**MP3 Report**
**Course Number: ECE484**
**Team Name: Carla McCarface**
**Team Members:**
**Peng (penghan2)**
**Amith (amithr3)**
**Tracy (tracymt2)**

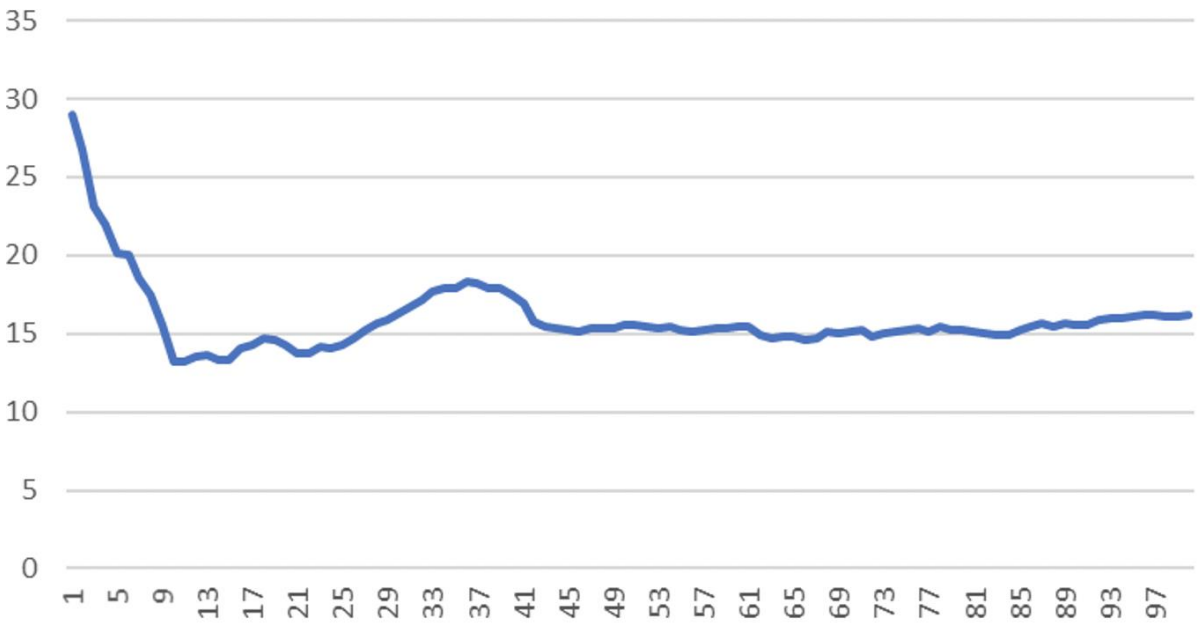
**Problem 4 (20 points).** Using 4 measurement directions, keep the sensor limit constant at 15, run your algorithm with the number of particles 500, 1000, 1500. Plot the error in position estimation (euclidean distance between actual position and the estimated position) and orientation estimation as a function of algorithm iterations. Since particle filtering is a randomized algorithm, run several instances of the same setup and plot the average error for each time. How does changing the number of particles influence the estimation accuracy, converging speed and computational cost of the algorithm? Record a video of one of the three runs. The video should include the turtle map window. Provide a link to the video and include it in the report

As the number of particles increases, the estimation state will be more accurate, the converging speed will be faster, but the computational cost will be more expensive.
The video link and the plots of errors are listed below.

