MEC 231B Final Project Report

Joohwan Seo and Youngsang Suh

April 2022

1 Problem formulation

1.1 Target system

In this project, we aim to implement a state estimator for a bicycle. We are given the bicycle model and several measurement signals such as position, steering angle, and pedal speed. The model is as follows:

$$\begin{aligned}\dot{x}_1(t) &= v(t)\cos\theta(t)\\ \dot{y}_1(t) &= v(t)\sin\theta(t)\\ \dot{\theta}(t) &= \frac{v(t)}{B}\tan\gamma(t)\end{aligned} \tag{1}$$

where (x_1, y_1) is the position of the rear wheel, θ is the heading, γ is the steering angle, $\bar{\omega}$ is the pedaling speed, r is the wheel radius, and the B is the wheelbase. We set $x = [x_1, y_1, \theta]^T$ as the system state and $u = [\gamma, \bar{\omega}]^T$ as our control inputs. Here, we are tasked to track the trajectory of a bicycle in uncertain model and measurements. Thus, we make some assumptions to tackle the uncertainties.

1.2 Modeling & Design decisions

Since bicycle's physical parameters are not perfectly known and some measurements are assumed to be piecewise constant, we regard them as process noise parameters $v(k)$. For measurement noise $w(k)$, we use a calibration data. It is the measurement data while the bicycle is staying still, hence calculating the expectation and the covariance matrix from (x_1, y_1) data corresponds to the measurement noise distribution.

$$\begin{aligned}v(k) &= [r\ B\ \bar{\omega}(k)\ \gamma(k)]^T\\ \Sigma_{vv} &= \mathrm{diag}[\sigma_r^2, \sigma_B^2, \sigma_{\bar{\omega}}^2, \sigma_\gamma^2]\\ \mathbb{E}[w(k)] &= [0.1875\ 0.3962]^T\\ \Sigma_{ww} &= \begin{bmatrix}1.08934 & 1.53329\\ 1.53329 & 2.98795\end{bmatrix}\end{aligned} \tag{2}$$

where σ_r^2 is the variance for the wheel radius, σ_B^2 is the variance for the wheelbase, $\sigma_{\bar{\omega}}^2$ is the variance for the pedal speed, and σ_γ^2 is the variance for the steering angle, respectively. $\sigma_{\bar{\omega}}^2$ and σ_γ^2 are our tuning parameters for estimator design, which indicates the uncertainty coming from a piecewise constant assumption, and a tire slip. Equation (2) shows our noise model. Since parameter r, B and measurement $\bar{\omega}$, γ contain uncertainty, assuming them as process noise, we would be able to better estimate the state of a bicycle. As mentioned, measurement noise $w(k)$ distribution is computed referring to the calibration data.

Taking the noise model into account, the system model can be rewritten as

$$\begin{aligned}\dot{x}(t) &= q(x(t), u(t), v(t))\\ z(t) &= h(x(t), w(t))\end{aligned} \tag{3}$$

We discretized the system equation using the forward difference method for the implementation, which is given by

$$\begin{aligned}x(k+1) &= q_k(x(k), u(k), v(k)) := x(k) + T \cdot q(x(k), u(k), v(k))\\ z(k) &= h(x(k), w(k)),\end{aligned} \tag{4}$$

where T is the sampling time (0.01s). We defined our noise models and indicated that $\sigma_{\bar{\omega}}^2$, σ_γ^2 are our design parameters. Following sections introduce our estimators that we implemented using different methods (Subsection 2.1 EKF design and Subsection 2.2 UKF design).

2 EKF and UKF

2.1 EKF design

2.1.1 Initialization

We initialize the state $x(k)$ as $x_m(0) = [0\ 0\ \frac{1}{4}\pi]^T$, as it is given that the cyclist starts near the origin and initially heading approximately North-East. Also, for $P_m(0)$, we choose this matrix our own. We arbitrarily tuned as $P_m(0) = I_{3\times3}$. Using the discretized system equation (4), we further describe the state equation. Note that the angular velocity of the rear wheel is 5 times the pedaling speed ($v(k) = 5r\bar{\omega}(k)$).

