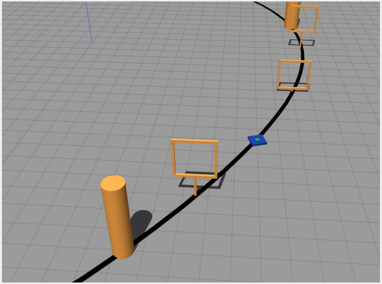
Autonomous Mobile Robotics

Robot Localization

Geesara Kulathunga

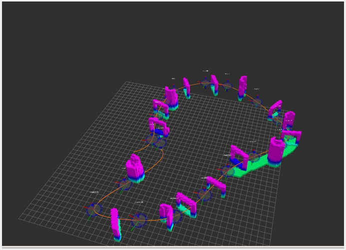


October 26, 2022

Robot Localization

Contents

- A Taxonomy of Localization Problems
- Markov localization
  - Environment Sensing
  - Motion in the Environment
  - Localization in the Environment
- EKF localization with known correspondence
- Particle filter localization with known correspondence



A Taxonomy of Localization Problems

- Local Versus Global
  - Position tracking where initial position is known (local tracking)
  - Robot position is unknown, initially has to assume that pose of robot is uniform in the most of the cases (global)
  - Kidnapped robot problem; anytime robot can be moved to different location without prior knowledge (global)
- Static Versus Dynamic Environments
  - In static environment, robot's pose is only the variable quantity
  - Dynamics environment, whole configuration can be changed over the time
- Passive Versus Active Approaches
  - In passive, robot is controlled through some other means, robot motion is not aiming at facilitating localization

Algorithm Markov_localization($bel(x_{t-1}), u_t, z_t, m$)**:**

for all x_t *do*

$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}, m)\ bel(x_{t-1})\ dx$

$bel(x_t) = \eta\ p(z_t \mid x_t, m)\ \overline{bel}(x_t)$

endfor

return $bel(x_t)$

- Markov localization is derived from the algorithm Bayes filter
- However, it requires information about the map to estimate the measurement model $p(z_t|x_t, m)$
- Markov localization addresses the global localization, the position tracking, and the kidnapped robot problem in static environment

Markov localization

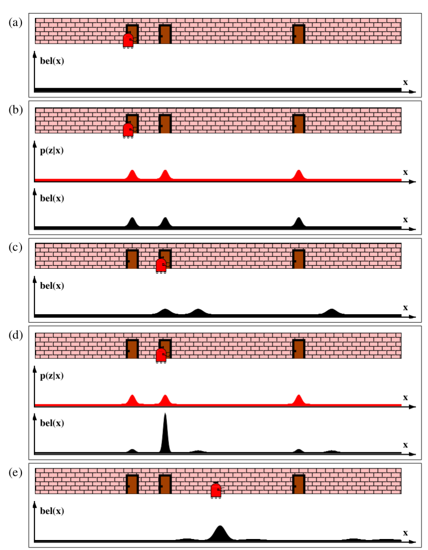
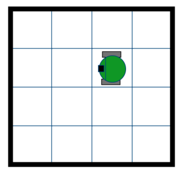


Illustration of the Markov localization algorithm, Thrun, Sebastian. "Probabilistic robotics." Communications of the ACM 45.3 (2002): 52-57.



.02	.05	.05	.05
.02	.05	.18	.05
.05	.05	.18	.05
.05	.05	.05	.05

robot initial belief

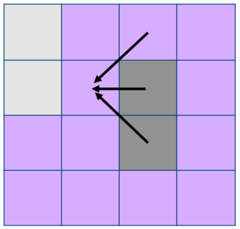
- The map is discretized into 16 cells, each of which has an area of $1m^2$
- Consider the initial belief of the robot position is given
- If control command to the robot is given by $\delta x, \delta y$ = -1.0 cells, 0.0 cells, what is the probability that robot be in the position (2,3)
- The following outcomes are possible when the control command is being applied

.00	.00	.00
.00	.00	1.0
.00	.00	.00

$\xrightarrow{(\Delta x,\Delta y)}$

.00	.20	.00
.00	.50	.10
.00	.20	.00

- How many possible ways to get to (2,3)?



  - Prediction step

$$p(x_k|z_{1:k-1}, u_{1:k-1}) = \Sigma_{x_{k-1} \in X}\ p(x_k|x_{k-1}, u_{k-1})p(x_{k-1}|z_{1:k-1}, u_{0:k-1}) \quad (1)$$

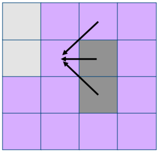
  - Correction step

$$p(x_k|z_{1:k}, u_{0:k-1}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1}, u_{0:k-1})}{p(z_k|z_{1:k-1}, u_{0:k-1})} \quad (2)$$

, where

$$p(z_k|z_{1:k-1}, u_{0:k-1}) = \Sigma_{x_k \in X}\ p(z_k|x_k)p(x_k|z_{1:k-1}, u_{0:k-1})$$

- How many possible ways to get to (2,3)?