https://youtu.be/AAfQPT1FK_Y

N = 500

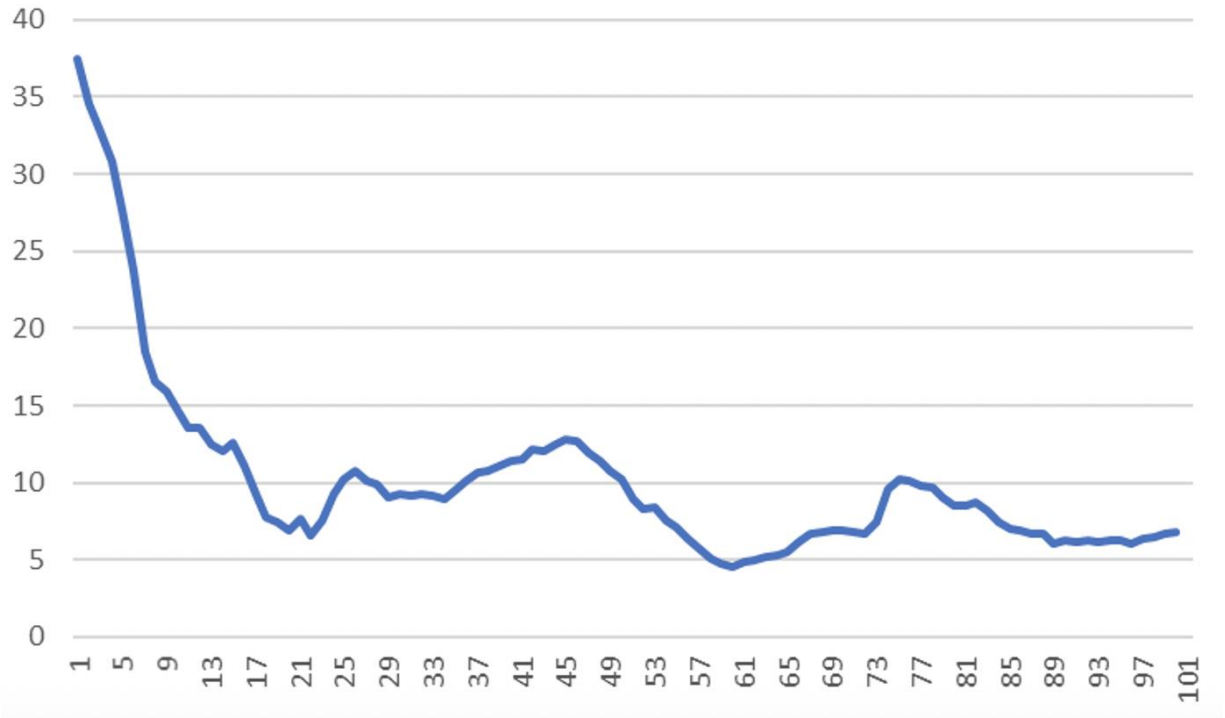
Position error



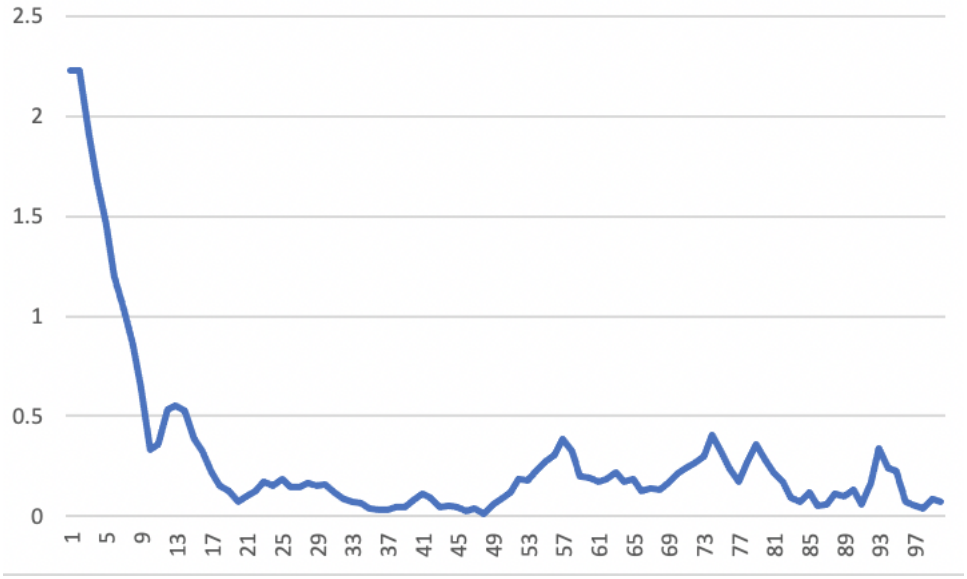

Orientation error


N = 1000

Position error

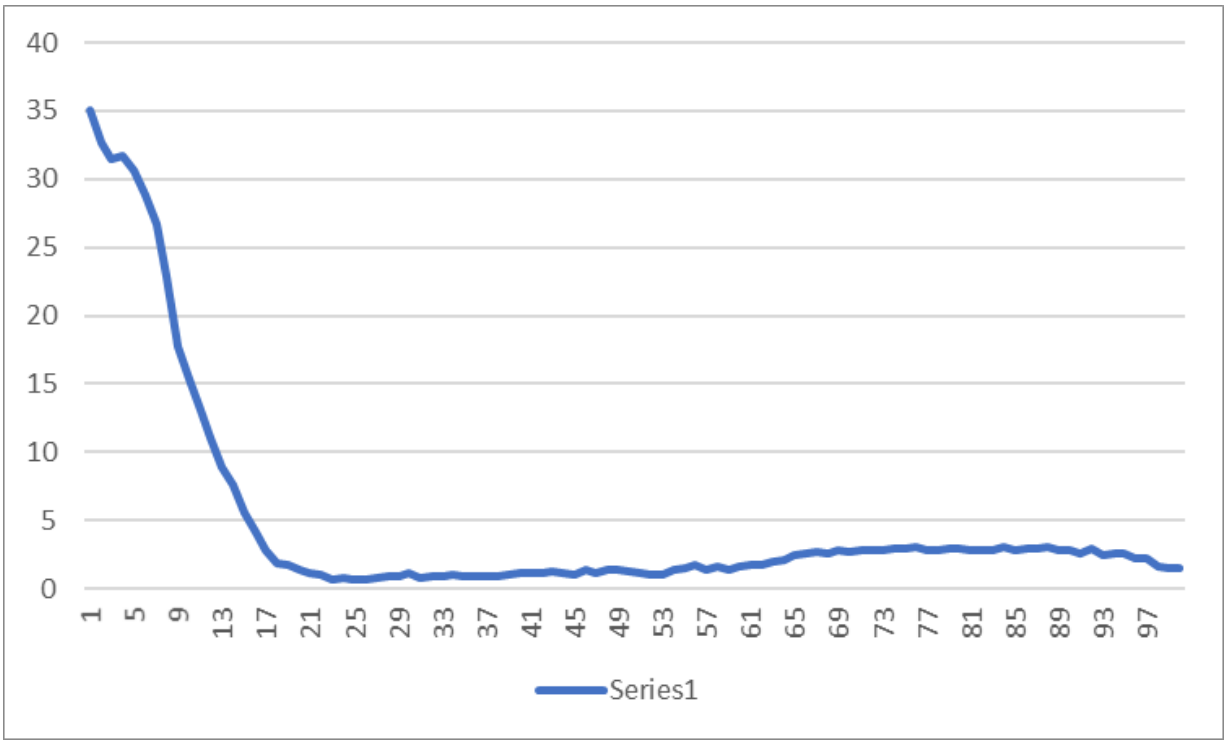



Orientation error

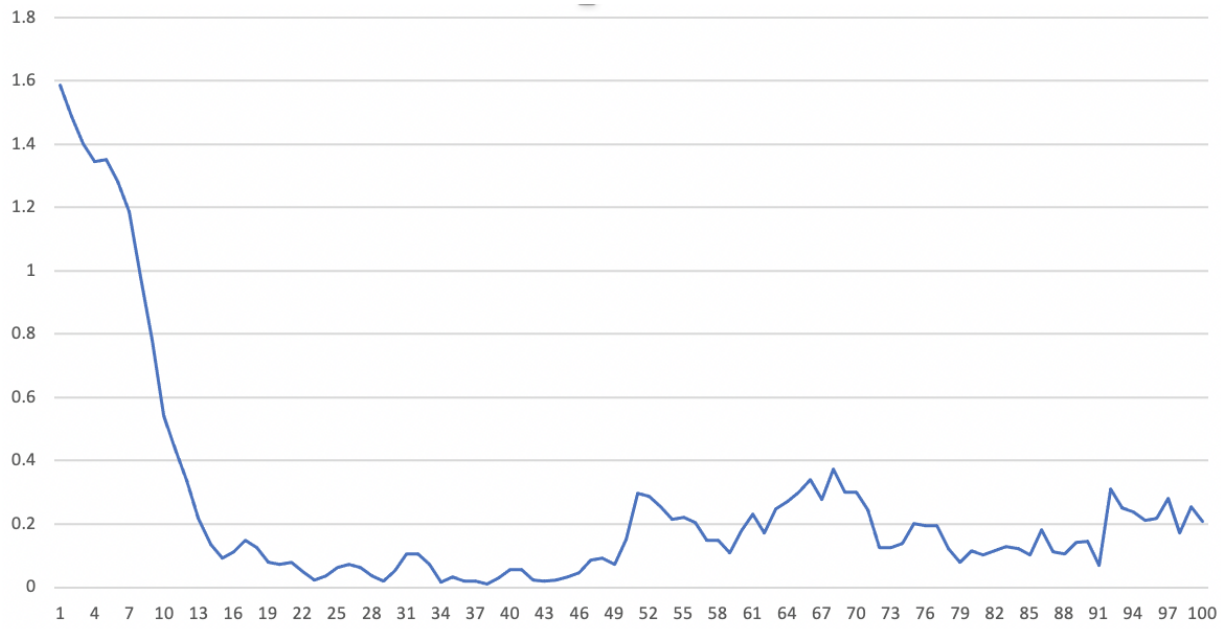



N = 1500

Position error




Orientation error




**Problem 5 (20 points).** Using 4 measurement directions, keep the number of particles constant at 1000, run your algorithm with sensor limit 15, 20, 25. Plot the error in position estimation (euclidean distance between actual position and the estimated position) and orientation estimation as a function of algorithm iterations. Since particle filtering is a randomized algorithm, run several instances of the same setup and plot the average error for each time. How does changing sensor limit influence the estimation accuracy, converging speed and computation cost of the algorithm?

