CS 369: Introduction to Robotics

Prof. Thao Nguyen

Spring 2026

HAVERFORD
COLLEGE

Admin

- Lab 4 grades posted on Moodle
- Lab 6 posted tonight
  - group work
  - due Wednesday (Apr 1)
- Research paper presentation sign-up

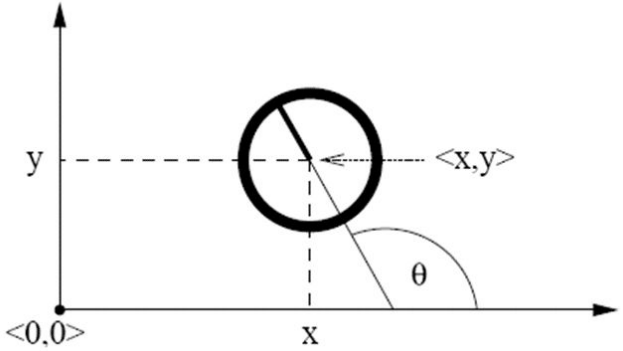
Outline for today

- Motion models
- Sensor models
- Localization algorithms

Outline for today

- Motion models
- Sensor models
- Localization algorithms

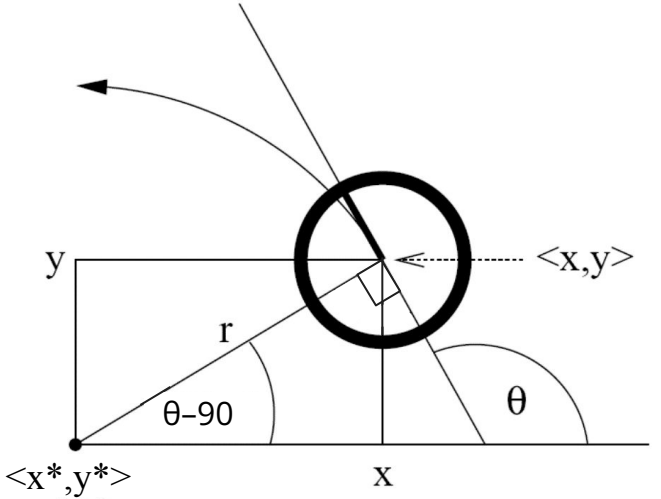
Motion models



- Consider robots operating on a planar surface
  - The state space is three-dimensional: (x,y,θ)
- In practice, one often finds two types of motion models:
  - Odometry-based
  - Velocity-based
- Odometry-based models are used when systems are equipped with wheel encoders
- Velocity-based models calculate the new pose based on the velocities and the time elapsed

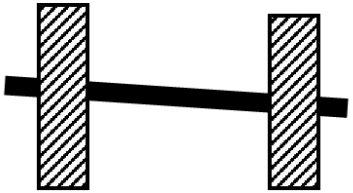
Velocity model

- $u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}$ → translational velocity; → rotational velocity
- $x^* = x - \frac{v}{\omega}\sin\theta$
- $y^* = y + \frac{v}{\omega}\cos\theta$
- $x_t = \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x^* + \frac{v}{\omega}\sin(\theta + \omega\Delta t) \\ y^* - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \theta + \omega\Delta t \end{pmatrix}$

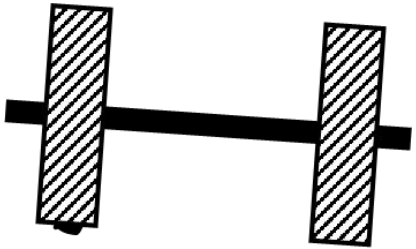


Motion errors

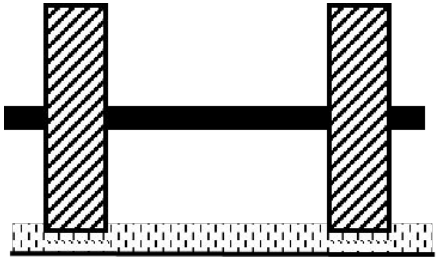
Many reasons:



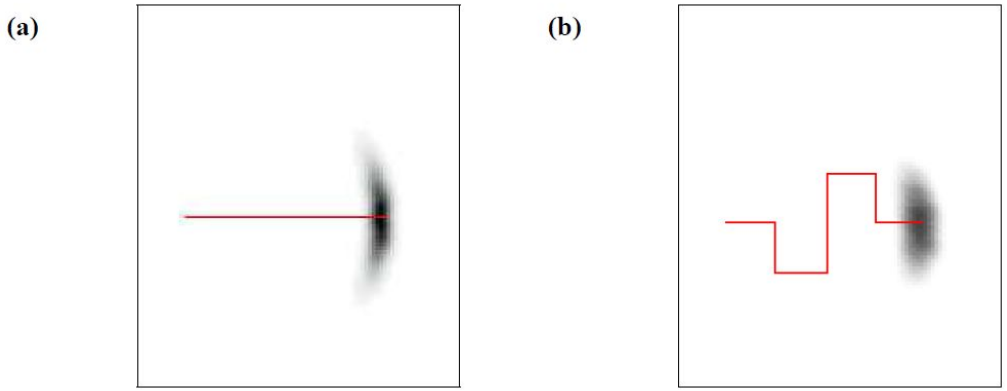
different wheel diameters



bump



carpet



Posterior distributions of the robot's pose upon executing the motion command illustrated by the solid line.

Noise model for velocity

The actual motion is given by the commanded motion corrupted with noise.

$$\hat{v} = v + \varepsilon_{\alpha_1 v^2 + \alpha_2 \omega^2}$$

$$\hat{\omega} = \omega + \varepsilon_{\alpha_3 v^2 + \alpha_4 \omega^2}$$

$$\hat{\gamma} = \varepsilon_{\alpha_5 v^2 + \alpha_6 \omega^2}$$

final rotation

Velocity motion model

Algorithm motion_model_velocity(x_t, u_t, x_{t-1}):

$$\mu = \frac{1}{2}\frac{(x-x')\cos\theta + (y-y')\sin\theta}{(y-y')\cos\theta - (x-x')\sin\theta}$$

$$x^* = \frac{x+x'}{2} + \mu(y-y')$$

$$y^* = \frac{y+y'}{2} + \mu(x'-x)$$

$$r^* = \sqrt{(x-x^*)^2 + (y-y^*)^2}$$

$$\Delta\theta = \text{atan2}(y'-y^*, x'-x^*) - \text{atan2}(y-y^*, x-x^*)$$

$$\hat{v} = \frac{\Delta\theta}{\Delta t} r^*$$

$$\hat{\omega} = \frac{\Delta\theta}{\Delta t}$$

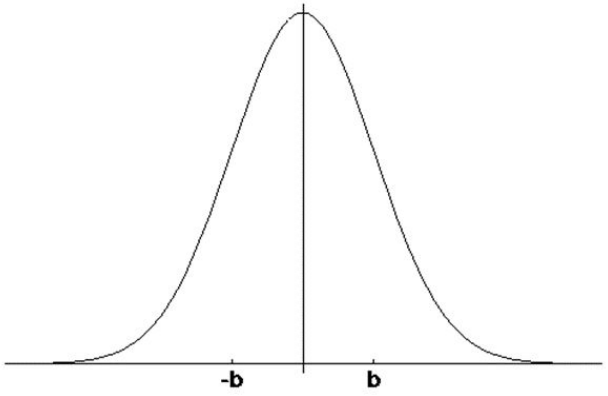
$$\hat{\gamma} = \frac{\theta'-\theta}{\Delta t} - \hat{\omega}$$

return $\textbf{prob}(v-\hat{v}, \alpha_1 v^2 + \alpha_2\omega^2) \cdot \textbf{prob}(\omega-\hat{\omega}, \alpha_3 v^2 + \alpha_4\omega^2) \cdot \textbf{prob}(\hat{\gamma}, \alpha_5 v^2 + \alpha_6\omega^2)$

