Robot Motion

Nart Shawash
ELEE 5810

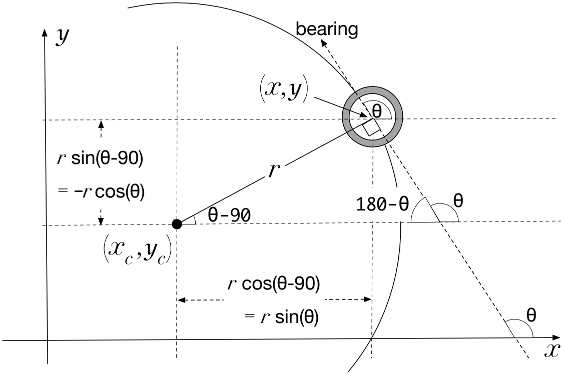
April 19, 2021

Table of contents

State and Control

1. Robot moves in 2D plane.
2. Robot's state in 2D plane is $\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$
    - $\begin{pmatrix} x \\ y \end{pmatrix}$ is the location.
    - θ is the orientation or bearing.
3. Control is $u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}$
    - v_t is the translational velocity.
    - ω_t is the angular velocity.
4. So robot is moving on a circle with radius $r = \left|\frac{v}{\omega}\right|$
5. If $\lim_{\omega \to 0} r = \infty$, then robot moves on a straight line.
6. During Δt, both v_t and ω_t are constant.

Velocity Motion Model §5.3
Robot moving on a circle



$$x = x_c + r \sin\theta$$
$$y = y_c - r \cos\theta$$

$$x_c = x - \frac{v}{\omega} \sin\theta$$
$$y_c = y + \frac{v}{\omega} \cos\theta$$

After Δt the robot will be at

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x_c + \frac{v}{\omega}\sin(\theta+\omega\Delta t) \\ y_c - \frac{v}{\omega}\cos(\theta+\omega\Delta t) \\ \theta+\omega\Delta t \end{pmatrix}$$

$$= \begin{pmatrix} \underbrace{x - \frac{v}{\omega}\sin\theta}_{x_c} + \frac{v}{\omega}\sin(\theta+\omega\Delta t) \\ \underbrace{y + \frac{v}{\omega}\cos\theta}_{y_c} - \frac{v}{\omega}\cos(\theta+\omega\Delta t) \\ \theta+\omega\Delta t \end{pmatrix}$$

$$= \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \underbrace{\begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta+\omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta+\omega\Delta t) \\ \omega\Delta t \end{pmatrix}}_{\text{state update}}$$

Real motion involves noise in the control

$$\underbrace{\begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix}}_{\text{actual}} = \begin{pmatrix} v \\ \omega \end{pmatrix} + \underbrace{\begin{pmatrix} \epsilon_{\alpha_1 v^2 + \alpha_2 \omega^2} \\ \epsilon_{\alpha_3 v^2 + \alpha_4 \omega^2} \end{pmatrix}}_{\text{noise}} \tag{5.10}$$

- Here $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are robot specific parameters.
- Common choices for the noise $\epsilon_{b^2}(a)$ are
  1. Normal with zero mean. $\epsilon_{b^2}(a) \sim \frac{1}{\sqrt{2\pi b^2}} e^{-\frac{a^2}{2b^2}}$
  2. Triangular with zero mean. $\epsilon_{b^2}(a) \sim \max\left\{0, \frac{1}{\sqrt{6}b} - \frac{|a|}{6b^2}\right\}$

Better model for pose

after execution of $u_t = \begin{pmatrix} v \\ \omega \end{pmatrix}$ at $x_{t-1} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$, new state is

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\hat{\omega}}\sin\theta + \frac{\hat{v}}{\hat{\omega}}\sin(\theta + \hat{\omega}\Delta t) \\ \frac{\hat{v}}{\hat{\omega}}\cos\theta - \frac{\hat{v}}{\hat{\omega}}\cos(\theta + \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t \end{pmatrix}$$

Problem

Support of $p(x_t|u_t, x_{t-1})$ is 2D, while x_t is 3D.

Solution

Assume the robot performs a rotation $\hat{\gamma}$ when it arrives at its final pose. So $\boxed{\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t}$ *and* $\boxed{\hat{\gamma} = \epsilon_{\alpha_5 v^2 + \alpha_6 \omega^2}}$

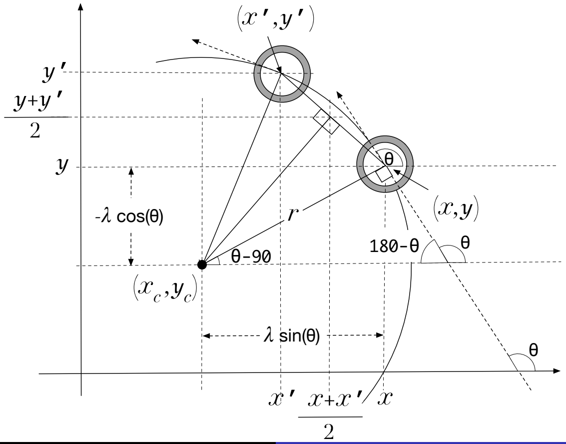
Better model for pose

Now that we have state update given by

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}} \sin(\theta + \hat{\omega}\Delta t) \\ \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}} \cos(\theta + \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t + \hat{\gamma}\Delta t \end{pmatrix}$$

Computation of $p(x_t|u_t, x_{t-1})$

Want to calculate $p(x_t|u_t, x_{t-1})$: control u_t carrying the robot from pose x_{t-1} to x_t in time Δt. First find $\hat{u}_t$ that takes x_{t-1} to x_t regardless of the final orientation.



