1 Introduction

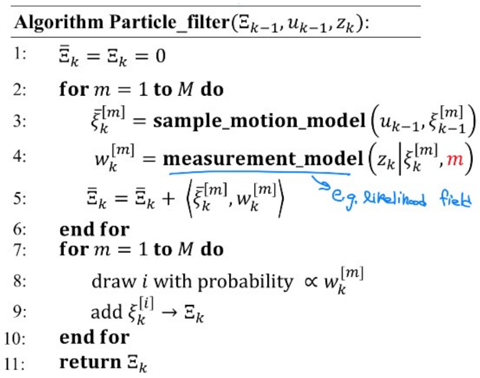
During the lab slot on Nov 11, we recorded LiDAR (/scan), /odom, and /tf pose data of our robot at 2 stationary points (points 2 and 5) on Maze 1. The exact pose of our robot during recording of data is shown in Appendix A.

Although we recorded the /odom ROS topic and thus already have an estimate of the robot, this position is with respect to the docking station. Ideally the docking station would match the provided map frame, but the docking station is not secured to the ground and therefore moved around during the lab session due to human error. Furthermore, the odometry of the robot may have accumulated drift so it is necessary instead to have an accurate estimate of the robots with respect to the true global (map) frame.

As such, the aim of this lab was to localize and estimate the pose of our robot with a particle filter, against the known map provided of Maze 1. Applying a Particle Filter algorithm to this localization problem is known as “Monte Carlo Localization”.

2 Methods

The pseudocode for Particle Filter is shown in the figure below. The corresponding implementation code is in the function `particle_filter_loop(self, posterior_samples)`. As the robot was stationary at each point, a sample motion model is not necessary since the predicted particles at subsequent iterations are the same as the belief particles of the previous iteration.



```
Algorithm Particle_filter($\Xi_{k-1}, u_{k-1}, z_k$):
1:  $\bar{\Xi}_k = \Xi_k = 0$
2:  for $m = 1$ to $M$ do
3:      $\bar{\xi}_k^{[m]} =$ sample_motion_model$\left(u_{k-1}, \xi_{k-1}^{[m]}\right)$
4:      $w_k^{[m]} = \underline{\text{measurement\_model}}\left(z_k \middle| \xi_k^{[m]}, m\right)$   e.g. likelihood field
5:      $\bar{\Xi}_k = \bar{\Xi}_k + \left\langle \bar{\xi}_k^{[m]}, w_k^{[m]} \right\rangle$
6:  end for
7:  for $m = 1$ to $M$ do
8:      draw $i$ with probability $\propto w_k^{[m]}$
9:      add $\xi_k^{[i]} \rightarrow \Xi_k$
10: end for
11: return $\Xi_k$
```

The approach taken is shown in the flow charts below:

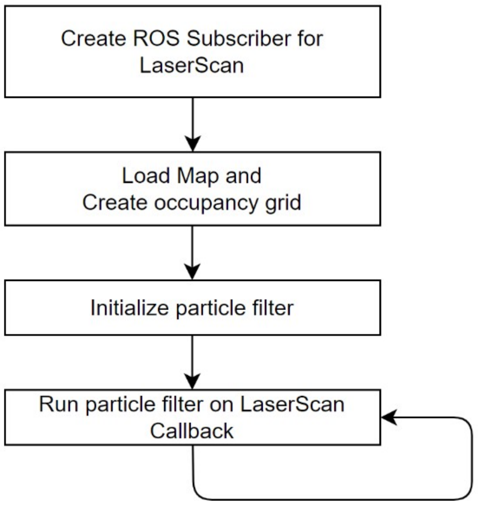

Create ROS Subscriber for LaserScan
Load Map and Create occupancy grid
Initialize particle filter
Run particle filter on LaserScan Callback

Figure 1: High-level Flowchart

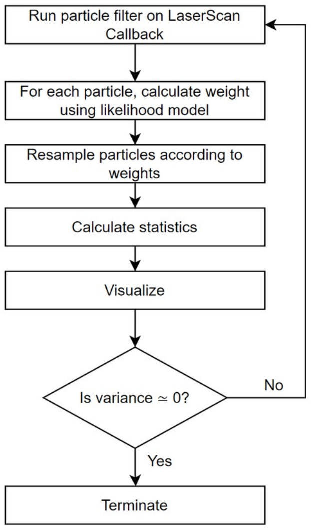

Run particle filter on LaserScan Callback
For each particle, calculate weight using likelihood model
Resample particles according to weights
Calculate statistics
Visualize
Is variance ≃ 0?
No
Yes
Terminate

Figure 2: Particle Filter flowchart

Using the known map, we first create an occupancy grid by loading the provided image and scaling by the map parameters.

First, 2500 particles are initialised in `initialize_particle_filter()` within a small window around the general area of the maze having random position and orientations using a uniform distribution. These particles are then used in the main loop of the particle filter to calculate the weights and redraw samples based on our sensor model. Through experimentation, we chose to use 2500 particles in order to balance a trade-off between accuracy and speed of computations.

The sensor model was constructed using the likelihood field model. The pseudocode of the algorithm is shown below. The corresponding implementation code is in the function `likelihood_field(self, predicted_sample)`.

