Machine Learning and Artificial Intelligence for Robotics

Assignment: 0

Boston Cleek

October 14, 2019

Problem 1

The kinematic motion model bellow describes the change in pose for a differential drive robot.

$$\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} u_1 cos(\theta) \\ u_1 sin(\theta) \\ u_2 \end{bmatrix} dt \tag{1}$$

Where u_1 is the linear velocity input, u_2 is the angular velocity input, θ is the current angular position, and dt is the change in time. The change in pose is defined by the horizontal position of the robot dx, the vertical position dy, and the yaw angle relative to the world x-axis is $d\theta$. In the frame of the robot the positive x-direction is forward and the positive y-direction is left. Positive angular velocity and angular position is considered counter clockwise. The model is not linear in the linear velocity control input and angular position input because of the trigonometric functions defining dx and dy. The model is linear in the angular velocity control input because $d\theta$ is linearly proportional to $u_2 dt$.

Problem 2

The control inputs listed bellow were applied to the motion model given in Eq.(1).

u_1 (m/s)	u_2 (rad/s)	t (s)
0.5	0	1
0	-1/2π	1
0.5	0	1
0	1/2π	1
0	0	1

Table 1: Control inputs and specified duration

The trajectory of the robot in global x and y coordinates given the control sequence in Table (1).

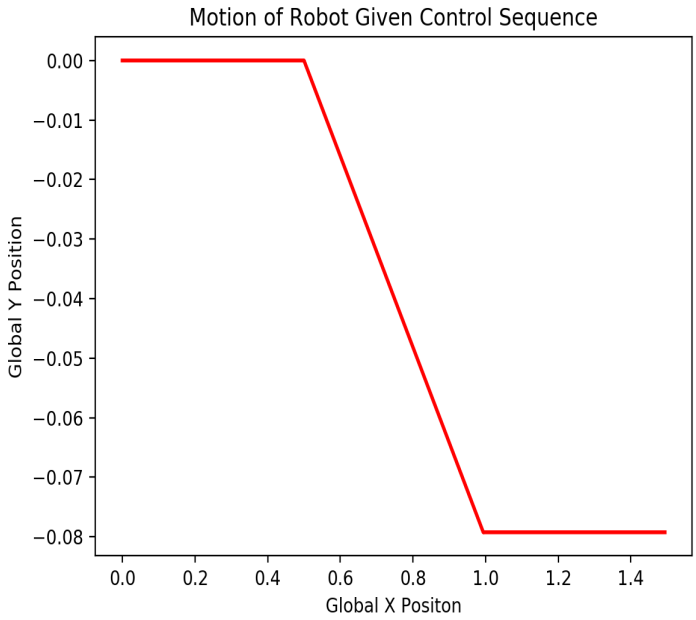

Figure 1: Trajectory of robot give control sequence

Problem 3

The control commands in data set 0 in *Odometry.dat* were given to the controller. The results comparing the dead-reckoned trajectory to the ground truth trajectory are shown bellow.

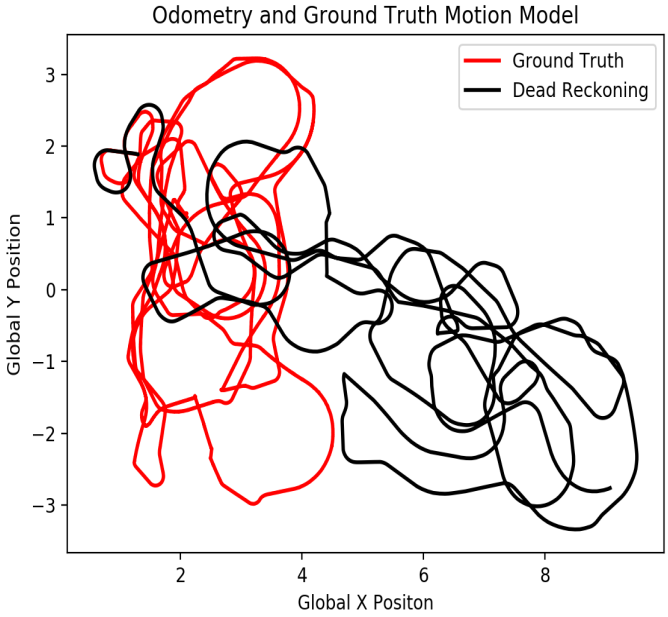

Figure 2: Ground truth and dead-reckoned trajectory of robot

The robot starts approximately at (x,y) = (1.3m, 1.9m). From Fig.(2) the dead-reckoned path initially follows the ground truth path. After approximately 300 control commands have been issued the dead-

reckoned path starts to drift from the ground truth. After all the controls from the odometry file have been issued the dead-reckoned path differs significantly from the ground truth path. The difference is due to the lack of stochastic modelling in the motion model.