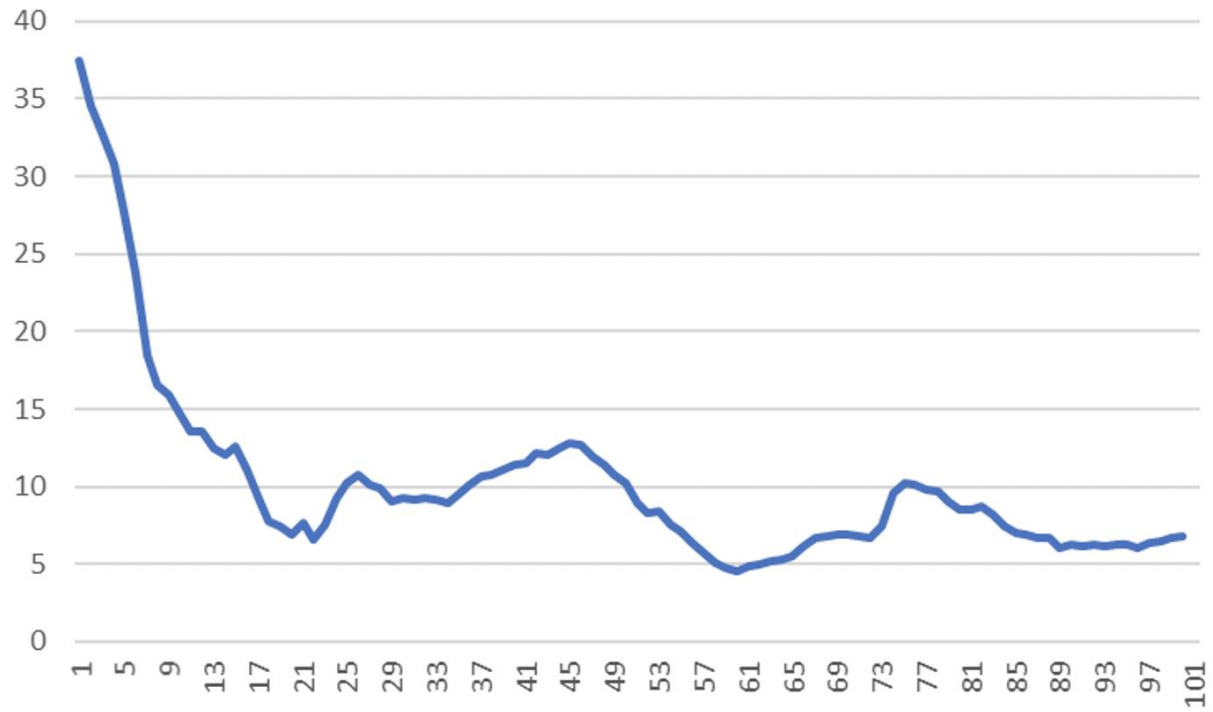
Record a video of one of the three runs. The video should include the turtle map window. Provide a link to the video and include it in the report.

As the value of the sensor limit increases, the estimation state will be more accurate, the converging speed will be faster, but the computational cost will be more expensive. The video link and the plots of errors are listed below.

