1 The Odometry Model

The state of the robot at time t is $q_t \in \mathbb{R}^3$, where

$$q_t = \begin{bmatrix} \theta_t \\ x_t \\ y_t \end{bmatrix}. \tag{1}$$

The state of the map is $m_t = \begin{bmatrix} m_{x1} & m_{y1} & \dots & m_{xn} & m_{yn} \end{bmatrix}^T \in \mathbb{R}^{2n}$

Let $\xi_t = \begin{bmatrix} q_t \\ m_t \end{bmatrix} \in \mathbb{R}^{2n+3}$ be the combined state vector.

The odometry model that we derived earlier in the class governs how the robot's state transitions from time $t-1$ to time t, given the twist

$$u_t = \begin{bmatrix} \Delta\theta_t \\ \Delta x_t \\ 0 \end{bmatrix}. \tag{2}$$

The Δy_t component of the twist is 0 for the diff-drive robot.

To perform the odometry update, the twist u may be computed from encoder readings (this is what we do with our system) or it can be given as the control input (for example, if no wheel encoder feedback were available).

The forward kinematics of the robot convert encoder readings to a twist. At each timestep, the odometry calculations calculations provide $T_{wb'} = T(q) = T(\theta, x, y)$, the transformation from the world frame to the updated body frame.

The state transition has the form:

$$\begin{bmatrix} q_t \\ m_t \end{bmatrix} = g(\xi_{t-1}, u_t, w_t). \tag{3}$$

The transition function $g(\xi_{t-1}, u_t, w_t)$ is the model of the robot's movement and the map's movement. The map remains stationary, and the robot moves depending on the twist.

There are two cases for the dynamics, depending on whether there is a non-zero rotational velocity.

1. Zero rotational velocity: $\Delta\theta_t = 0$:

$$T_{wb'} = T(\theta_{t-1}, x_{t-1} + \Delta x_t \cos\theta_{t-1}, y_{t-1} + \Delta x_t \sin\theta_{t-1}) \tag{4}$$

   The transformation $T_{wb'}$ shows how the robot moves from its previous configuration $(\theta_{t-1}, x_{t-1}, y_{t-1})$ to its new configuration (θ_t, x_t, yt).

   The full state transformation (including the stationary map state and process noise $w_t \sim \mathcal{N}(0, Q)$) yields

$$\begin{bmatrix} q_t \\ m_t \end{bmatrix} = \begin{bmatrix} q_{t-1} \\ m_{t-1} \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} 0 \\ \Delta x_t \cos\theta_{t-1} \\ \Delta x_t \sin\theta_{t-1} \end{bmatrix} \\ 0_{2n} \end{bmatrix} + \begin{bmatrix} w_t \\ 0_{2n} \end{bmatrix} \tag{5}$$

The process noise accounts for uncertainty in the movement of the robot. Their is no noise component for the landmarks because we know they remain stationary. The notation 0_m means a zero vector in $\mathbb{R}^m$.

2. Non-zero rotational velocity[1]. $\Delta\theta_t \neq 0$:

$$\begin{aligned} T_{wb'} &= T(a,b,c) \\ a &= \theta_{t-1} + \Delta\theta_t \\ b &= x_{t-1} - \frac{\Delta x_t}{\Delta\theta_t}\sin\theta_{t-1} + \frac{\Delta x_t}{\Delta\theta_t}\sin(\theta_{t-1}+\Delta\theta_t) \\ c &= y_{t-1} + \frac{\Delta x_t}{\Delta\theta_t}\cos\theta_{t-1} - \frac{\Delta x_t}{\Delta\theta_t}\cos(\theta_{t-1}+\Delta\theta_t) \end{aligned} \tag{6}$$

The transformation $T_{wb'}$ leads to the state transition equation $\xi_t = g(\xi_{t-1}, u_t, w_t)$:

$$\begin{bmatrix} q_t \\ m_{t-1} \end{bmatrix} = \begin{bmatrix} q_{t-1} \\ m_{t-1} \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} \Delta\theta_t \\ -\frac{\Delta x_t}{\Delta\theta_t}\sin\theta_{t-1} + \frac{\Delta x_t}{\Delta\theta_t}\sin(\theta_{t-1}+\Delta\theta_t) \\ \frac{\Delta x_t}{\Delta\theta_t}\cos\theta_{t-1} - \frac{\Delta x_b}{\Delta\theta_t}\cos(\theta_{t-1}+\Delta\theta_t) \end{bmatrix} \\ 0_{2n} \end{bmatrix} + \begin{bmatrix} w_t \\ 0_{2n} \end{bmatrix} \tag{7}$$

Let $\hat{\xi}_t = \begin{bmatrix} \hat{q}_t \\ \hat{m}_t \end{bmatrix}$ be the current state estimate

