Programming Assignment Report

231B Experiential Advanced Control

Shakshi Himmatramka, Jigisha Sampat

May 4, 2020

1 Introduction

The programming assignment requires the designing of a state estimator to track the position and heading of a bicycle as it moves. In doing so, the assignment allows us to utilize our learning from this course to analyze and identify what estimator structure would be most suited to a real-life problem. The goal is to implement the state estimator on a programming language (MATLAB or Python).

In this project, we compared two state estimation techniques, the Extended Kalman Filter and the Unscented Kalman Filter, both most suited for non-linear problems such as the one presented to us. We implemented both these techniques in MATLAB and then tested them for the 100 use cases provided to us. In this paper, we discuss how we modeled our system, the design choices and assumptions we made and our results for one case (Run 1) and our take-away from this assignment.

2 Modeling

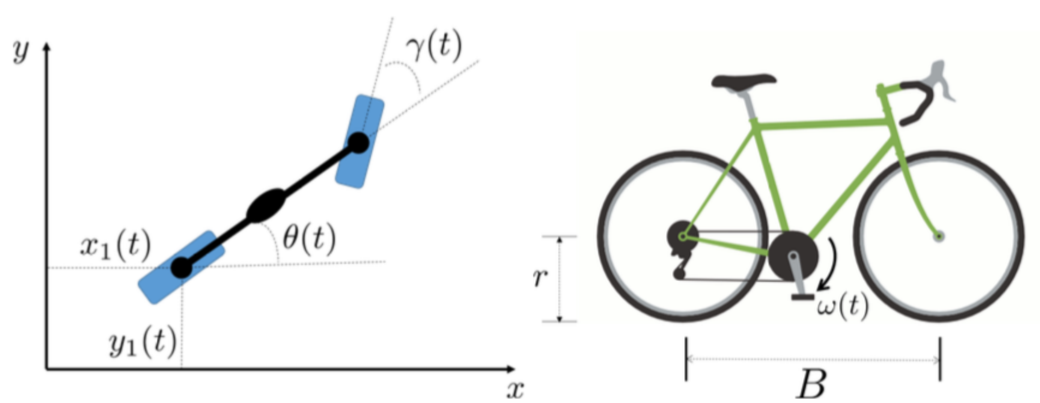

Figure 1: Bicycle Model

We modelled the system and measurement models individually. The system under consideration is a bicycle as shown in Fig[1] from the given problem statement. (x_1, y_1) represent the position of rear wheel, λ represent the front wheel steering angle, θ is the heading angle and ω is the pedaling speed. r represent wheel radius and B is the wheel base.

2.1 System Model

The continuous bicycle dynamics is given by:

$$\dot{x_1}(t) = v(t)cos(\theta(t)) \tag{1}$$

$$\dot{y_1}(t) = v(t)sin(\theta(t)) \tag{2}$$

$$\dot{\theta}(t) = \frac{v(t)}{B} sin(\lambda(t)) \tag{3}$$

where $v(t)$ is the bicycle linear velocity. As given in the problem statement, $\omega_r = 5 * \omega$, where ω_r is the rear wheel angular velocity.

Since,

$$v(t) = r\omega_r(t) \tag{4}$$

$$v(t) = r * 5 * \omega(t) \tag{5}$$

Further, we discretized the system using Euler discretization as given by:

$$\dot{x} = \frac{x(k+1) - x(k)}{d_t} \tag{6}$$

where x represent the states and d_t the sampling time.

Using 5 and 6 the continous system model is discretised as:

$$x_1(k+1) = x_1(k) + 5r\omega(k)cos(\theta(k))d_t \tag{7}$$

$$y_1(k+1) = y_1(k) + 5r\omega(k)sin(\theta(k))d_t \tag{8}$$

$$\theta(k+1) = \theta(k) + \frac{5r\omega(k)}{B} sin(\lambda(k))d_t \tag{9}$$

Here we assume that the steering angle $\lambda(t)$ and pedal speed $\omega(t)$ are both piece-wise constant and also known at all times.

