SLAM.

Part 1. Introduction

1. Terminologies.

① Mapping: a model of environment. Assume robot position is know.

② state : position of robot, position of landmark, etc.

State estimation: from noisy sensor data, estimate the position

③ Localization: location of the robot (x. y. orientation).

④ SLAM: Simultaneous localization and mapping.

chicken-or-egg problem, we don't know localization or mapping. We must calculate them at the same time. They will help each other for estimation.

⑤ Odometry: commands sent to wheel. like go left, etc.

⑥ Data Association: having obtained sensor data, association is about which object our observation corresponds to.

2. SLAM problem.

Given $u_{1:T}$ control sequence. $z_{1:t}$ observation

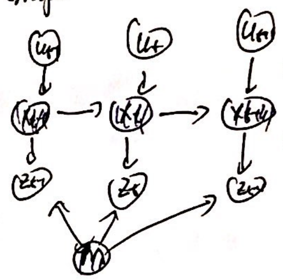
Want: mapping m, state: $x_{0:T}$

full SLam $p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$ All states

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

$$= \int_{x_0} \cdots \int_{x_{t-1}} p(x_{0:t}, m \mid z_{1:t}, u_{1:t})\, dx_{t-1} \cdots dx_0$$

The Graph



x and m are hidden.

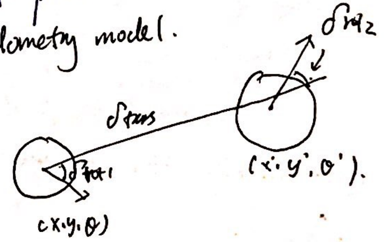
(i) mapping and state affects observation.

(ii) previous state and current command affect current state.

3. motion model $p(x_t \mid u_t, x_{t-1})$

motion model is to study how commands affect robot position.

Odometry model.



① state is characterized by (x, y, orientation).

② Command is characterized by (rotation, transfution, rotation)

③ relation.

$\delta_{rot1} = atan2(y'-y,\ x'-x) - \theta$

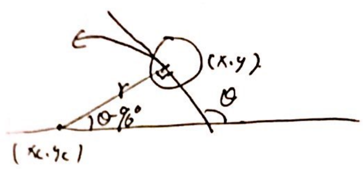
$\delta_{trans} = \sqrt{(x-x')^2 + (y-y')^2}$

$\delta_{rot2} = \theta' - \theta - \delta_{rot1}$

④ Command will not be perfected executed. we Assume $\delta_{rot1}, \delta_{trans}, \delta_{rot2}$ all to be Gaussian

$u \sim N(0, \Sigma)$

velocity Model.



① it assume in a small time, robot is doing circular move.

command $u = (v, \omega)^T$. $\omega = \frac{v}{r}$.

② relation

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta + \omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t + \underbrace{\gamma\Delta t}_{\text{account to final rotation}} \end{pmatrix}$$

4. Sensor model

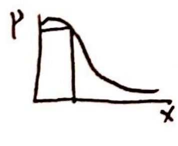
different sensors have different model. Here Assume laser ranger sensor,

① $Z_t = \{z_t^1 \cdots z_t^k\}$. K measurements.

$p(z_t | x_t, m) = \prod_{i=1}^{k} p(z_t^i | x_t, m)$ Assume independent.

② Beam Endpoint Model

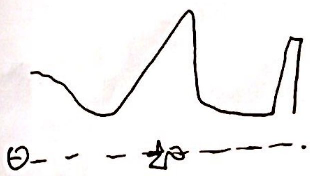
Just focus on beam endpoint.



Just convolve your obstacle a little bit, use that as a distribution.

Then just pick the x at the end of your laser beam.

③ Ray-Cast Model.



Gaussian over ~~obstacle~~ obstacle.
exponential decay : dynamic obstacle
peak : limitation of laser. maximum range of laser.
uniform : noise.