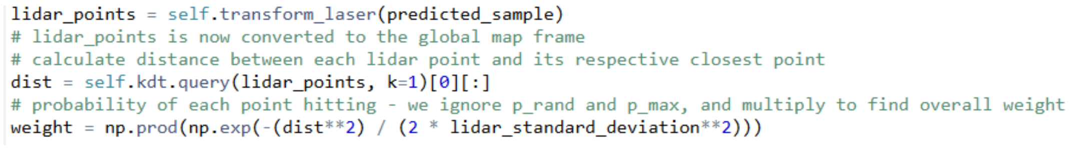
```
Algorithm likelihood_field_range_model($z_k, \xi_k, m$)
1: $q = 1$
2: for $k = 1, 2, \dots, N$ do
3:     if $z_k^i \neq z_{\max}$ then
4:         $x_{z_k^i} = x + x_{\text{sen}}^i \cos\theta - y_{\text{sen}}^i \sin\theta \quad (z_k^i \cos\theta_{\text{sen}}^i = x_{\text{sen}}^i,\ z_k^i \sin\theta_{\text{sen}}^i = y_{\text{sen}}^i)$
5:         $y_{z_k^i} = y + y_{\text{sen}}^i \cos\theta + x_{\text{sen}}^i \sin\theta$
6:         $\text{dist} = \min_{x',y'} \left\{ \sqrt{\left(x_{z_k^i} - x'\right)^2 + \left(y_{z_k^i} - y'\right)^2} \| (x', y') \text{occupied in } m \right\}$
7:         $q = q \cdot (\eta_1 \epsilon_{\sigma_{\text{hit}}}(\text{dist}) + \eta_2 p_{\max} + \eta_3 p_{\text{rand}})$
8:     end if
9: end for
10: return $q$
```

In `likelihood_field()`, we first convert the entire LiDAR scan data to the global map frame using our functions `transform_laser(predicted_sample)` and `lidar_to_cartesian()`. This is done by performing the following:

1. Lookup ROS transform between LiDAR frame and Turtlebot4's base frame (base_footprint) using the ROS2 tf library
2. Determine translation and yaw rotation from above to apply to the lidar points

3. Transform coordinates of each lidar beam from its range value to coordinates in the robot's frame
4. Perform coordinate transform of each lidar beam from the robot's frame to the global map frame based on particle's predicted robot global pose

Additionally, LaserScan measurements that are reported at the max range of the sensor are filtered out. After calculating the lidar scan in the global map frame, the points on the occupancy map closest to these scan points are found using an efficient KD Tree and then the distance is used to compute its likelihood as shown in the code snippet below:



```
lidar_points = self.transform_laser(predicted_sample)
# lidar_points is now converted to the global map frame
# calculate distance between each lidar point and its respective closest point
dist = self.kdt.query(lidar_points, k=1)[0][:]
# probability of each point hitting - we ignore p_rand and p_max, and multiply to find overall weight
weight = np.prod(np.exp(-(dist**2) / (2 * lidar_standard_deviation**2)))
```

The equation for the underlying probability for each point on the lidar scan is as follows:

$$p\left(z_t^k; x_t, m\right) = \eta_1 p_{hit} + \eta_2 p_{max} + \eta_3 p_{rand}$$

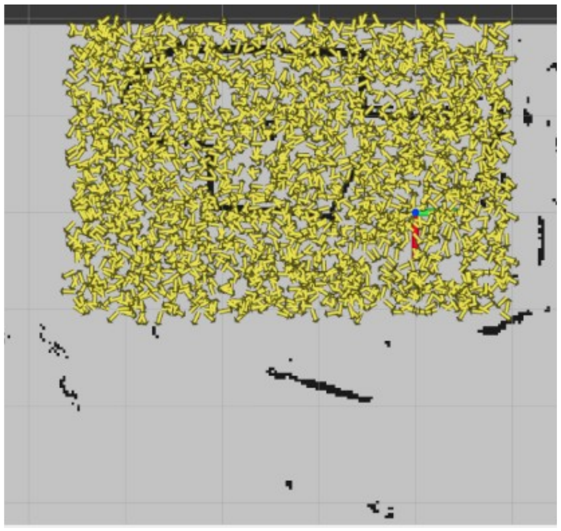
The probability of the laser range p_{hit} is modelled as a Gaussian. The RPLIDAR-A1's datasheet shows an accuracy of 1% for a full-scale range of 12m. Assuming that this accuracy corresponds to a full-scale acceptable measurement range of ±3 standard deviations, the LiDAR's standard deviation was set to be equal to (0.01*12)/3 = 0.04 meters. Using this setup, the likelihood is found and iterated over all the points in the lidar scan.

For the purposes of this lab, the effect of random noise p_{rand} was omitted due to robot being completely stationary during its scan duration reducing the chances of noise from motion. Hence η_3=0. Additionally, weight η_2 was also set to 0 as we explicitly filter out points with max range error, within our `lidar_to_cartesian`() function.

As shown in the flowchart, our particle filter algorithm iterated until the variance of particles is close to 0, as this can be used as an indicator of convergence. As the algorithm iterated, we also visualized the particle filter output in RViz to visually compare the particles sampled, the average pose, and the LaserScan at the average robot pose with respect to the map.

3 Results

The initial and final particle visualization for point 2 is shown below:



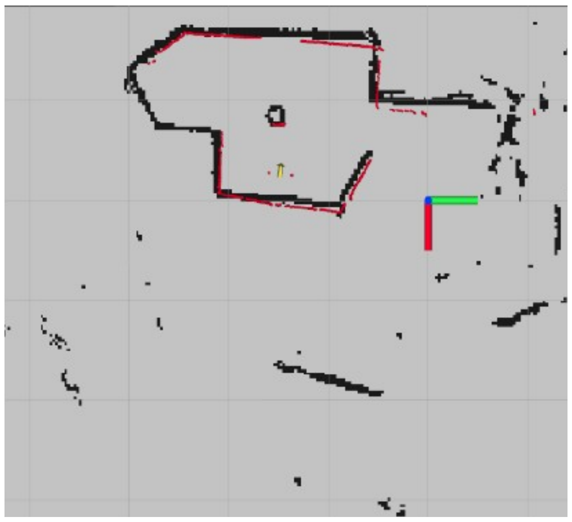
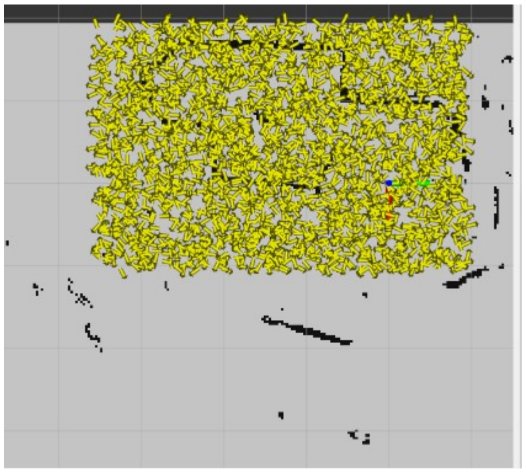


