ENMT482 Assignment 1

Oliver Dale and Josh Roberts, Group 18

1 Sensor fusion

1.1 Sensor models

Explain your sensor models (calibration plots should go in the appendix).

1.2 Motion model

Explain your motion model. A figure of the estimated robot speed versus the speed predicted using your motion model would be useful.

1.3 Bayes filter

Explain the Bayes filter you used.

1.4 Results

Include plots of how close your estimate was to the true position, for the datasets with true position. Include a plot of your estimate and its standard deviation for the test dataset. If you use a Kalman filter, it would be useful to show the weights as a function of time.

1.5 Discussion

Discuss what worked well and what improvements could be done.

2 Particle filter

2.1 Sensor model

The sensor model updated the particle weights based on the beacon measurements (Fig. X).

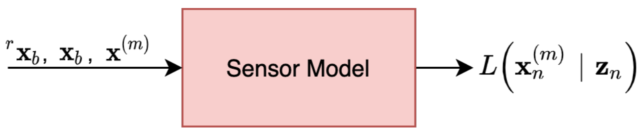

Figure 1: Sensor Model

The beacon pose was converted to a range and bearing to decouple the likelihood function. Each particle calculated its corresponding range and bearing to the beacons global location, which were compared with the measured beacon pose. The particles were then weighted on how close their calculated range and bearing matched the measurements. Particles that had less error were given higher weightings. The decoupled likelihood function was used to update the particle weights (1).

$$a_n^{((m)}) = a_{(n-1)}^m f_R(r_n - r_n^m) f_\phi(angdiff(\phi_n, \phi_n^m)) \tag{1}$$

The PDF distributions of the range and bearing error were assumed to be gaussian and were adjusted through trial and error. As the units of the range and bearing differed, the standard deviations were 0.08 and 0.05 respectively. This meant that a particle was given an equal weighting from both PDF's if it had a range error of 8 cm and an angular error of 3 degrees. A tighter variance was applied to the bearing error, stopping the particles from deviating away from the robots true path.

2.2 Motion model

A probabilistic motion model was used to account for the kinematic uncertainty between the local and global poses. The odometry motion model was selected over the velocity motion model as it proved to be more accurate by using an EKF to fuse odometry and gyroscopic measurements. The change in local position was converted to a global coordinate system by parameterising the change into two independent rotations and a translation (ϕ_1,ϕ_2 and d). Thus, a PDF could be created and sampled from for each movement, allowing the particles to spread out at each step. Each particles pose was updated using the following equations.

$$x_n = x_(n-1) + dcos(\theta_(n-1) + \phi_1) \tag{2a}$$

$$y_n = y_(n-1) + dsin(\theta_(n-1) + \phi_1) \tag{2b}$$

$$\theta_n = (\theta_(n-1) + \phi_1 + \phi_2) \tag{2c}$$

Where ϕ_1, ϕ_2 and d were each sampled from the joint PDF $f_D = (d; d^{'})$. The standard deviations were determined via trial and error. As the units of the range and bearing differed, the standard deviations were 0.02 and 0.001 respectively. If one standard deviation of noise was applied to the pose of a particle, it moved 2 cm and 0.1 degrees. Selecting the standard deviations was a balance, ensuring the particles covered the true location of the robot while reducing the uncertainty of the posterior belief.

2.3 Implementation

The particle filter was implemented to start from an unknown position. To ensure the map was sufficiently covered, one-hundred particles were uniformly spread of the entire domain. During the resampling process, ten particles were randomly placed around the posterior belief from a uniform distribution. The upper and lower bounds of the distribution were proportional to the inverse of the squared sum of the particle weights. Thus, when the particles weights were low, the distribution to sample from was larger. This allowed the spread of the random particles to dynamically change depending on how accurate the particle filter was. This stopped the particles from converging to an incorrect start location during the beginning of the localisation, as random particles were sampled from a wide distribution to cover the true position. By adding random particles, the chance of an unlucky series of numbers wiping out good particles (known as particle deprivation) was also reduced. As a last resort, if the sum of the particle weights dropped below a threshold (signifying a lost robot) the particles were randomly dispersed around the posterior belief to find the true location of the robot.

