EN4594 Autonomous Systems

Simultaneous Localization and Mapping (SLAM) - II

Peshala Jayasekara, PhD

Dept. of Electronic and Telecommunication Engineering

Faculty of Engineering

University of Moratuwa

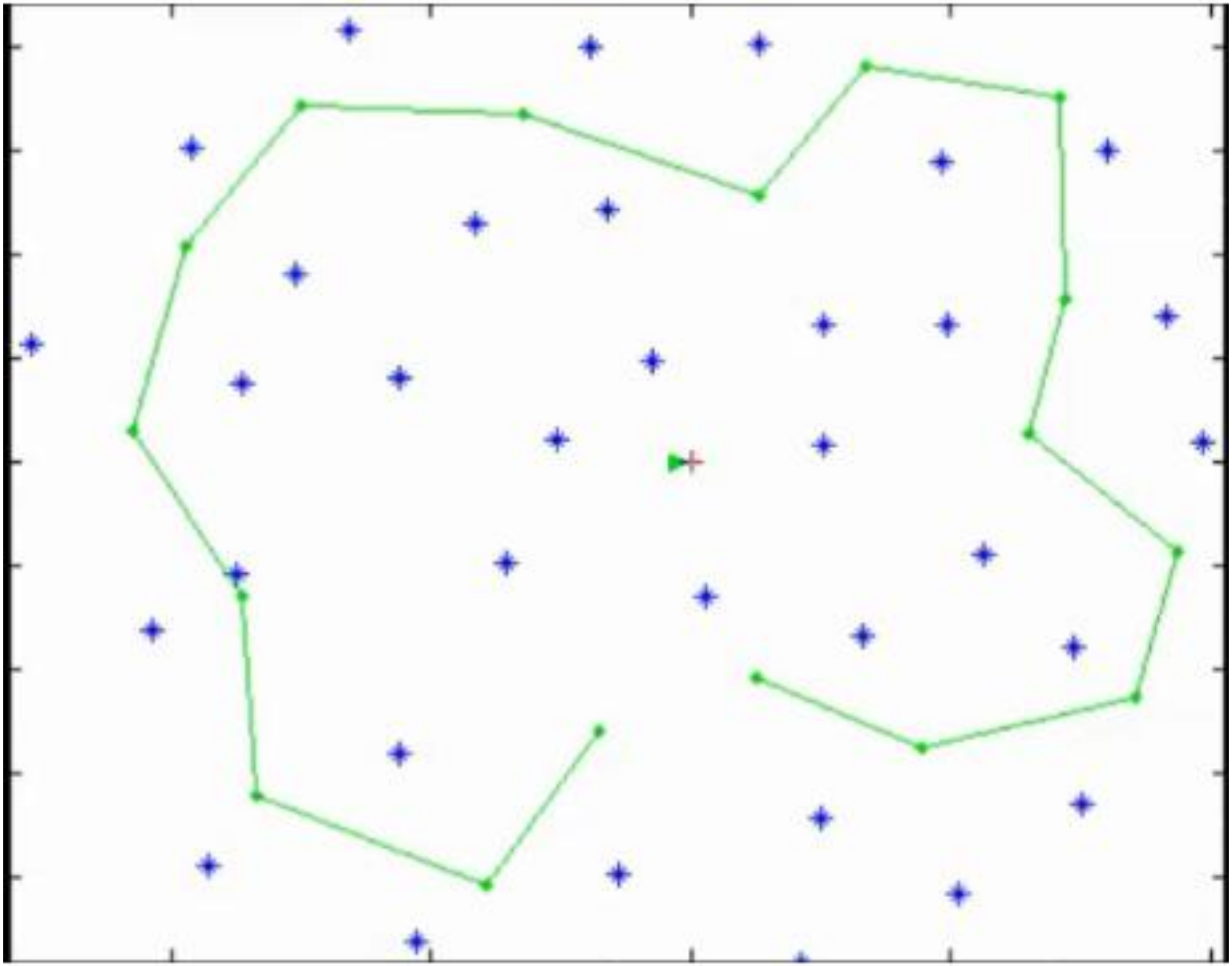
Outline

- Recap
- EKF SLAM with Known Data Associations
  - Initial Belief
  - State Transition
  - Measurement Model
  - Landmark Initialization
  - EKF SLAM Algorithm

Recap

- SLAM is the process of building a map of a static environment and to localize in the map at the same time
  - ➢ To localize, you need a map
  - ➢ To map, you need robot localization
- One of the earliest solutions to the SLAM problem is the application of EKF to landmark based online SLAM problem.
  - ➢ State transition only involves robot motion; landmark locations do not change
  - ➢ Wrong data associations can make the EKF SLAM diverge

Recap



www.youtube.com/watch?v=y7OnimZRj2w

EKF SLAM with Known Data Associations

Extended Kalman Filter

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

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

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

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

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

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

return μ_t, Σ_t

state $y_t = \left[\begin{array}{c} x_t \\ m \end{array}\right] \begin{array}{l} \text{robot pose} \\ \text{map} \end{array}$

