RLML Derivation

January 7, 2025

overview

- front-end model is to find the robot pose at current timestamp t
- back-end model is to optimize over all robot keyframe poses and all reflector positions

front-end model

- Notation
  - X_t denotes the robot pose at time t, which is (x, y, θ)
  - u_t denotes the control input at time t, which is (v, ω)(linea speed, and angular speed)
  - Z_t^R is the reflective marker measurement at time t, which is a vector of 2 dims, (x, y), denotes the position of marker in the robot frame
  - m^R denotes the reflective marker position in the map
  - Z_t^s is the scan measurement at time t, which is a number p denotes the probability of the laser scan touched an object at its end
  - m^s is the submap constructed from several previous scans
- Given
  - X_{t-1} is the robot pose at time $t-1$
  - u_t can be obtained by robot odometry and imu
  - Z_t^R can be obtained from measurement
  - m^R can be obtained from reflector position in submap
  - Z_t^s can be obtained from measurement
  - m^s can be obtained from submap(submap is constructed from several previous scans)

front-end model

$$
\begin{cases}
X_t = f(X_{t-1}, u_t) \\
Z_t^R = g(X_t, m^R) \\
Z_t^s = h(X_t, m^s)
\end{cases}
\tag{1}
$$

derive the loss

$$
\begin{cases}
e_o = X_t - f(X_{t-1}, u_t) \\
e_R = Z_t^R - g(X_t, m^R) \\
e_s = Z_t^s - h(X_t, m^s)
\end{cases}
\tag{2}
$$

we want to find the $\hat{X}_t$ which minimize the following cost:

$$
\arg\min_{X_t} (e_o^T \Sigma_o^{-1} e_o + e_R^T \Sigma_R^{-1} e_R + e_s^T \Sigma_s^{-1} e_s)
\tag{3}
$$

front-end model

next, we show how to propagate covariance Σ_o, first we can linearize the motion prediction equation at $\hat{X_{t-1}}$ and $u_t = 0$

$$X_t = f(\hat{X_{t-1}}) + \frac{\delta f}{\delta X_{t-1}}(X_{t-1} - \hat{X_{t-1}}) + \frac{\delta f}{\delta u_t} u_t \quad (4)$$

let G_x denotes $\frac{\delta f}{\delta X_{t-1}}$ and G_u denotes $\frac{\delta f}{\delta u_t}$

$$\Sigma_o = G_x \Sigma_{\hat{X_{t-1}}} G_x^T + G_u \Sigma_u G_u^T \quad (5)$$

where $\Sigma_{\hat{X_{t-1}}}$ is the covariance of $\hat{X_{t-1}}$ and Σ_u is the covariance of control input u_t, Σ_u can be empirically denoted as:

$$\Sigma_u = \begin{bmatrix} (a * v + b)^2 & 0 \\ 0 & (c * \omega + d)^2 \end{bmatrix} \quad (6)$$

a, b, c, d are hyper parameters which can be experimentally measured and can also be tuned, it denotes how precise are the speed and angular speed

front-end model

next, we show the explicit function of motion prediction and calculate the derivative. $X_t = f(X_{t-1}, u_t)$ can be written as:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t-1} + \begin{bmatrix} v\Delta t \cos(\theta + \omega\Delta t/2) \\ v\Delta t \sin(\theta + \omega\Delta t/2) \\ \omega\Delta t \end{bmatrix} \tag{7}$$

so G_x is:

$$G_x = \begin{bmatrix} 1 & 0 & -v\Delta t \sin(\theta + \omega\Delta t/2) \\ 0 & 1 & v\Delta t \cos(\theta + \omega\Delta t/2) \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

and G_u is:

$$G_u = \begin{bmatrix} \Delta t \cos(\theta + \omega\Delta t/2) & \frac{-v(\Delta t)^2 \sin(\theta+\omega\Delta t/2)}{2} \\ \Delta t \sin(\theta + \omega\Delta t/2) & \frac{v(\Delta t)^2 \cos(\theta+\omega\Delta t/2)}{2} \\ 0 & \Delta t \end{bmatrix} \tag{9}$$

front-end model

Σ_R and Σ_s are trivial and are all selected empirically, for example, reflective marker's observation are usually within 0.025m, so Σ_R can be

$$\begin{bmatrix} (0.025 * 0.025)^2 & 0 \\ 0 & (0.025 * 0.025)^2 \end{bmatrix} \tag{10}$$

as we need to estimate the covariance at $\hat{X}_t$, so we can linearize all the equation at $\hat{X}_t$, and put into covariance formula of linear least square:

$$\Sigma_{\hat{X}_t} = (J_o^T \Sigma_o^{-1} J_o + J_R^T \Sigma_R^{-1} J_R + J_s^T \Sigma_s^{-1} J_s)^{-1} \tag{11}$$

where $J_o = \frac{\delta e_o}{\delta X_t}$, $J_R = \frac{\delta e_R}{\delta X_t}$ and $J_s = \frac{\delta e_s}{\delta X_t}$
this is similar to kalman filter, first use $\Sigma_{X_{t-1}^{\hat{}}}$ propagate to Σ_o, and after optimization, we calculate covariance and get $\Sigma_{\hat{X}_t}$

back-end model

- Notation
  - X_i denotes the robot pose of keyframe i, which is (x, y, θ)
  - ε is the union of all the keyframe poses
  - Z_{ik}^{R} is the reflective marker measurement between keyframe i and reflector k
  - $\Delta\varepsilon_{ij}^{P}$ is the relative pose constraint between keyframe i and keyframe j
  - $\Delta\varepsilon_{ij}^{r}$ is the loop closure pose constraint between keyframe i and keyframe j

back-end model

$$\begin{cases} e_{ij}^P = (X_i - X_j) - \Delta\varepsilon_{ij}^P \\ e_{ij}^r = (X_i - X_j) - \Delta\varepsilon_{ij}^r \\ e_{ik}^R = g(X_i, m_k^R) - Z_{ik}^R \end{cases} \tag{12}$$

we want to find the $X_i, i = 1, 2, 3, ..., N$ and $m_k^R, k = 1, 2, 3, ..., M$ which minimize the following cost:

$$\arg\min({e_{ij}^P}^T {\Sigma_{ij}^P}^{-1} e_{ij}^P + {e_{ij}^r}^T {\Sigma_{ij}^r}^{-1} e_{ij}^r + {e_{ik}^R}^T {\Sigma_{ik}^R}^{-1} e_{ik}^R) \tag{13}$$

back-end model

for $\Delta\varepsilon_{ij}^{P} = (\hat{X}_i - \hat{X}_j)$, and $\hat{X}_i \sim N(\mu_i, \Sigma_i)$, this Σ_i is estimated through formula (11). so $\Delta\varepsilon_{ij}^{P} \sim N(\mu_i - \mu_j, \Sigma_i + \Sigma_j)$, and we can derive the covariance

$$\Sigma_{ij}^{P} = \Sigma_i + \Sigma_j \tag{14}$$

for Σ_{ij}^{r}, relocalization's covariance can be experimentally estimated. for Σ_{ik}^{R}, reflector measuring covariance is the same as Σ_R in front-end method.