2.2 Measurement Model

The measurement model encompasses the position measurements of the center of the bicycle that are captured at discrete time steps $t_k = 0.5$. This model can be described as below:

$$p(k) = \begin{bmatrix} x_1(t_k) + \frac{1}{2}B\cos\theta(t_k) \\ y_1(t_k) + \frac{1}{2}B\sin\theta(t_k) \end{bmatrix} \tag{10}$$

The biggest assumption that is made in the measurement model is that the measurements are unbiased and that the measurements are available at their discrete time steps even though in reality, the data does not necessarily come at exactly this rate.

2.3 Final Model

2.3.1 B and r treated as constant

In the first case, we modeled our system considering only the nominal values of r and B as constant values in our system model equations and derived the model equations.

$$q = \begin{bmatrix} x_1 & y_1 & theta \end{bmatrix} \tag{11}$$

$$u = \begin{bmatrix} \omega & \lambda \end{bmatrix} \tag{12}$$

where q represent system states and u system inputs.

System Model: $q(k+1) = f_{k-1}(q(k-1), v(k-1))$

$$\begin{bmatrix} x_1(k+1) \\ y_1(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) + 5r\omega(k)cos(\theta(k))d_t + v_1 \\ y_1(k) + 5r\omega(k)sin(\theta(k))d_t + v_2 \\ \theta(k) + \frac{5r\omega(k)}{B} sin(\lambda(t))d_t + v_3 \end{bmatrix} \tag{13}$$

where v_1, v_2, v_3 represent the process noise with mean zero and Co-variance matrix V.

Measurement Model: $z(k) = h_k(q(k), w(k))$

$$z(k) = \begin{bmatrix} x_1(t_k) + \frac{1}{2}B\cos\theta(t_k) + w_1 \\ y_1(t_k) + \frac{1}{2}B\sin\theta(t_k) + w_2 \end{bmatrix} \tag{14}$$

where w_1, w_2 represent the measurement noise with mean zero and Co-variance matrix W.

2.3.2 B and r uncertain

The measurement is corrupted with certain uncertainty which can be caused due to imperfect mechanical model of the bike. These imperfections can be attributed to mechanical tolerances, weight of the rider, different model of tires or varying tire pressure. The two main uncertainties considered in our model are the tire radius and the Baseline between the wheel. The nominal values and error bounds for these values are given below:

$$r = 0.425m \tag{15}$$

$$r_{err} = \pm 0.05 \tag{16}$$

$$B = 0.8m \tag{17}$$

$$B_{err} = \pm 0.1 \tag{18}$$

In case two, we considered our uncertainty values r and B as states with the nominal value being their initial mean and their error bound being the variance.

$$r_{max} = r + r_{err} * r = 0.44625 \tag{19}$$

$$r_{min} = r - r_{err} * r = 0.43500 \tag{20}$$

$$P_r = \frac{1}{12}(r_{max} - r_{min})^2 = 1.5 * 10^{-4} \tag{21}$$

$$B_{max} = B + B_{err} * B = 0.81 \tag{22}$$

$$B_{min} = B - B_{err} * B = 0.72 \tag{23}$$

$$P_B = \frac{1}{12}(B_{max} - B_{min})^2 = 2.13 * 10^{-3} \tag{24}$$

Our system model now goes from having 3 states to having 5 states, which changes our system dynamics while the measurement dynamics remains same.

System Model: $q(k+1) = f_{k-1}(q(k-1), v(k-1))$

$$\begin{bmatrix} x_1(k+1) \\ y_1(k+1) \\ \theta(k+1) \\ r(k+1) \\ B(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) + 5r(k)\omega(k)cos(\theta(k))d_t + v_1 \\ y_1(k) + 5r(k)\omega(k)sin(\theta(k))d_t + v_2 \\ \theta(k) + \frac{5r(k)\omega(k)}{B(k)}sin(\lambda(t))d_t + v_3 \\ r(k) + v_4 \\ B(k) + v_5 \end{bmatrix} \tag{25}$$

where v_1, v_2, v_3, v_4, v_5 represent the process noise with mean zero and Co-variance matrix V.