- We need $g, G, h, H, \mu_0, \Sigma_0, R_t, Q_t$

Initial Belief

- Gaussian Initial Belief
  - ➢ Either initialize with only robot pose variables (3 × 1 vector) Or include the landmarks as well, with zero initialization

$$\mu_0 = (0 \quad 0 \quad 0 \quad \cdots \quad 0)^T_{3+2N}$$

$$\Sigma_0 = \begin{bmatrix} 0 & 0 & 0 & & & & & \\ 0 & 0 & 0 & & & & 0 & \\ 0 & 0 & 0 & & & & & \\ & & & \infty & & & & \\ & & & & \infty & & & \\ & & & & & \infty & & \\ & & 0 & & & & \infty & \\ & & & & & & & \infty \\ & & & & & & & & \infty \end{bmatrix}_{(3+2N)\times(3+2N)}$$

robot pose

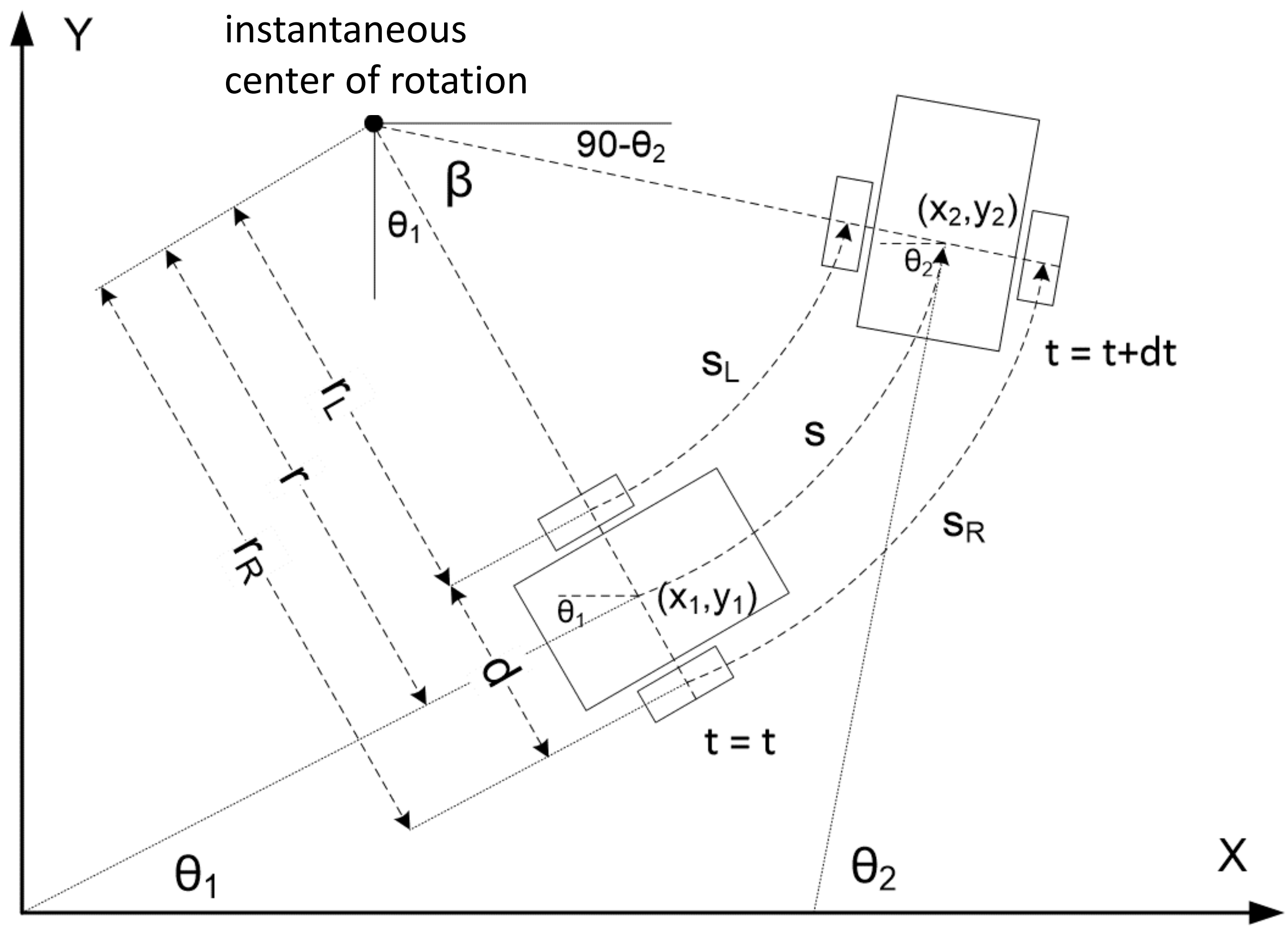
no correlation

very high initial uncertainty

State Transition

State Transition

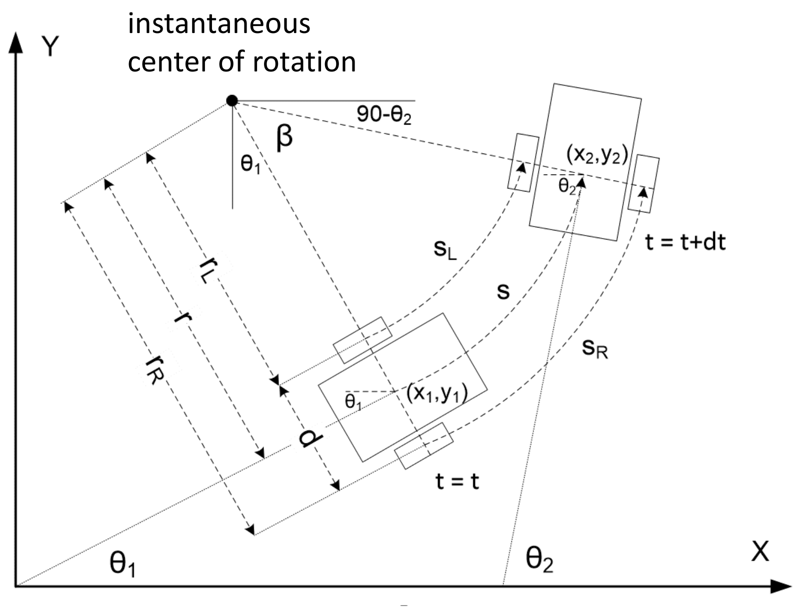
- E.g. Differential drive robot



State Transition

- Let the initial heading of the robot be θ_1 and the heading after one time step be θ_2 :

