# Particle Filter

Particle Filter Localisation (PFL) works by using a distribution of particles to estimate the robot’s location. Each particle acts as a hypothesis of the real robot’s position and orientation. Sensor readings are simulated for each of the particles and compared to actual sensor readings from the robot. This difference then corresponds to a weighting for each particle, which reflects the probability that the particle represents the robot’s true state. As such, the weighting then controls how much each particle is resampled. If a particle’s weighting is high then more particles in the next generation will be in its general vicinity. This continues until the particles converge upon the true location of the robot.

Initial Setup

The PF initialises with five hundred particles spread randomly across the map, but within 10cm of the borders like the true robot. The robot and all particles are configured to use a location signature of six scans as this is felt to reflect the limit of the real robot’s capabilities within the time frame.

Comparison Method and Weight Assignment

The initial method used to compare particle scans against the expected scan was the mean error. However, this was changed to the Euclidean distance between the scan vectors in order to emphasise any high difference between the scans. A Gaussian distribution is then used to assign a weight to each particle based on the difference calculated. The smaller the difference the higher the weight.

The variance and damping factor of the Gaussian distribution were set through trial and error from considering graphs such as seen in Figure 1 whilst running the simulation. The variance controls the level of difference a particle requires to be considered as a good estimate and to be assigned a weight from the Gaussian. It also controls the rate at which a decrease in difference gives an increase in weight. The damping factor is used to encourage greater accuracy from the PF. It means that the difference in weights of the better particles have to be a magnitude greater than the worst particles for the effect to be seen in resampling. Nonetheless, setting the damping factor too high lead to a lack of convergence and the PFL taking an inordinate amount of time, so a balance was struck.

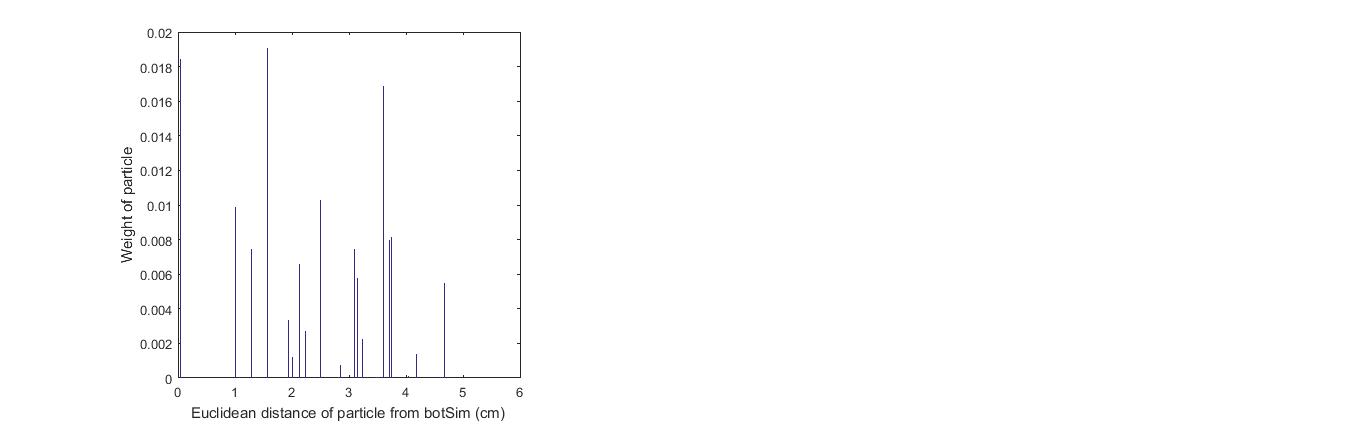
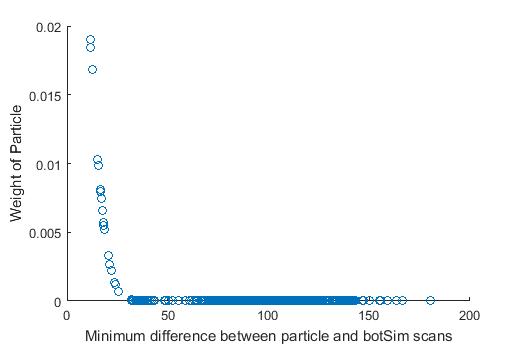


