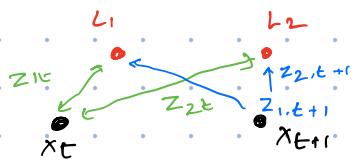


Problem 1: Extended Kalman Filter

In this exercise you will apply the extended Kalman filter to solve a basic 2D robot localization problem. To that end, we will consider a ground robot that is navigating in a 2D environment containing two landmarks at known positions $l_i = [l_{ix}, l_{iy}] \in \mathbb{R}^2$ for $i \in \{1, 2\}$.

- (a) We will model our robot as a simple point moving in the plane (i.e. ignoring its orientation), and assume that it is possible to directly control its velocity. Denoting its position and velocity at time t by $p(t) = (p_x(t), p_y(t)) \in \mathbb{R}^2$ and $v(t) = (v_x(t), v_y(t)) \in \mathbb{R}^2$ (respectively), write down the discrete-time state transition model that predicts the robot's position $p(t + \Delta t)$ at time $t + \Delta t$ given its position $p(t)$ at time t , assuming constant velocity and additive mean-zero Gaussian noise with covariance $R \in \mathbb{S}_+^2$ on the final position.
- (b) At each timestep, we assume that the robot is able to collect a noisy measurement of the range to each of the two landmarks in the environment, subject to additive mean-zero Gaussian noise with covariance $Q \in \mathbb{S}_+^2$. Write down the corresponding measurement model for the resulting measurements.
- (c) As we saw in class, implementing an extended Kalman filter requires linearizing any nonlinear state transition or measurement models about the current state estimate, and using the resulting linear approximations in the standard Kalman filter equations. Derive the Jacobians for the state transition and measurement models in parts (a) and (b).
- (d) Using your results from parts (a)–(c), derive the corresponding state propagation and measurement update equations for an extended Kalman filter to solve this localization problem.
- (e) Using the EKF that you designed in parts (a)–(d), write a script that applies it to solve the following simulated localization scenario:

- The landmarks are located at $l_1 = (5, 5)$ and $l_2 = (-5, 5)$.
- The covariances for the robot state transition and measurement models are given by $R = .1I_2$, $Q = .5I_2$ for all $t \in [0, 40]$.
- The robot's initial belief over its position is $p(0) \sim \mathcal{N}(0, I_2)$.
- The timestep interval is given by $\Delta t = .5$, and the sequence of velocity commands is given by:
 - $v = (1, 0)$ for all $0 \leq t \leq 10$,
 - $v = (0, -1)$ for all $10 < t \leq 20$,
 - $v = (-1, 0)$ for all $20 < t \leq 30$,



a) $p(t + \Delta t) = f[p(t), v] + \epsilon_t$ where $p(t) = \begin{bmatrix} p_x(t) \\ p_y(t) \end{bmatrix}$
 $v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$

$$\begin{bmatrix} p_x(t + \Delta t) \\ p_y(t + \Delta t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_x(t) \\ p_y(t) \end{bmatrix} + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \epsilon_t$$

where $\epsilon_t = (0, R_t)$ $\rightarrow R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$

b) $z(t) = h(x_t) + \delta_t$ where $x_t \sim \mathcal{N}(0, \Theta_t)$

$$\Rightarrow \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \sqrt{(p_x(t) - l_{1x})^2 + (p_y(t) - l_{1y})^2} \\ \sqrt{(p_x(t) - l_{2x})^2 + (p_y(t) - l_{2y})^2} \end{bmatrix} + \delta_t$$

where $\delta_t \sim \mathcal{N}(0, \Theta_t)$, $\Theta_t = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}$

c) Jacobian for state transition model

$$p(t + \Delta t) = p_x + v_x \Delta t \leftarrow f_1$$

$$p_y + v_y \Delta t \leftarrow f_2$$

$$\frac{\partial f_1}{\partial p_x} = 1 \quad \frac{\partial f_1}{\partial p_y} = 0 \quad \Rightarrow J(p(t + \Delta t)) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\frac{\partial f_2}{\partial p_x} = 0 \quad \frac{\partial f_2}{\partial p_y} = 1$$