Figure 3: Initial Particle locations

Figure 4: Converged Robot pose (yellow arrow) and projected LaserScan

The initial and final particle visualization for point 5 is shown below:



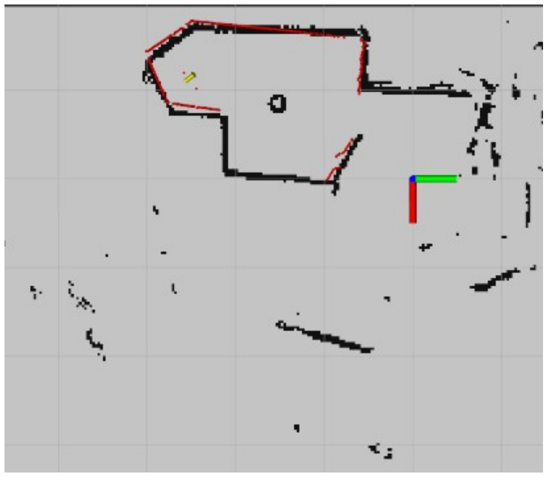


Figure 5: Initial Particle locations

Figure 6: Converged Robot pose (yellow arrow) and projected LaserScan

The final pose of the particle filter algorithm and the corresponding MSE for each of the points is shown in the table below. Note: the MSE was calculated as the mean-squared error between lidar points in the global frame at the last iteration and the occupancy map.

Data	**Point 2**	**Point 5**
X [m]	-0.23	-1.1
Y [m]	-1.49	-2.57
θ [rad]	3.016	2.213
MSE [m^2]	0.0026433835355585736	0.0014893705821526822

From Figures 4 and 6, we can see that the transformed scan matches the provided map quite closely. However, there does appear to be a slight rotation offset.

4 Discussion

The particle filter converges to a single pose with a low MSE, as shown in the table above and from Figures 4 and 6. The predicted pose matches the maze boundaries quite accurately, all within a few or even one iteration of the filter. However, this result is very dependent on the initial samples. As the initial pose of each particle is random, depending on the initial particles, the filter will either converge quickly to a very good candidate in the search space or continue to iterate until it eventually finds the best one. This is due to the nature of particle filters. As there are a discrete number of particles, all with some initial random pose, the particle filter algorithm eventually converges to one of these initial particles given that the robot is stationary. Increasing the number of particles would increase the chances of a good pose being drawn initially, at the trade-off of performance at each iteration of the filter.

The likelihood field model used a product of weights to determine the overall weight of the particle. It was noticed that due to this, the particles quickly converge due to the product of all the lidar points being taken. However, this method is prone to noise. For example, when there is even 1 erroneous lidar measurement, the particle will be classified as “poor” even though it may have been a good candidate. In this case, taking the sum of weights may be a better alternative though this approach would instead be susceptible to local minima. However, through testing, our scan did not appear to be very noisy as it matched well with the provided map.

Another filter that may be used instead of a Particle Filter is a Kalman Filter. The Kalman filter is an efficient and fast filter and requires much less computation and memory than a particle filter since the Kalman Filter does not need to keep track of many particles. However, using a Kalman Filter requires that the state transition and measurement probability must both be linear.

5 Conclusion

The aim of this lab was to localize and estimate the pose of our robot with a particle filter, against the known map provided of Maze 1. Overall, through our implementation, we were able to successfully implement and use a particle filter to localize the robot in the global map frame at both points on Maze 1.

Estimating the global location of a robot, while accounting for sensor noise and odometry drift, is necessary for accurate autonomous navigation.