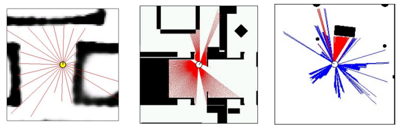


  - Prediction step

$$
\begin{aligned}
p(x_{i,t}|u_t) &= \Sigma_{j=1}^{n} p\left(x_{i,t}|x_{j,t-1}, u_t\right) p\left(x_{j,t-1}\right) \\
&= p\left(x_{i,t} = (2,3)|x_{j,t-1} = (3,3), u_t = (-1,0)\right) p\left(x_{j,t-1} = (3,3)\right) \\
&+ p\left(x_{i,t} = (2,3)|x_{j,t-1} = (2,3), u_t = (-1,0)\right) p\left(x_{j,t-1} = (2,3)\right) \\
&+ p\left(x_{i,t} = (2,3)|x_{j,t-1} = (3,2), u_t = (-1,0)\right) p\left(x_{j,t-1} = (3,2)\right) \\
&+ p\left(x_{i,t} = (2,3)|x_{j,t-1} = (3,4), u_t = (-1,0)\right) p\left(x_{j,t-1} = (3,4)\right) \\
&= 0.5 \cdot 0.18 + 0.1 \cdot 0.05 + 0.18 \cdot 0.2 + 0.05 \cdot 0.2 \\
&= 0.141
\end{aligned}
$$

Correction step

- How can we estimate the $p(z_t|x_{i,t})$?



- If each sensor reading consists of N measurements, i.e., $z = \{z_1, ..., z_n\}$, assuming each such measurement is independent given the robot pose,

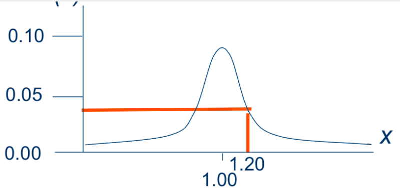
$$p(z_t|x_{i,t}) = \Pi_{j=1}^{n} p(z_j|x_{i,t}, m)$$

- Such measurements can be caused by known obstacles, dynamic obstacles, reflections, etc.

- Correction step

$$p(x_{i,t}|z_t) = \frac{p(z_t|x_{i,t})p(x_{i,t}|u_t)}{p(z_t)}$$

- $p(z_t|x_{i,t})$ getting measurement z_t from state $x_{i,t}$
- Let z_t be 1.2m and range sensor has the following distribution



- $p(z_t)$ probability of the sensor measurement z_t. Calculated so that the sum over all states $x_{i,t}$ equals 1

$$1 = \Sigma_{i=1}^{n} p(x_{i,t}|z_t = 1.2)$$

$$1 = \frac{\Sigma_{i=1}^{n} p(z_t = 1.2|x_{i,t}) p(x_{i,t}|u_{i,t})}{p(z_t = 1.2)}$$

$$p(z_t = 1.2) = \Sigma_{i=1}^{n} p(z_t = 1.2|x_{i,t}) p(x_{i,t}|u_{i,t})$$

-

$$p(x_{i,t}|z_t) = \frac{p(z_t|x_{i,t}) p(x_{i,t}|u_t)}{p(z_t)}$$

$$= \frac{p\Big(z_t = 1.2|x_{i,t} = (2,3)\Big) p(x_{i,t}|u_t)}{p(z_t = 1.2)} = \frac{0.04 \cdot 0.141}{p(z_t = 1.2)}$$

- Can we calculate this?

$$p(z_t = 1.2) = \Sigma_{i=1}^{n} p(z_t = 1.2|x_{i,t}) p(x_{i,t}|u_{i,t})$$

Let us consider there are four gates followed by landing gate. Drone does not know it's current location. Objective is to follow through several times and find out the location of the landing gate. Without loss of generality, let poses of the each places including landing gate be $x_1, x_2, x_3, x_4,$ and x_5. Camera can detect with probability of 0.8 whether is it a normal gate, and detect landing gate as the normal gate with error probability 0.1.

$$\begin{aligned} p(Z = n|X = x_n) = 0.8,\ n \in \{1, 2, 3, 4\} \\ p(Z = n|X = x_l) = 0.1,\ l \in \{5\} \end{aligned} \tag{3}$$

where index n indicated normal gates and index l indicated landing gate.

Example 01

- If drone does not know the its initial position what would be its initial belief?
- Estimate the location probability distribution after first measurement $p(X_1|Z = n)$?