Measurement Model: $z(k) = h_k(q(k), w(k))$

$$z(k) = \begin{bmatrix} x_1(t_k) + \frac{1}{2}B(k)\cos\theta(t_k) + w_1 \\ y_1(t_k) + \frac{1}{2}B(k)\sin\theta(t_k) + w_2 \end{bmatrix} \tag{26}$$

3 Design Decisions And Justification

To develop the estimator, we had to make design decision regarding the following term:

- States Initial Condition $q(0)$
- States Initial Co-variance Matrix $P(0)$
- Measurement/Sensor Noise Co-variance Matrix W
- Process Noise Co-variance Matrix V

1. States Initial Condition $q(0)$: As given in the problem statement, it has been given the bicycle initial position can be assumed to be at the origin heading in North-East Direction. Therefore,

$$\begin{align} x_1(0) &= 0 \tag{27}\\ y_1(0) &= 0 \tag{28}\\ \theta(0) &= \pi/4 \tag{29}\\ r(0) &= r \tag{30}\\ B(0) &= B \tag{31}\\ & \tag{32} \end{align}$$

   where r and B are nominal values of wheel radius and wheel base.

2. States Initial Co-variance Matrix $P(0)$: The initial variance for the states $r(k)$ and $B(k)$ is determined using their error bound. The initial variance for the position state and heading state is considered as for maximum ignorance. Therefore,

$$P(0) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & P_r & 0 \\ 0 & 0 & 0 & 0 & P_B \end{bmatrix} \tag{33}$$

3. Measurement/Sensor Noise Co-variance Matrix W: To determine the Measurement noise co-variance matrix, we assumed we perfectly know the system dynamics and used the calibration file (Experiment zero) to calculate W.

   The following summarize the process:

   - Loaded Experiment 0 data
   - Clean up the measurement data
   - Since the measurements were taken for a stationary bicycle, the values were distributed with the mean as true bicycle position due to noise. We then subtracted the data with the true position to get zero-mean data.
   - The Co-variance of this zero-mean data which represent sensor noise was used as W.

$$W = \begin{bmatrix} 1.0893 & 1.5333 \\ 1.5333 & 2.9880 \end{bmatrix} \tag{34}$$

4. Process Noise Co-variance Matrix V: To design the process noise co-variance matrix, we determined the relations among different states. Based on this we developed a basic outline of the matrix:

$$V = \begin{bmatrix} v_{11} & 0 & v_{13} & v_{14} & 0 \\ 0 & v_{22} & v_{23} & v_{24} & 0 \\ 0 & 0 & v_{33} & v_{34} & v_{35} \\ 0 & 0 & 0 & v_{44} & 0 \\ 0 & 0 & 0 & 0 & v_{55} \end{bmatrix} \tag{35}$$

   The varibles in the matrix were determined by tuning it to get minimum estimation error.

3.1 Choice of Estimator

Since our system model is non-linear, the most obvious choice of estimators was to use Extended Kalman Filter or Unscented Kalman Filter. Particle Filter could also be used but was deemed too computationally heavy for our purposes. Extended Kalman filters work best for "mildly" non-linear systems that are differentiable. Unscented Kalman Filters on the other hand are better at dealing with non-linearity than Extended Kalman Filters. For the sake of this project, we tried to implement both these methods and have described their implementation in detail in the next few sections.

4 Extended Kalman Filter

The EKF is derived by linearizing the nonlinear system equations about the latest state estimate and then applying the (standard) Kalman Filter prior and measurement update equations to the linearized equations. In the case of the Extended Kalman filter too, we considered the system model with r and B constant as well as with them as uncertainty.