6 Appendices

6.1 Appendix A: Ground Truth Pose of Robot

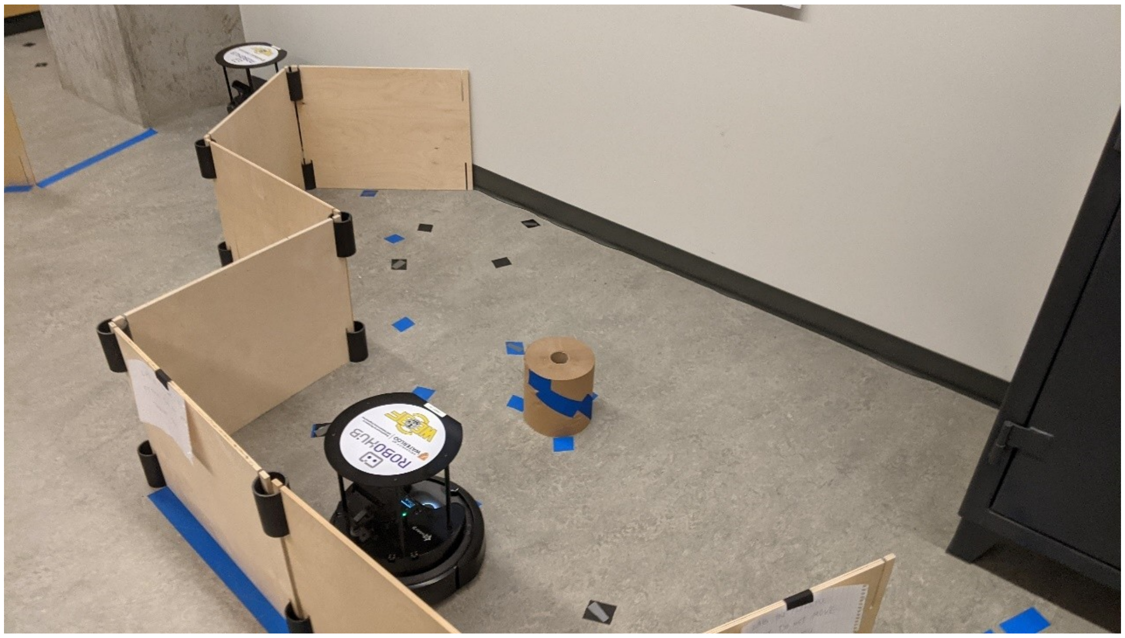


Figure 1: Pose of Robot at Point 2 on Maze 1

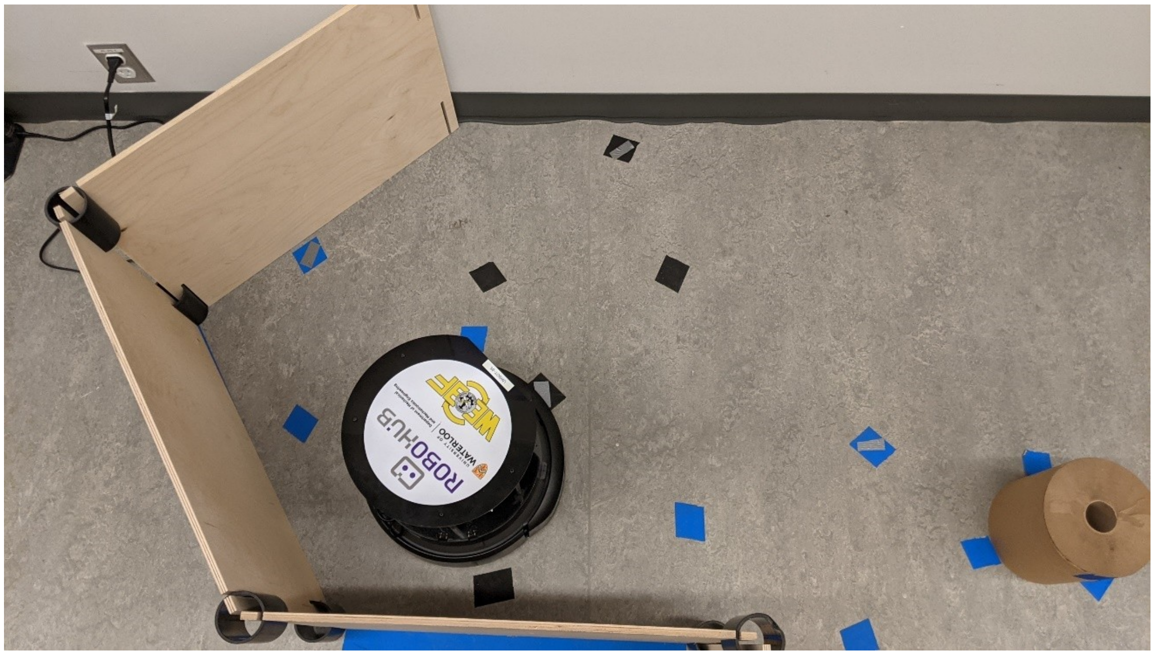
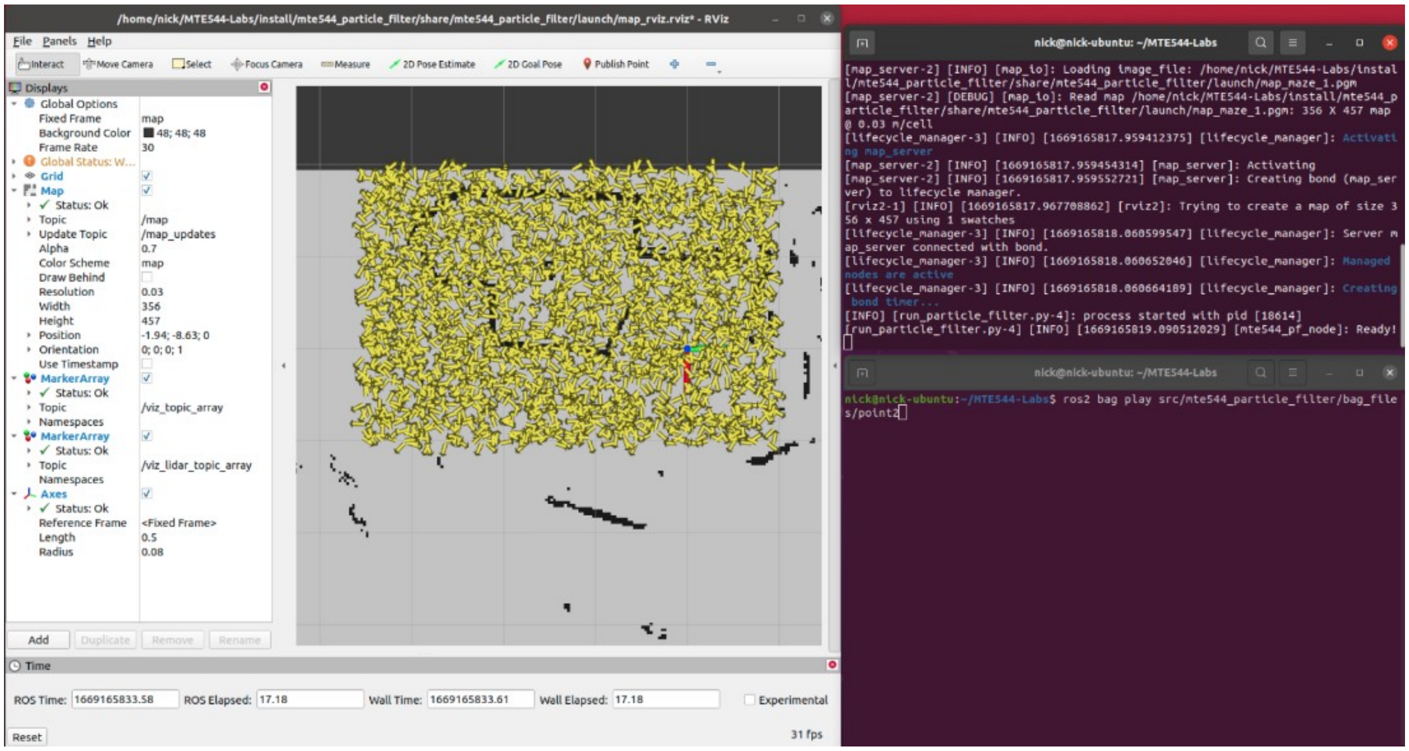


