ME8135 State Estimation for Robotics and Computer Vision HW2

Matthew Lisondra Sajad Saeedi

June 2, 2023

Question 1: Kalman Filter

Use pyGame, or any other similar libraries, to simulate a simplified 2D robot and perform state estimation using a Kalman Filter. Motion Model:

$$\dot{x} = \frac{r}{2}(u_r + u_l) + \omega_x \qquad \dot{y} = \frac{r}{2}(u_r + u_l) + \omega_y$$

$r = 0.1$ m is the radius of the wheel, u_r and u_l are control signals applied to the right and left wheels. $\omega_x = N(0, 0.1)$ and $\omega_y = N(0, 0.15)$. Simulate the system such that the robot is driven 1 m to the right. Assume the speed of each wheel is fixed and is 0.1 m/s. Use these initial values

$$x_0 = 0, y_0 = 0, P_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \text{ (initial covariance matrix)}$$

and assume the motion model is computed 8 times a second. Assume every second a measurement is given:

$$z = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r_x & 0 \\ 0 & r_y \end{pmatrix}$$

where $r_x = N(0, 0.05)$ and $r_y = N(0, 0.075)$.

SOLUTION: we need to discretize the system by first doing,

$$\begin{aligned} \dot{x} &= \frac{r}{2}(u_r + u_l) + \omega_x \\ \frac{x_k - x_{k-1}}{T} &= \frac{r}{2}(u_r + u_l) + \omega_x \\ x_k - x_{k-1} &= \frac{Tr}{2}(u_r + u_l) + T\omega_x \\ x_k &= x_{k-1} + \frac{Tr}{2}(u_r + u_l) + T\omega_x \end{aligned}$$

$$\begin{aligned} \dot{y} &= \frac{r}{2}(u_r + u_l) + \omega_y \\ \frac{y_k - y_{k-1}}{T} &= \frac{r}{2}(u_r + u_l) + \omega_y \\ y_k - y_{k-1} &= \frac{Tr}{2}(u_r + u_l) + T\omega_y \\ y_k &= y_{k-1} + \frac{Tr}{2}(u_r + u_l) + T\omega_y \end{aligned}$$

$$\Rightarrow \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} + \begin{pmatrix} \frac{Tr}{2} & \frac{Tr}{2} \\ \frac{Tr}{2} & \frac{Tr}{2} \end{pmatrix} \begin{pmatrix} u_r \\ u_l \end{pmatrix} + \begin{pmatrix} T\omega_x \\ T\omega_y \end{pmatrix}$$

and so the last line is our motion model. We set $T = 1/8$ s and hardcode $u_r = 1$ and $u_l = 0.1$ (we set our controls like this becuase we mainly want our robot to go right).

Below is the Kalman filter algorithm. Using our motion model above, we see what we must define matrices A_t, B_t and R_t.

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

$\quad \bar{\mu}_t = A_t\, \mu_{t-1} + B_t\, u_t$

$\quad \bar{\Sigma}_t = A_t\, \Sigma_{t-1}\, A_t^T + R_t$

$\quad K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$

$\quad \mu_t = \bar{\mu}_t + K_t(z_t - C_t\, \bar{\mu}_t)$

$\quad \Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$

$\quad$ return μ_t, Σ_t

The Kalman filter algorithm for linear Gaussian state transitions and measurements [1]-[2].

We define matrices A_t, B_t and R_t as follows:

$$A_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad B_t = \begin{pmatrix} \frac{Tr}{2} & \frac{Tr}{2} \\ \frac{Tr}{2} & \frac{Tr}{2} \end{pmatrix} \qquad R_t = \begin{pmatrix} T * 0.1 & 0 \\ 0 & T * 0.15 \end{pmatrix}$$