Example 01

- If drone does not know the its initial position what would be its initial belief? Since we have any prior clue about current location, we could assume that location can be described with the uniform distribution

$$p(X = x_i) = bel(x_i) = 0.2, i \in \{1, 2, ..., 5\}$$

Example 01

- Estimate the location probability distribution after the first measurement, i.e., $p(X_1|Z=n)$?

$$
\begin{aligned}
p(Z=n) &= \Sigma_i p(Z=n|X_1=x_i)P(X=x_i) \\
&= [0.8, 0.8, 0.8, 0.8, 0.1] \odot [0.2, 0.2, 0.2, 0.2, 0.2] = 0.66 \\
p(X_1|Z=n) &= \frac{p(Z=n|X_1)p(X_1)}{p(Z=n)} \\
&= \frac{[0.8, 0.8, 0.8, 0.8, 0.1] \cdot [0.2, 0.2, 0.2, 0.2, 0.2]}{0.66} \\
p(X_1|Z=n) &= [0.24, 0.240.240.24, 0.03]
\end{aligned} \tag{4}
$$

Example 02

- Determine if sensor gives three reading in the this sequence? normal, land, normal. What is the probability distribution in this scenario?

Example 02

- Determine if sensor gives three reading in the this sequence? normal, land, normal. What is the probability distribution in this scenario?

$$
\begin{aligned}
p(Z_2 = l|Z_1 = n) = \Sigma_{x_i}\ p(Z_2 = l|X_2 = x_i)P(X_2 = x_i|Z_1 = n)\\
= [0.2, 0.2, 0.2, 0.2, 0.9] \odot [0.24, 0.240.240.24, 0.03] = 0.219\\
p(X_2|z_1 = n, z_2 = l)\\
= \frac{[[0.2, 0.2, 0.2, 0.2, 0.9] \cdot [0.24, 0.240.240.24, 0.03]}{p(Z_2 = l|Z_1 = n)}\\
= [0.21, 0.21, 0.21, 0.21, 0.123]\\
p(X_3|Z_1 = n, Z_2 = l, Z_3 = n) = \text{repeat the same procedure}
\end{aligned}
\tag{5}
$$

Example 03

Now let's introduce some control actions. Theoretically, drone has to move gates proportional to control value, e.g., if u =2 it has to pass two gates in the direction of counter-clockwise. However, due to external disturbances, it has 80% possibility that can pass and move to desired pose, 10% can pass to one more gate that it desired, or one gate less than it desired. This can be described with the following transition probabilities:

$$p(X_k = x_i | X_{k-1} = x_j, U_{k-1} = u) = 0.8 \; for \; i = j + u$$

$$p(X_k = x_i | X_{k-1} = x_j, U_{k-1} = u) = 0.1 \; for \; i = j + u - 1$$

$$p(X_k = x_i | X_{k-1} = x_j, U_{k-1} = u) = 0.1 \; for \; i = j + u + 1$$

Example 03

- Determine drone belief after applying control u = 2, this time assume we know the initial location of the drone, i.e., $p(X_0) = [1, 0, 0, 0, 0]$

Example 03

- Determine drone belief after applying control u = 2, this time assume we know the initial location of the drone, i.e., $p(X_0) = [1, 0, 0, 0, 0]$

$$
\begin{aligned}
p(X_1 = x_1|U_0 = 2) &= \Sigma_{x_i}\, p(X_1 = x_1|X_0 = x_i, U_0 = 2)p(X_0 = x_i) \\
&= [0.0, 0.0, 0.1, 0.8, 0.1] \cdot [1.0, 0.0, 0.0, 0.0, 0.0] = 0 \\
p(X_1 = x_2|U_0 = 2) &= \Sigma_{x_i}\, p(X_1 = x_2|X_0 = x_i, U_0 = 2)p(X_0 = x_i) \\
&= [0.1, 0, 0, 0.1, 0.8] \odot [1.0, 0.0, 0.0, 0.0, 0.0] = 0.1 \\
p(X_1 = x_3|U_0 = 2) &= [0.8, 0.1, 0, 0, 0.1] \cdot [1.0, 0.0, 0.0, 0.0, 0.0] \\
p(X_1 = x_4|U_0 = 2) &= [0.1, 0.8, 0.1, 0, 0] \cdot [1.0, 0.0, 0.0, 0.0, 0.0] \\
p(X_1 = x_5|U_0 = 2) &= [0, 0.1, 0.8, 0.1, 0] \cdot [1.0, 0.0, 0.0, 0.0, 0.0] \\
p(X_1|U_0 = 2) &= [0, 0.1, 0.8, 0.1, 0]
\end{aligned} \tag{6}
$$

Example 04

This time, again assume that drone does not know its initial position. However, drone can observe the environment while moving around. If $u_k = 1$ and the sequence of measurement were $\{l, n, n\}$ calculate robot belief after k= 3?

