AA274 (Winter 2017-18): Problem Set 3

Anqi Fu

February 23, 2018

1 Extended Kalman Filter (EKF)

(i) See submitted code. Let the state variables be $\mathbf{x}(t) = (x(t), y(t), \theta(t))$ and the control variables be $\mathbf{u}(t) = (V(t), \omega(t))$. The differential drive model is

$$\begin{aligned}
\dot{x}(t) &= V(t)\cos(\theta(t)) \\
\dot{y}(t) &= V(t)\sin(\theta(t)) \\
\dot{\theta}(t) &= \omega(t),
\end{aligned}$$

which we can discretize by assuming a zero-order hold on the control, i.e. $\mathbf{u}(t)$ is constant over a time interval of length dt. Then we can integrate to get

$$\theta_t - \theta_{t-dt} = \int_{t-dt}^{t} \dot{\theta}(s)ds = \int_{t-dt}^{t} \omega(s)ds = \omega_t \int_{t-dt}^{t} ds = \omega_t dt.$$

If $\omega(t) = 0$ over the interval, $\theta(t)$ remains constant, so our coordinates can be found easily by integrating

$$\begin{aligned}
x_t - x_{t-dt} &= \int_{t-dt}^{t} \dot{x}(s)ds = \int_{t-dt}^{t} V(s)\cos(\theta(s))ds = V_t\cos(\theta_{t-dt})dt \\
y_t - y_{t-dt} &= \int_{t-dt}^{t} \dot{y}(s)ds = \int_{t-dt}^{t} V(s)\sin(\theta(s))ds = V_t\sin(\theta_{t-dt})dt.
\end{aligned}$$

With a slight abuse of notation, We may thus write

$$\mathbf{x}_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} + V_t\cos(\theta_{t-1})dt \\ y_{t-1} + V_t\sin(\theta_{t-1})dt \\ \theta_{t-1} + \omega_t dt \end{pmatrix} = g(\mathbf{x}_{t-1}, \mathbf{u}_t)$$

and the Jacobians of g with respect to $\mathbf{x}$ and $\mathbf{u}$ are, respectively,

$$G_x = \begin{pmatrix} 1 & 0 & -V_t\sin(\theta_{t-1})dt \\ 0 & 1 & V_t\cos(\theta_{t-1})dt \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad G_u = \begin{pmatrix} \cos(\theta_{t-1})dt & 0 \\ \sin(\theta_{t-1})dt & 0 \\ 0 & dt \end{pmatrix}.$$

If $\omega(t) \neq 0$, we exploit the fact that $\dot{\theta}(t) = \omega(t)$ to get

$$\begin{aligned}
x_t - x_{t-dt} &= \int_{t-dt}^{t} V(s)\cos(\theta(s))\frac{\dot{\theta}(s)}{\omega(s)}ds = \frac{V_t}{\omega_t}\int_{t-dt}^{t}\cos(\theta(s))\dot{\theta}(s)ds \\
&= \frac{V_t}{\omega_t}(\sin(\theta_t) - \sin(\theta_{t-dt})) = \frac{V_t}{\omega_t}(\sin(\theta_{t-dt} + \omega_t dt) - \sin(\theta_{t-dt})) \\
y_t - y_{t-dt} &= \frac{V_t}{\omega_t}\int_{t-dt}^{t}\sin(\theta(s))\dot{\theta}(s)ds = \frac{V_t}{\omega_t}(-\cos(\theta_{t-dt} + \omega_t dt) + \cos(\theta_{t-dt}))
\end{aligned}$$

Our discrete model becomes

$$\mathbf{x}_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} + \frac{V_t}{\omega_t}(\sin(\theta_{t-1} + \omega_t dt) - \sin(\theta_{t-1})) \\ y_{t-1} - \frac{V_t}{\omega_t}(\cos(\theta_{t-1} + \omega_t dt) - \cos(\theta_{t-1})) \\ \theta_{t-1} + \omega_t dt \end{pmatrix} = g(\mathbf{x}_{t-1}, \mathbf{u}_t)$$

with Jacobians

$$\begin{aligned}
G_x &= \begin{pmatrix} 1 & 0 & \frac{V_t}{\omega_t}(\cos(\theta_{t-1} + \omega_t dt) - \cos(\theta_{t-1})) \\ 0 & 1 & \frac{V_t}{\omega_t}(\sin(\theta_{t-1} + \omega_t dt) - \sin(\theta_{t-1})) \\ 0 & 0 & 1 \end{pmatrix} \\
G_u &= \frac{1}{\omega_t}\begin{pmatrix} \sin(\theta_t) - \sin(\theta_{t-1}) & -\frac{V_t}{\omega_t}(\sin(\theta_t) - \sin(\theta_{t-1})) + V_t\cos(\theta_t)dt \\ -\cos(\theta_t) + \cos(\theta_{t-1}) & \frac{V_t}{\omega_t}(\cos(\theta_t) - \cos(\theta_{t-1})) + V_t\sin(\theta_t)dt \\ 0 & \omega_t dt \end{pmatrix}.
\end{aligned}$$

where $\theta_t := \theta_{t-1} + \omega_t dt$.

