Programming Project of ME C231B

Xu Shen
Dept. of Mechincal Engineering
UC Berkeley
Berkeley, CA, USA
xu_shen@berkeley.edu

Mingguang Zhou
Dept. of Mechincal Engineering
UC Berkeley
Berkeley, CA, USA
mingguang_zhou@berkeley.edu

***Abstract*—This document is a report for the programming project of ME C231B in Spring, 2020. The project comes with the problem of estimating the position for a riding bicycle. To solve the problem, three different non-linear estimator algorithms (the Extended Kalman Filter, the Unscented Kalman Filter, and the Particle Filter) have been implemented specifically to the given bicycle system. With given measurement data, the performance of estimator algorithms has been tuned and compared. In result, the Particle Filter has been validated to provide the most satisfactory estimation for the problem.**

I. Problem Statement

A. Provided System

A simplified bicycle model is given and illustrated in Fig. 1. The position of the rear wheel in the x-y 2D plane (x_1, y_1)

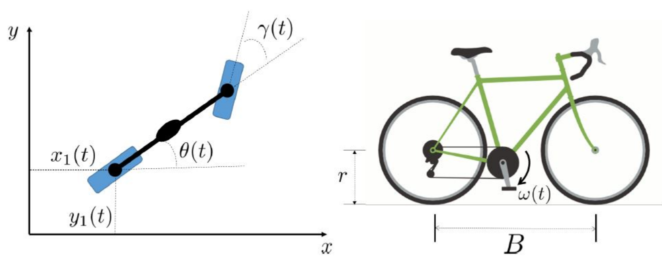

Fig. 1. Simplified bicycle system

shall be estimated with the GPS measurement for the bicycle ride. Here $\theta(t)$ is the heading, $\gamma(t)$ is the steering angle relative to the bicycle body, r is the wheel radius, and B is the wheel baseline. The system dynamics, inputs, and the discrete time form are further described below:

1) Bicycle Dynamics in Continuous Time: The dynamics for the bicycle system are given as below:

$$\dot{x_1}(t) = v(t)\cos(\theta(t)) \tag{1}$$
$$\dot{y_1}(t) = v(t)\sin(\theta(t)) \tag{2}$$
$$\dot{\theta}(t) = \frac{v(t)}{B}\tan(\gamma(t)) \tag{3}$$

where $v(t)$ is the linear velocity of the bicycle, for which the angular velocity of the rear wheel is always 5 times the pedaling speed due to the gear ratio.

2) System Input: The system inputs are summarized as:

$$u(t) = \begin{bmatrix}\gamma(t)\\ \omega(t)\end{bmatrix} \tag{4}$$

for which $\omega(t)$ is the pedal speed, and $\gamma(t)$ is the steering angle.

3) Time Discretization: Due to the discrete time nature of computer processors, the given dynamics system and inputs are combined and transformed into the discrete time form as below:

$$x_1(k) = x_1(k-1) + v(k-1)\cos(\theta(k-1))\Delta t \tag{5}$$
$$y_1(k) = y_1(k-1) + v(k-1)\sin(\theta(k-1))\Delta t \tag{6}$$
$$\theta(k) = \theta(k-1) + \frac{v(k-1)}{B}\tan(\gamma(k-1))\Delta t \tag{7}$$
$$v(k) = 5r\omega(k) \tag{8}$$
$$u(k) = \begin{bmatrix}\gamma(k)\\ \omega(k)\end{bmatrix} \tag{9}$$

B. Known Signals and Provided Data

1) The position measurement equations are given as below:

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

t_k is the discrete time, each measurement period is about 0.5 second.

2) The manufacturer's nominal values for the bicycle are $\mu_r = 0.425[\text{m}], \mu_B = 0.8[\text{m}]$.
3) Initial position and heading values are $\mu_{x_1(0)} = 0, \mu_{y_1(0)} = 0, \mu_{\theta(0)} = \frac{\pi}{4}$.
4) 100 different bicycle rides data are each given as a comma separated text file, including current time, steering angle, pedal speed, (x, y) measurement of bicycle center, and true pose (x, y, θ) at each time step with $\Delta t = 0.1[\text{s}]$. The measurement and true pose might not be available at all time steps and they will be "NaN" when no data.