4.1 B and r treated as constant

In the case when B and r are treated constant, we have 3 states as described by Equation 13 in the Final Model. To implement our Extended Kalman Filter, we need to linearize this final system at each time step which gives us the following matrices for the system and measurement models:

$$A = \begin{bmatrix} 1 & 0 & -5\omega r(\sin(\theta(t_k))d_t \\ 0 & 1 & 5\omega r\cos(\theta(t_k))d_t \\ 0 & 0 & 1 \end{bmatrix} \tag{36}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{37}$$

Similarly, linearizing the measurement model gave us the following matrices:

$$H = \begin{bmatrix} 1 & 0 & -\frac{1}{2}B\sin(\theta(t_k)) \\ 0 & 1 & -\frac{1}{2}B\cos(\theta(t_k)) \end{bmatrix} \tag{38}$$

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{39}$$

4.2 B and r uncertain

In the case when B and r are treated as variable states, we have 5 states as described by Equation 25 in the Final Model. To implement our Extended Kalman Filter, we need to linearize this final system at each time step which gives us the following matrices for the system and measurement models:

$$A = \begin{bmatrix} 1 & 0 & -5(\sin(\theta(t_k))d_t & 5d_t\omega\cos(\theta(t_k)) & 0 \\ 0 & 1 & 5\cos(\theta(t_k))d_t & 5d_t\omega\cos(\theta(t_k)) & 0 \\ 0 & 0 & 1 & (5dt\omega\tan(\lambda)/B & -(5d_t r\omega\tan(\lambda)/B^2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{41}$$

Similarly, linearizing the measurement model gave us the following matrices:

$$H = \begin{bmatrix} 1 & 0 & -\frac{1}{2}B\sin(\theta(t_k) & 0 & \frac{1}{2}\cos(\theta(t_k)) \\ 0 & 1 & -\frac{1}{2}B\cos(\theta(t_k) & 0 & \frac{1}{2}\sin(\theta(t_k)) \end{bmatrix} \tag{42}$$

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{43}$$

4.3 Final Algorithm

Once we had linearized our dynamics, at each time step, the linearized dynamics were updated with the current state values. We calculated the prior model without considering measurement values and then updated our model based on the gains that were calculated from the measurement values. These models are described below:

PREDICTION STEP:

$$\hat{x_p} = f_{k-1}(q(k-1), v(k-1)) \tag{44}$$

$$P_p(k) = A(k-1)P_m(k-1)A^T(k-1) + L(k-1)V(k-1)L^T(k-1) \tag{45}$$

MEASUREMENT UPDATE STEP:

$$K(k) = P_p H^T(H(k)P_p H^T + M(k)W(k)M(k)^T)^{-1} \tag{46}$$

$$\hat{x_m}(k) = \hat{x_p} + K(k)(z(k) - h_k(\hat{x_p}, 0) \tag{47}$$

$$P_m(k) = (I - K(k)H(k))P_p(k) \tag{48}$$

4.4 RESULTS

The model was run for 1000 iterations for about a 100 different cases. The results for case 1 (Experiment 1 data file) has been discussed in the report.

4.4.1 B and r Constant

The error from the true value for case 1 is:

$$pos_x = -1.0042m \tag{49}$$
$$pos_y = -0.41174m \tag{50}$$
$$angle = 0.57755rad \tag{51}$$
$$\tag{52}$$

It was observed that at certain time-steps in Fig[2] the estimated x position and y position did not perfectly track the measurements hence as seen in Fig[3] the estimate was not closely following the trajectory. Also, the final error is large than the reference error data provided. But, there is a smoother change in the input heading angle.

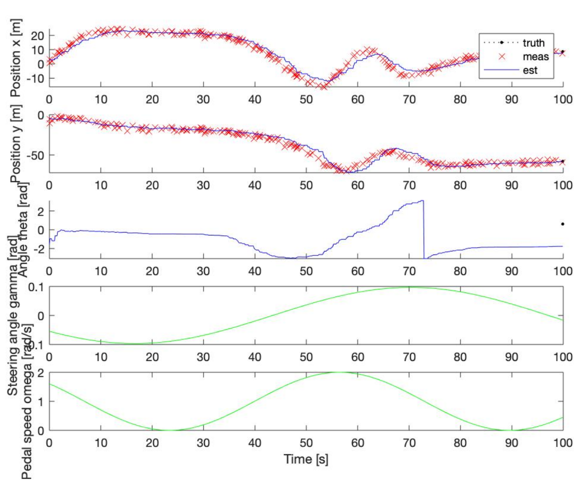

Figure 2: EKF (r and B constant)