Example 04

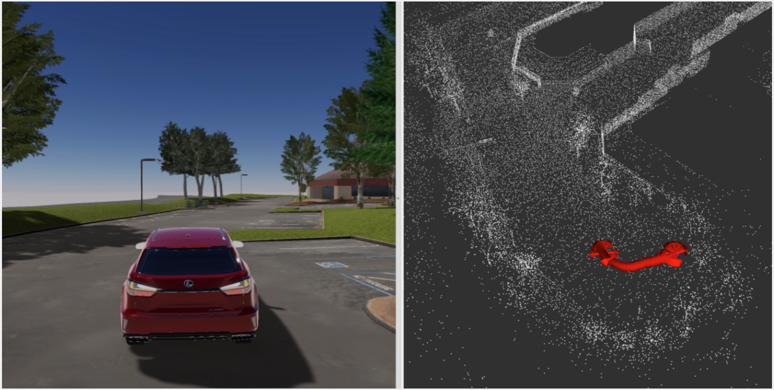
$$
\begin{aligned}
bel_p(X_1 = x_1) &= \Sigma_{x_1}\ p(X_1 = x_1 | X_0 = x_i, u_0) bel(X_0 = x_i) \\
&= [0.1, 0, 0, 0.1, 0.8] \cdot [0.2, 0.2, 0.2, 0.2, 0.2] = 0.2 \\
bel_p(X_1 = x_2) &== [0.8, 0.1, 0, 0, 0.1] \cdot [0.2, 0.2, 0.2, 0.2, 0.2] = 0.2 \\
bel_p(X_1 = x_3) &== [0.1, 0.8, 0.1, 0, 0] \cdot [0.2, 0.2, 0.2, 0.2, 0.2] = 0.2 \\
bel_p(X_1 = x_4) &== [0, 0.1, 0.8, 0.1, 0] \cdot [0.2, 0.2, 0.2, 0.2, 0.2] = 0.2 \\
bel_p(X_1 = x_5) &== [0, 0, 0.1, 0.8, 0.1] \cdot [0.2, 0.2, 0.2, 0.2, 0.2] = 0.2 \\
bel_p(X_1) &= [0.2, 0.2, 0.2, 0.2, 0.2]
\end{aligned} \tag{7}
$$

Example 04

After the measurement is obtained, the correction step is determined as follows:

$$\begin{aligned}
bel(X_1 = x_1) &= \eta\, p(z_1 = l|x_1) bel_p(X_1 = x_1) = \eta 0.2 \cdot 0.2 = \eta 0.4 \\
bel(X_1 = x_2) &= \eta\, p(z_1 = l|x_2) bel_p(X_1 = x_2) = \eta 0.2 \cdot 0.2 = \eta 0.4 \\
bel(X_1 = x_3) &= \eta\, p(z_1 = l|x_3) bel_p(X_1 = x_3) = \eta 0.2 \cdot 0.2 = \eta 0.4 \\
bel(X_1 = x_4) &= \eta\, p(z_1 = l|x_4) bel_p(X_1 = x_4) = \eta 0.2 \cdot 0.2 = \eta 0.4 \\
bel(X_1 = x_5) &= \eta\, p(z_1 = l|x_5) bel_p(X_1 = x_5) = \eta 0.9 \cdot 0.2 = \eta 0.18 \\
\eta &= \frac{1}{0.4 + 0.4 + 0.4 + 0.4 + 0.18} = 0.56 \\
bel(X_1) &= [0.22, 0.22, 0.22, 0.22, 0.89]
\end{aligned} \tag{8}$$

EKF Localization



https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/ekf-localization-howto.html

- Specific case of Markov localization
- Represents beliefs $bel(x_t)$ by their first and second moment, the mean μ_t and the covariance Σ_t
- Map is represented by a collection of features and those are known

EKF Localization

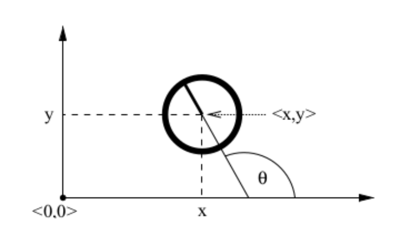
- Specific case of Markov localization
- Represents beliefs $bel(x_t)$ by their first and second moment, the mean μ_t and the covariance Σ_t
- Map is represented by a collection of features and those are known
- Initially, it requires following information:
  - robot pose at time $k-1$ with μ_{t-1}, Σ_{t-1}
  - Control input u_k
  - Map and a set of features $z_t = \{z_t^1, z_t^2, ...\}$ measured at time k and those are corresponded to variables $c_t = \{c_t^1, c_t^2, ...\}$
- Output is a new, revised estimation: μ_t and Σ_t