$$\beta = \theta_2 - \theta_1 \qquad (1)$$
$$= change\ of\ heading$$



- Let, the distance traveled by the left wheel be s_L and the distance traveled by the right wheel be s_R

$$s_L = r_L\beta \qquad (2)$$
$$s_R = r_R\beta \qquad (3)$$

- Eq. (3) – (2) gives

$$s_R - s_L = (r_R - r_L)\beta = d \cdot \beta \qquad (4)$$

Therefore, $$\beta = \frac{s_R - s_L}{d} \qquad (5)$$

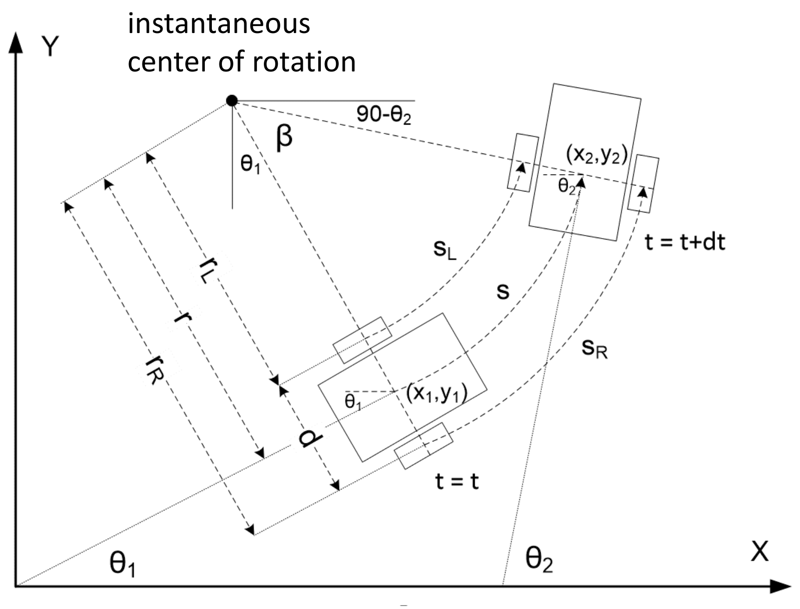
State Transition

- If the distance traveled by the center of the robot is given by s,

$$s = \frac{s_L + s_R}{2} = r\beta \tag{6}$$

- Thus,

$$r = \frac{s_L + s_R}{2\beta} \tag{7}$$



- The final pose of the robot can be calculated as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} rsin(\theta_1 + \beta) - rsin(\theta_1) \\ -rcos(\theta_1 + \beta) + rcos(\theta_1) \\ \beta \end{bmatrix} \tag{8}$$