$z_t^i = (r_t^i, \phi_t^i)^T$ pose $x : (x, y, \theta)^T$

range orientation. maps : (m_{jx}, m_{jy})

Euclidean.

$$z_t^i = \begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix} + Q_t$$

noise.

robot pose affects observation.

Part II EKF SLAM

1. Idea

idea of EKF SLAM is to use the EKF algorithm to solve SLAM problem. So we treat state here as the robot pose + landmarks.

Quick reminder. Given $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

$$\begin{cases} \bar{\mu}_t = g(u_t, \mu_{t-1}) \\ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \end{cases}$$

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

$$\begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \\ \Sigma_t = (I - K_t H_t)\bar{\Sigma}_t \end{cases}$$

return μ_t, Σ_t.

2. Basic set up.

① State
$x_t = (x, y, \theta, m_{1x}, m_{1y}, \dots m_{nx}, m_{ny})^T$
2D plane, $3+2n$ dim space.

② Assumption:
data association is known. or else, there will be extra dimension.

③ $\mu = (x, y, \theta, m_{1x}, \cdots m_{ny})^T = (X, M)^T$.
the notation is a little sloppy. we use probability distribution to express our believe. and assume they are Gaussian.
Here instead of using $\mu_x, \mu_y \dots$ we use $x, y, \theta \dots$

④ $$\Sigma = \left(\begin{array}{ccc|ccc} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xm_{1x}} & \cdots & \\ \hline \sigma_{m_{1x}x} & \sigma_{m_{1x}\theta} & & \sigma_{m_{1x}m_{1x}} & \cdots & \\ \vdots & \vdots & & \vdots & & \end{array} \right)$$

$$= \left(\begin{array}{c|c} \Sigma_{xx} & \Sigma_{xm} \\ \hline \Sigma_{mx} & \Sigma_{mm} \end{array} \right).$$

$\underbrace{\Sigma_{xx} \text{ is } 3\times3}_{\text{small}}$. Σ_{xm} is $3\times n$ Σ_{mx} is $n\times3$. Σ_{mm} is $n\times n$.
most expensive!

⑤ workflow.
state prediction → measurement prediction → $\underbrace{\text{obtain measurement} \rightarrow \text{data association} \rightarrow \text{update}}_{\text{correction}}$
changes Σ_{xx}, Σ_{xm}, Σ_{mx}
complexity n^2. Change Σ

Assumption. robot does not change environment m.

⑥ we use velocity motion model and range-bearing sensor model.

3. Initialization
At start time, $x=0, y=0, \theta=0$. $\Sigma_{xx}=0$. All landmarks are unknown. so initialize with 0.

$\mu = (0, \dots 0)^T$ $\Sigma_0 = \begin{pmatrix} 0 & 0 & \\ 0 & \infty & \\ & & \ddots \, \infty \end{pmatrix}$

uncertainty is ∞, since m_i could be any value!

4. Prediction.

① $\bar{\mu}_t = g(u_t, \mu_{t-1})$

Recall $\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\Delta t \end{pmatrix}$

now $\mu_s = \begin{pmatrix} x \\ y \\ \theta \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ we have

$$\begin{pmatrix} x' \\ y' \\ \theta' \\ \vdots \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \\ \vdots \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\Delta t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

dimension lifting.
denote $\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$. we want to lift 3-dim vector into $2N+3$ dim

$\begin{pmatrix} u_1 \\ \vdots \\ u_3 \end{pmatrix} \rightarrow \begin{pmatrix} u_1 \\ \vdots \\ 0 \end{pmatrix}$. $F_x = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \cdots \\ 0 & 1 & 0 & \vdots \\ 0 & 0 & 1 & \cdots 0 \end{array} \right)$

$F_x^T \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$ will do the lifting.