Problem 4

The linearization performed by the UKF is accomplished by representing the belief as a Gaussian and extracting sigma points from it. The sigma points are located at the mean μ and along the axis of the covariance matrix Σ. The equation for the sigma points is shown in line 2 bellow. There are $2n+1$ sigma points for an n-dimensional Gaussian. The sigma points are passed through the nonlinear state transition function g. In line 2 the parameter λ is defined as $\alpha^2(n+\kappa)-n$, where α and κ are scaling parameters that define how far the sigma points are away from the mean.

After a Gaussian is passed through a nonlinear function it is no longer a Gaussian. However, the predicted Gaussian is recovered by applying weights to each of the sigma points. Each sigma point has two weights. One weight w_m is used when recovering the mean and the other w_c is used when recovering the covariance. The predicted mean and predicted covariance are shown in lines 4 and 5 respectively. They key point is the UKF avoids linearizing around the mean using a Taylor series like the EKF does.

Note the motion model noise covariance matrix R_t is applied to the predicted covariance matrix in line 5 and the measurement model noise covariance matrix Q_t is applied in line 9 to the uncertainty matrix S_t.

1: **Algorithm Unscented_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):**

2: $\mathcal{X}_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} \quad \mu_{t-1} - \gamma\sqrt{\Sigma_{t-1}})$

3: $\bar{\mathcal{X}}_t^* = g(u_t, \mathcal{X}_{t-1})$

4: $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$

5: $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + R_t$

6: $\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} \quad \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$

7: $\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$

8: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

9: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

10: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

11: $K_t = \bar{\Sigma}_t^{x,z} \; S_t^{-1}$

12: $\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$

13: $\Sigma_t = \bar{\Sigma}_t - K_t \; S_t \; K_t^T$

14: *return* μ_t, Σ_t

Figure 3: The unscented Kalman filter algorithm according to [1]

The sigma points are sampled again after the Gaussian is recovered in lines 4 and 5. The new sigma points are passed through the measurement model in line 7. The cross-covariance matrix in line 10 determines the relationship between state and observation. The Kalman gain K_t is computed in line 11 based on the cross-covariance and the uncertainty matrix. Note in line 12 the difference between the actual measurement z_t and the predicted observation $\hat{z}_t$ is used to determine the affect the Kalman gain will have upon updating the mean. The Kalman gain and the uncertainty matrix are used to update the covariance matrix in line 13.

Problem 5

The measurement model outputs the range and bearing of an observed landmark relative to the robot [2]. The range is defined as the distance from the robot to the landmark. The bearing is angular position of the landmark relative to the robots x-axis frame.

$$\begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x_t - x_i)^2 + (y_t - x_i)^2} \\ \arctan(\frac{y_i - y_t}{x_i - x_t}) - \theta_t \end{bmatrix} \tag{2}$$

Where x_i and y_i are the known global coordinates of each landmark that is observed. The variables x_t, y_t, and θ_t are the global coordinates of the robot at time t.

The outputs are range r (m) and bearing ϕ (rad) respectively. The illustration bellow depicts the outputs of the measurement model relative to the robot.

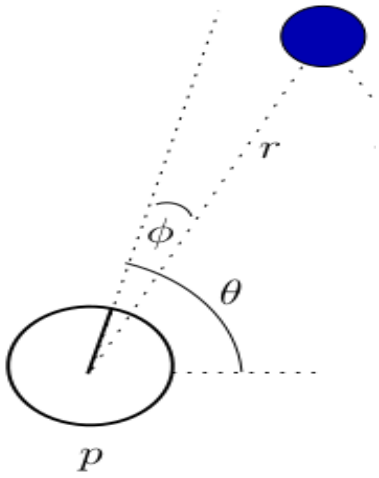

Figure 4: The mobile robot observing a landmark (p is the position of the robot), illustration from [3]

Problem 6

The measurement model from Eq.(2) was used to compute the range and bearing on a set of landmarks when the robot is at the positions described bellow. The range and bearing outputs from the measurement

model can be converted into global coordinates and used to compute the error in the estimated landmark positions. The error will be zero unless noise is added to the measurement model. However, if noise is added the error will be the noise added to the model.