(ii) See submitted code. Let the belief state at $t-1$ be distributed as $\mathcal{N}(\mathbf{x}_{t-1}, P_{t-1})$. We model the dynamic certainty with white noise $\nu \sim \mathcal{N}(\mathbf{0}, Q)$ applied to the control input. For a time step of dt, the EKF prediction step is

$$\begin{aligned}
\bar{\mathbf{x}}_t &= g(\mathbf{x}_{t-1}, \mathbf{u}_t) \\
\bar{P}_t &= G_x P_{t-1} G_x^T + dt \cdot G_u Q G_u^T.
\end{aligned}$$

(iii) Let $\mathbf{m} = (\alpha, r)$ be the polar coordinate parameters of a line in the world frame. We are given the robot's state $\mathbf{x}_{rob} = (x_{rob}, y_{rob}, \theta_{rob})$, which is the offset/yaw of the robot's base frame with respect to the world frame, and $\mathbf{x}_{cam} = (x_{cam}, y_{cam}, \theta_{cam})$, the offset/yaw of the camera's frame with respect to the robot's base frame. To transform between frames, we must first translate by the (x_0, y_0) coordinates of the new origin, then rotate by the relative angle θ.

Define the (counterclockwise) rotation matrix

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

TODO: Finish frame mapping and compute Jacobian.

(iv)

(v)

(vi)

(vii)

2 EKF Simultaneous Localization and Mapping (SLAM)

(i) In EKF SLAM, the belief state is augmented with

$$\mathbf{x}(t) = (x(t), y(t), \theta(t), \alpha^1, r^1, \dots, \alpha^J, r^J)$$

where the map features are assumed to be static in the world frame, so that $\dot{\alpha}^j = 0$ and $\dot{r}^j = 0$ for all $j = 1, \dots, J$. We can readily extend the discrete model from Problem 1. If $\omega(t) = 0$, the state transition function is

$$\mathbf{x}_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \\ \alpha_t^1 \\ r_t^1 \\ \vdots \\ \alpha_t^J \\ r_t^J \end{pmatrix} = \begin{pmatrix} x_{t-1} + V_t \cos(\theta_{t-1}) dt \\ y_{t-1} + V_t \sin(\theta_{t-1}) dt \\ \theta_{t-1} + \omega_t dt \\ \alpha_{t-1}^1 \\ r_{t-1}^1 \\ \vdots \\ \alpha_{t-1}^J \\ r_{t-1}^J \end{pmatrix} = g(\mathbf{x}_{t-1}, \mathbf{u}_t)$$

with Jacobians $G_x = \frac{\partial g(\mathbf{x}_{t-1}, \mathbf{u}_t)}{\partial \mathbf{x}}$ and $G_u = \frac{\partial g(\mathbf{x}_{t-1}, \mathbf{u}_t)}{\partial \mathbf{u}}$ given by

$$G_x = \begin{pmatrix} 1 & 0 & -V_t \sin(\theta_{t-1}) dt & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & V_t \cos(\theta_{t-1}) dt & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \quad \text{and} \quad G_u = \begin{pmatrix} \cos(\theta_{t-1}) dt & 0 \\ \sin(\theta_{t-1}) dt & 0 \\ 0 & dt \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Similarly, when $\omega(t) \neq 0$, we have

$$\mathbf{x}_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \\ \alpha_t^1 \\ r_t^1 \\ \vdots \\ \alpha_t^J \\ r_t^J \end{pmatrix} = \begin{pmatrix} x_{t-1} + \frac{V_t}{\omega_t}(\sin(\theta_{t-1} + \omega_t dt) - \sin(\theta_{t-1})) \\ y_{t-1} - \frac{V_t}{\omega_t}(\cos(\theta_{t-1} + \omega_t dt) - \cos(\theta_{t-1})) \\ \theta_{t-1} + \omega_t dt \\ \alpha_{t-1}^1 \\ r_{t-1}^1 \\ \vdots \\ \alpha_{t-1}^J \\ r_{t-1}^J \end{pmatrix} = g(\mathbf{x}_{t-1}, \mathbf{u}_t)$$

and Jacobians

$$G_x = \begin{pmatrix} 1 & 0 & \frac{V_t}{\omega_t}(\cos(\theta_{t-1} + \omega_t dt) - \cos(\theta_{t-1})) & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \frac{V_t}{\omega_t}(\sin(\theta_{t-1} + \omega_t dt) - \sin(\theta_{t-1})) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$G_u = \frac{1}{\omega_t} \begin{pmatrix} \sin(\theta_t) - \sin(\theta_{t-1}) & -\frac{V_t}{\omega_t}(\sin(\theta_t) - \sin(\theta_{t-1})) + V_t \cos(\theta_t) dt \\ -\cos(\theta_t) + \cos(\theta_{t-1}) & \frac{V_t}{\omega_t}(\cos(\theta_t) - \cos(\theta_{t-1})) + V_t \sin(\theta_t) dt \\ 0 & \omega_t dt \\ 0 & dt \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

where $\theta_t := \theta_{t-1} + \omega_t dt$.

(ii)

(iii)

3 Turtlebot Localization

(i) See submitted code.

(ii) See submitted code.