② $g(u_t, \mu_{t-1}) = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ \frac{v_t}{w_t}\cos\theta - \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ w_t\Delta t \end{pmatrix}$

$$G_t^x = \frac{\partial}{\partial (x, y, \theta)^T} g(u_t, \mu_{t-1})$$

$$= I + \underbrace{\begin{pmatrix} 0 & 0 & -\frac{v_t}{w_t}\cos\theta + \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ 0 & 0 & -\frac{v_t}{w_t}\sin\theta + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ 0 & 0 & 0 \end{pmatrix}}_{3\times 3}$$

dim lift : $G_t = \begin{pmatrix} G_t^x & 0 \\ 0 & \underbrace{I}_{2N\times 2N} \end{pmatrix}$

$$R_t = F_x^T \underbrace{R_t^x}_{3\times 3} F_x$$

lifted!

$$\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$$

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

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

updated

5 correction.

① initialize landmarks if no previous information.

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \underbrace{\begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix}}_{\text{estimated robot location}} + \underbrace{\begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \end{pmatrix}}_{\text{observation}}$$

estimated robot location + observation $\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = z_t^i$.

② measurement prediction

now we assume we know robot location and landmark position, we want to predict what we will see. $\hat{z}_t^i$

$$\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \quad q = \delta^T \delta \quad \text{Euclidean..}$$

$$\therefore \hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \underbrace{\bar{\mu}_{t,\theta}}_{\text{orientation of robot!}} \end{pmatrix} = h(\bar{\mu}_t)$$

$${}^{low}H_t^i = \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} = \begin{pmatrix} \frac{\partial\sqrt{q}}{\partial x} & \frac{\partial\sqrt{q}}{\partial y} & \frac{\partial\sqrt{q}}{\partial \theta} & \frac{\partial\sqrt{q}}{\partial m_{j,x}} & \frac{\partial\sqrt{q}}{\partial m_{j,y}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

we assume state vector here contains only non-zero entries. $(x, y, \theta, m_{j,x}, m_{j,y})$

$$= \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix}$$

in reality μ_t is a high dim vector

$(x, y, \theta, m_{1,x}, m_{1,y}, \ldots, m_{j,x}, m_{j,y}, \ldots, m_{N,x}, m_{N,y})$.

so H_t^i looks like

$$\begin{pmatrix} \frac{\partial\sqrt{q}}{\partial x} & \frac{\partial\sqrt{q}}{\partial y} & \frac{\partial\sqrt{q}}{\partial \theta} & 0 \cdots 0 & \frac{\partial\sqrt{q}}{\partial m_{j,x}} & \frac{\partial\sqrt{q}}{\partial m_{j,y}} & 0 \cdots 0 \\ - & - & - & - & - & - & - \end{pmatrix}$$

$$\underset{2\times 5}{{}^{low}H_t^i} \longrightarrow \underset{2\times(2N+3)}{H_t^i}$$

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

$$H_t^i = {}^{low}H_t^i \cdot F_{x,j}.$$

note this is w.r.t the i^{th} observation.

③ $k_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + k_t (z_t - h(\bar{\mu}_t))$ } correct ~~update~~ measurement and update state.

$\Sigma_t = (I - k_t H_t) \bar{\Sigma}_t$

So far this is the EKF SLAM.

$\mu_t, \mu_{t-1}, \Sigma_{t-1} \rightarrow \bar{\mu}_t, \bar{\Sigma}_t, z_t \rightarrow k_t, \mu_t, \Sigma_t$

One observation is that covariance matrix is usually dense while information matrix is "sparser"

off-diagonal elements are near zero but not zero. so Ω is not sparse.

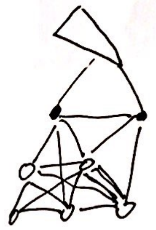
This gives us the idea of extended information filter to do slam but if we can try to sparsify the information matrix, we get const time computation complexity and hopefully get a not-so-bad approximation.

Part III SEIF SLAM.