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ y_1(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} q1 \\ q2 \\ q3 \end{bmatrix} = \begin{bmatrix} x_1(k) + 5r\bar{\omega}(k)\cos\theta(k)\cdot T \\ y_1(k) + 5r\bar{\omega}(k)\sin\theta(k)\cdot T \\ \theta(k) + \frac{5r\bar{\omega}(k)}{B}\tan\gamma(k)\cdot T \end{bmatrix} \tag{5}$$

2.1.2 Prior Update

Implementing EKF, we first run prior update. Following equations describe the process:

$$\begin{aligned} &\hat{x}_p(k+1) = q(\hat{x}_m(k), \mathbb{E}[v(k)]) \\ &A(k) = \frac{\partial q}{\partial x}(\hat{x}_m(k), \mathbb{E}[v(k)]) = \begin{bmatrix} \frac{\partial q_1}{\partial x_1} & \frac{\partial q_1}{\partial y_1} & \frac{\partial q_1}{\partial \theta} \\ \frac{\partial q_2}{\partial x_1} & \frac{\partial q_2}{\partial y_1} & \frac{\partial q_2}{\partial \theta} \\ \frac{\partial q_3}{\partial x_1} & \frac{\partial q_3}{\partial y_1} & \frac{\partial q_3}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -5r\bar{\omega}(k)\sin\theta(k)\cdot T \\ 0 & 1 & 5r\bar{\omega}(k)\cos\theta(k)\cdot T \\ 0 & 0 & 1 \end{bmatrix} \\ &L(k) = \frac{\partial q}{\partial v}(\hat{x}_m(k), \mathbb{E}[v(k)]) = \begin{bmatrix} 5\bar{\omega}(k)\cos\theta(k)\cdot T & 0 & 5\gamma\cos\theta(k)\cdot T & 0 \\ 5\bar{\omega}(k)\sin\theta(k)\cdot T & 0 & 5\gamma\sin\theta(k)\cdot T & 0 \\ \frac{5\bar{\omega}(k)}{B}\tan\gamma(k)\cdot T & \frac{-5r\bar{\omega}(k)}{B^2}\tan\gamma(k)\cdot T & \frac{5r}{B}\tan\gamma(k) & \frac{5r\bar{\omega}(k)}{B}\sec\gamma(k)^2\cdot T \end{bmatrix} \end{aligned} \tag{6}$$

Here, we derived matrices $A(k)$ and $L(k)$. From the matrices, we can compute matrix $P_p(k)$.

$$P_p(k+1) = A(k)P_m(k)A^T(k) + L(k)\Sigma_{vv}L^T(k) \tag{7}$$

where we selected $\Sigma_{vv} = \text{diag}[0.00015, 0.00213, 0.1, 0.1]$.

2.1.3 Measurement Update

We receive the measurement of the bicycle center $p(k)$. Thus, matrices $H(k)$ and $M(k)$ are derived as follows:

$$\begin{aligned} z(k) &= h(x(k), w(k)) = \begin{bmatrix} x_1(k) + 0.5B\cos\theta(k) + w_1(k) \\ y_1(k) + 0.5B\sin\theta(k) + w_2(k) \end{bmatrix} \\ H(k) &= \frac{\partial h}{\partial x}(\hat{x}_p(k), \mathbb{E}[w(k)]) = \begin{bmatrix} 1 & 0 & -0.5B\sin\theta(k) \\ 0 & 1 & 0.5B\cos\theta(k) \end{bmatrix} \\ M(k) &= \frac{\partial h}{\partial w}(\hat{x}_p(k), \mathbb{E}[w(k)]) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{8}$$

Now, we can compute the Kalman gain $K(k)$, referring to the measurement noise distribution from equation (2). Using the Kalman gain, we can finally update the posteriors.

$$\begin{aligned} &K(k) = P_p(k)H^T(k)\left(H(k)P_p(k)H^T(k) + M(k)\Sigma ww M^T(k)\right)^{-1} \\ &\hat{x}_m(k) = \hat{x}_p(k) + K(k)\left(\bar{z}(k) - h(\hat{x}_p(k), \mathbb{E}[w(k)])\right) \\ &P_m(k) = (I_{3\times3} - K(k)H(k))P_p(k) \end{aligned} \tag{9}$$