Comparison between KF and EKF

KF	EKF
	$\boxed{\Phi_k = \left.\frac{\partial f(\mathbf{x}_k, t)}{\partial \mathbf{x}}\right\rvert_{\mathbf{x}_k}}$
$\hat{\mathbf{x}}_k^- = \Phi_k \mathbf{x}_k$	$\boxed{\hat{\mathbf{x}}_k^- = f(\mathbf{x}_k, \mathbf{t})}$
$\mathbf{P}_k^- = \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{Q}_k$	$\mathbf{P}_k^- = \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{Q}_k$
	$\boxed{\mathbf{H} = \left.\frac{\partial h(\hat{\mathbf{x}}_k^-)}{\partial \hat{\mathbf{x}}}\right\rvert_{\hat{\mathbf{x}}_k^-}}$
$\mathbf{y} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-$	$\mathbf{y} = \mathbf{z}_k - \boxed{h(\hat{x}_k^-)}$
$\mathbf{K_k} = \mathbf{P_k^-} \mathbf{H_k}^T (\mathbf{H_k} \mathbf{P_k^-} \mathbf{H_k}^T + \mathbf{R}_k)^{-1}$	$\mathbf{K_k} = \mathbf{P_k^-} \mathbf{H_k}^T (\mathbf{H_k} \mathbf{P_k^-} \mathbf{H_k}^T + \mathbf{R}_k)^{-1}$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K_k} \mathbf{y}$	$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K_k} \mathbf{y}$
$\mathbf{P}_k = (\mathbf{I} - \mathbf{K_k} \mathbf{H_k}) \mathbf{P_k^-}$	$\mathbf{P}_k = (\mathbf{I} - \mathbf{K_k} \mathbf{H_k}) \mathbf{P_k^-}$

Probabilistic Motion Model

- Motion models comprise the state transition probability $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1})$ (prediction step of the Bayes filter)
- Robot pose $[x \quad y \quad \theta]^\top$, shown in a global coordinate system

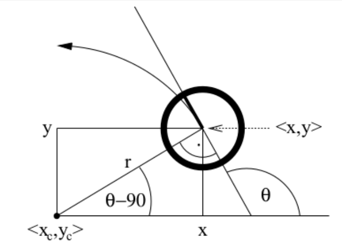


- Probabilistic kinematic model, or motion model (velocity motion model or odometry motion model), describes the posterior distribution over kinematic states that a robot assumes when executing the motion command $\mathbf{u}_t$ at $\mathbf{x}_{t-1}$

Velocity Motion Model (Noise-free)

- A robot can be control through linear and angular velocities $\mathbf{u}_t = [v_t \quad \omega_t]^\top$
- Differential drives, Ackerman drives, and synchro-drives can be controlled in this way
- Let $\mathbf{x}_{t-1} = [x_{t-1} \quad y_{t-1} \quad \theta_{t-1}]^\top, \mathbf{x}_t = [x_t \quad y_t \quad \theta_t]^\top$ be pose and time $t-1$ and successor pose, respectively, after applying applying control u_t for δt duration
- If both velocities are kept at a fixed value for the entire time interval, [t-1, t], robot moves on a circle with radius $r = |\frac{v}{u}|$
- For linear motion, r becomes infinite
- After δt units of time, the noise-free robot has progressed $v\delta t$ along the circle, which caused its heading direction to turn by $\omega\delta t$

Velocity Motion Model (Noise-free)



- The center of the circle is at

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x - \frac{v}{\omega} sin(\theta) \\ y + \frac{v}{\omega} cos(\theta) \end{bmatrix}$$

- After δt time, ideal robot will be at $\mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} & y_{t+1} & \theta_{t+1} \end{bmatrix}$

$$= \begin{bmatrix} x_c + \frac{v}{\omega} sin(\theta_t + \omega\delta t) \\ y_c - \frac{v}{\omega} cos(\theta_t + \omega\delta t) \\ \theta_t + \omega\delta t \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} -\frac{v}{\omega} sin(\theta_t) + \frac{v}{\omega} sin(\theta_t + \omega\delta t) \\ \frac{v}{\omega} cos(\theta_t) - \frac{v}{\omega} cos(\theta_t + \omega\delta t) \\ \omega\delta t \end{bmatrix}$$

- In reality, robot motion is subject to noise, to model such noise, which is formed a zero-centered random variable with finite variance, we can do the following approach

$$\begin{pmatrix} \hat{v}_t \\ \hat{\omega}_t \end{pmatrix} = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} + \begin{pmatrix} \varepsilon_{\alpha_1 v_t^2 + \alpha_2 \omega_t^2} \\ \varepsilon_{\alpha_3 v_t^2 + \alpha_4 \omega_t^2} \end{pmatrix} = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} + N(0, M_t) \qquad (9)$$

where $M_t = \begin{pmatrix} \varepsilon_{\alpha_1 v_t^2 + \alpha_2 \omega_t^2} & 0 \\ 0 & \varepsilon_{\alpha_3 v_t^2 + \alpha_4 \omega_t^2} \end{pmatrix}$

- Real motion model

$$\underbrace{\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix}}_{\mathbf{x}_{t+1}} = \underbrace{\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}_t}{\hat{\omega}_t} sin(\theta) + \frac{\hat{v}_t}{\hat{\omega}_t} sin(\theta + \hat{\omega}_t \delta t) \\ \frac{\hat{v}_t}{\hat{\omega}_t} cos(\theta) - \frac{\hat{v}_t}{\hat{\omega}_t} cos(\theta + \hat{\omega}_t \delta t) \\ \hat{\omega}_t \delta t + \hat{\gamma} \delta t \end{pmatrix}}_{f(u_t, \mathbf{x}_t)} \quad (10)$$