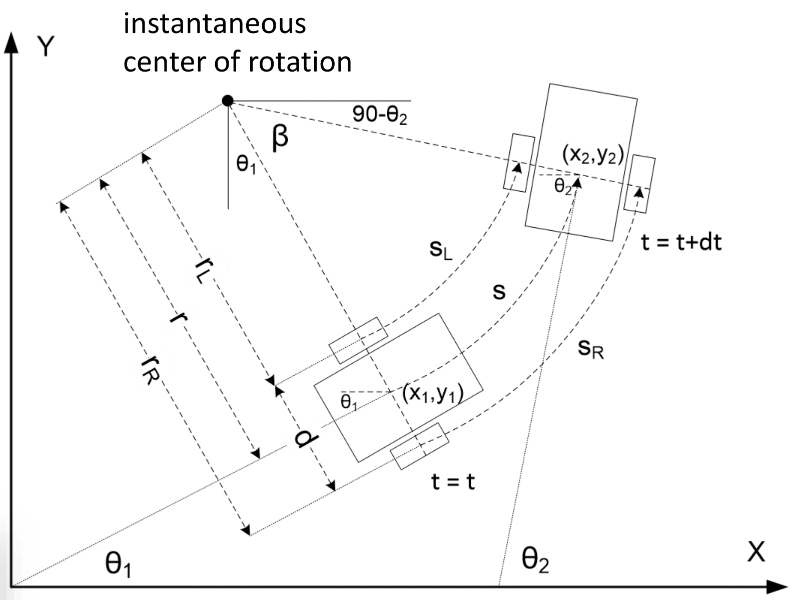
State Transition

- When $(s_L - s_R) \rightarrow 0$, we see that

  $\beta \rightarrow 0$ and $r \rightarrow \infty$ $\because$ (5) and (7)

$$\beta = \frac{s_R - s_L}{d} \quad (5)$$

$$r = \frac{s_L + s_R}{2\beta} \quad (7)$$



- For this condition it can be shown that eq. (8) be transformed as follows:

$$(8): \begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} r\sin(\theta_1 + \beta) - r\sin(\theta_1) \\ -r\cos(\theta_1 + \beta) + r\cos(\theta_1) \\ \beta \end{bmatrix}$$

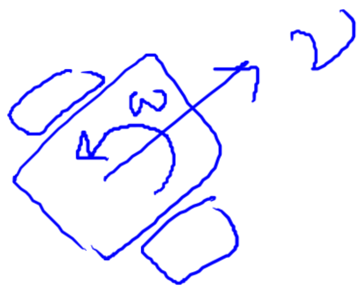
$$\Rightarrow \begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} r\beta\cos(\theta_1) \\ r\beta\sin(\theta_1) \\ \beta \end{bmatrix} \quad (9)$$

exercise

State Transition

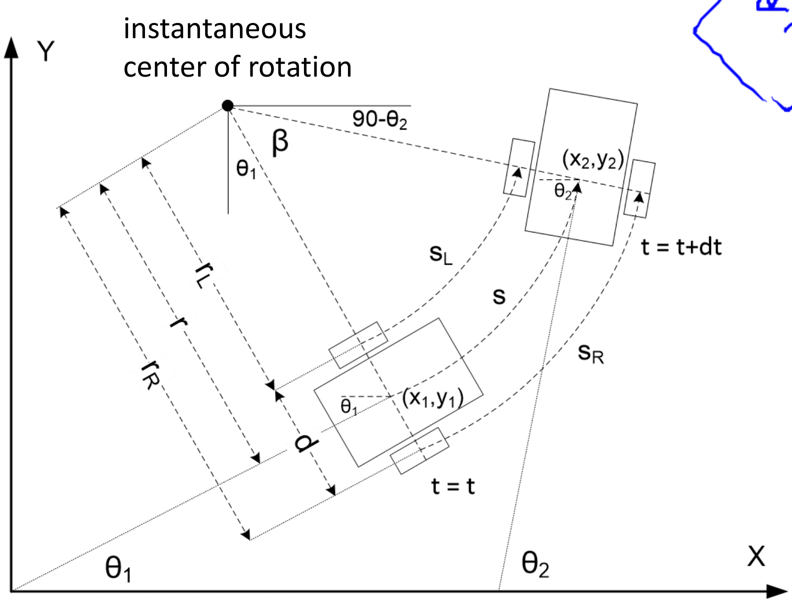
- State transition as a function of control input u_t and previous state x_t
  - ➢ Control input for a differential drive mobile robot

$$u_t = [v_t \quad \omega_t]^T$$



v: linear velocity

ω: angular velocity



$$v = \frac{s}{dt} \qquad \omega = \frac{\beta}{dt}$$

$$\frac{v}{\omega} = \frac{s}{\beta} = r$$

State Transition

- State transition as a function of control input u_t and previous state x_t

$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} rsin(\theta_1 + \beta) - rsin(\theta_1) \\ -rcos(\theta_1 + \beta) + rcos(\theta_1) \\ \beta \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} \frac{v}{\omega} \sin(\theta_1 + \omega \Delta t) - \frac{v}{\omega} \sin\theta_1 \\ -\frac{v}{\omega} \cos(\theta_1 + \omega \Delta t) + \frac{v}{\omega} \cos\theta_1 \\ \omega \Delta t \end{bmatrix}$$