2.2 UKF design

Before we directly describe the UKF implementation, we present the system modification for UKF. We can rewrite the system equation (4) as follows:

$$\begin{aligned} x(k+1) &= q_k(x(k), u(k), v(k)) \\ y(k) &= h_k(x(k)) + w(k) \end{aligned} \tag{10}$$

Note that in the measurement, the uncertainty $w(k)$ is decoupled from the measurement function $h_k(\cdot)$. Therefore, $v(k)$ should be considered as the nonlinear noise case, which is covered in our lecture note 11.3.2, but $w(k)$ can be considered as additive noise case, which is described in 11.3.1 [1]. Therefore, we define augmented states ξ as

$$\xi = [x, y, \theta, r, B, \bar{\omega}, \gamma]^T \in \mathbb{R}^{n_\xi} \tag{11}$$

where $n_\xi = n + \dim(v)$ in our case. Then, augmented system functions can be written as

$$\begin{aligned} \bar{q}_k(\xi) &:= [q_k(x, u, v)^T, v^T]^T \\ \bar{h}_k(\xi) &= h_k(x) + w(k) \end{aligned} \tag{12}$$

2.2.1 Initialization

Following the assumption of the EKF design, we also initialize the state and uncertainty variance as

$$x_m(0) = [0,\ 0,\ \frac{1}{4}\pi]^T, \quad P_m(0) = I_{3\times 3}.$$

2.2.2 Prior Update

First, we let

$$\hat{\xi}_m(k-1) = [\hat{x}_m(k-1)^T,\ 0,\ 0]^T, \quad \mathrm{Var}[\hat{\xi}_m(k-1)] = \text{block diag}[P_m(k-1), \Sigma_{vv}(k-1)]. \tag{13}$$

Then, the $2n+1$ sigma points are generated as

$$\begin{aligned} s_{\hat{\xi}_m(k-1),0} &= \hat{x}_m(k-1) \\ s_{\hat{\xi}_m(k-1),i} &= \hat{x}_m(k-1) + \left(\sqrt{n_\xi \mathrm{Var}[\hat{\xi}_m(k-1)]}\right)_i \\ s_{\hat{\xi}_m(k-1),n+i} &= \hat{x}_m(k-1) + \left(\sqrt{n_\xi \mathrm{Var}[\hat{\xi}_m(k-1)]}\right)_i \end{aligned} \tag{14}$$

where $i = 1, \cdots, n$. Note that we define the square root of the matrix as $\sqrt{A} = T\sqrt{\Lambda}T^{-1}$, where $A = T\Lambda T^{-1}$ can be obtained by the eigenvalue decomposition.
Then, the prior sigma points are computed as

$$\begin{aligned} s_{\hat{\xi}_p(k),i} &= q_{k-1}\left(s_{\hat{\xi}_m(k-1),i}\right), \quad \text{for } i \in \{0, 1, \ldots, 2n_\xi\} \\ &=: [s_{x_p(k)}^T,\ \cdot\ ,\ \cdot]^T, \end{aligned} \tag{15}$$

with $\Sigma_{vv} = \mathrm{diag}[0.00015, 0.00213, 0.15, 0.01]$. Using these sigma points, the piror statistics are calculated as

$$\begin{aligned} \hat{x}_p(k) &= \sum_{i=0}^{2n_\xi} \frac{1}{2n_\xi + 1} s_{x_p(k),i}, \\ P_p(k) &= \sum_{i=0}^{2n\xi} \frac{1}{2n_\xi + 1} (s_{x_p(k),i} - \hat{x}_p(k))(s_{x_p(k),i} - \hat{x}_p(k))^T. \end{aligned} \tag{16}$$

2.2.3 Measurement Update

Using the sigma points for x_p, σ_{x_p}, the sigma points for the measurements are computed as

$$s_{z(k),i} = h_k(s_{x_p(k),i}, \mathbb{E}[w(k)]), \quad \text{for } \in \{0, 1, \ldots, 2n_\xi\}. \tag{17}$$

