MPC-MAP Assignment No. 3 - Report

Author: Šimon Prasek

Date: 12.03.2025

Task 1

Using the values that we measured in the first report I implemented pose prediction and then distributed the particles all around the map – I didn't implement functions to check if the particles are in the wall, but if the particle leaves the map, it gets punished with very low weight.

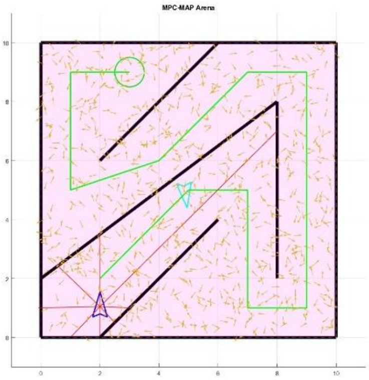
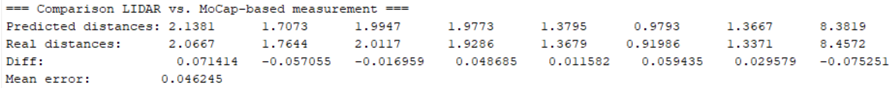


Figure 1 - Distributed particles

Task 2

After implementing the compute_lidar_function I created another function that compares LiDAR and MoCap data. For the weight_particles function I used the Euclidean distance.



```
=== Comparison LIDAR vs. MoCap-based measurement ===
Predicted distances: 2.1381     1.7073     1.9947     1.9773     1.3795      0.9793     1.3667     8.3819
Real distances:      2.0667     1.7644     2.0117     1.9286     1.3679     0.91986     1.3371     8.4572
Diff:                 0.071414  -0.057055  -0.016959   0.048685   0.011582   0.059435   0.029579  -0.075251
Mean error:         0.046245
```

Figure 2 - Comparison of LiDAR and MoCap

Task 3

In the resample_particles function I used algorithm Low Variance Systematic Resampling and implemented injection of random particles for solving the kidnapped robot problem that helped the particle filter to work even better.

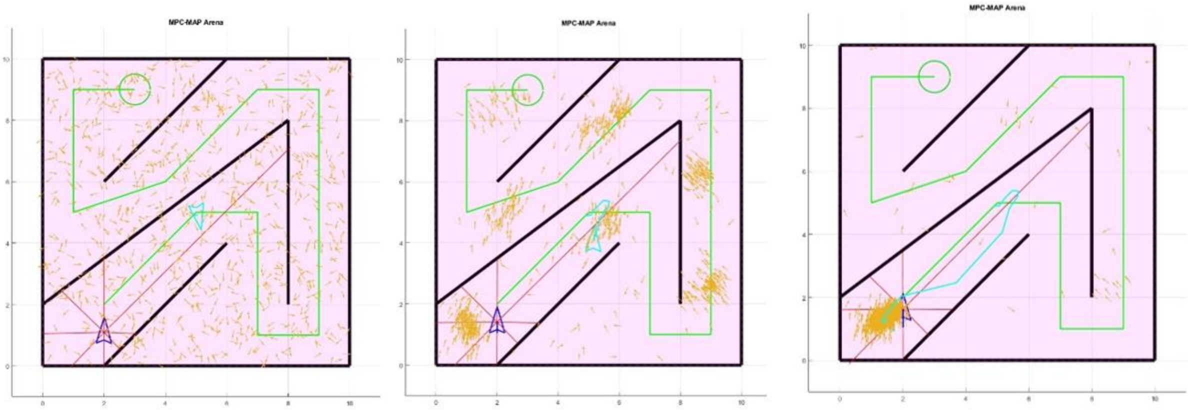


Figure 3 - Stepping of the code - Step 1,10,20

Task 4

For testing, I used the first and third maps. In the figure below, you can see a close-up of the robot and particles. Ive also implemented the estimate_pose function, which calculates the robots pose using the median of all particles.

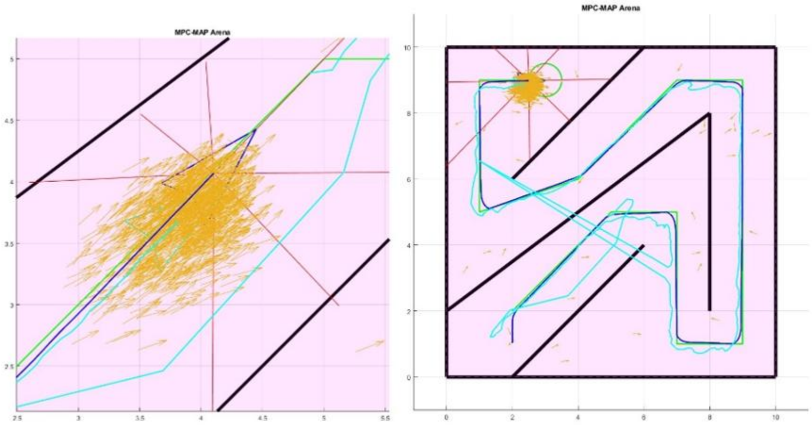


Figure 4 - Close-up and how robot made it to the finish