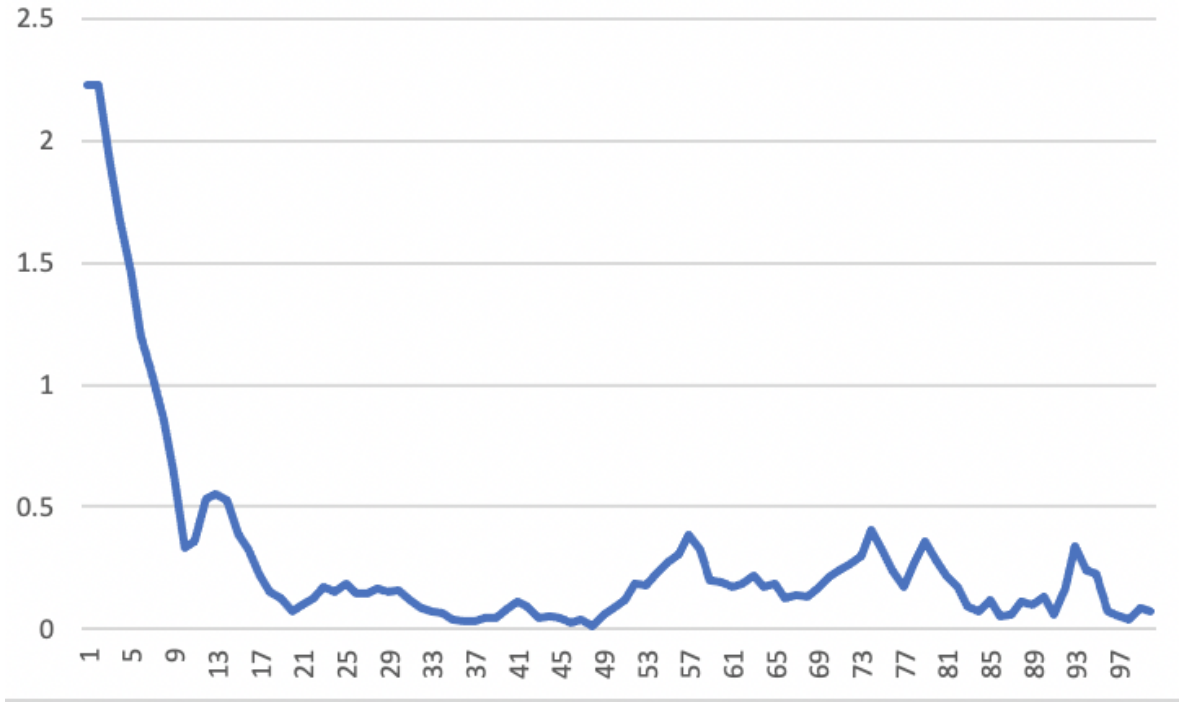
https://youtu.be/rsxJv8A4WA0

S = 15

Position error

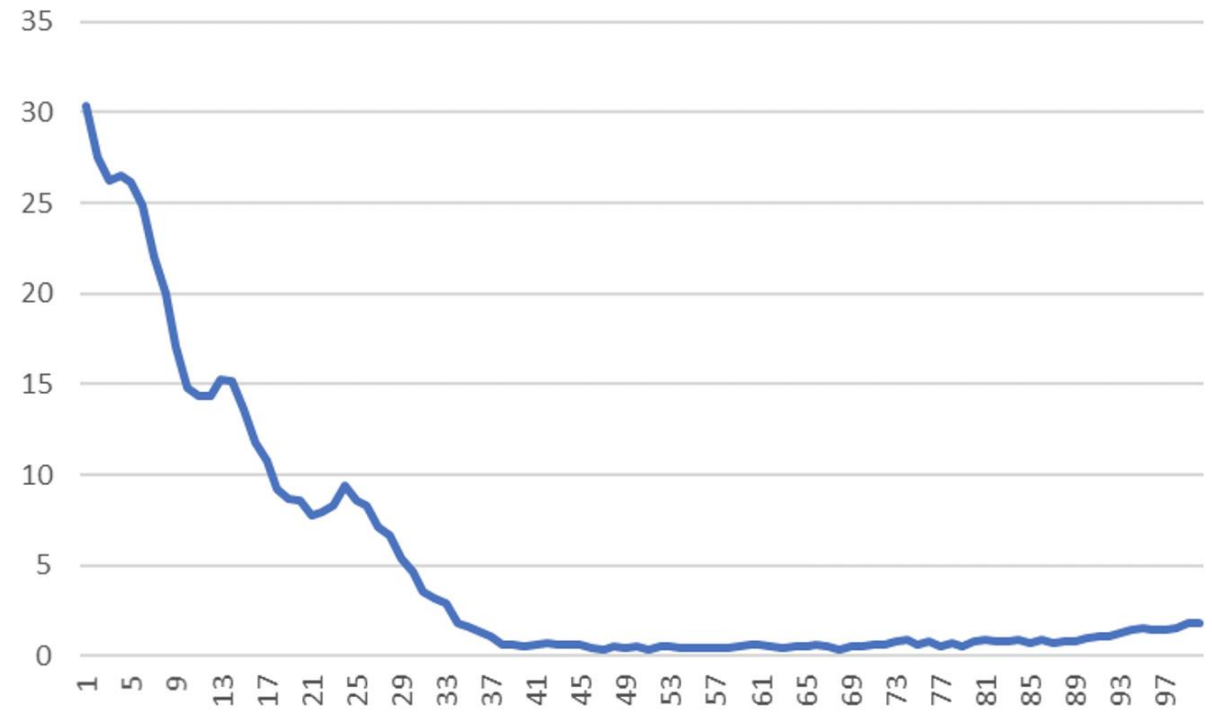



Orientation error

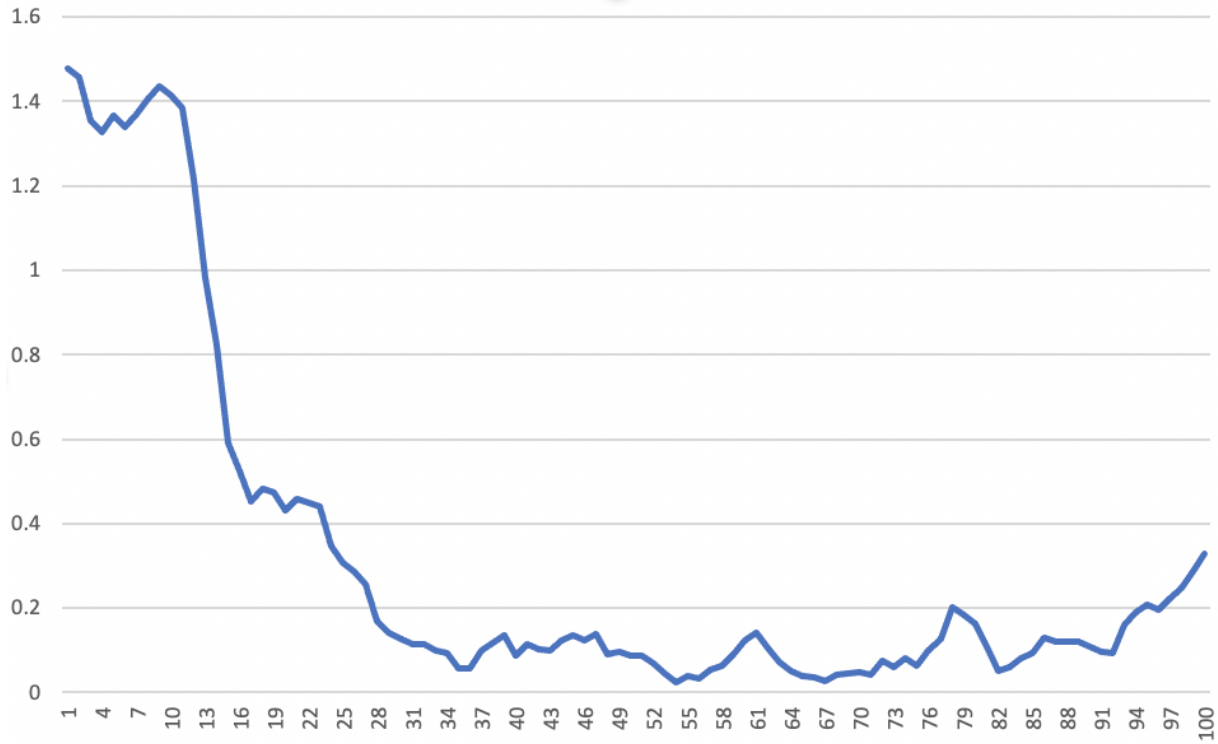



S = 20

Position error