Figure 1b: Difference vs weight

Figure a: Euclidian distance vs weight

As the order of a sensor scan is dependent on the particle’s orientation, simply checking each particle’s scan once might disregard a particle which is in the correct location, but in the wrong orientation. So the scan of each particle is cyclically shifted, the weight of the particle in each orientation calculated and then the particle is turned to the orientation giving the highest weight.

Resampling and Movement Model

The weights of the particles are normalised and then a roulette wheel selector is used on their cumulative sum to pick the location and orientation of the next generation of particles. The position of the respawned particles are then calculated. Their standard deviations are then compared against a convergence threshold to see if they have converged upon a location in Cartesian space and a location estimate will be returned. If not, the botSim and all the particles perform a motion and the process repeats.

Estimating Location

Both the mean and the mode of the particle positions in Cartesian space are used to estimate position as experiments never found one was overwhelmingly better than the other. As such both are estimated and compared against the observed scan and the once with the lowest distance is chosen as the final estimate of position.

# PFL Problems

Location Inaccuracy

To protect the path planning from inaccuracies in the PFL position estimate, the following safeguards were put into place:

* After turning, but before the movement, compare the distance of the step against observed scan readings and reinitialise the PFL if performing the movement would lead to a collision.
* After each step, compare the expected scan against the observed scan and reinitialise the PFL if the difference is above a set threshold.

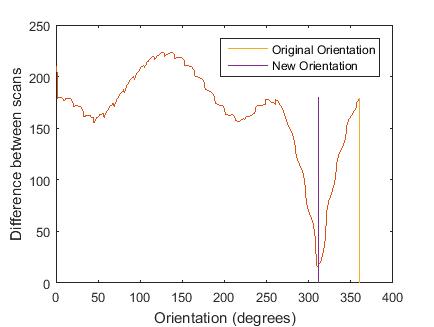
Orientation Inaccuracy

Figure : Orientation vs Difference

For the path planning and navigation, even moderate inaccuracies in the orientation estimate could cause problems and the PF was found to be quite inaccurate. In order to better the orientation estimate, once the PFL has converged upon an estimate, that estimate is compared in all possible orientations (to the degree) and the orientation with the least difference to the observed scan is chosen. Figure 2 demonstrates how the original orientation of the estimate can be quite different to the best orientation. Additionally, it shows how there is clearly a best choice of orientation.

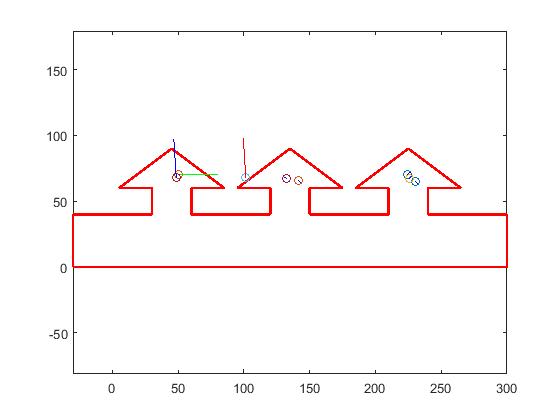
Localising in Symmetric Maps

Figure 3: Symmetric Map

When a map has symmetric areas as seen in Figure 3, the PFL struggles to differentiate between them as the scans in all these areas would be very similar. To move the botSim away from such areas as quickly as possible and keep the botSim exploring from the whole map, a movement strategy was adopted. The strategy is to locate the direction with the greatest reading from the scan and then move in that direction for a random fraction of that distance. However, this ran into yo-yoing problems, where the botSim would move back and forth in a straight line and not perform any exploration. To resolve this, the above strategy was set to happen 70% of the time and the rest of the time the direction of travel would be chosen at random from the scan.