Position (x, y, θ)	Landmark (Subject #)
(2,3,0)	6
(0,3,0)	13
(1,-2,0)	16

Table 2: Position of robot when landmark is observed

The measurement outputs:

r (m)	ϕ (rad)
8.093	-1.758
2.572	-1.206
5.063	1.106

Table 3: Range and bearing outputs

Problem 7

The full implementation of the unscented Kalman filter algorithm can be seen in $filter.py$ starting on line 402. There are functions above line 402 corresponding to each step in the algorithm described in Table(3).

The filter uses the following scaling parameters ($\beta = 2$, $\alpha = 10^{-5}$, and $\kappa = 0$). The two matrices bellow represent the motion and measurement noise covariance matrices used in the implementation of the algorithm. The values used in R_t are ($\sigma_x = 0.004$, $\sigma_y = 0.004$, and $\sigma_\phi = 0.0085$) and values used in Q_t are ($\sigma_r = 0.002$ and $\sigma_\phi = 0.0085$) unless noted otherwise. The reasoning for the choice of these values is described in further detail in problem 9.

$$R_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \quad Q_t = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix}$$

Problem 8

The same controls in Table (1) were applied to the UKF. The output from the UKF appears identical to the output from the motion model. No measurements were included therefore the algorithm returns the predicted mean and covariance from lines 4 and 5 Fig.(3).

The controls from *Odometry.dat* and the measurements from $Measurement.dat$ were applied to the UKF. The trajectory output from the UKF more closely follows the ground truth trajectory. The landmarks observed by the robot were used to compute the posterior belief. In the presence of a measurement the uncertainty matrix and the cross-covariance matrix are computed to determine the Kalman gain in line 11 of the algorithm in Fig.(3).

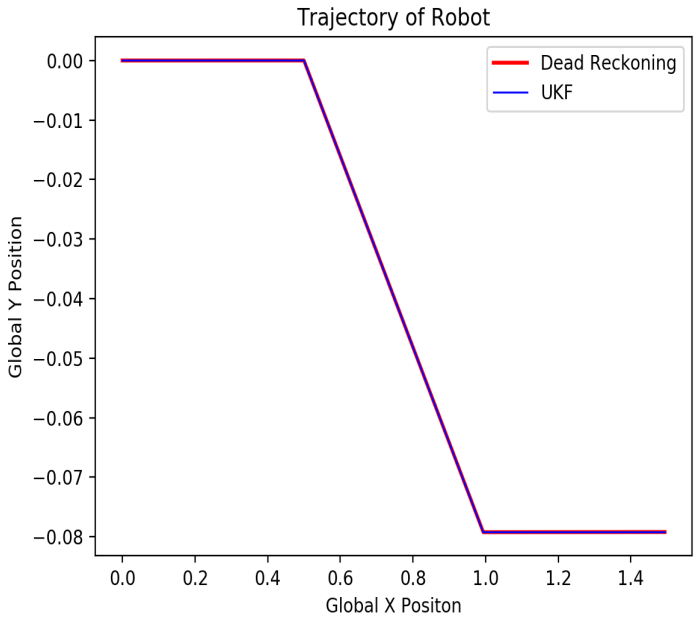

Figure 5: Trajectory given control sequence for motion model and UKF

The actual measurement z_t is used is compare to the predicted observation $\hat{z}_t$ in line 11 when updating the mean μ_t. The Kalman gain is used along with the uncertainty matrix when updating the covariance matrix Σ_t. The measurements reduce the magnitude of the covariance matrix therefore, the robot is more certain about its position. Initially, there are no measurements and the filtered path closely resembles the dead-reckoned path. When a measurement is detected the filtered path converges to the ground truth path.