1. A sparse matrix is one that has finite number of non-zero off-diagonal entries independent of matrix size.

Inversion of a matrix is quadratic operation on dense matrix, but on sparse matrix, it is of constant time

2. Information matrix is "near" sparse. Most off-diagonals are close to zero. Information matrix can be interpreted as a graph of links between nodes.

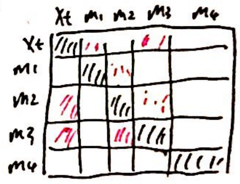


node: robot and landmarks

link: connection/correlation

missing links: independence.

Information matrix

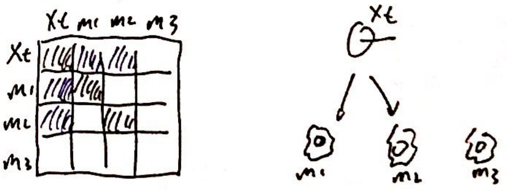


diagonals are strong correlation

off-diagonals characterise strength of links.

So most off-diagonal entries are close to 0. this means most landmark are independent and robot only sees limited number of landmarks at given time. This intuition makes sense

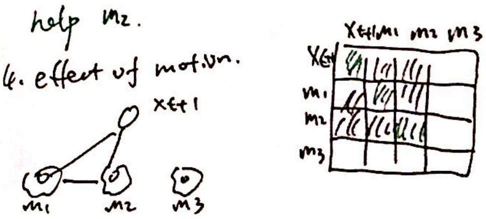
3. Effect of measurement on information matrix.



robot observes m_1, m_2. but the landmarks are independent.

Intuition.. If I know robot pose, then knowing m_1 does not help m_2.

4. effect of motion.



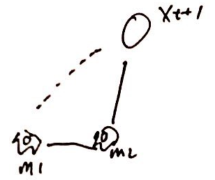
- landmarks become correlated on motion update

Because now x_{t+1} is considered uncertain and needs prediction.

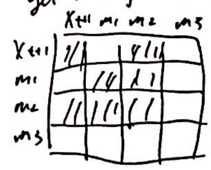
So knowing m_1 will tell something about m_2. Since x_{t+1} is uncertain

5. Sparsification step.

As x_t moves on, information will fill in Ω. This is not what we want. Let's "sparsify" the correlation.



get rid of $x_{t+1} \to m_1$



this will result in sparsity in matrix.

Intuition: ~~[illegible]~~ robot only observes a certain number of landmarks.

We define active landmarks: currently observed ones + some more (small).

passive landmarks: all others (large)

$\Rightarrow$ { robot pose is linked to active landmarks only
landmarks only have links to nearby landmarks (landmarks that have been active at the same time). m_1 and m_2.

Next we go to the algorithm details.

1. $\Sigma \sim \Omega^{-1}$ $\mu = \Omega^{-1}\xi$. $\Omega = \Sigma^{-1}$ $\xi = \Sigma^{-1}\mu$.

$$(R + PQP^T)^{-1} = R^{-1} - R^{-1}P(Q^{-1} + P^TR^{-1}P)^{-1}P^TR^{-1}.$$

2. The SEIF SLAM Algorithm

① $\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t$ = motion update($\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t$)

② ξ_t, Ω_t = measurement update($\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t$).

③ ~~update state estimation~~ μ_t = update_state-estimate($\xi_t, \Omega_t, \bar{\mu}_t$).

④ $\tilde{\xi}_t, \tilde{\Omega}_t$ = sparsification(ξ_t, Ω_t, μ_t)

Remark:

(i) motion update, measurement update is what we have in EKF-SLAM. Here we also need to maintain $\bar{\mu}_t$, since in some terms in EIF, $\bar{\mu}_t$ is needed.

(ii) Generally we go from $\mu_{t-1} \to \bar{\mu}_t \to \mu_t$. But to go from $\bar{\mu}_t$ to μ_t, that's an estimation like ξ_t, Ω_t. but needs information from canonical form. So step ③ is an extremely difficult step.

