10

Simultaneous Localization and Mapping

★ Introduction

⇒ Also known as Concurrent Mapping and Localization, or CML.

⇒ In SLAM, the robot acquires a map of its environment while simultaneously localizing itself relative to this map.

⇒ From a probabilistic perspective, there are two main forms of the SLAM problem, which are both of equal practical importance.

- online SLAM
  - It involves estimating the posterior over the momentary pose along with the map

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

- full SLAM problem
  - Calculate a posterior over the entire path x_{1:t} along with the map, instead of just the current pose x_t

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

⇒ In particular, the online SLAM problem is the result of integrating out past poses from the full SLAM problem:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int\cdots\int p(x_{1:t}, m \mid z_{1:t}, u_{1:t})\, dx_1\, dx_2 \ldots\, dx_{t-1}$$

{ From Law of total Probability }

⇒ SLAM problems possess a continuous and a discrete component.

- The continuous estimation problem pertains to
  1. Location of the objects in the map
     (Objects may be landmarks in feature-based representation)
  2. Robot's own pose variables

- The discrete nature has to do with correspondence:
  - When an object is detected, a SLAM algorithm must reason about the relation of this object to previously detected objects.

{ Either the object is the same as a previously detected one, or it is not. }

⇒ At times, it will be useful to make the correspondence variables explicit:

- The online SLAM posterior is then given by

$$p(x_t, m, c_t \mid z_{1:t}, u_{1:t})$$

- and the full SLAM posterior by

$$p(x_{1:t}, m, c_{1:t} \mid z_{1:t}, u_{1:t})$$

.

- If we denote the feature extractor as a function f , the features extracted from a range measurement are given by f(z_t).
  - Most feature extractors extract a small number of features from high-dimensional sensor measurements
    - A key advantage of this approach is the enormous reduction of computational complexity

- Features correspond to distinct objects in the physical world.
- In robotics, it is common to call those physical objects landmarks
- The most common model for processing landmarks assumes that the sensor can measure the range and the bearing of the landmark relative to the robot's local coordinate frame.
- In addition, the feature extractor may generate a signature.
- If we denote the range by r, the bearing by φ, and the signature by s, the feature vector is given by a collection of triplets

$$f(z_t) \quad = \quad \{f_t^1, f_t^2, \ldots\} \quad = \quad \{\begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^1 \\ s_t^2 \end{pmatrix}, \ldots\}$$

- The number of features identified at each time step is variable
- Maps consist of list of features, m = {m_1 , m_2 , . . .}
  - Each feature may possess a signature and a location coordinate
- We need to define a variable that establishes correspondence between the feature f_t^i and the landmark m_j in the map.
- This variable will be denoted by c^i_t with c^i_t ∈ {1, . . . , N + 1}; N is the number of landmarks in the map m.
  - If c^i_t = j ≤ N , then the i-th feature observed at time t corresponds to the j-th landmark in the map.
  - When c^i_t = N + 1: Here a feature observation does not correspond to any feature in the map m.

- The online posterior is obtained from the full posterior by integrating out past robot poses and summing over all past correspondences:

$$p(x_t, m, c_t \mid z_{1:t}, u_{1:t}) \\ = \int \int \cdots \int \sum_{c_1} \sum_{c_2} \cdots \sum_{c_{t-1}} p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \, dx_1 \, dx_2 \ldots \, dx_{t-1}$$

- In practice, calculating a full posterior is usually infeasible. Problems arise from two sources:
  - The high dimensionality of the continuous parameter space
  - The large number of discrete correspondence variables

★ SLAM with EKF (Known Correspondence)

⇒ In a nutshell, the EKF SLAM algorithm applies the EKF to online SLAM using maximum likelihood data association.

⇒ EKF SLAM is subject to a number of approximations and limiting assumptions:

1. Feature-based maps.
2. Gaussian noise.
3. Positive measurements.

⇒ The SLAM algorithm for the case with known correspondence addresses the continuous portion of the SLAM problem only.

⇒ For convenience, let us call the state vector comprising robot pose and the map the combined state vector, and denote this vector y_t .

$$\begin{aligned} y_t &= \begin{pmatrix} x_t \\ m \end{pmatrix} \qquad (10.7) \\ &= (\ x\ \ y\ \ \theta\quad m_{1,x}\ m_{1,y}\ s_1\quad m_{2,x}\ m_{2,y}\ s_2\ \ \ldots\ \ m_{N,x}\ m_{N,y}\ s_N\ )^T \end{aligned}$$

⇒ Here x, y, and θ denote the robot's coordinates at time t, m_{i,x} , m_{i,y} are the coordinates of the i-th landmark, for i = 1, . . . , N , and s_i is its signature.

⇒ EKF SLAM calculates the online posterior

$$p(y_t \mid z_{1:t}, u_{1:t})$$

⇒ For algorithm shown below:

- → Lines 2 through 5 apply the motion update
- → Lines 6 through 20 incorporate the measurement update

.

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

$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & \underbrace{0 \cdots 0}_{2N} \end{pmatrix}$$

$$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

$$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & \frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

$$\bar{\Sigma}_t = G_t \ \Sigma_{t-1} \ G_t^T + F_x^T \ R_t \ F_x$$

$$Q_t = \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}$$

for all observed features $z_t^i = (r_t^i \ \ \phi_t^i \ \ s_t^i)^T$ do

$j = c_t^i$

if landmark j never seen before

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{pmatrix} + r_t^i \begin{pmatrix} \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \\ 0 \end{pmatrix}$$

endif

$$\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$$

$$q = \delta^T \delta$$

$$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{j,s} \end{pmatrix}$$

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & \underbrace{0 \cdots 0}_{2j-2} & 0 & 0 & 1 & \underbrace{0 \cdots 0}_{2N-2j} \end{pmatrix}$$

$$H_t^i = \frac{1}{q} \begin{pmatrix} \sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & -\sqrt{q}\delta_x & \sqrt{q}\delta_y & 0 \\ \delta_y & \delta_x & -1 & -\delta_y & -\delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} F_{x,j}$$

$$K_t^i = \bar{\Sigma}_t \ H_t^{iT} (H_t^i \ \bar{\Sigma}_t \ H_t^{iT} + Q_t)^{-1}$$

endfor

$$\mu_t = \bar{\mu}_t + \sum_i K_t^i (z_t^i - \hat{z}_t^i)$$

$$\Sigma_t = (I - \sum_i K_t^i \ H_t^i) \ \bar{\Sigma}_t$$

return μ_t, Σ_t