$x_{t-1} = (x, y, \theta)^T$
$x_t = (x', y', \theta')^T$
$u = (v, \omega)^T$

Noise distributions

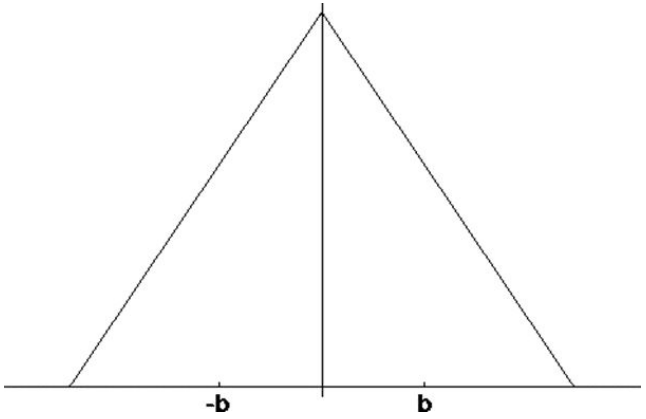
Normal



Algorithm prob_normal_distribution(a, b^2):

return $\frac{1}{\sqrt{2\pi b}}\ e^{-\frac{1}{2}\frac{a^2}{b}}$

Triangular



Algorithm prob_triangular_distribution(a, b^2):

if $|a| > \sqrt{6b}$

return 0

else

return $\frac{\sqrt{6b}-|a|}{6b}$

Sampling

1: **Algorithm sample_motion_model_velocity**(u_t, x_{t-1}):

2: $\hat{v} = v + \textbf{sample}(\alpha_1 v^2 + \alpha_2 \omega^2)$

3: $\hat{\omega} = \omega + \textbf{sample}(\alpha_3 v^2 + \alpha_4 \omega^2)$

4: $\hat{\gamma} = \textbf{sample}(\alpha_5 v^2 + \alpha_6 \omega^2)$

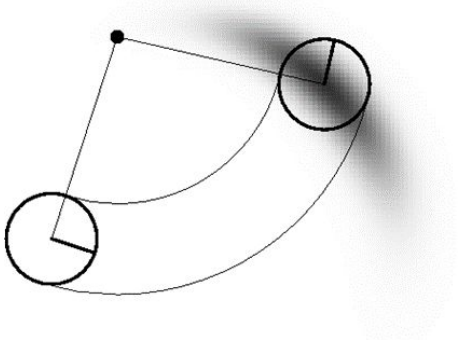
5: $x' = x - \frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}} \; sin(\theta + \hat{\omega}\Delta t)$

6: $y' = y + \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}} \; cos(\theta + \hat{\omega}\Delta t)$

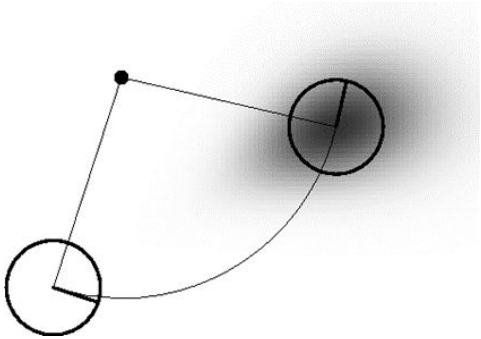
7: $\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$