Figure 2: Pose of Robot at Point 5 on Maze 1

6.2 Appendix B: Particle Filter Visualization

Initial Particle filter visualization for point 2:

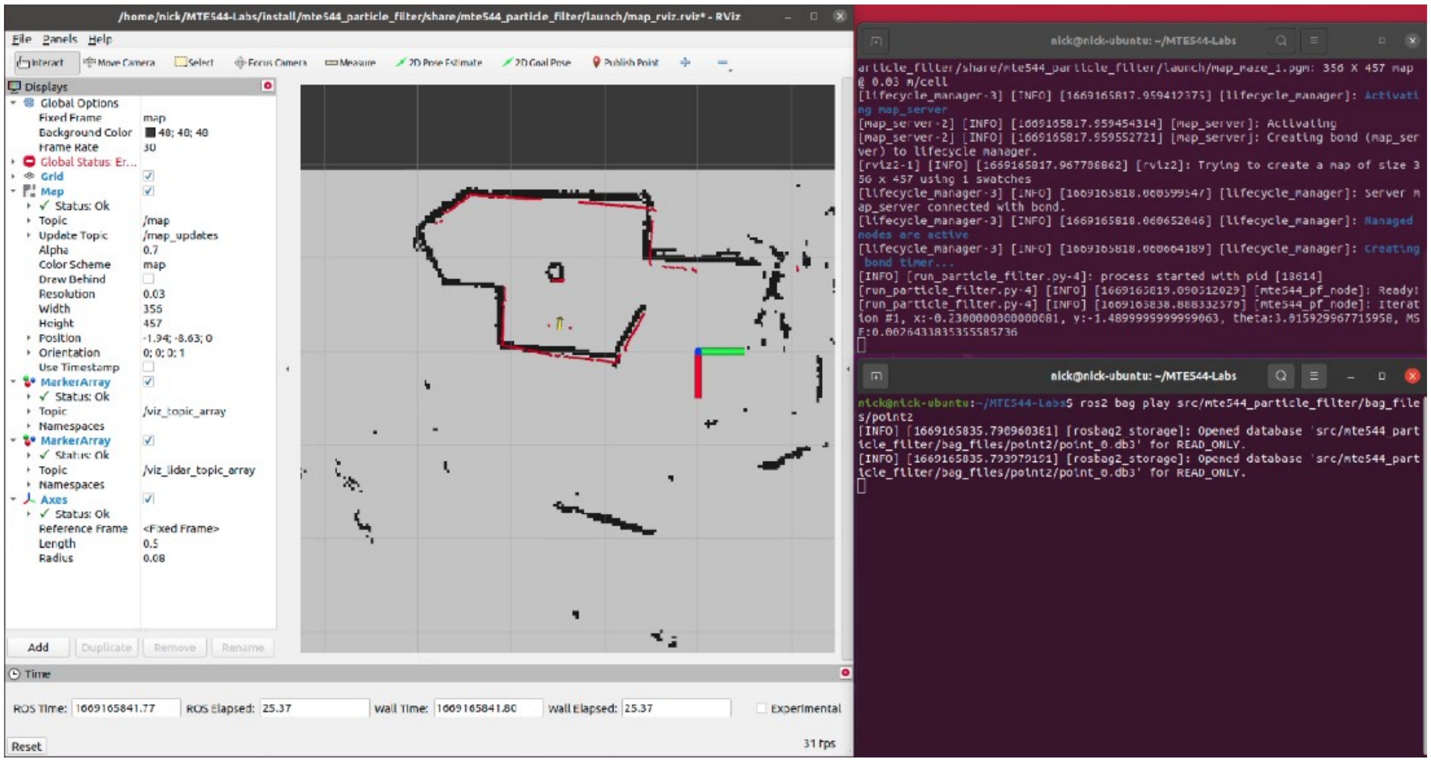


```
/home/nick/MTE544-Labs/install/mte544_particle_filter/share/mte544_particle_filter/launch/map_rviz.rviz* - RViz

Global Options
  Fixed Frame        map
  Background Color   48; 48; 48
  Frame Rate         30
Global Status: W...
Grid
Map
  Status: Ok
  Topic              /map
  Update Topic       /map_updates
  Alpha              0.7
  Color Scheme       map
  Draw Behind
  Resolution         0.03
  Width              356
  Height             457
  Position           -1.94; -8.63; 0
  Orientation        0; 0; 0; 1
  Use Timestamp
MarkerArray
  Status: Ok
  Topic              /viz_topic_array
  Namespaces
MarkerArray
  Status: Ok
  Topic              /viz_lidar_topic_array
  Namespaces
Axes
  Status: Ok
  Reference Frame    <Fixed Frame>
  Length             0.5
  Radius             0.08

ROS Time: 1669165833.58   ROS Elapsed: 17.18   Wall Time: 1669165833.61   Wall Elapsed: 17.18
31 fps
```

```
nick@nick-ubuntu: ~/MTE544-Labs
[map_server-2] [INFO] [map_io]: Loading image_file: /home/nick/MTE544-Labs/instal
l/mte544_particle_filter/share/mte544_particle_filter/launch/map_maze_1.pgm
[map_server-2] [DEBUG] [map_io]: Read map /home/nick/MTE544-Labs/install/mte544_p
article_filter/share/mte544_particle_filter/launch/map_maze_1.pgm: 356 X 457 map
@ 0.03 m/cell
[lifecycle_manager-3] [INFO] [1669165817.959412375] [lifecycle_manager]: Activati
ng map_server
[map_server-2] [INFO] [1669165817.959454314] [map_server]: Activating
[map_server-2] [INFO] [1669165817.959552721] [map_server]: Creating bond (map_ser
ver) to lifecycle manager.
[rviz2-1] [INFO] [1669165817.967708862] [rviz2]: Trying to create a map of size 3
56 x 457 using 1 swatches
[lifecycle_manager-3] [INFO] [1669165818.060599547] [lifecycle_manager]: Server m
ap_server connected with bond.
[lifecycle_manager-3] [INFO] [1669165818.060652046] [lifecycle_manager]: Managed
nodes are active
[lifecycle_manager-3] [INFO] [1669165818.060664189] [lifecycle_manager]: Creating
 bond timer...
[INFO] [run_particle_filter.py-4]: process started with pid [18614]
[run_particle_filter.py-4] [INFO] [1669165819.090512029] [mte544_pf_node]: Ready!
```

```
nick@nick-ubuntu: ~/MTE544-Labs
nick@nick-ubuntu:~/MTE544-Labs$ ros2 bag play src/mte544_particle_filter/bag_file
s/point2
```