We can obtain expected measurement, the associated covariance matrix $P_z z(k)$ as well as cross covariance $P_{xz}(k)$ given by

$$\begin{aligned} \hat{z}(k) &= \sum_{i=0}^{2n_\xi} \frac{1}{2n_\xi + 1} s_{z(k),i}, \\ P_{zz}(k) &= \sum_{i=0}^{2n_\xi} \frac{1}{2n_\xi + 1} (s_{z(k),i} - \hat{z}(k))(s_{z(k),i} - \hat{z}(k))^T, \\ P_{xz}(k) &= \sum_{i=0}^{2n_\xi} \frac{1}{2n_\xi + 1} (s_{x_p(k),i} - \hat{x}_p(k))(s_{z(k),i} - \hat{z}(k))^T. \end{aligned} \tag{18}$$

Finally, the Kalman filter gain is calculated as

$$\begin{aligned} K(k) &= P_{xz}(k)P_{zz}(k)^{-1} \\ \hat{x}_m(k) &= \hat{x}_p(k) + K(k)(z(k) - \hat{z}(k)) \\ P_m(k) &= P_p(k) - K(k)P_z z(k)K(k)^T \end{aligned} \tag{19}$$

3 Experiment and discussion

We have tested the aforementioned two estimators with the same test data set. To avoid over-fitting, we ran over 1 $\sim$ 99 data set and tuned our design parameters. For analysis, the average of final x, y, θ errors are computed. We also extracted the variance of the final errors so that we can choose the better estimator that has small average error with small variance. Table 1 shows our EKF and UKF estimator performance over 1 $\sim$ 99 run data. As a result, we decided to choose **EKF** as our final estimator.

Table 1: Evaluation result of EKF and UKF estimators over 1 99 run data set

Estimator	**EKF**	**UKF**
$\mathbb{E}[e_x]$	-0.0676	0.0450
$\mathbb{E}[e_y]$	-0.0265	0.1258
$\mathbb{E}[e_\theta]$	-0.0272	0.8338
Var$[e_x]$	0.0170	0.0104
Var$[e_y]$	0.0480	0.0425
Var$[e_\theta]$	0.0589	0.3985

To help visualize our estimator performance, we also show the final error of the EKF estimator over 1 $\sim$ 5 run data. For discussion, we select data run_001. Fig. 1 is the trajectory estimation result for run_001. We also plotted UKF estimator's result for reference in Fig. 2. From Table 2, we notice that final e_x, e_y, and e_θ are -0.0459, -0.2369, 0.0800, respectively. Comparing to the typical scores of the instructor's estimator, we conclude that our EKF estimator has a comparable performance. However, we still have some possibilities of bad performance since we are only comparing the final error, not along the whole trajectory. Fortunately, the estimator performance holds for run_002 $\sim$ 005 data, showing that our estimator is not over-fitted to a specific run data.

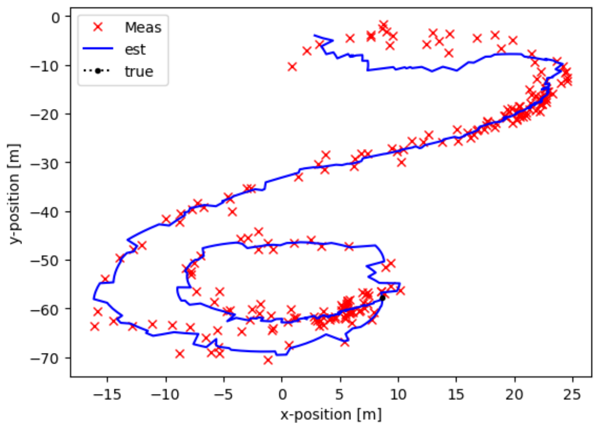
We figured out that the EKF estimator performs worse when our input (γ, $\bar{\omega}$) variance gets small. Setting our input variance smaller compared to the measurement variance means that we should trust the bicycle model, i.e.,prediction, more than the measurement. This intuition can be clarified by looking at Fig. 1 (a). Here, the estimated trajectory of the EKF showed noisy response since we designed our input variance to be large, thus the estimation follows the noisy outputs. On the other hand, for UKF estimator, it turns out that final error decreases when we trust the model more than the measurement data, which is why we designed our input variance to be small. This may be because the UKF is relying on more accurate nonlinear model, while the EKF is based on linearized model.