C. Uncertainty

The measurements contain uncertainty due to electrical noise in the sensor, timing imprecision, and atmospheric disturbances that warp the path of the GPS signals. The dynamic system is also not deterministic due to the imperfect knowledge of system model and input, wheel slip, and the variation of parameters. The baseline B and the tire radius r are also corrupted with uncertainty. The uncertainty is represented as $\pm 10\%$ uniformly distributed tolerance around nominal baseline μ_B, and $\pm 5\%$ uniformly distributed tolerance around nominal tire radius μ_r.

II. Modelling for Estimation

A. System Dynamics

According to the problem statement, there are 5 uncertain state variables that we are interested to estimate: x_1, y_1, θ, B, r. We use an augmented state representation:

$$x(k) = [x_1(k), y_1(k), \theta(k), B(k), r(k)]^\top$$

and the dynamics is therefore extended to

$$x(k) = q_{k-1}(x(k-1), u(k-1), v(k-1)) \tag{11}$$

$$= \begin{bmatrix} x_1(k-1) + 5r(k-1)\omega(k-1)\cos(\theta(k-1))\Delta t \\ y_1(k-1) + 5r(k-1)\omega(k-1)\sin(\theta(k-1))\Delta t \\ \theta(k-1) + \frac{5r(k-1)\omega(k-1)}{B(k-1)}\tan(\gamma(k-1))\Delta t \\ B(k-1) \\ r(k-1) \end{bmatrix} + v(k-1) \tag{12}$$

where the underlying dynamics of B and r is just constant, $v(k) \in \mathbb{R}^5$ is the simplified additive process noise.

The measurement is

$$z(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \end{bmatrix} = h_k(x(k), w(k)) \tag{13}$$

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

where $w(k) \in \mathbb{R}^2$ is the simplified additive measurement noise.

B. Initial Belief of the States

According to the problem statement, we assume that the initial system state is normally distributed:

$$x(0) \sim (\mu_{x(0)}, P(0)) \tag{15}$$

where

$$\mu_{x(0)} = [\mu_{x_1(0)}, \mu_{y_1(0)}, \mu_{\theta(0)}, \mu_B, \mu_r]^\top$$
$$P(0) = \mathrm{diag}([\sigma^2_{x_1(0)}, \sigma^2_{y_1(0)}, \sigma^2_{\theta(0)}, \sigma^2_{B(0)}, \sigma^2_{R(0)}])$$

C. Parameterization of Noise

We assume that both the process and measurement noise are normally distributed with zero mean:

$$v(k) \sim (0, V), V = \mathrm{diag}([\sigma^2_{\Delta x_1}, \sigma^2_{\Delta y_1}, \sigma^2_{\Delta\theta}, \sigma^2_{\Delta B}, \sigma^2_{\Delta R}]) \tag{16}$$

$$w(k) \sim (\mathbb{E}[w], W), W = \mathrm{diag}([\sigma^2_{\Delta z_x}, \sigma^2_{\Delta z_y}]) \tag{17}$$

and $\{v(\cdot)\}$, $\{w(\cdot)\}$ and $\{x(\cdot)\}$ are mutually independent.

III. Estimator Design

Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF) have been developed to solve the position estimation for the system. Through each iteration of each algorithm, a posteriori update for the state $\hat{x}_m$ at time k will be obtained, which contains the estimation for the bicycle position at each time k. The detailed description for each algorithm is demonstrated below:

A. EKF

1) Initialization:

$$\hat{x}_m(0) = \mu_{x(0)} \tag{18}$$
$$P_m(0) = P(0) \tag{19}$$

2) Process Update:

$$\hat{x}_p(k) = q_{k-1}(\hat{x}_m(k-1), u(k-1), 0) \tag{20}$$
$$P_p(k) = A(k-1)P_m(k-1)A^T(k-1) + V(k-1) \tag{21}$$

where $A(k-1)$ is computed as Eq.(23).

$$A(k-1) = \frac{\delta q_{k-1}(\hat{x}_m(k-1), u(k-1), 0)}{\delta x} \tag{22}$$

$$= \begin{bmatrix} 1 & 0 & -vs & 0 & 5c\omega \\ 0 & 1 & vc & 0 & 5s\omega \\ 0 & 0 & 1 & -\frac{v}{B^2}t & \frac{5}{B}t\omega \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

where v, B, ω, s, c, t are short notation for $v(k-1)$, $B(k-1)$, $\omega(k-1)$, $\sin(\theta(k-1))\Delta t$, $\cos(\theta(k-1))\Delta t$, and $\tan(\gamma(k-1))\Delta t$ respectively due to the space limitation.