Final Particle filter visualization for point 2:

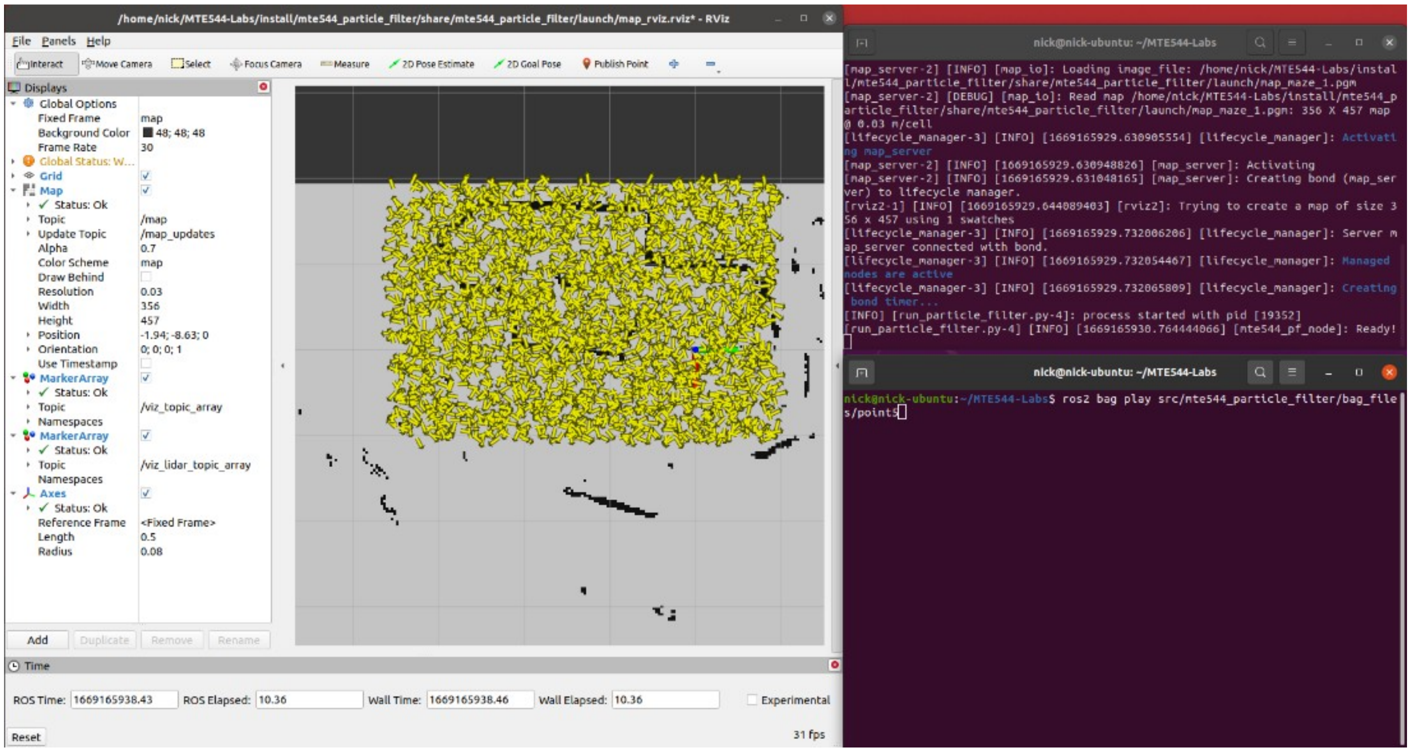


```
/home/nick/MTE544-Labs/install/mte544_particle_filter/share/mte544_particle_filter/launch/map_rviz.rviz* - RViz

Global Options
  Fixed Frame        map
  Background Color   48; 48; 48
  Frame Rate         30
Global Status: Er...
Grid
Map
  Status: Ok
  Topic              /map
  Update Topic       /map_updates
  Alpha              0.7
  Color Scheme       map
  Draw Behind
  Resolution         0.03
  Width              356
  Height             457
  Position           -1.94; -8.63; 0
  Orientation        0; 0; 0; 1
  Use Timestamp
MarkerArray
  Status: Ok
  Topic              /viz_topic_array
  Namespaces
MarkerArray
  Status: Ok
  Topic              /viz_lidar_topic_array
  Namespaces
Axes
  Status: Ok
  Reference Frame    <Fixed Frame>
  Length             0.5
  Radius             0.08

ROS Time: 1669165841.77   ROS Elapsed: 25.37   Wall Time: 1669165841.80   Wall Elapsed: 25.37
31 fps
```

```
nick@nick-ubuntu: ~/MTE544-Labs
article_filter/share/mte544_particle_filter/launch/map_maze_1.pgm: 356 X 457 map
@ 0.03 m/cell
[lifecycle_manager-3] [INFO] [1669165817.959412375] [lifecycle_manager]: Activati
ng map_server
[map_server-2] [INFO] [1669165817.959454314] [map_server]: Activating
[map_server-2] [INFO] [1669165817.959552721] [map_server]: Creating bond (map_ser
ver) to lifecycle manager.
[rviz2-1] [INFO] [1669165817.967708862] [rviz2]: Trying to create a map of size 3
56 x 457 using 1 swatches
[lifecycle_manager-3] [INFO] [1669165818.060599547] [lifecycle_manager]: Server m
ap_server connected with bond.
[lifecycle_manager-3] [INFO] [1669165818.060652046] [lifecycle_manager]: Managed
nodes are active
[lifecycle_manager-3] [INFO] [1669165818.060664189] [lifecycle_manager]: Creating
 bond timer...
[INFO] [run_particle_filter.py-4]: process started with pid [18614]
[run_particle_filter.py-4] [INFO] [1669165819.090512029] [mte544_pf_node]: Ready!
[run_particle_filter.py-4] [INFO] [1669165838.888332570] [mte544_pf_node]: Iterat
ion #1, x:-0.2300000000000081, y:-1.4899999999999063, theta:3.015929967715958, MS
E:0.0026433835355585736
```

```
nick@nick-ubuntu: ~/MTE544-Labs
nick@nick-ubuntu:~/MTE544-Labs$ ros2 bag play src/mte544_particle_filter/bag_file
s/point2
[INFO] [1669165835.790960381] [rosbag2_storage]: Opened database 'src/mte544_part
icle_filter/bag_files/point2/point_0.db3' for READ_ONLY.
[INFO] [1669165835.793079191] [rosbag2_storage]: Opened database 'src/mte544_part
icle_filter/bag_files/point2/point_0.db3' for READ_ONLY.
```