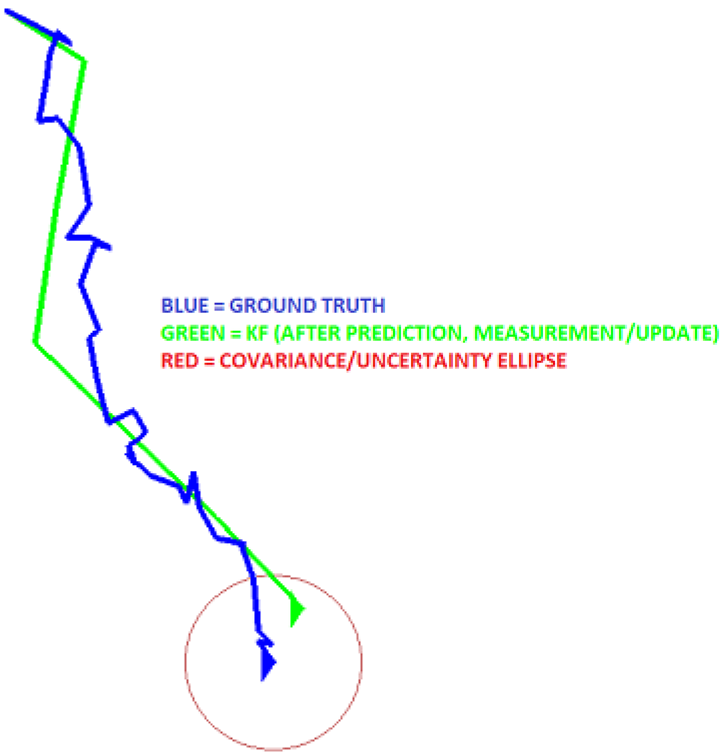
The R_t matrix is the covariance of the Gaussian random vector that models the randomness in the state transition given by the last term (the noise) in the motion model above $(T\omega_x, T\omega_y)$.

We now see from the measurement model we must define matrices C_t and Q_t as follows:

$$C_t = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \qquad Q_t = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.075 \end{pmatrix}$$

The Q_t matrix is the covariance of the Gaussian random vector that models the randomness in the measurement given by the last term (the noise) in the measurement model above (r_x, r_y).

Below is a trajectory along with the covariance ellipse for the ground-truth motion model (in dark blue) and the KF motion (in green) after eight predictions and one update per second.



Robot is moving to the right (as in right side of screen) (ground-truth and KF trajectories shown) [1].

The plot and animation is in a separate file. Just analyzing the trajectories, we see that the KF trajectory is smoother and still within a bound around the ground truth.

Question 2: Extended Kalman Filter

Repeat the previous assignment, this time with a classic motion model and range observations made from a landmark located at $M = [10, 10]$. L is the distance between the wheel, known as wheelbase, and is 0.3 m.

$$\dot{x} = \frac{r}{2}(u_r + u_l)\cos(\theta) + \omega_x \qquad \dot{y} = \frac{r}{2}(u_r + u_l)sin(\theta) + \omega_y \qquad \dot{\theta} = \frac{r}{L}(u_r - u_l)$$

Assume

$$u_\omega = \frac{1}{2}(u_r + u_l), \qquad u_\psi = (u_r - u_l)$$

Then the equations become:

$$\dot{x} = ru_\omega\cos(\theta) + \omega_\omega \qquad \dot{y} = ru_\omega sin(\theta) + \omega_\omega \qquad \dot{\theta} = \frac{r}{L}u_\psi + \omega_\psi$$

$\omega_\psi = N(0, 0.01)$ and $\omega_\omega = N(0, 0.1)$. Program the robot such that it loops around point M.

a) Compute the EKF with the linear measurement model in the previous assignment.

SOLUTION: Refer to the other files.

b) Compute the EKF with the range/bearing measurements of point M. Assume range noise is $N(0, 0.1)$ and bearing noise is $N(0, 0.01)$. Range is in meters, and bearing is in radians. Visualize the measurements as well.

SOLUTION: Refer to the other files.

References

[1] Thrun, Sebastian. "Probabilistic robotics." Communications of the ACM 45.3 (2002): 52-57.

[2] Barfoot, Timothy D. State estimation for robotics. Cambridge University Press, 2017.