(iii) Our goal is to get $\tilde{\xi}_t$, $\tilde{\Omega}_t$ which are sparse. We start to assume Ω_{t-1} is sparse.

3. prediction step. $(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}; u_t) \to \bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t$

① Recall EKF SLAM.

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

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

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

Remark:

F_x, $\bar{\mu}_t$, G_t are const operation. $\bar{\mu}_t$ is done!

next. we need Φ_t. $\bar{\Omega}_t$.

① notation.

$$\delta = \begin{pmatrix} -\frac{v_t}{w_t}\sin\mu_{t-1,\theta} + \frac{v_t}{w_t}\sin(\mu_{t-1,\theta} + w_t\Delta t) \\ \frac{v_t}{w_t}\cos\mu_{t-1,\theta} - \frac{v_t}{w_t}\cos(\mu_{t-1,\theta} + w_t\Delta t) \\ w_t\Delta t \end{pmatrix}$$

$$\Delta = \begin{pmatrix} 0 & 0 & \frac{-v_t}{w_t}\cos\mu_{t-1,\theta} - \frac{v_t}{w_t}\cos(\mu_{t-1,\theta} + w_t\Delta t) \\ 0 & 0 & \frac{v_t}{w_t}\sin\mu_{t-1,\theta} - \frac{v_t}{w_t}\sin(\mu_{t-1,\theta} + w_t\Delta t) \\ 0 & 0 & 0 \end{pmatrix}$$

$$\bar{\Omega}_t = \bar{\Sigma}_t^{-1} = (G_t\Omega_{t-1}^{-1}G_t^T + R_t)^{-1} \equiv (\Phi_t^{-1} + R_t)^{-1}.$$

$$\Phi_t \equiv (G_t\Omega_{t-1}^{-1}G_t^T)^{-1} = (G_t^T)^{-1}\Omega_{t-1}G_t^{-1}.$$

③ now.

$$\bar{\Omega}_t = (\Phi_t^{-1} + F_x^T R_t^x F_x)^{-1}$$

$$= \Phi_t - \Phi_t \underbrace{F_x^T}_{\text{sparse}} \underbrace{(R_t^{x-1} + F_x\Phi_t F_x^T)^{-1}}_{3\times3 \text{ inversion}} \underbrace{F_x}_{\text{sparse.}} \Phi_t.$$

now we only need Φ_t to be sparse

$\equiv \Phi_t - k_t$

④ now $\Phi_t = (G_t^T)^{-1}\Omega_{t-1}G_t^{-1}$. since we've got Ω_{t-1} from previous step as sparse. problem is inverse of G_t.

$$G_t^{-1} = (I + F_x^T\Delta F_x)^{-1} = \begin{pmatrix} \Delta + I_3 & 0 \\ 0 & I_{2N} \end{pmatrix}^{-1}.$$

$$= \begin{pmatrix} (\Delta + I_3)^{-1} & 0 \\ 0 & I_{2N} \end{pmatrix}$$

inversion of 3x3 matrix.

$$= I_{3+2N} + \begin{pmatrix} (\Delta + I_3)^{-1} - I_3 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= I + \underbrace{F_x^T ((I+\Delta)^{-1} - I) F_x}_{\Psi_t}$$

$$= I + \Psi_t.$$

$\rightarrow \Phi_t$ is const time computation.