3) Measurement Update:

$$K(k) = P_p(k)H^T(k)(H(k)P_p(k)H^T(k) + W(k))^{-1} \tag{24}$$
$$\hat{x}_m(k) = \hat{x}_p(k) + K(k)(\bar{z}(k) - h_k(\hat{x}_p(k), \mathbb{E}[w])) \tag{25}$$
$$P_m(k) = (I - K(k)H(k))P_p(K) \tag{26}$$

where $\bar{z}(k)$ is the actual measurement reading at k, and $H(k)$ is computed as Eq.(28).

$$H(k) = \frac{\delta h_k(\hat{x}_p(k), \mathbb{E}[w])}{\delta x} \tag{27}$$

$$= \begin{bmatrix} 1 & 0 & -\frac{B(k)}{2}\sin(\theta(k)) & \frac{\cos(\theta(k))}{2} & 0 \\ 0 & 1 & \frac{B(k)}{2}\cos(\theta(k)) & \frac{\sin(\theta(k))}{2} & 0 \end{bmatrix} \tag{28}$$

B. UKF

1) Initialization:

$$\hat{x}_m(0) = \mu_{x(0)} \tag{29}$$
$$P_m(0) = P(0) \tag{30}$$

2) Process Update: Generate $2n$ sigma-points:

$$s_{x_m(k-1),i} = \hat{x}_m(k-1) + (\sqrt{nP_m(k-1)})_i \tag{31}$$
$$s_{x_m(k-1),n+i} = \hat{x}_m(k-1) - (\sqrt{nP_m(k-1)})_i \tag{32}$$

for $n = \dim(x) = 5$ and $i \in \{0, 1, ..., n-1\}$.

Then compute the prior sigma-points:

$$s_{x_p(k),n+i} = q_{k-1}(s_{x_m(k-1),i}, u(k-1), 0) \tag{33}$$

for $i \in \{0, 1, ..., 2n-1\}$.

Compute the prior statistics:

$$\hat{x}_p(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} s_{x_p(k),i} \tag{34}$$

$$P_p(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} (s_{x_p(k),i} - \hat{x}_p(k))(s_{x_p(k),i} - \hat{x}_p(k))^T + V(k-1) \tag{35}$$

3) Measurement Update: Compute sigma-points for measurements:

$$s_{z(k),i} = h_k(s_{x_p(k),i}, \mathbb{E}[w]) \tag{36}$$

Then compute the expected measurement $\hat{z}(k)$, covariance $P_{zz}(K)$ and cross variance $P_{xz}(K)$:

$$\hat{z}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} s_{z(k),i} \tag{37}$$

$$P_{zz}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} (s_{z(k),i} - \hat{z}(k))(s_{z(k),i} - \hat{z}(k))^T + W(k-1) \tag{38}$$

$$P_{xz}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n} (s_{x_p(k),i} - \hat{x}_p(k))(s_{z(k),i} - \hat{z}(k))^T \tag{39}$$

Lastly apply the kalman filter gain:

$$K(k) = P_{xz}(K) P_{zz}(K)^{-1} \tag{40}$$

$$\hat{x}_m(k) = \hat{x}_p(k) + K(k)(\bar{z}(k) - \hat{z}(k)) \tag{41}$$

$$P_m(k) = P_p(k) - K(k) P_{zz}(k) K(k)^T \tag{42}$$

where $\bar{z}(k)$ is the actual measurement reading at k.

C. PF

1) Initialization: Draw N samples $\{x_m^n(0)\}$

$$\{x_m^n(0)\} \sim \mathcal{N}(\mu_{x(0)}, P(0)) \tag{43}$$

2) Process Update: Simulate all particles $\{x_m^n(k-1)\}$ through the process equation:

$$x_p^n(k) = q_{k-1}(x_m^n(k-1), u(k-1), v^n(k-1)) \tag{44}$$

where we also have samples $\{v^n(k-1)\} \sim \mathcal{N}(0, V)$.

3) Measurement Update: Scale each particle by the measurement likelihood

$$\beta_n = \alpha f_{z(k)|x(k)} \left(\bar{z}(k) | x_p^n(k)\right) \tag{45}$$

$$= \alpha f_w \left(\bar{z}(k) - h_k(x_p^n(k), \mathbb{E}[w])\right) \tag{46}$$

here α is the normalization constant chosen such that $\sum_{n=1}^N \beta_n = 1$.