- ➢ Still, this is not a function of full previous state
  - State contains landmarks as well

State Transition

- Robot motion model is extended to the augmented state vector

$$y_t = y_{t-1} + \begin{bmatrix} \frac{v_t}{\omega_t}\sin(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\sin\theta_{t-1} \\ -\frac{v_t}{\omega_t}\cos(\theta_{t-1}+\omega_t\Delta t) + \frac{v_t}{\omega_t}\cos\theta_{t-1} \\ \omega_t \cdot \Delta t \\ \left.\begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix}\right\} 2N \end{bmatrix}$$

  - More compactly

$$y_t = y_{t-1} + F_x^T \begin{bmatrix} \frac{v_t}{\omega_t}\sin(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\sin\theta_{t-1} \\ -\frac{v_t}{\omega_t}\cos(\theta_{t-1}+\omega_t\Delta t) + \frac{v_t}{\omega_t}\cos\theta_{t-1} \\ \omega_t \cdot \Delta t \end{bmatrix}$$

where $F_x = \begin{bmatrix} 1 & 0 & 0 & \overbrace{0 \quad 0 \quad \cdots \quad 0}^{2N} \\ 0 & 1 & 0 & 0 \quad 0 \quad \cdots \quad 0 \\ 0 & 0 & 1 & 0 \quad 0 \quad \cdots \quad 0 \end{bmatrix}$

State Transition

- Full state transition model with noise

$$\underbrace{y_t = y_{t-1} + F_x^T \begin{bmatrix} \frac{v_t}{\omega_t}\sin(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\sin\theta_{t-1} \\ -\frac{v_t}{\omega_t}\cos(\theta_{t-1}+\omega_t\Delta t) + \frac{v_t}{\omega_t}\cos\theta_{t-1} \\ \omega_t \cdot \Delta t \end{bmatrix}}_{g(u_t,\, y_{t-1})} + \mathcal{N}(0,\ F_x^T R_t F_x)$$

➢ $F_x^T R_t F_x$ extends the process noise covariance matrix to the dimension of the full state vector

Extended Kalman Filter

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

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$

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

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

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$

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

return μ_t, Σ_t

Linearizing State Transition

- State transition function g is linearized using the 1st order Taylor expansion

$$g(u_t, y_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t\,(y_{t-1} - \mu_{t-1})$$

Jacobian

$$G_t = \left.\frac{\partial g(u_t, y_{t-1})}{\partial y_{t-1}}\right|_{u_t, \mu_{t-1}}$$

State Transition: Jacobian

- Jacobian

$$G_t = \frac{\partial}{\partial y_{t-1}} \left\{ y_{t-1} + F_x^T \underbrace{\begin{bmatrix} v_t/\omega_t \sin(\theta_{t-1} + \omega_t \Delta t) - \frac{v_t}{\omega_t} \sin\theta_{t-1} \\ -v_t/\omega_t \cos(\theta_{t-1} + \omega_t \Delta t) + \frac{v_t}{\omega_t} \cos\theta_{t-1} \\ \omega_t \cdot \Delta t \end{bmatrix}}_{} \right.$$

$$\begin{array}{l} R_1 \\ R_2 \\ R_3 \end{array}$$

$$G_t = I_{3+2N} + F_x^T \begin{bmatrix} \partial/\partial x (R_1) & \partial/\partial y (R_1) & \partial/\partial \theta (R_1) & \frac{\partial}{x_1}(R_1) & \frac{\partial}{\partial y_1}(R_1) \cdots \\ \partial/\partial x (R_2) & \partial/\partial y (R_2) & \partial/\partial \theta (R_2) & \partial/\partial x_1 (R_2) & \partial/\partial y_1 (R_2) \cdots \\ \partial/\partial x (R_3) & \partial/\partial y (R_3) & \partial/\partial \theta (R_3) & \partial/\partial x_1 (R_3) & \partial/\partial y_1 (R_3) \cdots \end{bmatrix}$$

State Transition: Jacobian

- Jacobian

$$G_t = \frac{\partial}{\partial y_{t-1}} \left\{ y_{t-1} + F_x^T \begin{bmatrix} \frac{v_t}{\omega_t} \sin(\theta_{t-1} + \omega_t \Delta t) - \frac{v_t}{\omega_t} \sin\theta_{t-1} \\ -\frac{v_t}{\omega_t} \cos(\theta_{t-1} + \omega_t \Delta t) + \frac{v_t}{\omega_t} \cos\theta_{t-1} \\ \omega_t \cdot \Delta t \end{bmatrix} \right\}$$

$$G_t = I + F_x^T \left. \begin{bmatrix} 0 & 0 & \left[\frac{v_t}{\omega_t} \cos(\theta_{t-1} + \omega_t \Delta t) - \frac{v_t}{\omega_t} \cos\theta_{t-1}\right] & 0 & 0 & \cdots \\ 0 & 0 & \left[+\frac{v_t}{\omega_t} \sin(\theta_{t-1} + \omega_t \Delta t) - \frac{v_t}{\omega_t} \sin\theta_{t-1}\right] & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots \end{bmatrix} \right|_{\mu_{t-1}}$$