The Extended Kalman filter uses a state transition model linearized about the current estimate. Therefore we Taylor expand $g(\xi_{t-1}, u_t, 0)$ about our estimate $\hat{\xi}_t$:

$$g(\xi_{t-1}, u_t) \approx g(\hat{\xi}_{t-1}, u_t) + g'(\hat{\xi}_{t-1}, u_t)(\xi_{t-1} - \hat{\xi}_{t-1}). \tag{8}$$

The function $g'(\xi_{t-1}, u_t)$ is the derivative of g with respect to the state ξ: There are two cases

1. $\Delta\theta_t = 0$:

$$A_t = g'(\xi_{t-1}, u_t) = I + \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -\Delta x_t \sin\theta_{t-1} & 0 & 0 \\ \Delta x_t \cos\theta_{t-1} & 0 & 0 \end{bmatrix} & 0_{3\times 2n} \\ 0_{2n\times 3} & 0_{2n\times 2n} \end{bmatrix} \tag{9}$$

2. $\Delta\theta_t \neq 0$:

$$A_t = g'(\xi_{t-1}, u_t) = I + \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -\frac{\Delta x_t}{\Delta\theta_t}\cos\theta_{t-1} + \frac{\Delta x_t}{\Delta\theta_t}\cos(\theta_{t-1}+\Delta\theta_t) & 0 & 0 \\ -\frac{\Delta x_t}{\Delta\theta_t}\sin\theta_{t-1} + \frac{\Delta x_t}{\Delta\theta_t}\sin(\theta_{t-1}+\Delta\theta_t) & 0 & 0 \end{bmatrix} & 0_{3\times 2n} \\ 0_{2n\times 3} & 0_{2n\times 2n} \end{bmatrix} \tag{10}$$

[1] Remember, floating point arithmetic is inexact so in practice you want to check for small rotational velocity rather than expected it to be identically 0

2 The Measurement Model

Let r_j be the distance to landmark j (this is the range measurement). Let ϕ_j be the relative bearing of landmark j (this is the bearing measurement).

Relative cartesian $\bar{x}, \bar{y}$ measurements can be transformed into range bearing measurements:

$$r_j = \sqrt{\bar{x}^2 + \bar{y}^2} \tag{11}$$

$$\phi_j = \text{atan2}(\bar{y}, \bar{x}) \tag{12}$$

(The laser scanner gives range-bearing measurements but the landmark detection algorithm provides relative $\bar{x}, \bar{y}$ measurements). Either can be used but we express the equations in terms of range-bearing here because noise is more naturally expressed in these coordinates and it is more generally applicable. You could follow similar procedures to derive the equations directly for relative x, y measurements however (in which case the measurement model would be linear).

The measurement model relates the system states to the measurements. The measurement for range and bearing to landmark j is:

$$z_j(t) = h_j(\xi_t) + v_t, \tag{13}$$

where

$$h_j(\xi_t) = \begin{bmatrix} r_j \\ \phi_j \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{x,j} - x_t)^2 + (m_{y,j} - y_t)^2} \\ \text{atan2}(m_{y,j} - y_t, m_{x,j} - x_t) - \theta_t \end{bmatrix} \tag{14}$$

and $v_t \sim N(0, R)$ is the sensor noise.

We can approximate $h_j(\xi_t)$ using a Taylor expansion about the estimate $\hat{\xi}_t$:

$$h_j(\xi_t) \approx h_j(\hat{\xi}_t) + h_j'(\hat{\xi}_t)(\xi_t - \hat{\xi}_t). \tag{15}$$

Let the estimated relative x and y distances be given by

$$\delta_x = (\hat{m}_{x,j} - \hat{x}_t) \tag{16}$$

$$\delta_y = (\hat{m}_{y,j} - \hat{y}_t) \tag{17}$$

and let $d = \hat{\delta}_x^2 + \hat{\delta}_y^2$ (this is the estimated squared distance between the robot and landmark j at time t).

Then the derivative with respect to the state is the following block matrix:

$$H_j = h_j'(\xi_t) = \begin{bmatrix} \begin{bmatrix} 0 & \frac{-\delta_x}{\sqrt{d}} & \frac{-\delta_y}{\sqrt{d}} \\ -1 & \frac{\delta_y}{d} & \frac{-\delta_x}{d} \end{bmatrix} & \begin{bmatrix} 0_{1\times2(j-1)} \\ 0_{1\times2(j-1)} \end{bmatrix} & \begin{bmatrix} \frac{\delta_x}{\sqrt{d}} & \frac{\delta_y}{\sqrt{d}} \\ -\frac{\delta_y}{d} & \frac{\delta_x}{d} \end{bmatrix} & \begin{bmatrix} 0_{1\times(2n-2j)} \\ 0_{1\times(2n-2j)} \end{bmatrix} \end{bmatrix} \tag{18}$$

The matrix $H_j \in \mathbb{R}^{2\times(3+2n)}$, and has zeros corresponding to the landmark states that have not been measured.