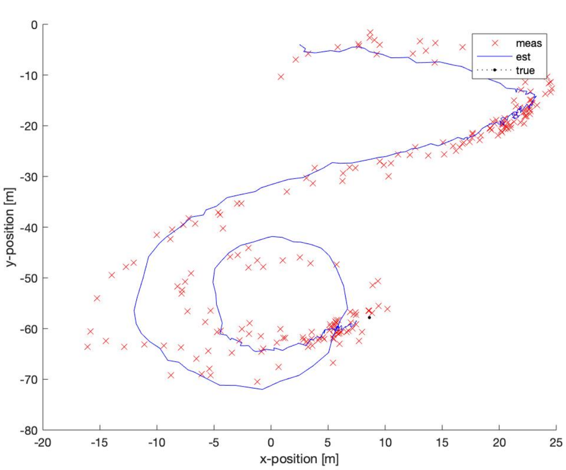

Figure 3: EKF (r and B constant)

4.4.2 B and r Uncertain

The error from the true value for case 1 is:

$$pos_x = -0.57797m \tag{53}$$
$$pos_y = -0.12314m \tag{54}$$
$$angle = 0.0077984rad \tag{55}$$
$$\tag{56}$$

It was observed in Fig[4] that the estimated x position and y position perfectly track the measurements and as seen in Fig[5] the estimate followed the measured trajectory and the trajectory estimated is smooth. There is a smoother change in the input heading angle. The final error is quite close to the reference error data provided.

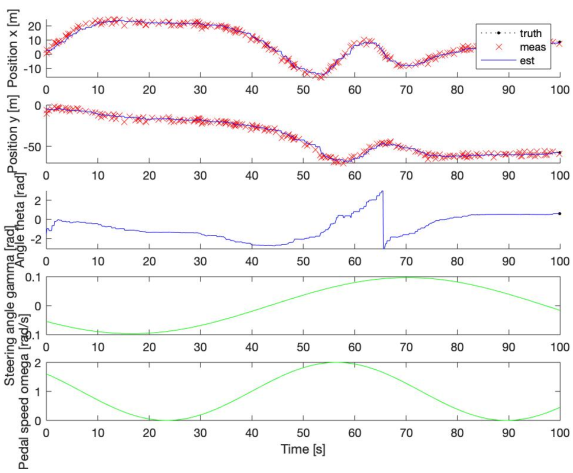

Figure 4: EKF

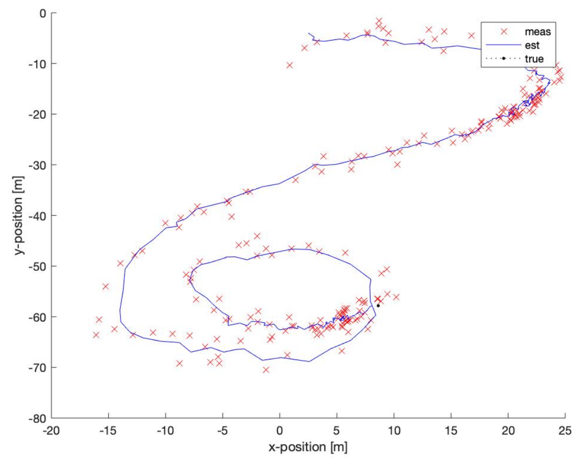

Figure 5: EKF

5 Unscented Kalman Filter

The Unscented Kalman Filter uses a specially selected set of points to approximate the random variables. These points are then transformed through the (full) nonlinear functions, and from the transformed points the relevant statistics are recovered.

5.1 Algorithm

The sigma points for the Unscented Kalman Filter are calculated using the mean and variance of the states. For the two cases of B and r constant vs uncertain, the difference was the size of the covariance matrices and mean matrices. Otherwise, the steps to solve the UKF algorithm are the same and can be described by the following equations:

SIGMA POINTS:

$$s_{x,i} = q_m + \sqrt{nP_m} \tag{57}$$
$$s_{x,n} = q_m - \sqrt{nP_m} \tag{58}$$

$$s_{y,i} = f(s_{x,i})) \tag{59}$$

PREDICTION STEP:

$$\hat{x_p}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} s_{y,i} \tag{60}$$

$$\hat{P}_p(k) = \sum_{i=0}^{2n-1} \frac{1}{2n}(s_{y,i} - x_p\hat{(k)})(s_{y,i} - \hat{x_p}(k))^T + V_m \tag{61}$$