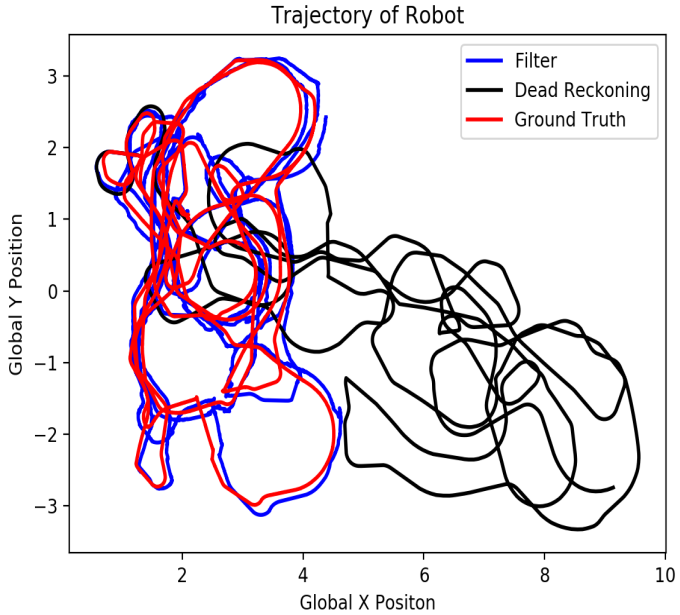

Figure 6: Trajectory of robot comparing ground truth, dead reckoning, and UKF

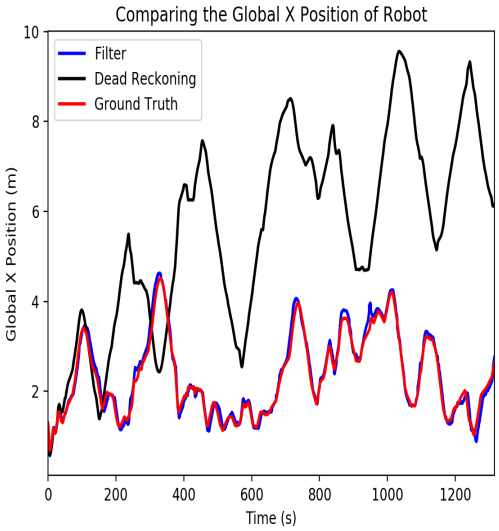

Figure 7: Global x position

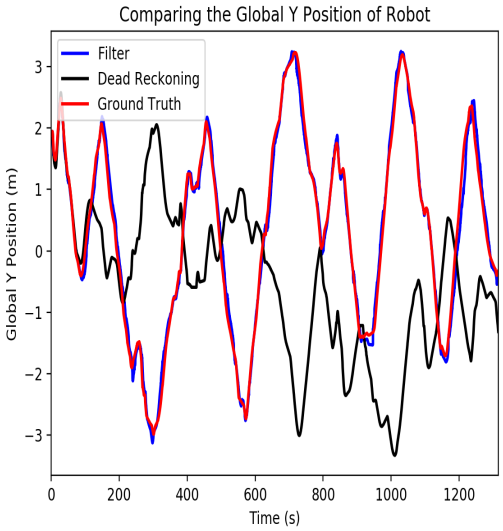

Figure 8: Global y position

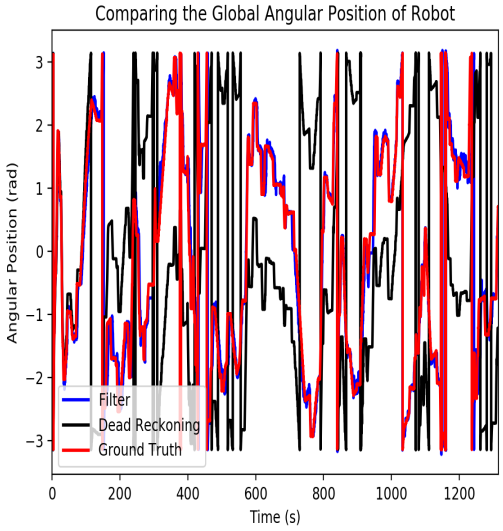

Figure 9: Global angular position

The trajectory in Fig.(6) represents the full path the robot. The x position of the robot versus time is shown in Fig.(7). The y position of the robot versus time is shown in Fig.(8). The angular position of the robot versus time is shown in Fig.(9).

Problem 9

In order to examine the affect the noise covariance matrices have on the filter the first 500 control commands and corresponding measurements were applied to the UKF.