, where $\hat{\gamma} \sim \varepsilon_{\alpha_5 v_t^2 + \alpha_6 \omega_t^2}$

- Approximated motion model, i.e., replacing true motion $\hat{v}_t$ and $\hat{\omega}_t$ by executed control (v_t, ω_t)

$$\underbrace{\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix}}_{\mathbf{x}_{t+1}} = \underbrace{\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t} sin(\theta) + \frac{v_t}{\omega_t} sin(\theta + \omega_t \delta t) \\ \frac{v_t}{\omega_t} cos(\theta) - \frac{v_t}{\omega_t} cos(\theta + \omega_t \delta t) \\ \omega_t \delta t \end{pmatrix}}_{f(u_t, \mathbf{x}_t)} + N(0, Q_t) \quad (11)$$

Estimate transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t, u_{t+1})$

The formulated motion model assumes that the robot travels with a fixed velocity during δt, resulting in a circular trajectory. For a robot that moved from x_{t+1} to x_t, the center of the circle is defined as x_c, y_c and given by

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} -\lambda sin(\theta) \\ \lambda cos(\theta) \end{bmatrix} = \begin{bmatrix} \frac{x_t + x_{t+1}}{2} + \mu(y_t - y_{t+1}) \\ \frac{y_t + y_{t+1}}{2} + \mu(x_t - x_{t+1}) \end{bmatrix}$$

, where $\gamma, \mu \in \mathbb{R}$, and $\mu = \frac{1}{2} \frac{(x_t - x_{t+1})cos(\theta) + (y_t - y_{t+1})sin(\theta)}{(y_t - y_{t+1})cos(\theta) - (x_t - x_{t+1})sin(\theta)}$. Then x_c, y_c can be reformulated as

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \frac{x_t + x_{t+1}}{2} + \frac{1}{2} \frac{(x_t - x_{t+1})cos(\theta) + (y_t - y_{t+1})sin(\theta)}{(y_t - y_{t+1})cos(\theta) - (x_t - x_{t+1})sin(\theta)} (y_t - y_{t+1}) \\ \frac{y_t + y_{t+1}}{2} + \frac{1}{2} \frac{(x_t - x_{t+1})cos(\theta) + (y_t - y_{t+1})sin(\theta)}{(y_t - y_{t+1})cos(\theta) - (x_t - x_{t+1})sin(\theta)} (x_t - x_{t+1}) \end{bmatrix}$$

Hence, $r = \sqrt{(x_t - x_c)^2 + (y_t - y_c)^2}$ and $\delta\theta = atan2(y_{t+1} - y_t, x_{t+1} - x_t)$ can also be determined

Estimate transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t, u_{t+1})$

- Since $\delta dis = r \cdot \delta\theta$ can be determined, control input

$$\hat{u} = \begin{bmatrix} \hat{v} \\ \hat{\omega} \end{bmatrix} = \delta t^{-1} \begin{bmatrix} \delta dist \\ \delta\theta \end{bmatrix}$$

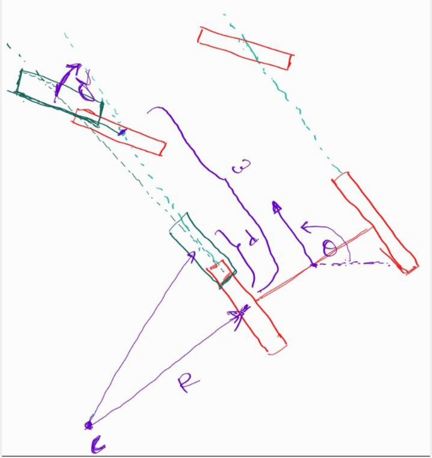
- Final heading

$$\hat{\gamma} = \delta t^{-1}(\theta t + 1 - \theta_t) - \hat{\omega}$$

- Motion error $v_{err} = v - \hat{v}, \omega_{err} = \omega - \hat{\omega}$, and $\gamma_{err} = \hat{\gamma}$
- Corresponding motion error probabilities $\varepsilon_{\alpha_1 v_t^2 + \alpha_2 \omega_t^2}(v_{err}), \varepsilon_{\alpha_3 v_t^2 + \alpha_4 \omega_t^2}(\omega_{err})$, and $\varepsilon_{\alpha_5 v_t^2 + \alpha_6 \omega_t^2}(\gamma_{err})$
- The desired transition probability

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, u_{t+1}) = \varepsilon_{\alpha_1 v_t^2 + \alpha_2 \omega_t^2}(v_{err}) \cdot \varepsilon_{\alpha_3 v_t^2 + \alpha_4 \omega_t^2}(\omega_{err}) \cdot \varepsilon_{\alpha_5 v_t^2 + \alpha_6 \omega_t^2}(\gamma_{err})$$

Consider the following robot model



The front tire is pointing in direction α relative to the wheelbase. Over a short time period the car moves forward and the rear wheel ends up further ahead and slightly turned inward, as depicted with the green dotted tire. Over such a short time frame we can approximate this as a turn around a C .