MEASUREMENT UPDATE STEP:

$$s_{z(k),i} = h(s_{x_p(k),i})) \tag{62}$$

$$\hat{z}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} s_{z,i} \tag{63}$$

COVARIANCE:

$$P_{zz}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} (s_{z(k),i} - \hat{z}(k))(s_{z(k),i} - \hat{(}k))^T + W_m \tag{64}$$

$$P_{xz}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} (s_{y,i} - x_p\hat{(}k))(s_{z(k),i} - \hat{z}(k))^T \tag{65}$$

KALMAN FILTER GAIN:

$$K(k) = P_{xz}(k) P_{zz}(k)^{-1} \tag{66}$$

$$\hat{x_m}(k) = \hat{x_p} + K(k)(z(k) - \hat{z}(k)) \tag{67}$$

$$P_m(k) = P_p(k) - K(k) P_{zz}(k) K(k)^T \tag{68}$$

5.2 RESULTS

The model was run for 1000 iterations for about a 100 different cases. The results for case 1 are shown below.

5.2.1 B and r Constant

The error from the true value for case 1 is:

$$pos_x = -1.1397m \tag{69}$$
$$pos_y = -0.79447m \tag{70}$$
$$angle = -2.3714rad \tag{71}$$
$$\tag{72}$$

It was observed in Fig[6] that the estimated x position and y position perfectly track the measurements hence as seen in Fig[7] the estimate quite closely following the trajectory. But, the resulting trajectory was not smooth and the final error is quite larger than the reference error data provided.

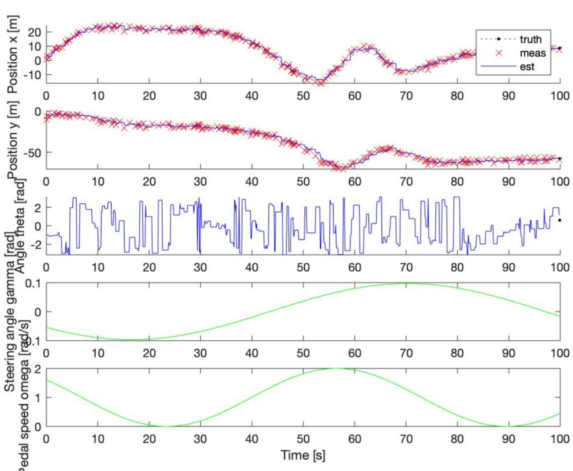

Figure 6: UKF (r and B constant)

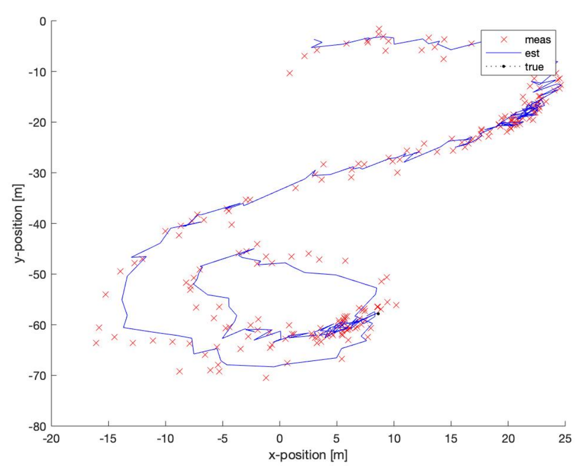

Figure 7: UKF (r and B constant)

5.2.2 B and r Uncertain

The error from the true value for case 1 is:

$$pos_x = -0.5668m \tag{73}$$
$$pos_y = -0.6009m \tag{74}$$
$$angle = -0.24443rad \tag{75}$$
$$\tag{76}$$

It was observed in Fig[8] that the estimated x position and y position perfectly track the measurements hence as seen in Fig[9] the estimate quite closely followed the trajectory. But, the trajectory is rough and angle is impulsively changing. The final error is closer to the reference error data provided.