State Transition: Jacobian

- Jacobian

$$G_t = I + F_x^T \left[\begin{matrix} 0 & 0 & \left[\frac{v_t}{\omega_t}\cos(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\cos\theta_{t-1}\right] & 0 & 0 & \cdots \\ 0 & 0 & \left[+\frac{v_t}{\omega_t}\sin(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\sin\theta_{t-1}\right] & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots \end{matrix}\right] \Bigg|_{\mu_{t-1}}$$

  - More compactly

$$G_t = I + F_x^T \overbrace{\begin{bmatrix} 0 & 0 & v_t/\omega_t\cos(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\cos\theta_{t-1} \\ 0 & 0 & v_t/\omega_t\sin(\theta_{t-1}+\omega_t\Delta t) - \frac{v_t}{\omega_t}\sin\theta_{t-1} \\ 0 & 0 & 0 \end{bmatrix}}^{p} F_x$$

$$G_t = I + F_x^T p F_x$$

verify : exercise

Extended Kalman Filter

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

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

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

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

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

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

return μ_t, Σ_t

- Predicted covariance
  - Motion noise covariance should be transformed to state dimensions

$$R_t \rightarrow \underbrace{F_x^T R_t F_x}_{3+2N \times 3+2N}$$

$$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

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

EKF SLAM Prediction

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

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$
$$\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + R_t$$
$$K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$
$$\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$$
return μ_t, Σ_t

- Plug the equations into EKF

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

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

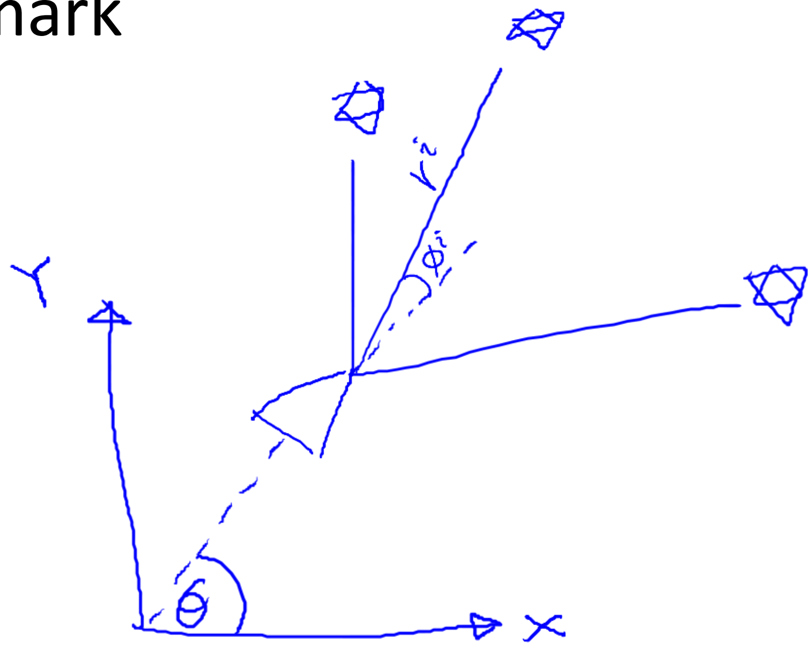
$$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

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

Measurement Model

Measurement Model

- The landmark sensor will observe the range and bearing to a landmark



$\{r_t^i, \phi_t^i\}$

- ➢ assume that i^{th} feature in the list of observations corresponds to the j^{th} landmark in the map

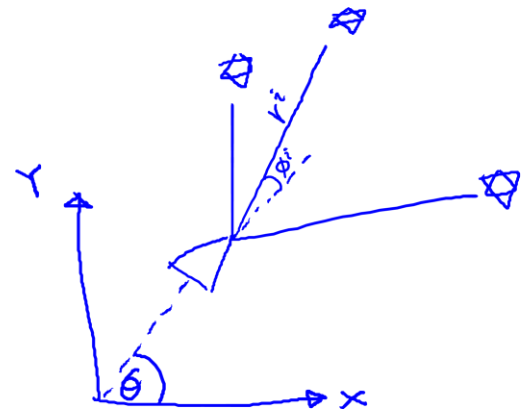
Known Data Association

Measurement Model

- Let

  robot pose : $x_t = [x \quad y \quad \theta]^T$

  landmark : $m_j = [m_{j,x} \quad m_{j,y}]$



$$\underbrace{\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix}}_{z_t^i} = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ \operatorname{atan2}(m_{j,y}-y,\ m_{j,x}-x)-\theta \end{pmatrix}}_{h(y_t,\,j)} + \underbrace{N\left(0, \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix}\right)}_{Q_t}$$