Prove that

- $\beta = \frac{d}{w} tan(\alpha)$
- $R = \frac{d}{\beta}$, where $d = \delta tv$, if robot robot move with v forward velocity for δt time
- The position of the C is given by
  $Cx = x - Rsin(\theta), \quad Cy = y + Rcos(\theta)$
- If robot move forward for time δt, the new pose is given by
  $$\begin{bmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{bmatrix} = \begin{bmatrix} x - Rsin(\theta) + Rsin(\theta + \beta) \\ y + Rcos(\theta) - Rcos(\theta + \beta) \\ \theta + \beta \end{bmatrix}$$

Remark: $sin(-\theta) = -sin(\theta), cos(-\theta) = cos(\theta)$

How can we define the state variables, control inputs, and the system model?

How can we define the measurement model?

EKF Localization with known correspondence

State variables, control inputs, and the system model

- state variables: $\mathbf{x} = [x\ , y\ , \theta]$
- control inputs: $\mathbf{u} = [v,\ \alpha]$
- $\bar{\mathbf{x}} = \mathbf{x} + f(\mathbf{x}, \mathbf{u}) + N(0, Q)$, where Q is the white noise

Measurement model

If the installed sensor gives a noisy bearing and range to multiple known landmarks, bearing and range can be estimated in the following way, e.g., let p_x, p_y be a landmark location,

$$r = \sqrt{(p_x - x)^2 + (p_y - y)^2}, \quad \phi = arctan(\frac{p_y - y}{p_x - x}) - \theta \qquad (12)$$
$$\mathbf{z} = h(\mathbf{x}, P) + N(0, R),$$

R is the white noise

EKF Localization with known correspondence

- Approximated motion model, i.e., replacing true motion $\hat{v}_t$ and $\hat{\omega}_t$ by executed control (v_t, ω_t)

$$\underbrace{\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix}}_{\mathbf{x}_{t+1}} = \underbrace{\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t} sin(\theta) + \frac{v_t}{\omega_t} sin(\theta + \omega_t \delta t) \\ \frac{v_t}{\omega_t} cos(\theta) - \frac{v_t}{\omega_t} cos(\theta + \omega_t \delta t) \\ \omega_t \delta t \end{pmatrix}}_{f(u_t, \mathbf{x}_t)} + N(0, Q_t) \tag{13}$$

- Let μ_{t-1}, Σ_{t-1} be the previous optimal state $(\hat{\mathbf{x}}^{-}_{t-1})$ as a Gaussian distribution

- Taylor expansion is used to linearize the function $f(u_t, \mathbf{x}_{t-1})$

$$f(u_t, \mathbf{x}_{t-1}) \approx f(u_t, \hat{\mathbf{x}}^{-}_{t-1}) + \Phi_t(\mathbf{x}_{t-1} - \hat{\mathbf{x}}^{-}_{t-1}) \tag{14}$$

$$\Phi_t = \frac{\partial f(u_t, \hat{\mathbf{x}}^{-}_{t-1})}{\partial \hat{\mathbf{x}}^{-}_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \hat{\mathbf{x}}^{-}_{t-1,x}} & \frac{\partial x'}{\partial \hat{\mathbf{x}}^{-}_{t-1,y}} & \frac{\partial x'}{\partial \hat{\mathbf{x}}^{-}_{t-1,\theta}} \\ \frac{\partial y'}{\partial \hat{\mathbf{x}}^{-}_{t-1,x}} & \frac{\partial y'}{\partial \hat{\mathbf{x}}^{-}_{t-1,y}} & \frac{\partial y'}{\partial \hat{\mathbf{x}}^{-}_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \hat{\mathbf{x}}^{-}_{t-1,x}} & \frac{\partial \theta'}{\partial \hat{\mathbf{x}}^{-}_{t-1,y}} & \frac{\partial \theta'}{\partial \hat{\mathbf{x}}^{-}_{t-1,\theta}} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & \frac{v_t}{\omega_t}(-cos(\hat{\mathbf{x}}^{-}_{t-1,\theta}) + cos(\hat{\mathbf{x}}^{-}_{t-1,\theta} + \omega_t \delta t)) \\ 0 & 1 & \frac{v_t}{\omega_t}(-sin(\hat{\mathbf{x}}^{-}_{t-1,\theta}) + sin(\hat{\mathbf{x}}^{-}_{t-1,\theta} + \omega_t \delta t)) \\ 0 & 0 & 1 \end{pmatrix}$$

where $\hat{\mathbf{x}}^{-}_{t-1} = \hat{\mathbf{x}}^{-}_{t-1,x}, \hat{\mathbf{x}}^{-}_{t-1,y}, \hat{\mathbf{x}}^{-}_{t-1,\theta}$ denotes the mean estimate factored into its individual three values