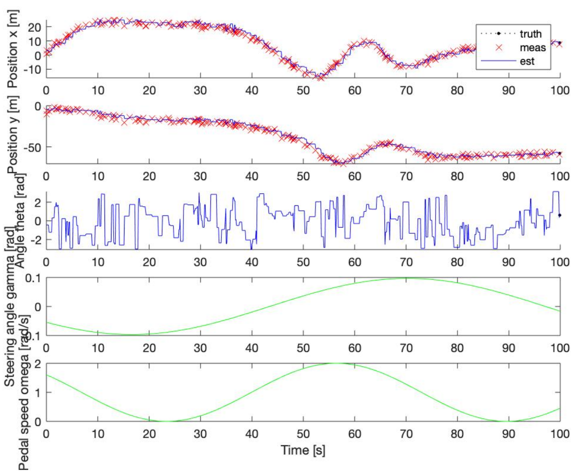

Figure 8: UKF

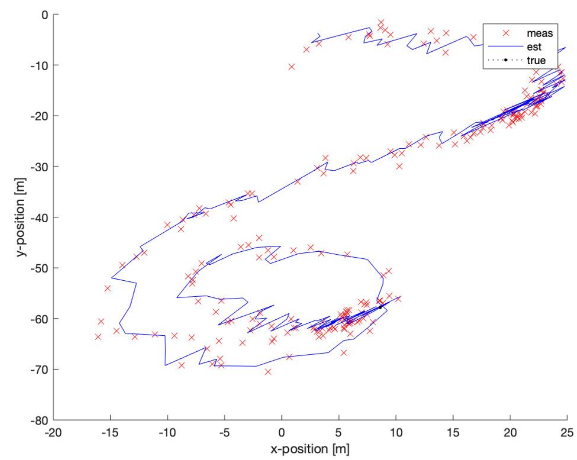

Figure 9: UKF

6 Evaluation and Discussion

Although the Unscented Kalman Filter is supposed to give better results for non-linear problems than the Extended Kalman Filter, in our case, we saw that Extended Kalman Filter showed better results.

Following summarize the evaluation and discussions

- It was very apparent that considering r and B as a variable state played a huge role in the accuracy of our result. As can be seen in the UKF as well as EKF case, the estimates are much closer to the real measurements, and the errors are much smaller when the uncertainty of r and B are accounted for.
- EKF resulted in a smoother change in the input and hence a smoother estimated trajectory. While UKF had impulsive changes in the heading angle input making the estimated trajectory rough.
- UKF was more sensitive to changes in measurement and hence resulted in good tracking of measurements. On the other hand, EKF did a fairly good job tracking the measurements but was less sensitive (hence smoother) than UKF
- Tuning of process noise Co-variance played an important role in determining the final estimated error. The co-variance matrix for EKF was highly tunable which provided us with the flexibility to reduce the error. In case of UKF, majority values resulted in singularity of co-variance matrix which limited the values for which the error can be tuned. Therefore, we were able to get smaller errors with EKF as compared to UKF.

The table given below summarize the error for each of the implemented errors.

Characteristic	Reference Error	EKF (r and B Constant)	EKF (r and B Uncertain)	UKF (r and B Constant)	UKF (r and B Uncertain)
x position	0.246	-1.0042	-0.57797	-1.1397	-0.5668
y position	0.304	-0.41174	-0.12314	-0.79447	-0.6009
heading angle	0.066	0.57755	0.0077984	-2.3714	-0.24443

It is clearly evident that Extended Kalman Filter considering the r and B uncertainity gave the lowest error. The error of the position y and heading angle is lower than the given reference values while x position is slightly larger than the reference error.

Similar behavior was observed for other data sets and runs.

7 Improvements

- There could be a better way to identify the process noise co-variance matrix. Right now, this matrix was tuned to improve the final result but that might not be the most realistic or ideal solution.
- Since in a lot of cases in the UKF runs, the co-variance matrix was not positive definite, the singularity resulted in restrictions to values the co-variance matrix could take. Finding a way to circumvent this singularity would allow to tune the control better and hence allow to harness the full potential of the Unscented Kalman Filter.