2.4 Results

The particle filter was able to closely follow the SLAM trajectory and had an average range and bearing error of 0.1 m and 2 degrees. Figure X, shows two artifacts of the particle filter. During the initial resampling process, the estimated path zigzags around as the particles converge onto the true path. The estimated path also overshoots around the corner where they are no sensors in sight. This shows the inherent inaccuracies within the motion model. Compared to the odometry alone, the beacon measurements remove the introduced skew to the path.

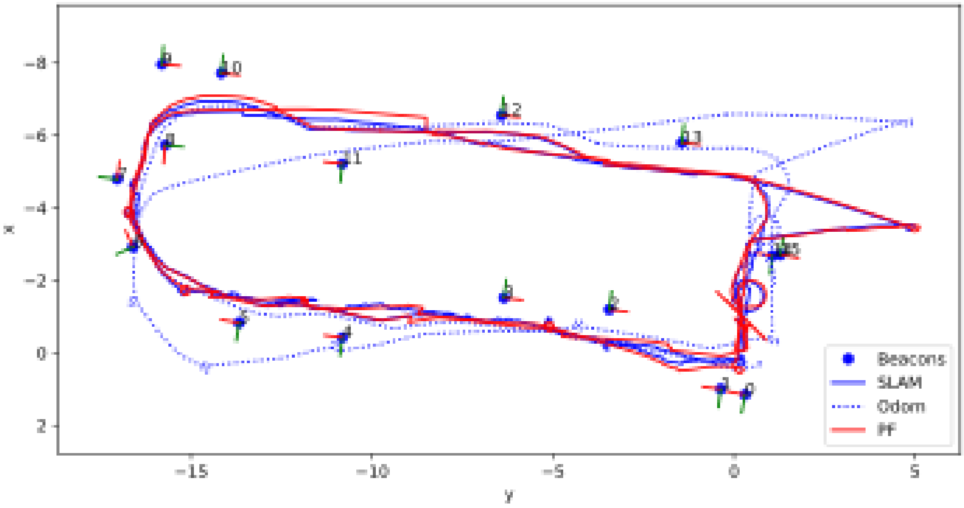

Figure 2: Particle Filter Path

2.5 Discussion

The particle filter was able to successfully track the robot without a known starting position. The random resampling worked well and reduced the effect of particle deprivation. This approach could be further improved by altering the quantity of random particles injected, based on the sum of the particle weightings. If the weighted sum of the particles is low, injecting more random particles increases the chance of finding the location of the robot. When the weighted sum of the particles is high, the posterior belief becomes less affected by the random particles. Further improving this approach, an adaptive resampling approach could be implemented such as KLD-Sampling. This controls the quantity of the total particles by measuring the particles distribution across the state space using bins. By adaptively changing the particles the computational cost and variance of the estimated likelihood can both be minimized. Another area of improvement is the motion model. While the odometry model provided accurate measurements for the particles to move during periods with no beacons in sight. It could be improved by fusing the model with the command velocities via an estimator such as BLUE. This is useful information that can improve the motion models prediction.

3 SLAM

Show the map you obtained from the Lab using the ‘gmapping‘ program and provide your observations regarding ‘gmapping‘’s performance.

Instructions

1. The reports can be created in Word or LaTeX. Use the appropriate template but delete all the instructions in blue.
2. The page limit is five pages with an optional one page appendix. No title pages please. We will deduct 10% for every page over the page limit. Do not squeeze the margin or use small fonts (12pt please).
3. Ensure your names and group number are in the title block.
4. No abstract, introduction, or conclusion is required.
5. Submit your reports as a PDF document through Learn. We will deduct 10% for non PDF documents.
6. Have a read of my guidelines for writing a report, `https://eng-git.canterbury.ac.nz/mph/report-guidelines/blob/master/report-guidelines.pdf`