Table 2: Final error of 1 $\sim$ 5 run data (EKF)

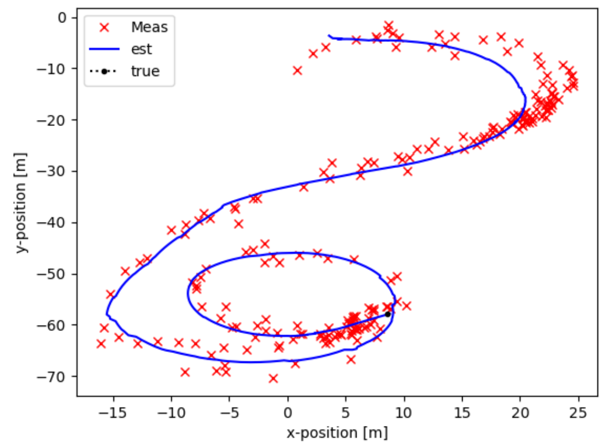
Run data	**run_001**	**run_002**	**run_003**	**run_004**	**run_005**
final e_x	-0.0459	0.0711	-0.3561	-0.2690	0.3297
final e_y	-0.2369	-0.3298	0.2222	-0.6250	-1.2886
final e_θ	0.0800	0.1794	0.0522	-0.0762	0.0860

References

[1] Mark, W, M., 2022. *Lecture Notes on ME231B.* (Online).

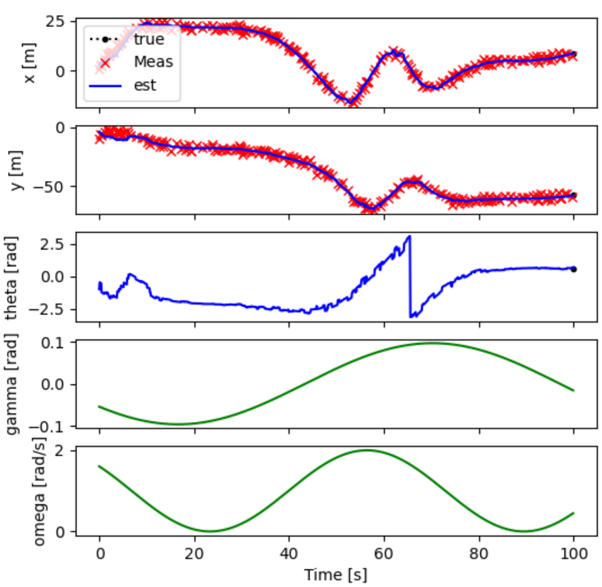


(a)

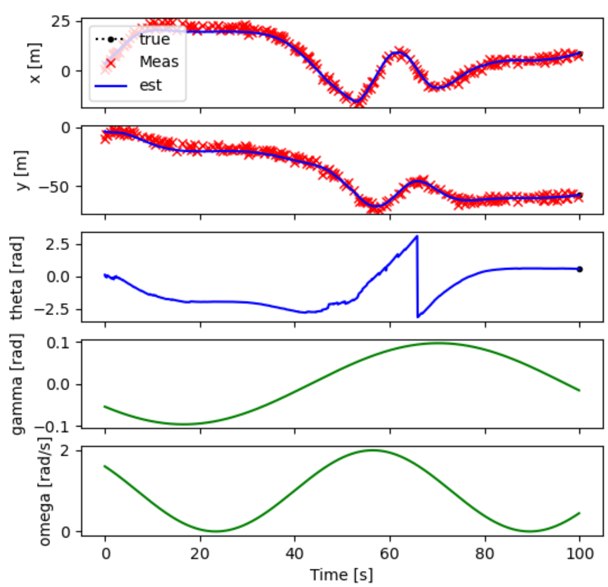


(b)

Figure 1: Trajectory estimation result for run_001. (a) EKF result, (b) UKF result.



(a)



(b)

Figure 2: State estimation results for run_001. (a) EKF result, (b) UKF result.