The motion model noise covariance matrix was the same in all plots with corresponding values of ($\sigma_x =$

0.004, $\sigma_y = 0.004$, and $\sigma_\phi = 0.0085$). The standard deviations for x and y position correspond to 4mm and the standard deviation for angular position is 0.5deg. The values are realistic for the motion model because smaller values will not capture stochasticity in the environment.

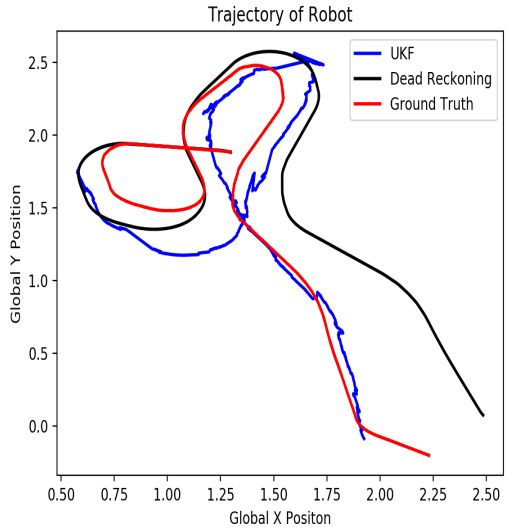

Figure 10: Smaller Q_t for UKF

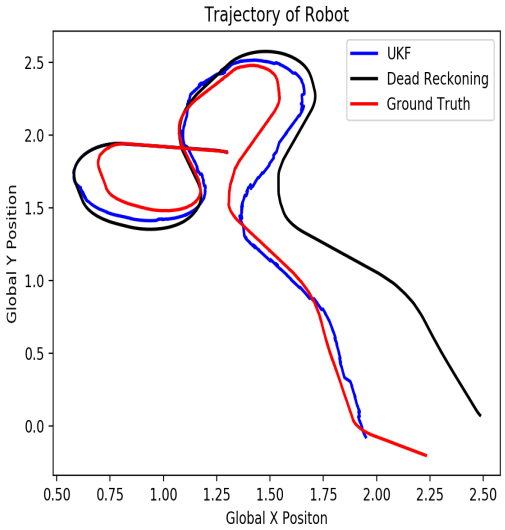

Figure 11: Larger Q_t for UKF

The standard deviations for range and bearing for Fig.(10) are ($\sigma_r = 0.001$ and $\sigma_\phi = 0.0017$). The standard deviations for range and bearing for Fig.(11) are ($\sigma_r = 0.002$ and $\sigma_\phi = 0.0085$). It is unlikely that the camera used in the experiment has a smaller uncertainty for range than 1mm and 0.5deg for bearing.

The filtered trajectory in Fig.(10) appears more jagged and diverges from the ground truth trajectory more than the filtered trajectory in Fig.(11). The small measurement noise covariance matrix places more trust on the incoming measurements than a slightly larger noise covariance matrix. This results in a trajectory that will diverge more from the ground truth.

The number of measurements observed by the robot were altered to examine the affect on the resulting trajectory. The first 500 controls and measurements were applied to the filter. In both figures bellow if the robot observed multiple landmarks at the same time only the first landmark was used in the filter. All experiments bellow use the noise parameters ($\sigma_x = 0.004$, $\sigma_y = 0.004$, and $\sigma_\phi = 0.0085$) and ($\sigma_r = 0.002$ and $\sigma_\phi = 0.0085$).