Initial Particle filter visualization for point 5:

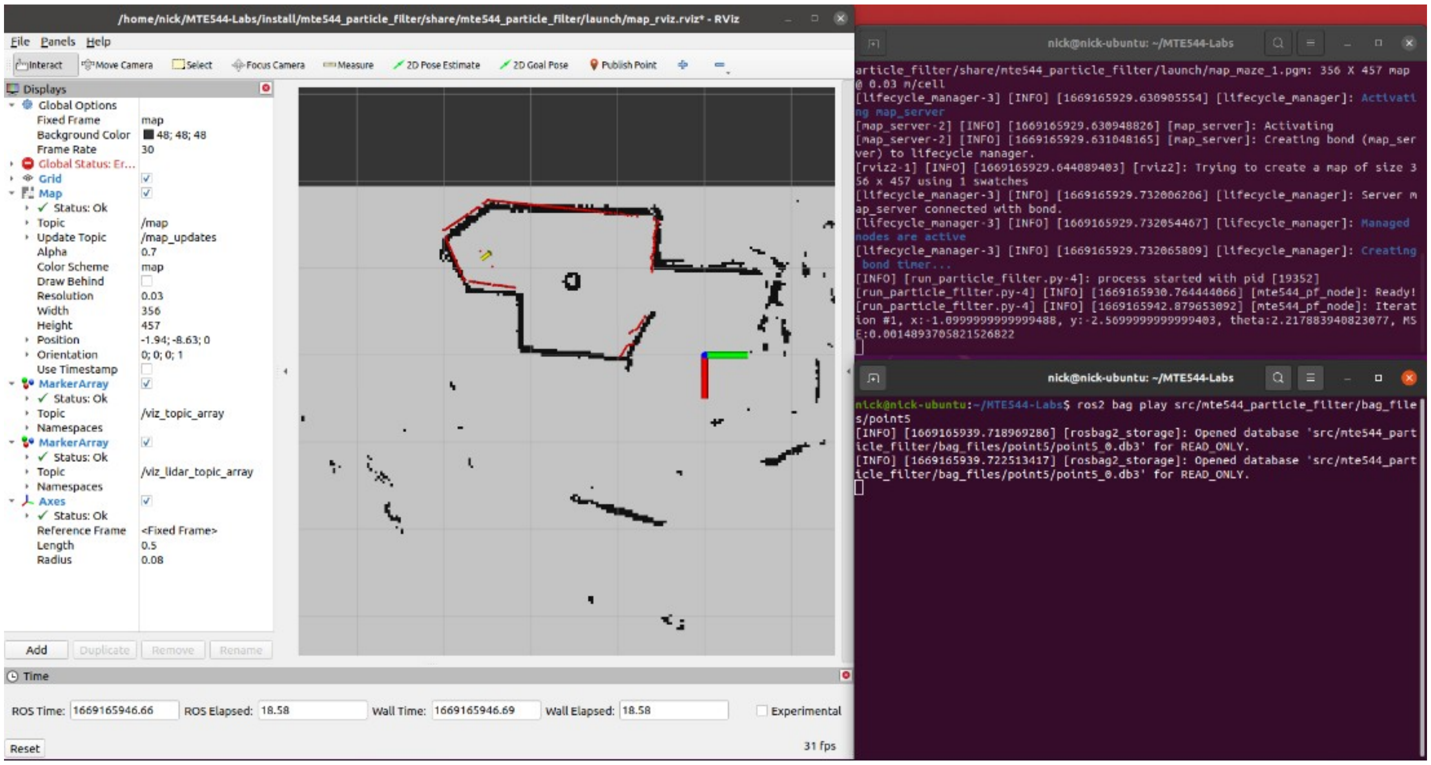


```
/home/nick/MTE544-Labs/install/mte544_particle_filter/share/mte544_particle_filter/launch/map_rviz.rviz* - RViz

Global Options
  Fixed Frame          map
  Background Color     48; 48; 48
  Frame Rate           30
Global Status: W...
Grid                   ✓
Map                    ✓
  Status: Ok
  Topic                /map
  Update Topic         /map_updates
  Alpha                0.7
  Color Scheme         map
  Draw Behind
  Resolution           0.03
  Width                356
  Height               457
  Position             -1.94; -8.63; 0
  Orientation          0; 0; 0; 1
  Use Timestamp
MarkerArray            ✓
  Status: Ok
  Topic                /viz_topic_array
  Namespaces
MarkerArray            ✓
  Status: Ok
  Topic                /viz_lidar_topic_array
  Namespaces
Axes                   ✓
  Status: Ok
  Reference Frame      <Fixed Frame>
  Length               0.5
  Radius               0.08

ROS Time: 1669165938.43   ROS Elapsed: 10.36   Wall Time: 1669165938.46   Wall Elapsed: 10.36
31 fps
```

```
nick@nick-ubuntu: ~/MTE544-Labs
[map_server-2] [INFO] [map_io]: Loading image_file: /home/nick/MTE544-Labs/instal
l/mte544_particle_filter/share/mte544_particle_filter/launch/map_maze_1.pgm
[map_server-2] [DEBUG] [map_io]: Read map /home/nick/MTE544-Labs/install/mte544_p
article_filter/share/mte544_particle_filter/launch/map_maze_1.pgm: 356 X 457 map
@ 0.03 m/cell
[lifecycle_manager-3] [INFO] [1669165929.630905554] [lifecycle_manager]: Activati
ng map_server
[map_server-2] [INFO] [1669165929.630948826] [map_server]: Activating
[map_server-2] [INFO] [1669165929.631048165] [map_server]: Creating bond (map_ser
ver) to lifecycle manager.
[rviz2-1] [INFO] [1669165929.644089403] [rviz2]: Trying to create a map of size 3
56 x 457 using 1 swatches
[lifecycle_manager-3] [INFO] [1669165929.732006206] [lifecycle_manager]: Server m
ap_server connected with bond.
[lifecycle_manager-3] [INFO] [1669165929.732054467] [lifecycle_manager]: Managed
nodes are active
[lifecycle_manager-3] [INFO] [1669165929.732065809] [lifecycle_manager]: Creating
 bond timer...
[INFO] [run_particle_filter.py-4]: process started with pid [19352]
[run_particle_filter.py-4] [INFO] [1669165930.764444066] [mte544_pf_node]: Ready!
```

```
nick@nick-ubuntu: ~/MTE544-Labs
nick@nick-ubuntu:~/MTE544-Labs$ ros2 bag play src/mte544_particle_filter/bag_file
s/point5
```