Resample N new particles $x_m^n(k)$ based on the distribution of $\{\beta_n\}$.

After resampling, apply roughening as

$$x_m^n(k) \leftarrow x_m^n(k) + \Delta x^n(k) \tag{47}$$

where $\Delta x^n(k)$ is drawn from a zero-mean, finite-variance distribution. The standard deviation of the i-th element of $\Delta x^n(k)$ is $\sigma_i = K_r E_i N^{-\frac{1}{d}}$.

D. Algorithm for Asynchronous Measurement

Due to the fact that the measurement is not always available at each time step, the vanilla state estimation is slightly modified as Algorithm. 1.

Algorithm 1: Async Estimation

Input: est $\in$ {EKF, UKF, PF}, $u(k)$, and $\bar{z}(k)$
Output: Posterior Estimation $x_m(k)$ and $P_m(k)$

```
1 est.Initialization();
2 for Each time step k do
3 |  x_p(k), P_p(k) =
   |    est.Process_Update(x_m(k), u(k), P_m(k));
4 |  if z̄(k) ≠ NaN then
5 |  |  x_m(k), P_m(k) =
   |  |    est.Measurement_Update(x_p(k), z̄(k), P_p(k));
6 |  else
7 |  |  x_m(k), P_m(k) = x_p(k), P_p(k);
8 end
```

The most important change is line 7, which means that if a measurement is not available, the "best" estimation result we can possibly acquire at the current time step is the updated prior with system dynamics.

IV. Experiments

A. Parameter Identification and Tuning

1) Tuning of Initial Belief: The uncertainty of r and B is treated as normal distribution with the same expected values $\mu_r = 0.425[m], \mu_B = 0.8[m]$ and the variance:

$$\sigma^2_{B(0)} = \frac{2 \times 10\% \mu_B}{12} = 0.0021 \tag{48}$$

$$\sigma^2_{R(0)} = \frac{2 \times 5\% \mu_r}{12} = 1.5 \times 10^{-4} \tag{49}$$

We tried the following values for initial variance of $x_1(0), y_1(0)$ and $\theta(0)$):

$$\sigma^2_{x(0)} = \sigma^2_{y(0)} \in \{0.005, 0.01, 0.05, 0.1, 0.5\} \tag{50}$$

$$\sigma^2_{\theta(0)} \in \{0.001\pi, 0.005\pi, 0.01\pi, 0.05\pi\} \tag{51}$$

For PF specific, the parameter K_r for roughening is tuned from

$$K_r \in \{1 \times 10^{-6}, 1 \times 10^{-4}, 0.001, 0.01, 0.1\} \tag{52}$$

2) Tuning of Process Noise: We tried the following values for the variance of v:

$$\sigma^2_{\Delta x_1} = \sigma^2_{\Delta y_1} \in \{0.001, 0.005, 0.01, 0.05, 0.1\} \tag{53}$$

$$\sigma^2_{\Delta\theta} \in \{0.001\pi, 0.005\pi, 0.01\pi\} \tag{54}$$

3) Measurement Noise Identification: We use the Run #0 to identify the distribution of measurement noise. The GPS measurement of (x, y) when bicycle is static at $(0, 0)$ is visualized in Fig. 2. Therefore, a multivariate Gaussian is a natural fit with

$$\mathbb{E}[w] = \begin{bmatrix} 0.188 \\ 0.396 \end{bmatrix}, \mathrm{Var}[w] = W = \begin{bmatrix} 1.09 & 1.53 \\ 1.53 & 2.98 \end{bmatrix} \tag{55}$$

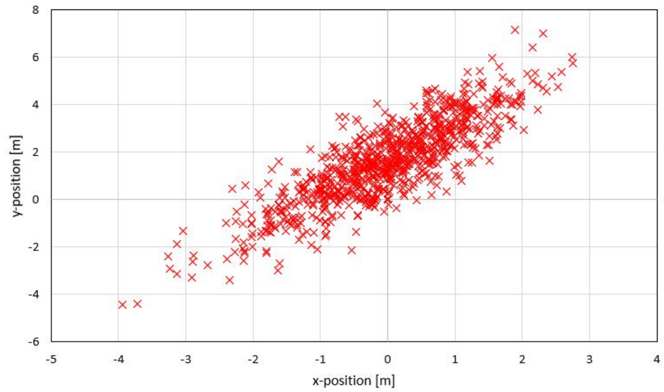

Fig. 2. Position measurements data for calibration run

B. Results from all Estimators and Estimator Selection

From Part A, as different combinations of $P(0)$ and V have been run with each estimator algorithm for the first 5 given runs of the bicycle ride, the best results of each algorithm by the sum of absolute errors are summarised in Tables I (EKF), II (UKF), and III (PF).