8: *return* $x_t = (x', y', \theta')^T$

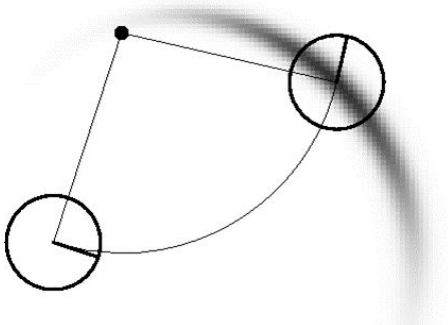
Noise parameters



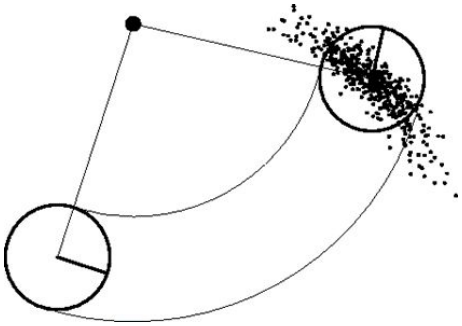
Moderate parameters

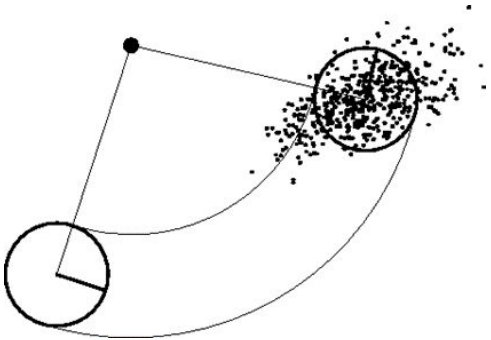


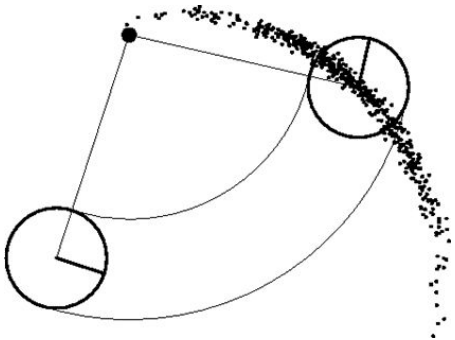
Large translational error (α_1, α_2)
small angular error (α_3, α_4)



Small translational and
large angular error







Map-consistent motion model

$$p(x_t \mid u_t, x_{t-1}) \quad \neq \quad p(x_t \mid u_t, x_{t-1}, m)$$

Approximation: $p(x_t \mid u_t, x_{t-1}, m) \quad = \quad \eta\, p(x_t \mid u_t, x_{t-1})\, p(x_t \mid m)$

Outline for today

- Motion models
- Sensor models
- Localization algorithms

Feature-based sensor model

- Landmarks: distinct objects used for robot navigation
- Sensor provides: landmark's range, bearing, signature

$$f(z_t) = \{f_t^1, f_t^2, \ldots\} = \{\begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^1 \\ s_t^2 \end{pmatrix}, \ldots\}$$

- $p(f(z_t) \mid x_t, m) = \prod_i p(r_t^i, \phi_t^i, s_t^i \mid x_t, m)$

Feature-based map

- A map is a list of objects in the environment along with their properties

$$m = \{m_1, m_2, \ldots, m_N\}$$

- Each object may possess a signature and location coordinate $(m_{i,x}, m_{i,y})$

- $$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \\ s_j \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_r^2} \\ \varepsilon_{\sigma_\phi^2} \\ \varepsilon_{\sigma_s^2} \end{pmatrix}$$

Sensor model

1: **Algorithm landmark_model_known_correspondence(f_t^i, c_t^i, x_t, m):**

2: $j = c_t^i$ → correspondence between feature f_t^i and landmark m_j

3: $\hat{r} = \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2}$

4: $\hat{\phi} = \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta$

5: $q = \textbf{prob}(r_t^i - \hat{r}, \sigma_r^2) \cdot \textbf{prob}(\phi_t^i - \hat{\phi}, \sigma_\phi^2) \cdot \textbf{prob}(s_t^i - s_j, \sigma_s^2)$

6: *return* q

Outline for today

- Motion models
- Sensor models
- Localization algorithms

Grid localization

1: **Algorithm Grid_localization**$(\{p_{k,t-1}\}, u_t, z_t, m)$**:**
2: for all k do
3: $\bar{p}_{k,t} = \sum_i p_{i,t-1}$ **motion_model**$(\text{mean}(\mathbf{x}_k), u_t, \text{mean}(\mathbf{x}_i))$
4: $p_{k,t} = \eta$ **measurement_model**$(z_t, \text{mean}(\mathbf{x}_k), m)\bar{p}_{k,t}$
5: endfor
6: return $\{p_{k,t}\}$

Monte Carlo localization

1: **Algorithm MCL**($\mathcal{X}_{t-1}, u_t, z_t, m$)**:**
2: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
3: for $m = 1$ to M do
4: $x_t^{[m]} = \textbf{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$
5: $w_t^{[m]} = \textbf{measurement\_model}(z_t, x_t^{[m]}, m)$
6: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
7: endfor
8: for $m = 1$ to M do
9: draw i with probability $\propto w_t^{[i]}$
10: add $x_t^{[i]}$ to $\mathcal{X}_t$
11: endfor
12: return $\mathcal{X}_t$

map