Final Particle filter visualization for point 5:



```
/home/nick/MTE544-Labs/install/mte544_particle_filter/share/mte544_particle_filter/launch/map_rviz.rviz* - RViz

Global Options
  Fixed Frame          map
  Background Color     48; 48; 48
  Frame Rate           30
Global Status: Er...
Grid                   ✓
Map                    ✓
  Status: Ok
  Topic                /map
  Update Topic         /map_updates
  Alpha                0.7
  Color Scheme         map
  Draw Behind
  Resolution           0.03
  Width                356
  Height               457
  Position             -1.94; -8.63; 0
  Orientation          0; 0; 0; 1
  Use Timestamp
MarkerArray            ✓
  Status: Ok
  Topic                /viz_topic_array
  Namespaces
MarkerArray            ✓
  Status: Ok
  Topic                /viz_lidar_topic_array
  Namespaces
Axes                   ✓
  Status: Ok
  Reference Frame      <Fixed Frame>
  Length               0.5
  Radius               0.08

ROS Time: 1669165946.66   ROS Elapsed: 18.58   Wall Time: 1669165946.69   Wall Elapsed: 18.58
31 fps
```

```
nick@nick-ubuntu: ~/MTE544-Labs
article_filter/share/mte544_particle_filter/launch/map_maze_1.pgm: 356 X 457 map
@ 0.03 m/cell
[lifecycle_manager-3] [INFO] [1669165929.630905554] [lifecycle_manager]: Activati
ng map_server
[map_server-2] [INFO] [1669165929.630948826] [map_server]: Activating
[map_server-2] [INFO] [1669165929.631048165] [map_server]: Creating bond (map_ser
ver) to lifecycle manager.
[rviz2-1] [INFO] [1669165929.644089403] [rviz2]: Trying to create a map of size 3
56 x 457 using 1 swatches
[lifecycle_manager-3] [INFO] [1669165929.732006206] [lifecycle_manager]: Server m
ap_server connected with bond.
[lifecycle_manager-3] [INFO] [1669165929.732054467] [lifecycle_manager]: Managed
nodes are active
[lifecycle_manager-3] [INFO] [1669165929.732065809] [lifecycle_manager]: Creating
 bond timer...
[INFO] [run_particle_filter.py-4]: process started with pid [19352]
[run_particle_filter.py-4] [INFO] [1669165930.764444066] [mte544_pf_node]: Ready!
[run_particle_filter.py-4] [INFO] [1669165942.879653092] [mte544_pf_node]: Iterat
ion #1, x:-1.0999999999999488, y:-2.5699999999999403, theta:2.217883940823077, MS
E:0.0014893705821526822
```

```
nick@nick-ubuntu: ~/MTE544-Labs
nick@nick-ubuntu:~/MTE544-Labs$ ros2 bag play src/mte544_particle_filter/bag_file
s/point5
[INFO] [1669165939.718969286] [rosbag2_storage]: Opened database 'src/mte544_part
icle_filter/bag_files/point5/point5_0.db3' for READ_ONLY.
[INFO] [1669165939.722513417] [rosbag2_storage]: Opened database 'src/mte544_part
icle_filter/bag_files/point5/point5_0.db3' for READ_ONLY.
```