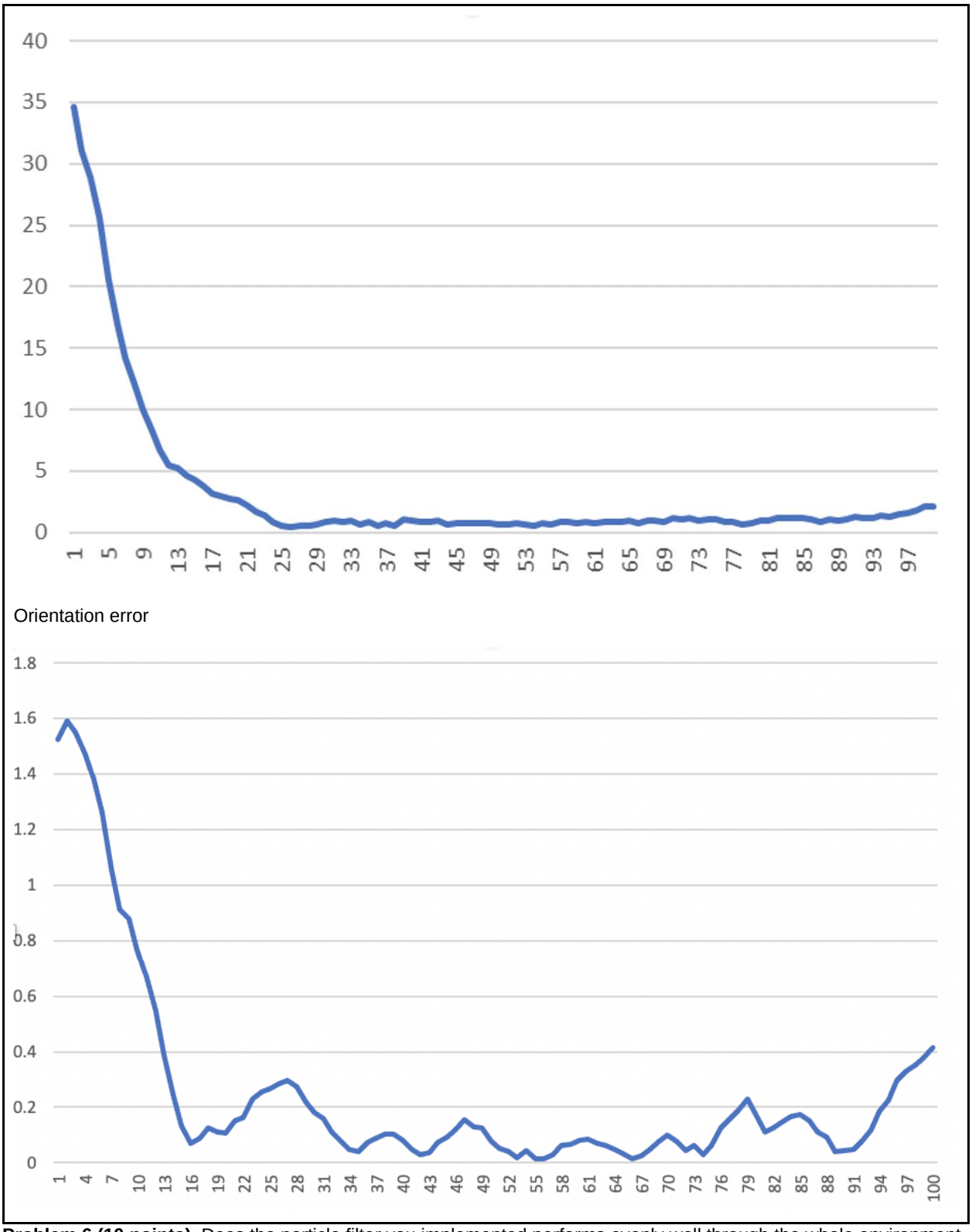



Orientation error




S = 25

Position error



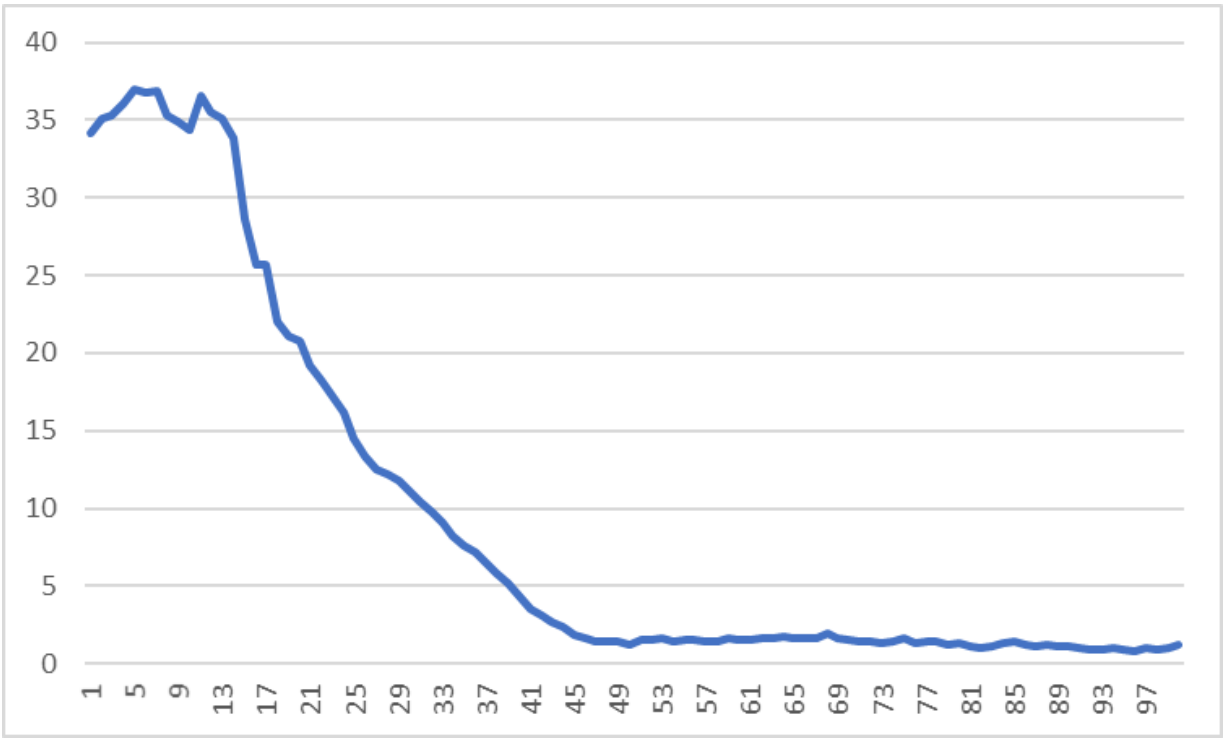

**Problem 6 (10 points).** Does the particle filter you implemented performs evenly well through the whole environment after converging? More specifically, does the your particle filter have larger prediction error in some regions of the environment than other regions? If yes, can you explain why this is happening?