Linearizing Measurement Model

- Measurement model h is linearized using the 1st order Taylor expansion

$$h(y_t, j) \approx h(\bar{\mu}_t, j) + H_t^j (y_t - \bar{\mu}_t)$$

$$H_t^j \xrightarrow{\text{Jacobian}} \left. \frac{\partial h}{\partial y_t} \right|_{\bar{\mu}_t}$$

Measurement Model: Jacobian

- Jacobian

$$H_t^j = \frac{\partial}{\partial y_t}\left\{\begin{pmatrix}\sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ \operatorname{atan2}(m_{j,y}-y,\ m_{j,x}-x)-\theta\end{pmatrix}\right\}\Bigg|_{\bar{\mu}_t} \quad \begin{matrix} h_1 \\ h_2 \end{matrix}$$

$$= \begin{bmatrix} \frac{\partial}{\partial x}() & \frac{\partial}{\partial y}() & \frac{\partial}{\partial \theta}() & \frac{\partial}{\partial m_{1,x}}() & \frac{\partial}{\partial m_{1,y}}() & \cdots & \frac{\partial}{\partial m_{j,x}}() & \frac{\partial}{\partial m_{j,y}}() & \cdots & \frac{\partial}{\partial m_{N,x}}() & \frac{\partial}{\partial m_{N,y}}() \\ \frac{\partial}{\partial x}() & \frac{\partial}{\partial y}() & \frac{\partial}{\partial \theta}() & \frac{\partial}{\partial m_{1,x}}() & \frac{\partial}{\partial m_{1,y}}() & \cdots & \frac{\partial}{\partial m_{j,x}}() & \frac{\partial}{\partial m_{j,y}}() & \cdots & \frac{\partial}{\partial m_{N,x}}() & \frac{\partial}{\partial m_{N,y}}() \end{bmatrix}$$

Measurement Model: Jacobian

- Jacobian

$$\frac{\partial}{\partial x} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \Big|_{\bar{\mu}_t}$$

$$= \frac{1}{\cancel{2}\sqrt{q_t}} \cancel{2}(m_{j,x} - x)(-1) \Big|_{\bar{\mu}_t}$$

$$= \frac{\bar{\mu}_{t,x} - \bar{\mu}_{j,x}}{\sqrt{q_t}} \qquad q_t = (\bar{\mu}_{j,x} - \bar{\mu}_{t,x})^2 + (\bar{\mu}_{j,y} - \bar{\mu}_{t,y})^2$$

similarly, $\frac{\partial}{\partial y}(\ ) = \frac{\bar{\mu}_{t,y} - \bar{\mu}_{j,y}}{\sqrt{q_t}}$

Measurement Model: Jacobian

- Jacobian

$$\frac{\partial}{\partial m_{j,x}} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2}\,\Big|_{\bar{\mu}_t}$$

$$= \frac{1}{\cancel{2}\sqrt{q_t}}\,\cancel{2}\,(m_{j,x}-x)\,\Big|_{\bar{\mu}_t}$$

$$= \frac{\bar{\mu}_{j,x}-\bar{\mu}_{t,x}}{\sqrt{q_t}}$$

Similarly, $\frac{\partial}{\partial m_{j,y}}(\;\;) = \frac{\bar{\mu}_{j,y}-\bar{\mu}_{t,y}}{\sqrt{q_t}}$

Measurement Model: Jacobian

- Jacobian

$$\frac{\partial}{\partial x}\left[\operatorname{atan2}\left(m_{j,y}-y,\ m_{j,x}-x\right)-\theta\right]$$

exercise

$$=\frac{\bar{\mu}_{j,y}-\bar{\mu}_{t,y}}{q_t}$$

Similarly, $\frac{\partial}{\partial y}(\ ) = \frac{\bar{\mu}_{t,x}-\bar{\mu}_{j,x}}{q_t}$

- Jacobian

$$H_t^j = \begin{bmatrix} \frac{\partial}{\partial x}() & \frac{\partial}{\partial y}() & \frac{\partial}{\partial \theta}() & \frac{\partial}{\partial m_{1,x}}() & \frac{\partial}{\partial m_{1,y}}() & \cdots & \frac{\partial}{\partial m_{j,x}}() & \frac{\partial}{\partial m_{j,y}}() & \cdots & \frac{\partial}{\partial m_{N,x}}() & \frac{\partial}{\partial m_{N,y}}() \\ \frac{\partial}{\partial x}() & \frac{\partial}{\partial y}() & \frac{\partial}{\partial \theta}() & \frac{\partial}{\partial m_{1,x}}() & \frac{\partial}{\partial m_{1,y}}() & \cdots & \frac{\partial}{\partial m_{j,x}}() & \frac{\partial}{\partial m_{j,y}}() & \cdots & \frac{\partial}{\partial m_{N,x}}() & \frac{\partial}{\partial m_{N,y}}() \end{bmatrix}$$