Jacobian for measurement model

$$z(t) = \sqrt{(p_x(t) - l_{1x})^2 + (p_y(t) - l_{1y})^2} \leftarrow f_1$$

$$\sqrt{(p_x(t) - l_{2x})^2 + (p_y(t) - l_{2y})^2} \leftarrow f_2$$

$$\frac{\partial f_1}{\partial p_x} = \frac{2(p_x - l_{1x})}{2\sqrt{(p_x(t) - l_{1x})^2 + (p_y(t) - l_{1y})^2}} = \frac{p_x - l_{1x}}{z_1}$$

$$\frac{\partial f_1}{\partial p_y} = \frac{2(p_y - l_{1y})}{2\sqrt{(p_x(t) - l_{1x})^2 + (p_y(t) - l_{1y})^2}} = \frac{p_y - l_{1y}}{z_1}$$

where $z_1 = \sqrt{(p_x(t) - l_{1x})^2 + (p_y(t) - l_{1y})^2}$

$$\frac{\partial f_2}{\partial P_x} = \frac{2(P_x - l_{2x})}{2\sqrt{(P_x(t) - l_{2x})^2 + (P_y(t) - l_{2y})^2}} = \frac{P_x - l_{2x}}{z_2}$$

$$\frac{\partial f_2}{\partial P_y} = \frac{P_y - l_{2y}}{z_2}$$

$$\Rightarrow H_t = \begin{bmatrix} \frac{P_x - l_{2x}}{z_1} & \frac{P_x - l_{2x}}{z_1} \\ \frac{P_x - l_{2x}}{z_2} & \frac{P_y - l_{2y}}{z_2} \end{bmatrix}$$

$$G_T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

d) Algorithm : Predict

$$x_{t+1} = g_t(x_t, u_t) + \epsilon_t \Rightarrow \begin{bmatrix} P_x(t+\Delta t) \\ P_y(t+\Delta t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_x(t) \\ P_y(t) \end{bmatrix} + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \epsilon_t$$

where $\epsilon_t = (0, R_t) \rightarrow R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$

$$\therefore x_t = N(\mu_t, \Sigma_t)$$

$$\Sigma_{t+1} = G_t \Sigma_t G_T^T + R_t \quad \text{and } G_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{Jacobian from process model})$$

Update step : to update current belief \rightarrow plugging $x_t(\hat{u}, \hat{\Sigma}) = x_{t+1}$ (from update step)

$$x_t = \hat{x}_t + k_t (z - h_t(\hat{x}_t)), \quad \Sigma_t = (I - k_t H_t) \hat{\Sigma}_t \quad \text{from update step}$$

$$\text{and Kalman gain } k_t = \hat{\Sigma}_t H_t^T (H_t \hat{\Sigma}_t H_t^T + Q_t)^{-1}$$

$$\text{where } H_t \text{ is } \begin{bmatrix} \frac{P_x - l_{2x}}{z_1} & \frac{P_x - l_{2x}}{z_1} \\ \frac{P_x - l_{2x}}{z_2} & \frac{P_y - l_{2y}}{z_2} \end{bmatrix}$$

$(x_t, \Sigma_t) \rightarrow$ output of update step is kalman filtered belief.

\rightarrow Plug (x_t, Σ_t) into predict step and continue loop for num-iter.

Problem 3

a) $r(t) = x_0 \exp(t - \frac{1}{2})$ where $x_0 = x @ t=0$

$$\ddot{J} \begin{bmatrix} 0 & -r/w \left((\dot{\varphi}_r - \dot{\varphi}_e + \varepsilon_r + \varepsilon_e) \right) & \frac{r}{2} \left((\dot{\varphi}_r - \dot{\varphi}_e) + \varepsilon_r + \varepsilon_e \right) \\ \frac{r}{w} (\dot{\varphi}_r - \dot{\varphi}_e) + \varepsilon_r + \varepsilon_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

b) $f(x | \mu, \sigma^2)$

$$f(z | x_t) = \frac{1}{\sqrt{2\sigma_p^2 \pi}} e^{-\frac{(z - x_t)^2}{2\sigma_p^2}}$$

we use l_t instead
of $x_t(l_t, R_t)$
since GPS only
measures l_t

c \rightarrow g : In code