3 Extended Kalman Filter Slam

At each timestep t, Extended Kalman filter SLAM takes odometry u_t and sensor measurements z_i and generates an estimate of the full state vector $\hat{\xi}_t$.

3.1 Initialization

We start out with some guess as to the robot state (usually $(0,0,0)$) and covariance:

$$\Sigma_0 = \begin{bmatrix} \Sigma_{0,q} & 0_{3\times 2n} \\ 0_{2n\times 3} & \Sigma_{0,m} \end{bmatrix} \tag{19}$$

Often, the initial guess covariance $\Sigma_{q,0} \in \mathbb{R}^{3\times 3}$ is initialized to zero (indicating that we are certain that the robot is at its initial position. $\Sigma_{0,m} \in \mathbb{R}^{2n\times 2n}$ is diagonal, with infinity (or very high numbers) on its diagonal, indicating that the robot does not yet know about any landmarks.

3.2 Prediction

First, update the estimate using the model:

$$\hat{\xi}_t^- = g(\hat{\xi}_{t-1}, u_t, 0) \tag{20}$$

Next, propagate the uncertainty using the linearized state transition model:

$$\hat{\Sigma}_t^- = A_t \hat{\Sigma}_{t-1} A_t^T + \bar{Q}, \tag{21}$$

Where

$$\bar{Q} = \begin{bmatrix} Q & 0_{3\times 2n} \\ 0_{2n\times 3} & 0_{2n\times 2n} \end{bmatrix} \tag{22}$$

is the process noise for the robot motion model, expanding to fill the whole state. Notice that moving does not modify the robot's knowledge of the landmark locations.

3.3 Update

A separate sensor measurement step is completed for each landmark that has been observed.[2].

There are practical steps that must occur, beyond what happens in an ordinary EKF, prior to incorporating the measurements.

1. Data association: Each incoming measurement must be associated with an existing landmark state. If a measurement does not correspond to a previously seen landmark, a new landmark is added/initialized. For now, assume that we know the correspondence between measurements and states. That is, we know that measurement i corresponds to landmark j.

2. Landmark Initialization: when a new landmark is encountered it must be added to the state vector and initialized. If the measurement model is invertible: then you invert the measurement model to get the state; otherwise you may need to accumulate multiple measurements to determine

[2]Technically all landmarks could be updated in a single step but this is both harder to implement and, due to the sparsity of the matrices involved, introduces many unnecessary computations

the initial landmark postion. In the range-bearing case, the measurement model is invertable so landmark j can be initialized as

$$\hat{m}_{x,j} = \hat{x}_t + r_j \cos(\phi_j + \hat{\theta}_t) \tag{23}$$

$$\hat{m}_{y,j} = \hat{y}_t + r_j \sin(\phi_j + \hat{\theta}_t) \tag{24}$$

Once the landmarks are initialized, we incorporate each landmark measurement one-by-one (even if we've received multiple landmark measurements in the same time-step). Since measurements improve our estimate, our linearized model depends on the estimate, and our measurement of landmark i is not affected by the states of the other landmarks, it makes sense to incrementally incorporate each measurement.

If we have M measurements, then $z \in \mathbb{R}^{2M}$ (because measuring a landmark provides range and bearing in this example). The range/bearing of landmark j is then $z_t^i \in \mathbb{R}^2$.

3.3.1 For each measurement i

Let j be the landmark associated with measurement i. There can be multiple measurements of the same landmark, but the idea is that we know the correspondence between the measurements and the landmarks.

1. Compute the theoretical measurement, given the current state estimate:

$$\hat{z}_t^i = h_j(\hat{\xi}_t^-) \tag{25}$$

2. Compute the Kalman gain from the linearized measurement model:

$$K_i = \Sigma_t^- H_i^T (H_i \Sigma_t^- H_i^T + R)^{-1} \tag{26}$$

3. Compute the posterior state update

$$\hat{\xi}_t = \hat{\xi}_t^- + K(z_t^i - \hat{z}_t^i) \tag{27}$$

4. Compute the posterior covariance

$$\Sigma_t = (I - K_i H_i)\Sigma_t^- \tag{28}$$

When we incorporate the next measurement $(i+1)$ we use the updated state and covariance $(\Sigma_t, \hat{\xi}_t)$ for Σ_t^- and $\hat{\xi}_t^-$.

4 Practical Tips

1. When subtracting angles, always make sure to normalize to between $-\pi$ and π. This ensures that you always get the shortest angular distance. Otherwise you can get discontinuities.

Further reading: [1], [2]

References

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.