$2 \times (3+2N)$

- ➢ Transform a low dimensional matrix to H_t^j using $F_{x,j}$

$$h_t^j = \begin{bmatrix} \frac{\partial}{\partial x}() & \frac{\partial}{\partial y}() & \frac{\partial}{\partial \theta}() & \frac{\partial}{\partial m_{j,x}}() & \frac{\partial}{\partial m_{j,y}}() \\ \frac{\partial}{\partial x}() & \frac{\partial}{\partial y}() & \frac{\partial}{\partial \theta}() & \frac{\partial}{\partial m_{j,x}}() & \frac{\partial}{\partial m_{j,y}}() \end{bmatrix}_{2 \times 5}$$

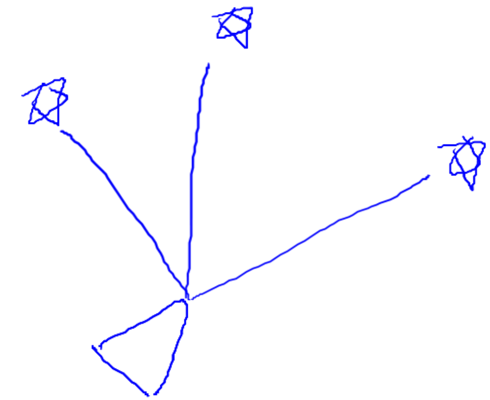
$$F_{x,j} = \begin{bmatrix} 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & \underbrace{0 \cdots 0}_{2j-2} & \underbrace{0 \quad\quad 1}_{2} & & \underbrace{0 \cdots 0}_{2N-2j} \end{bmatrix} \quad 5 \times (3+2N)$$

$$H_t^j = h_t^j \cdot F_{x,j}$$

EKF SLAM Correction

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

$$
\begin{aligned}
&\bar{\mu}_t = g(u_t, \mu_{t-1}) \\
&\bar{\Sigma}_t = G_t\ \Sigma_{t-1}\ G_t^T + R_t \\
&K_t = \bar{\Sigma}_t\ H_t^T (H_t\ \bar{\Sigma}_t\ H_t^T + Q_t)^{-1} \\
&\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \\
&\Sigma_t = (I - K_t\ H_t)\ \bar{\Sigma}_t \\
&\text{return } \mu_t, \Sigma_t
\end{aligned}
$$



- Plug the equations into EKF
  - Iterate through all landmark observations

iterate

$$
\begin{aligned}
&\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \\
&q = \delta^T \delta \\
&\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{j,s} \end{pmatrix} \\
&H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y & 0 \\ \delta_y & -\delta_x & -q & -\delta_y & +\delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & q \end{pmatrix} F_{x,j} \\
&K_t^i = \bar{\Sigma}_t\ H_t^{iT} (H_t^i\ \bar{\Sigma}_t\ H_t^{iT} + Q_t)^{-1} \\
&\bar{\mu}_t = \bar{\mu}_t + K_t^i(z_t^i - \hat{z}_t^i) \\
&\bar{\Sigma}_t = (I - K_t^i\ H_t^i)\ \bar{\Sigma}_t
\end{aligned}
$$

$$
F_{x,j} = \begin{pmatrix}
1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0 & 0\cdots0 \\
0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0 & 0\cdots0 \\
0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0 & 0\cdots0 \\
0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0 & 0\cdots0 \\
0 & 0 & 0 & 0\cdots0 & 0 & 1 & 0 & 0\cdots0 \\
0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 0 & 1 & \underbrace{0\cdots0}_{2N-2j}
\end{pmatrix}
$$

Landmark Initialization

Landmark Initialization

- State was initialized as:

$$\mu_0 = (0, 0, 0 \ldots\ldots 0\ 0)^T$$

  - this will lead to a poor linearization

- A better landmark initialization and thereby a better linearization point would be to use:
  - expected robot pose and measurement at the first sight of the landmark

Inverse measurement model

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

Landmark Initialization

- Measurement model

$$\underbrace{\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix}}_{z_t^i} \begin{matrix} = \\ = \end{matrix} \underbrace{\begin{bmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ \operatorname{atan2}(m_{j,y}-y,\ m_{j,x}-x)-\theta \end{bmatrix}}_{h(y_t,j)}$$

- Inverse measurement model

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

EKF SLAM Algorithm

EKF SLAM Algorithm

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

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

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

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

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

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

return μ_t, Σ_t

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

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

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

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

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

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

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

$j = c_t^i$

if landmark j *never seen before*

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

endif

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

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

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

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

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

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

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

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

endfor

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

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

return μ_t, Σ_t

*check Moodle for a clear algorithm document

Summary

- SLAM is the process of building a map of a static environment and to localize in the map at the same time
  - ➢ EKF SLAM with known data associations is presented
- State transition depends on the robot motion mechanism
  - ➢ Differential drive system was discussed
- Observation model depends on the type of sensors used
  - ➢ Range, bearing measurements were discussed