# Part B

* A brief explanation of your motion and sensor models.

The odometry motion model for the root was utilised due to its improved accuracy over the velocity motion model by utilising actual measurements. The odometry measurements given are fused readings from the odometry wheel sensors and IMU which estimate a 2-dimensional local pose for the robot as in Equation X:

(X)

Given the current and previous local poses of the robot, the probability density of the transition between poses could be modelled. The error in the transition was assumed to be small. Therefore, the global pose was parameterised identically to the local pose change as in Equation X:

(X)

Where is calculated as in Equation (X):

(X)

To determine and , the angle from the global x-axis to the direction of travel between the two poses (where the magnitude of the line is ) must be calculated first as in Equation (X):

Giving in Equation (X):

(X)

And giving in Equation (X):

(X)

The angle subtractions were calculated by an angle difference function which ensured the result was between .

The first particle pose for the 100 defined particles was the initial slam pose to give an accurate starting point. The next estimate of the particle poses was given by adding the transition between poses to the original particle estimate. Noise was also added to the estimates due to the residual error in the motion model. The errors were Gaussian modelled with error means . The variance of each direction were altered to best model the true behaviour of the estimate error in each respective direction giving;

Due to the motion model, the particles move a certain distance with some additive noise. For each of the particle measurements from the motion model, the weightings of each particle need to be updated.

Firstly, the relative pose measurements are broken into range and bearing as in Equations (X) and (X):

(X)

(X)

Where and are the measured pose of the beacon in the robot’s camera coordinate frame. The range and bearing measured by the particle are described in Equations (X) and (X):

(X)

(X)

Where and are the true locations of the currently visible beacon in the map coordinate system. , and is the mth particles 2D location and orientation in the map coordinates. Each particle determines its range and bearing to the measured beacon. The particles weight depends on how close its range and bearing are to the measured values. The particles with smaller discrepancies get higher weights according to Equation X:

(X)

Where is the range error PDF and is the bearing error PDF. The particle weightings therefore get proportionally smaller as the range increases. Hence, if a beacon is not located for a longer period of time the estimate of the robots position varies greater.

* A description of what you thought worked well about your estimation approach, and what you could do to improve it.

The first stage of the estimation approach was to determine a motion model which accurately tracked the odometry. Thus, confirming the equations implemented for the pose transitions were correct. Within the estimation this was very accurate. Therefore, the particles could be updated with the motion model estimation alongside the defined noise values. The resulting estimation plot is shown in Figure 3. Again, this closely follows the odometry’s estimated path. Whilst this is expected, the true path of the robot as estimated by the SLAM algorithm is much different, especially on the second lap. By implementing the sensor model to update the weightings of the particles to ensure the positioning was accurate with reference to the beacons located around the path. From the result of the sensor models implementation in Figure 1, it is observed that the robots position estimate is often corrected towards the SLAM estimation of the robot’s position when the robot gets close to a beacon. This confirms that the sensor model is working as expected, but without the perfect weightings to implement the correct adjustment every time. Thus, during the estimation of the second lap, the position is very similar to the odometry estimation of the position which is less than desirable. To improve this requires adjustment of both the particle estimation errors in the motion model and the adjustment of the PDF mean and variances for the sensor model. The confirmation that the first lap is much more accurate than the second lap, the erros between the estimated position and the SLAM estimate position is plotted in Figure 2. It can be observed that around halfway through the iterations the error becomes more erratic in both directions.

* Plots of your estimated trajectory alongside the position from SLAM.

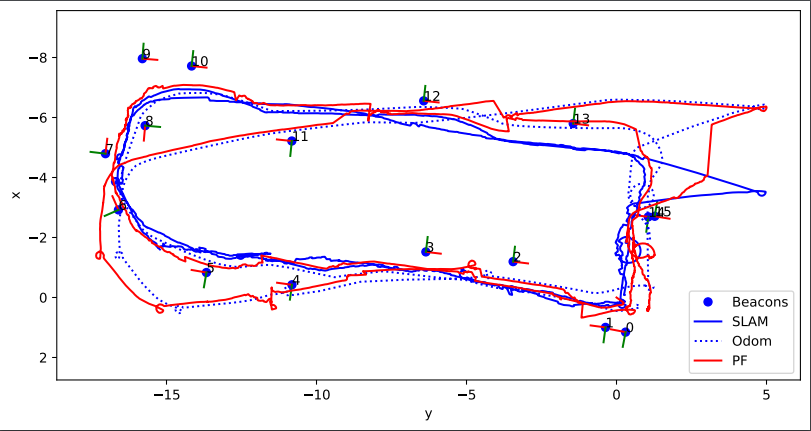


Figure : Resulting plot from the particle filter model implemented where the red line is the estimated route of the Turtlebot.

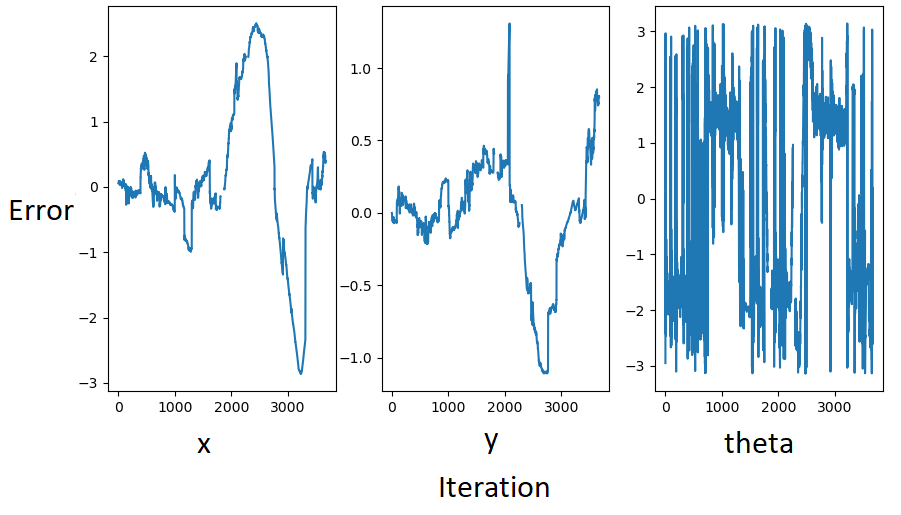


Figure : The corresponding error plots between the estimated route and the route estimated by the SLAM algorithm.

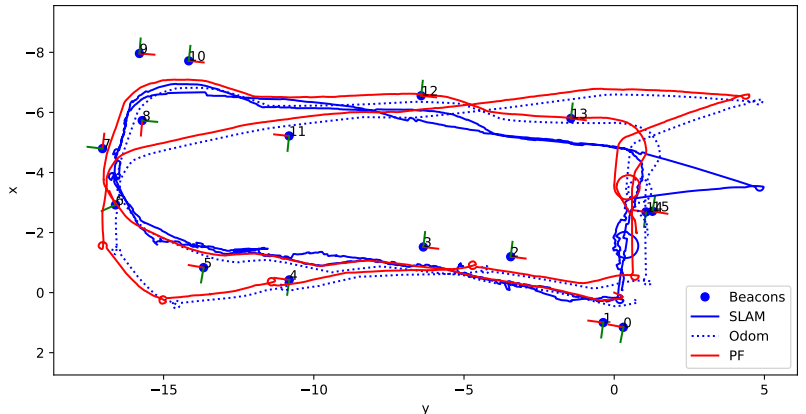


Figure : The estimation from only the motion model.