TABLE I
BEST EKF RESULTS FOR THE 1-5 RUNS

Run #	abs(x) [m]	abs(y) [m]	abs(θ) [rad]
1	0.576	0.308	0.287
2	0.024	0.197	0.161
3	0.216	0.071	0.092
4	0.744	0.871	0.101
5	0.666	1.213	0.224
Average	0.445	0.532	0.173

TABLE II
BEST UKF RESULTS FOR THE 1-5 RUNS

Run #	abs(x) [m]	abs(y) [m]	abs(θ) [rad]
1	0.422	0.646	0.038
2	0.163	0.026	0.003
3	0.098	0.263	0.233
4	0.921	0.753	0.113
5	0.666	1.213	0.224
Average	0.394	0.354	0.110

TABLE III
BEST PF RESULTS FOR THE 1-5 RUNS

Run #	abs(x) [m]	abs(y) [m]	abs(θ) [rad]
1	0.498	0.278	0.214
2	0.136	0.073	0.010
3	0.085	0.116	0.113
4	0.288	0.098	0.115
5	0.482	1.072	0.101
Average	0.298	0.327	0.110

According to the tables, the Particle Filter (PF) has the lowest absolute errors for all the components x, y and θ, and thus is selected for the final estimator algorithm to solve the problem in this project.

C. Best Result for Submission

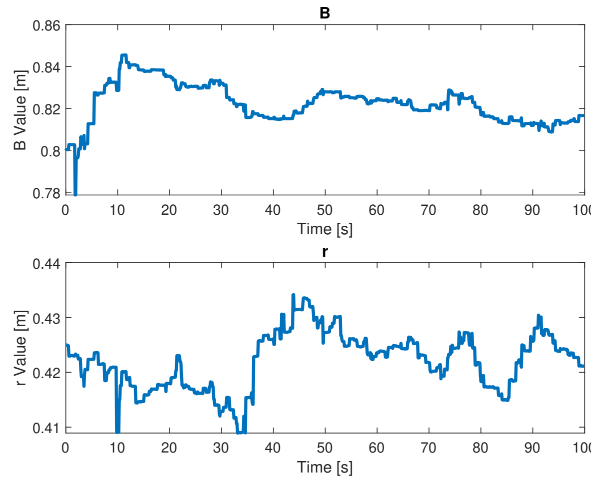

Fig. 3. Tire Radius r and Baseline B estimation of Run 3 with the Final PF

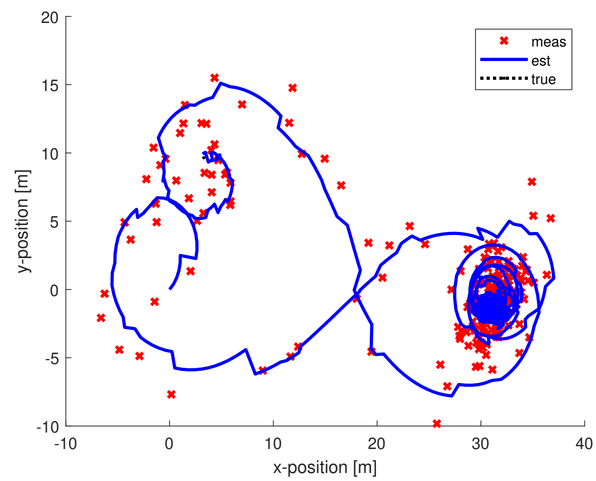

Fig. 4. Measurements and Estimation Results of Run 3 with the Final PF

With the Particle Filter (PF) chosen, different values of $P(0)$ and V have been run through to decide the optimal combination which derives smallest sum of absolute errors for the first 5 runs. As a result, the final chosen values for $P(0)$ and V are:

$$P(0) = \mathrm{diag}([\sigma^2_{x_1(0)}, \sigma^2_{y_1(0)}, \sigma^2_{\theta(0)}, \sigma^2_{B(0)}, \sigma^2_{R(0)}]) \tag{56}$$

$$= \mathrm{diag}([0.05, 0.05, 0.05\pi, \sigma^2_{B(0)}, \sigma^2_{R(0)}]) \tag{57}$$

$$V = \mathrm{diag}([\sigma^2_{\Delta x_1}, \sigma^2_{\Delta y_1}, \sigma^2_{\Delta\theta}, \sigma^2_{\Delta B}, \sigma^2_{\Delta R}]) \tag{58}$$

$$= \mathrm{diag}([0.01, 0.01, 0.001\pi, 1 \times 10^{-6}, 1 \times 10^{-6}) \tag{59}$$