$\rightarrow \bar{\Omega}_t$ is const time computation.

we write the computation out.

$$\Phi_t = (I + \Psi_t^T)\Omega_{t-1}(I + \Psi_t)$$

$$= \Omega_{t-1} + \Psi_t^T\Omega_{t-1} + \Omega_{t-1}\Psi_t + \Psi_t^T\Omega_{t-1}\Psi_t.$$

$$\equiv \Omega_{t-1} + \lambda_t$$

So the computation flow is.

$$\begin{matrix} F_x \\ \delta \\ \Delta \end{matrix} \rightarrow \Psi_t \rightarrow \lambda_t \rightarrow \Phi_t \rightarrow k_t \rightarrow \bar{\Omega}_t$$

⑤ Finally. $\bar{\xi} = \bar{\Omega}_t \bar{\mu}_t$ "const time."

explicitly $\qquad \bar{\Omega}_t \downarrow$ const $\qquad \bar{\mu}_t \downarrow$ const

$$\bar{\xi}_t = \bar{\Omega}_t (\mu_{t-1} + F_x^T \delta_t)$$

$$= \bar{\Omega}_t (\Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t).$$

$$= \bar{\Omega}_t \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t.$$

$$= (\underbrace{\bar{\Omega}_t - \Phi_t}_{-k_t} + \underbrace{\Phi_t - \Omega_{t-1}}_{\lambda_t} + \Omega_{t-1}) \underbrace{\Omega_{t-1}^{-1} \xi_{t-1}}_{\mu_{t-1}} + \bar{\Omega}_t F_x^T \delta_t.$$

$$= \xi_{t-1} + (\lambda_t - k_t) \mu_{t-1} + \underbrace{\bar{\Omega}_t}_{\text{sparse}} \underbrace{F_x^T \delta_t}_{3\times 3}$$

4. measurement update.

① Recall from EKF slam.

$Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$ $z_t^i = \begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix}$. $j = c_t^i$ data associate.

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

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

② $H_t^i = \frac{1}{q} \left(\begin{array}{ccccccc} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \boxed{\begin{matrix} 0 \cdots 0 \\ 0 \cdots 0 \end{matrix}} & \sqrt{q}\delta_x & \sqrt{q}\delta_y & \boxed{0} \\ \delta_y & -\delta_x & -q & & -\delta_y & \delta_x & \\ & & & 2j-2 & & & 2N-2j \end{array} \right)$

③ $\left.\begin{aligned} \xi_t &= \bar{\xi}_t + \sum_i H_t^{iT} Q_t^{-1} (z_t^i - \hat{z}_t^i + H_t^i \mu_t) \\ \Omega_t &= \bar{\Omega}_t + \sum_i H_t^{iT} Q_t^{-1} H_t^i \end{aligned}\right\}$ from EIF Filter.

inversion is 2x2 matrix. H_t^i is sparse. Everything is const time. but Ω_t is not sparse anymore! We have $\sum_i$ here. H_t^i have different sparse area.

5. mean update.

① Recall from EKF. $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$.

Kalman Gain ~~[illegible]~~ only be computed using Σ_t. in canonical form. So, this is complicated.

② Another observation.

$\mu_t = \Omega_t^{-1} \xi_t$. but Ω_t is not sparse.

we only have Ω_{t-1} as sparse.

so μ_t is costly..

③. idea. $\hat{\mu} = \arg\max p(\mu) = \arg\max_{\mu} \exp(-\frac{1}{2}\mu^T \Omega \mu + \xi^T \mu)$

use estimation.

In general this is not const time. But if we are only interested in robot pose and active landmarks, ~~[illegible]~~ things gets easier.

6. sparsification. $(\xi_t, \Omega_t, \mu_t) \rightarrow \tilde{\xi}_t, \tilde{\Omega}_t$

① Sparsification in general.

