Lecture 5 - EKF SLAM

SLAM Problem Definition

Given;

- robot's controls

$$u_{1:T} = \{u_1, u_2, \dots u_T\}$$

- observations

$$z_{1:T} = \{z_1, z_2, \dots z_T\}$$

Want;

- map of environment

$$m$$

- Path (Poses) of robot

$$x_{0:T} = \{x_0, x_1 \dots x_T\}$$

EKF SLAM - State Representation

- estimate robot's pose & locations of landmarks in environment
- Assumption: Known correspondences
- State space for 2D plane;

$$x_t = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{m_{1,x}, m_{1,y}}_{\text{landmark 1}}, \dots \underbrace{m_{n,x}, m_{n,y}}_{\text{landmark } n})^T$$

- map with n landmarks; $(3+2n)$-dimensional Gaussian
- Belief represented by;

$$\mu = \begin{bmatrix} x_R \\ m \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{bmatrix}$$

where $(x, y, \theta) \rightarrow x_R$
$(m_{1,x}, m_{1,y}) \rightarrow m$

↳ lets take this representation through the Kalman Filter

Covariance(A, A) = variance(A)

Covariance(A, B) ≡ Covariance(B, A)

$\Sigma^T = \Sigma$

Std deviation = $\sqrt{\text{variance}}$

- variance is avg of $(\text{std dev})^2$

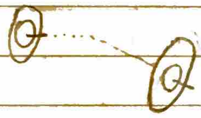
EKF SLAM - Filter Cycle

1) State Prediction

Correction Step

2) Measurement Prediction -- evaluating h @ predicted μ
3) Measurement
4) Data association
5) Update

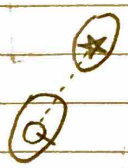
State Prediction



$\mu = \begin{bmatrix} x_R \\ m \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{bmatrix}$ → update

↳ linear time complexity in # of landmarks

Measurement Prediction



μ → no change yet

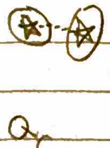
Σ → no change yet

Obtained Measurement

μ → no change yet

Σ → no change yet

Data Association & difference between h(x) & Z



μ → no change

Σ → no change

Update Step

$\mu = \begin{bmatrix} x_R \\ m \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{bmatrix}$

→ possibly update everything

→ n^2 complexity.

↳ expensive with large state space.

Concrete Example

Setup:

- Robot moves in 2D
- velocity-based motion model ←
- observes point landmarks (x, y)
- Range-bearing sensor
- Known data association
- Known # of landmarks

$$\underbrace{\begin{bmatrix}x'\\ y'\\ \theta'\end{bmatrix} = \begin{bmatrix}x\\ y\\ \theta\end{bmatrix} + \begin{bmatrix}-\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t)\\ \frac{v_t}{w_t}\cos\theta - \frac{v_t}{w_t}\cos(\theta + w_t\Delta t)\\ w_t\Delta t\end{bmatrix}}_{g_{x,y,\theta}(u_t,(x,y,\theta)^T)}$$

→ of EKF Algorithm

Initialization; - **Line 2. EKF Alg**: 2. $\bar{\mu}_t = g(u_t, \mu_{t-1})$

- Robot starts in it's own reference frame - all landmarks unknown
- 2N+3 dimensions

$$\mu_0 = [0\ 0\ 0 \dots 0]^T \qquad \Sigma_0 = \begin{bmatrix}0 & 0 & 0 & \dots & 0\\ 0 & 0 & \infty & \dots & 0\\ \vdots & \vdots & \vdots & & \vdots\\ 0 & 0 & 0 & \dots & \infty\end{bmatrix}$$

- how to map $g_{x,y,\theta}$ to the 2N+3 dimensional space? → only affect first 3 dimensions (not landmarks)

$$\underbrace{\begin{bmatrix}x'\\ y'\\ \theta'\\ \vdots\end{bmatrix} = \begin{bmatrix}x\\ y\\ \theta\\ \vdots\end{bmatrix} + \underbrace{\begin{bmatrix}1 & 0 & 0 & 0 & \dots & 0\\ 0 & 1 & 0 & 0 & \dots & 0\\ 0 & 0 & 1 & \underbrace{0 & \dots & 0}_{2N\ \text{cols}}\end{bmatrix}^T}_{F_x^T} \begin{bmatrix}-\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t)\\ \frac{v_t}{w_t}\cos\theta - \frac{v_t}{w_t}\cos(\theta + w_t\Delta t)\\ w_t\Delta t\end{bmatrix}}_{\boxed{g(u_t, x_t)}}$$

(pose update)

$\boxed{g(u_t, x_t)}$ → non-lin. func g

→ of EKF algorithm

Line 3: $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

↳ find this Jacobian

→ Jacobian of motion

$$G_t = \begin{bmatrix}G_t^x & 0\\ 0 & I\end{bmatrix}$$

↳ identity (2N × 2N)

→ this update only adjusts pose
- remember we need Jacobian in order to perform 1'st order taylor on non lin func g to linearize it. → KF assumes lin.