In principle, $\sigma^2_{\Delta B} = \sigma^2_{\Delta R} = 0$, corresponding to the fact that these two parameters do not change during the run. But we

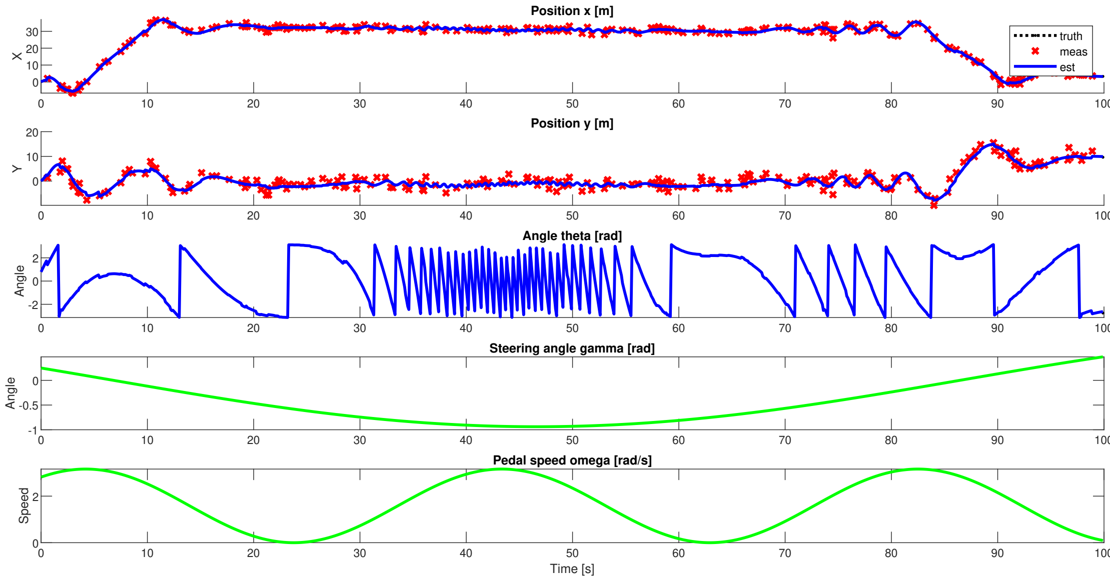

Fig. 5. States and Inputs of Run 3 with the Final PF

set them to be a very small number to maintain the positive definite property of V. The roughening parameter is chosen as $K_r = 0.001$. The number of particles N is chosen to be $N = 1000$ considering the trade-off between running time and precision.

The final errors for the first 5 given runs of the PF with the above parameters are indicated in Table IV and the running time is about 0.005s per step.

TABLE IV
FINAL PF RESULTS FOR THE 1-5 RUNS

Run #	x [m]	y [m]	θ [rad]
1	-0.574	-0.371	0.220
2	0.028	0.381	0.126
3	0.029	0.601	0.017
4	-0.831	-1.240	-0.139
5	0.649	-1.229	0.235

The visualization for the final PF to estimate the run 3 is illustrated in Fig. 4 to further demonstrate its performance. In the figure, the red cross dots are the measured position data, the blue line is the estimated path for the bicycle ride, and the single black dot is the final true position. As a result, the chosen PF estimator produces a reasonable path and generally tracks the final position. Fig. 3 and Fig. 5 also indicate how the states and inputs evolve with the time.

V. Conclusion and Future Work

During this project, three different estimator algorithms (the Extended Kalman Filter, the Unscented Kalman Filter, and the Particle Filter) have been implemented to estimate the rear wheel position of a simplified bicycle system in the 2-D plane. After tuning and comparing the performance of each estimator with the given measurement data, we choose the Particle Filter (PF). With further uncertainty tuning, the final PF estimator has delivered a satisfactory result for position estimation.

In speaking of the future work, research on the relation between uncertainty components and final errors shall be conducted to improve the estimator performance. Moreover, we could combine different estimator algorithms simultaneously to potentially optimize estimation performance.