- Motion model with respect to control

$$V_t = \frac{\partial f(u_t, \hat{\mathbf{x}}_{t-1}^{-})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{-sin(\theta)+sin(\theta+\omega_t\delta t)}{\omega_t} & \frac{v_t(sin(\theta)-sin(\theta+\omega_t\delta t))}{\omega_t^2} + \frac{v_t(cos(\theta+\omega_t\delta t)\delta t}{\omega_t} \\ \frac{cos(\theta)-cos(\theta+\omega_t\delta t)}{\omega_t} & -\frac{v_t(cos(\theta)-cos(\theta+\omega_t\delta t))}{\omega_t^2} + \frac{v_t(sin(\theta+\omega_t\delta t)\delta t}{\omega_t} \\ 0 & \delta t \end{pmatrix} \tag{15}$$

Let’s calculate using sympy
https://colab.research.google.com/drive/1Zd3ymJ0Cq83X_G1eTQXpJxPFiLBtS8Bh?usp=sharing

- Correction step: sensor reading

$$\underbrace{\begin{bmatrix} r_t^i \\ \theta_t^i \end{bmatrix}}_{z_t^i} = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ atan2(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix}}_{h(x_{t,j,m})} + N(0, R) \tag{16}$$

, where $m_{j,x}, m_{j,y}$ denotes the coordinates of jth landmark detection at time t, $R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$, and $\mathbf{x}_{t,x}^- = x, \mathbf{x}_{t,y}^- = y$

- The Taylor approximation of the measurement model $h(x_t, j, m) \approx h(\hat{\mathbf{x}}_t^-, j, m) + H_t^i(x_t - \hat{\mathbf{x}}_t^-)$

$$
\begin{aligned}
H_t^i = \frac{\partial h(\hat{\mathbf{x}}_t^-, j, m)}{\partial \hat{\mathbf{x}}_t^-} &= \begin{pmatrix} \frac{\partial r_t^i}{\partial \hat{\mathbf{x}}_{t,x}^-} & \frac{\partial r_t^i}{\partial \hat{\mathbf{x}}_{t,y}^-} & \frac{\partial r_t^i}{\partial \hat{\mathbf{x}}_{t,\theta}^-} \\ \frac{\partial \Phi_t^i}{\partial \hat{\mathbf{x}}_{t,x}^-} & \frac{\partial \Phi_t^i}{\partial \hat{\mathbf{x}}_{t,y}^-} & \frac{\partial \Phi_t^i}{\partial \hat{\mathbf{x}}_{t,\theta}^-} \end{pmatrix} \\
&= \begin{pmatrix} -\frac{m_{j,x} - \hat{\mathbf{x}}_{t,x}^-}{\sqrt{q}} & -\frac{m_{j,y} - \hat{\mathbf{x}}_{t,y}^-}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \hat{\mathbf{x}}_{t,y}^-}{q} & -\frac{m_{j,x} - \hat{\mathbf{x}}_{t,x}^-}{q} & -1 \end{pmatrix}
\end{aligned} \tag{17}
$$

where $q = (m_{j,x} - \hat{\mathbf{x}}_{t,x}^-)^2 + (m_{j,y} - \hat{\mathbf{x}}_{t,y}^-)^2$

We can formulate the location problem with EKF with known correspondence

EKF

$$\boxed{\Phi_t = \left.\frac{\partial f(\mathbf{x}_t, t)}{\partial \mathbf{x}}\right|_{\mathbf{x}_t}}$$

$$\boxed{\hat{\mathbf{x}}_k^- = f(\mathbf{x}_t, \mathbf{t})}$$

$$\mathbf{P}_t^- = \Phi_t \mathbf{P}_t \Phi_t^\top + \mathbf{Q}_t$$

$$\boxed{\mathbf{H} = \left.\frac{\partial h(\hat{\mathbf{x}}_t^-)}{\partial \hat{\mathbf{x}}}\right|_{\hat{\mathbf{x}}_t^-}}$$

$$\mathbf{y} = \mathbf{z}_t - \boxed{h(\hat{x}_t^-)}$$

$$\mathbf{K_t} = \mathbf{P_t^-} \mathbf{H_t}^\top (\mathbf{H_t} \mathbf{P_t^-} \mathbf{H_t}^\top + \mathbf{R}_t)^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_t^- + \mathbf{K_t} \mathbf{y}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K_t} \mathbf{H_t}) \mathbf{P_t^-}$$

References

Clark Chris.
Markovlocalization, September 2022.
[Online; posted 2-September-2022].

Gregor Klancar, Andrej Zdesar, Saso Blazic, and Igor Skrjanc.
Wheeled mobile robotics: from fundamentals towards autonomous systems.
Butterworth-Heinemann, 2017.

Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza.
Introduction to autonomous mobile robots.
MIT press, 2011.

Sebastian Thrun.
Probabilistic robotics.
Communications of the ACM, 45(3):52–57, 2002.