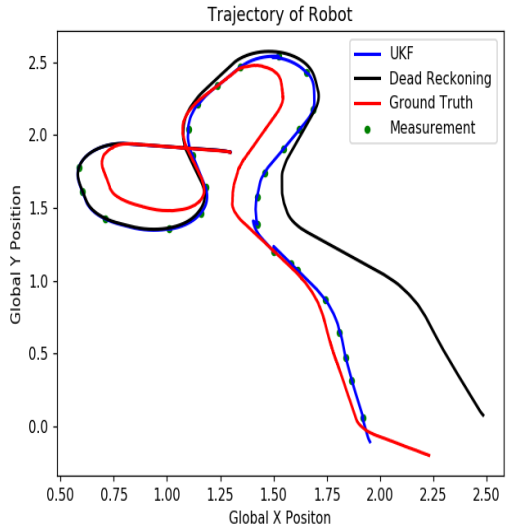

Figure 12: 27 landmarks observed in UKF

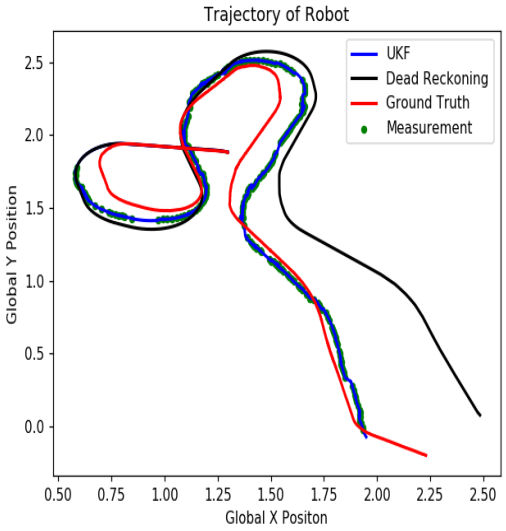

Figure 13: 270 landmarks observed in UKF

In Fig.(12) every 10th landmark observed was used and in Fig.(13) all landmarks were used to estimate the position. When measurements are skipped the filtered trajectory drifts towards the dead-reckoned path. This is expected because with more measurements the uncertainty in the robots position is reduced, this corresponds to a smaller covariance matrix.

Consider the case when the robot observes multiple landmarks at the same time. The control command and the first measurement are applied. However, the following measurements at the same timestamp are considered but the control step is ignored. These measurements enter the algorithm in Fig.(3) at line 6. The sigma points in line 7 are determined using the current mean and covariance.

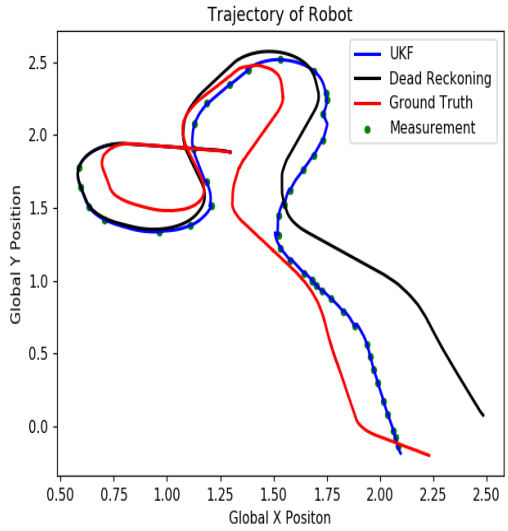

Figure 14: 45 landmarks observed in UKF

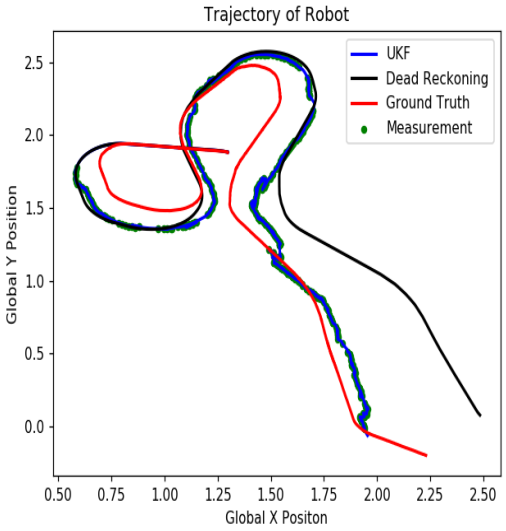

Figure 15: 450 landmarks observed in UKF

In Fig.(14) every 10th landmark is used all others are ignored. In Fig.(15) all landmarks including multiple observations at the same timestamp are used to estimate the pose. As expected in the case when all landmarks observed are considered the filtered trajectory more closely follows ground truth. The trajectory in Fig.(14) diverges more from ground truth than the trajectory in Fig.(15).

However, the filtered trajectory in Fig.(15) with 450 landmarks observed diverges more from ground truth than the filtered trajectory in Fig.(13) with 270 landmarks observed. Perhaps the covariance matrix becomes too small relative to the noise added by the measurement noise covariance matrix causing the robots trajectory to diverge from ground truth.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[3] J. Schwertfeger, "Tutorial on a probabilistic measurement model based on landmark range and bearing information," pp. 1–6, 2007.