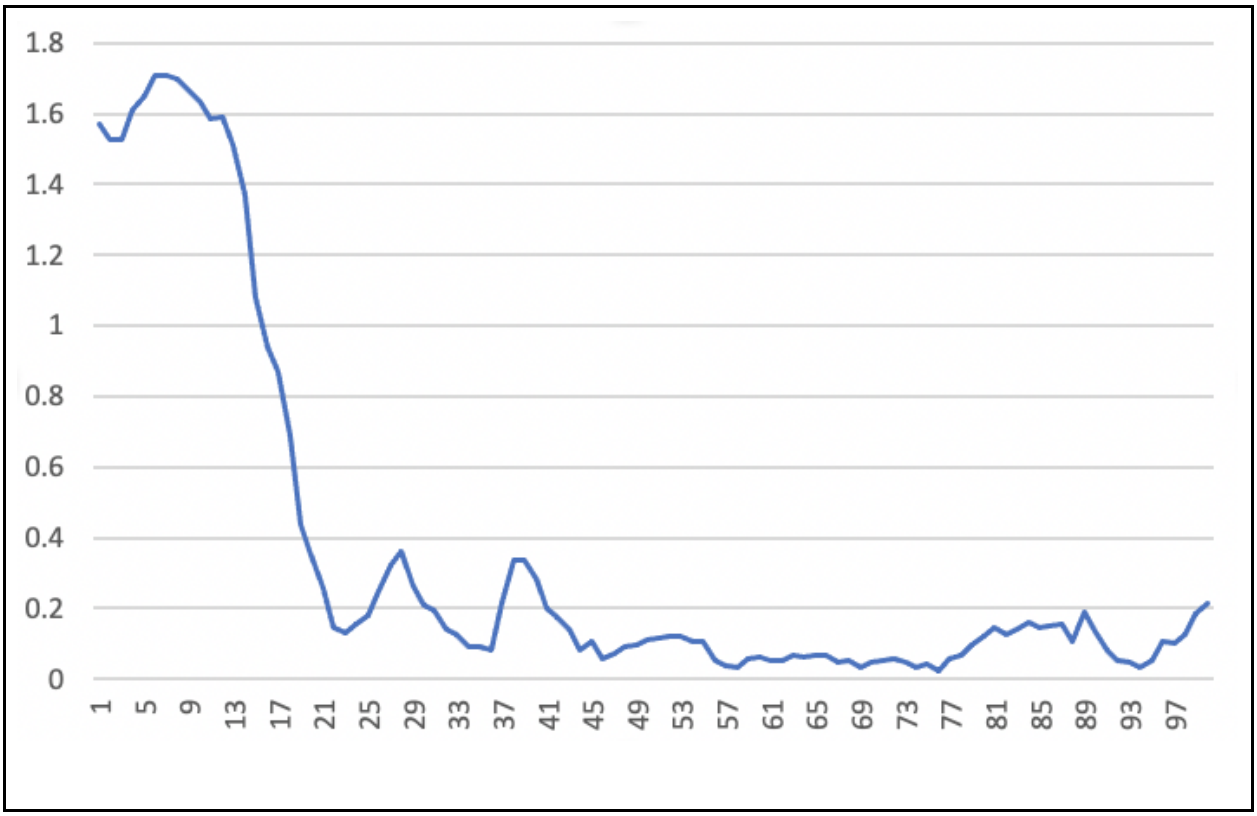
In some symmetric regions, the particle filter has larger prediction error. Because the region is symmetric and there are more positions that are similar to the actual position. In addition, the prediction error increases in regions where there are more boundaries. This is due to the low number of dimensions. If we increased the number of dimensions, we would see this happen less. Also, when the car moves in a large free space, the error will increase, because the distance to the walls are longer than the sensor limit so the sensor data will always be the sensor limit. When we increase the sensor limit, the error will be smaller.

**Problem 7 (10 points).** Modify the LidarProcessing module and the sensor model so that they can make measurements in 8 directions. Run your algorithm with number of particles 1000 and sensor limit 20. Plot the error in position estimation (euclidean distance between actual position and the estimated position) and orientation estimation as a function of algorithm iterations. Since particle filtering is a randomized algorithm, run several instances of the same setup and plot the average error for each time. How does having more sensor data influence the estimation accuracy and converging speed of the algorithm? Record a video of the run. Besides the turtle map window, the video should include the RViz window of the sensor
measurements. Provide a link to the video and include it in the report.

When we change the number directions from 4 to 8, the estimation state will be more accurate, and the converging speed will be faster. That is because we have more information to estimate the actual state, so the filter can quickly locate the position with higher accuracy. The video and the plots of errors are listed below.

https://youtu.be/VQsxATw5GBI




1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97