$$G_t^x = \frac{\partial g(u_t, x_t)}{\partial (x_t)}$$

$$G_t^x = \frac{\partial}{\partial (x,y,\theta)^T}\left(\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \underbrace{\begin{bmatrix} -\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ \frac{v_t}{w_t}\cos\theta - \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ w_t \Delta t \end{bmatrix}}_{} \right)$$

$$= I + \frac{\partial}{\partial (x,y,\theta)^T}\begin{bmatrix} \\ " \\ \\ \end{bmatrix}$$

$$= I + \begin{bmatrix} 0 & 0 & -\frac{v_t}{w_t}\cos\theta + \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ 0 & 0 & -\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ 0 & 0 & 0 \end{bmatrix}$$

→ no dependancies on x or y

$$= \begin{bmatrix} 1 & 0 & -\frac{v_t}{w_t}\cos\theta + \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ 0 & 1 & -\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ 0 & 0 & 1 \end{bmatrix} \rightarrow G_t = \begin{bmatrix} G_t^x & 0 \\ 0 & I \end{bmatrix}$$

↳ ✱ linearizes dependance of sin & cos of heading

line 3; $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

$$= \begin{bmatrix} G_t^x & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{bmatrix}\begin{bmatrix} (G_t^x)^T & 0 \\ 0 & 1 \end{bmatrix} + R_t$$

$$= \begin{bmatrix} G_t^x \Sigma_{xx} (G_t^x)^T & G_t^x \Sigma_{xm} \\ (G_t^x \Sigma_{xm})^T & \Sigma_{mm} \end{bmatrix} + R_t$$

→ large block not touched

↳ can see that only pose-related elements are updated

Prediction step completed ///

EKF SLAM Prediction Step - Alg

1. $\text{EKF\_SLAM\_Prediction}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, R_t)$:

2. $$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 \end{bmatrix}$$

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

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

5. $$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + \underbrace{F_x^T R_t^x F_x}_{R_t}$$

EKF SLAM - Correction Step

- Known data association → assumption
- $c_t^i = j$: i^{th} measurement at time t observes landmark j
- initialize landmark if un-observed
- Compute expected observations
- Compute Jacobian of h
- Proceed w/ computing Kalman gain K

Range - Bearing Observation

$$z_t^i = (r_t^i, \phi_t^i)^T$$

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

↳ predicted location of landmark ↳ estimate of robot's location ⟶ relative measurement

- Now compute predicted (expected) observ. → EKF compares predicted observ. with what you actually observe

Predicted / Expected Observation

$$\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{bmatrix}$$ → diff. between robot & landmark poses

$$q = \delta^T \delta$$ → squared Euclidean distance ($A^2 + B^2 = C^2$)

$$\hat{z}_t^i = \begin{bmatrix} \sqrt{q} \\ \operatorname{atan}(\delta_y, \delta_x) + \bar{\mu}_{t,\theta} \end{bmatrix} = h(\bar{\mu}_t)$$ → predicted observation. $= \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix}$

Jacobian of Observation

$${}^{low}H_t^i = \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} = \begin{bmatrix} \frac{d\sqrt{q}}{dx} & \frac{d\sqrt{q}}{dy} & \cdots \\ \frac{d\,\mathrm{atan2}(\ldots)}{dx} & \frac{d\,\mathrm{atan2}(\ldots)}{dy} & \cdots \end{bmatrix}$$ → 2 x 5

↳ low dim space, everything is 0 except $(x, y, \theta, m_{j,x}, m_{j,y})$, (only 1 landmark & robot pose)

$$= \frac{1}{q} \begin{bmatrix} -\sqrt{q}\,\delta_x & -\sqrt{q}\,\delta_y & 0 & \sqrt{q}\,\delta_x & \sqrt{q}\,\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix}$$ → only dimensions that matter here

Map to High-Dim Space ; $H_t^i = {}^{low}H_t^i F_{x,j}$

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

Can Now Do:

- Line 4.
- Line 5.
- Line 6.

} of EKF algorithm

↳ return μ_t, Σ_t

EKF SLAM - Correction

EKF_SLAM_Correction($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, R_t$)

$$Q_t = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix}$$

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

$j = c_t^i$

if j never seen before:

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

endif

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

this will be 0 if you observe landmark for first time since you init. it to the physical observ., then Kalman gain K gets canceled & you fall back to same predicted belief. (no update in mean estimate)

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

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

$$F_{x,j} = \begin{bmatrix} \langle \text{Prev Page} \rangle \end{bmatrix}$$

$$H_t^i = \frac{1}{q} \begin{bmatrix} \langle \text{Prev. Page} \rangle \end{bmatrix} F_{x,j}$$

$$\left.\begin{aligned} K_t^i &= \bar{\Sigma}_t H_t^{iT} (H_t^i \bar{\Sigma}_t H_t^{iT} + Q_t)^{-1} \\ \bar{\mu}_t &= \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^i) \\ \bar{\Sigma}_t &= (I - K_t^i H_t^i) \bar{\Sigma}_t \end{aligned}\right\} \text{Standard KF stuff}$$

endfor

$\mu_t = \bar{\mu}_t$

$\Sigma_t = \bar{\Sigma}_t$

return μ_t, Σ_t