$p(a,b,c)$ we ~~[illegible]~~ make an ~~[illegible]~~ assumption of p, where given c, a and b are independent.

$$\begin{cases} \tilde{p}(a|b,c) = \tilde{p}(a|c). \\ \tilde{p}(b|a,c) = \tilde{p}(b|c) \end{cases}$$

$$\Rightarrow p(a,b,c) = p(a|b,c)\, p(b|c)\, p(c)$$

$$\simeq p(a|c)\, p(b|c)\, p(c).$$

$= p(a|c)\frac{p(c)}{p(c)} p(b|c)p(c)$

$= \frac{p(a,c)\,p(b,c)}{p(c)}$

② Let's divide our landmarks into 3 sets

$m = m^+ + m^0 + m^-$

active | active & passive | passive

sparsification: remove links between robot's pose and "active → passive" landmarks. (Recall m_1, m_2, m_3 example)

⇒ Given m^+, m^-, assume x_t and m_0 are independent.

active landmarks go into passive ones because I observed them in past, now I observe new landmarks but I only can have certain number of active landmarks. So some of them go to passive

⇒ $p(x_t, m | z_{1:t}, u_{1:t}) = p(x_t, m^+, m_0, m^- | z_{1:t}, u_{1:t})$

$= p(x_t | m^+, m^0, m^-, u_{1:t}, z_{1:t})\, p(m^+, m^0, m^- | z_{1:t}, u_{1:t})$

$= p(x_t | m^+, m^-, u_{1:t}, z_{1:t})\, p(m^+, m^0, m^- | z_{1:t}, u_{1:t})$.

only direct links between x_t and m^+. so ~~no~~ direct links between x_t and m^-

⇒ Given active landmarks, passive ones does not matter.

should set $m^- = 0$ given m^+, m^0

$= p(x_t | m^+, m^- = 0, u_{1:t}, z_{1:t})\, p(m^+, m^0, m^- | z_{1:t}, u_{1:t})$

$= \frac{p(x_t, m^+ | m^- = 0, z_{1:t}, u_{1:t})}{p(m^+ | m^- = 0, u_{1:t}, z_{1:t})}\, p(m^+, m^0, m^- | z_{1:t}, u_{1:t})$

③ $\tilde\Omega_t = \Omega_t^1 - \Omega_t^2 + \Omega_t^3$

$\Omega_t \Leftrightarrow p(x_t, m | z_{1:t}, u_{1:t})$.

(i) $p(x_t, m^+, m^0 | m^- = 0, z_{1:t}, u_{1:t}) \rightarrow \Omega_t^0$ condition Ω_t on $m^- = 0$.

(ii) $\int_{m^0} p(x_t, m^+, m^0 | m^- = 0, z_{1:t}, u_{1:t}) = p(x_t, m^+ | m^- = 0, z_{1:t}, u_{1:t})$ marginalizing m^0 from Ω_t^0 get Ω_t^1

(iii) similarly. marginalizing x, m^0 from Ω_t^0 yields Ω_t^2. marginalizing x from Ω_t yields Ω_t^3

(iv) $\tilde\Omega_t = \Omega_t^1 - \Omega_t^2 + \Omega_t^3$

$\tilde\xi_t = \tilde\Omega_t \mu_t = (\Omega_t - \Omega_t + \tilde\Omega_t)\mu_t = \xi_t + (\tilde\Omega_t - \Omega_t)\mu_t$

few entries non-zero

④ conditioning as projection.

$\Omega_t^0 = F_{x,m^+,m^0} F_{x,m^+,m^0}^T\, \Omega_t\, F_{x,m^+,m^0} F_{x,m^+,m^0}^T$

$\tilde\Omega_t = \Omega_t - \Omega_t^0 F_{m^0}(F_{m^0}^T \Omega_t^0 F_{m^0})^{-1} F_{m^0}^T \Omega_t^0$
$+ \Omega_t^0 F_{x,m^0}(F_{x,m^0}^T \Omega_t^0 F_{x,m^0})^{-1} F_{x,m^0}^T \Omega_t^0$
$- \Omega_t F_x (F_x^T \Omega_t F_x)^{-1} F_x^T \Omega_t$.

$\tilde\xi_t = \xi_t + (\tilde\Omega_t - \Omega_t)\mu_t$.

F_x: every ~~zero~~ except corresponding to robot pose.

F_{m^0}: every zero except corresponding to entries of m_0