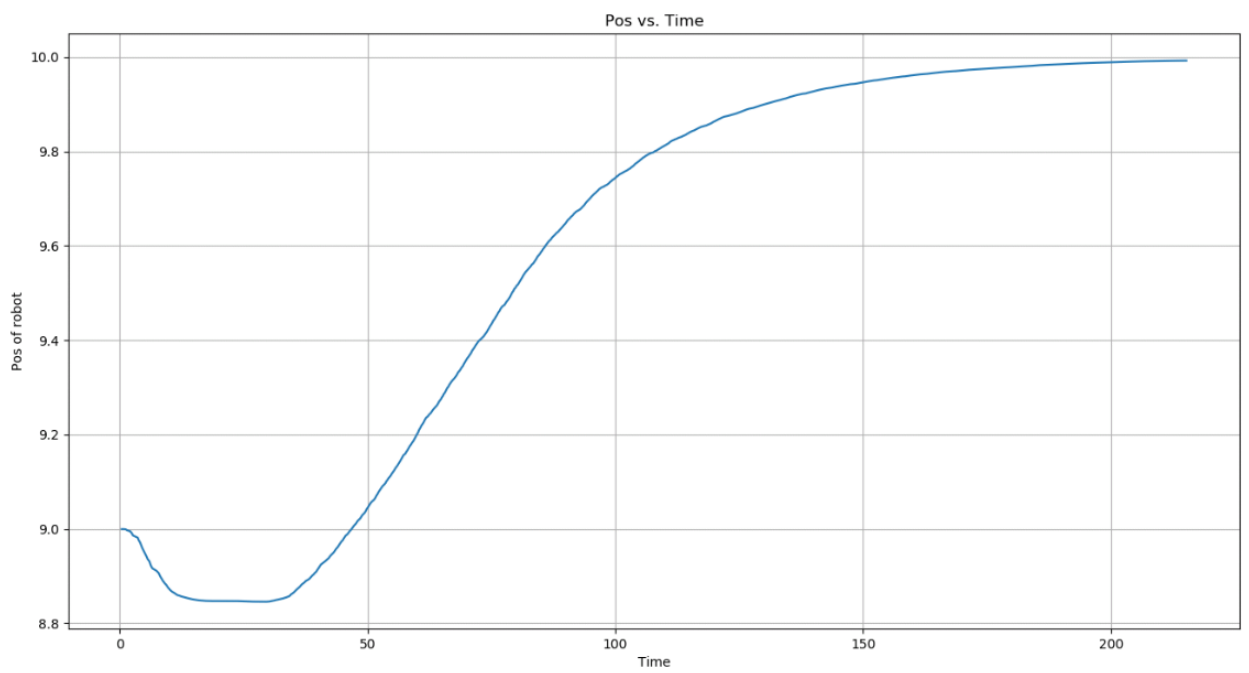
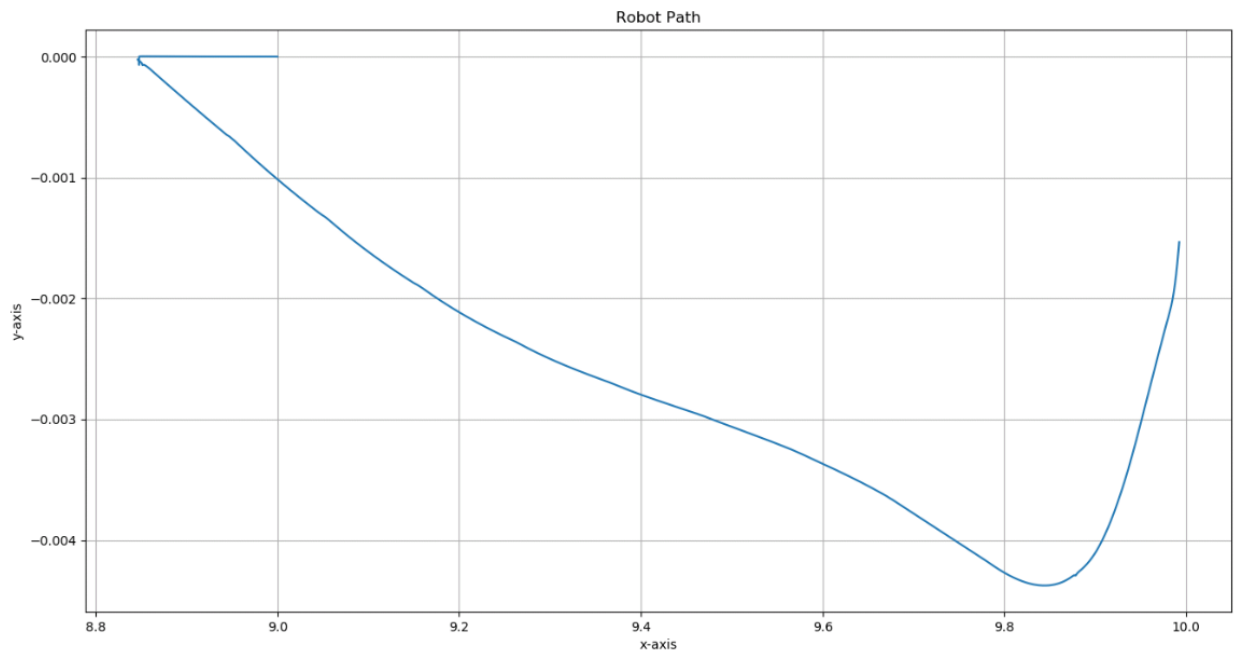


SC627  
Assignment-4

Roll No. - 213234002

Name – Sourav Chatterjee

Bot\_6



SC627  
Assignment-4

Roll No. - 213234002

Name – Sourav Chatterjee

Bot\_7