From the figure above

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin\theta \\ \lambda\cos\theta \end{pmatrix}}_{x_c = x - r\sin\theta \text{ and } y_c = y + r\cos\theta} = \begin{pmatrix} \frac{x+x'}{2} + \mu(y-y') \\ \frac{y+y'}{2} + \mu(x'-x) \end{pmatrix} \quad (5.17)$$

Right hand side term: center of the circle $(x_c \ y_c)^T$ lies on a ray that lies on the half-way point between $(x \ y)^T$ and $(x' \ y')^T$ and is orthogonal to the line between these coordinates. (page 130)
To solve for μ:

- Eliminate λ first.
- Then solve the resulting equation for μ. (assuming $\omega \neq 0$)

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

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} \frac{x+x'}{2} + \frac{1}{2}\frac{(x-x')\cos\theta+(y-y')\sin\theta}{(y-y')\cos\theta-(x-x')\sin\theta}(y+y') \\ \frac{y+y'}{2} + \frac{1}{2}\frac{(x-x')\cos\theta+(y-y')\sin\theta}{(y-y')\cos\theta-(x-x')\sin\theta}(x'-x) \end{pmatrix} \tag{5.19}$$

Moreover, notice that

$$r = \sqrt{(x-x_c)^2+(y-y_c)^2} \overset{\text{or}}{=} \sqrt{(x'-x_c)^2+(y'-y_c)^2}$$

$$\Delta\theta = \underbrace{\text{atan2}(y'-y_c, x'-x_c)}_{\theta_t} - \underbrace{\text{atan2}(y-y_c, x-x_c)}_{\theta_{t-1}}$$

$$\text{atan2} = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0 \\ \text{sign}(y)\left(\pi - \arctan\left|\frac{y}{x}\right|\right) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sign}(y)\,\frac{\pi}{2} & \text{if } x = 0,\ y \neq 0 \end{cases}$$

$$\Delta\text{dist} = r \cdot \Delta\theta$$

From Δdist, $\Delta\theta$ and Δt can compute the velocities $\hat{v}$ and $\hat{\omega}$

$$\hat{u}_t = \begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix} = \begin{pmatrix} \frac{\Delta \text{dist}}{\Delta t} \\ \frac{\Delta\theta}{\Delta t} \end{pmatrix}$$

The rotational velocity $\hat{\gamma}$ is obtained from $\boxed{\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t}$

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

Computation of $p(x_t | u_t, x_{t-1})$

The *motion error* is the deviation of $\hat{u}_t$ and $\hat{\gamma}_t$ from the command $u_t = (v \ \omega)^T$ and $\gamma = 0$

$$v_{\text{err}} = v - \hat{v}$$
$$\omega_{\text{err}} = \omega - \hat{\omega}$$
$$\gamma_{\text{err}} = \hat{\gamma}$$

From (5.10) and $\boxed{\hat{\gamma} = \epsilon_{\alpha_5 v^2 + \alpha_6 \omega^2}}$, probabilities of these errors are

$$\epsilon_{\alpha_1 v^2 + \alpha_2 \omega^2}(v_{\text{err}})$$
$$\epsilon_{\alpha_3 v^2 + \alpha_4 \omega^2}(\omega_{\text{err}})$$
$$\epsilon_{\alpha_5 v^2 + \alpha_6 \omega^2}(\gamma_{\text{err}})$$

Assuming independence, we get

$$p(x_t | u_t, x_{t-1}) = \epsilon_{\alpha_1 v^2 + \alpha_2 \omega^2}(v_{\text{err}}) \cdot \epsilon_{\alpha_3 v^2 + \alpha_4 \omega^2}(\omega_{\text{err}}) \cdot \epsilon_{\alpha_5 v^2 + \alpha_6 \omega^2}(\gamma_{\text{err}})$$

Motion-Model-Velocity(x_t, u_t, x_{t-1}), page 123

1. Algorithm Motion-Model-Velocity(x_t, u_t, x_{t-1}):
2. $\mu = \frac{1}{2}\frac{(x-x')\cos\theta+(y-y')\sin\theta}{(y-y')\cos\theta-(x-x')\sin\theta}$
3. $x_c = \frac{x+x'}{2} + \mu(y - y')$
4. $y_c = \frac{y+y'}{2} + \mu(x' - x)$
5. $r_c = \sqrt{(x - x_c)^2 + (y - y_c)^2}$
6. $\Delta\theta = \text{atan2}(y' - y_c, x' - x_c) - \text{atan2}(y - y_c, x - x_c)$
7. $\hat{v} = \frac{\Delta\theta}{\Delta t} r_c$
8. $\hat{\omega} = \frac{\Delta\theta}{\Delta t}$
9. $\hat{\gamma} = \frac{\theta' - \theta}{\Delta t} - \hat{\omega}$
10. return:

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

Odometry Motion Model §5.4

- Practice shows odometry to be more accurate than velocity.
- Odometry is available only after the robot has moved.
- Odometric information are sensor measurements, not controls.

Problem

Modeling odometry as measurements , requires Bayes filter to include actual velocity as state variables–which increases the dimension of the state space.

Solution

To keep state space small, it is common to consider odometry data as as if it were control signals.

Format of control information

- Odometry model uses *relative motion information*, as measured by robot's internal odometry.
- In the time interval $(t-1, t]$ robot advances from pose x_{t-1} to pose x_t.
- The odometry reports back a related advance

  from $\bar{x}_{t-1} = \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{\theta} \end{pmatrix}$ to $\bar{x}_t = \begin{pmatrix} \bar{x}' \\ \bar{y}' \\ \bar{\theta}' \end{pmatrix}$.
- Relation between bar coordinates and global coordinates is unknown.
- Relative difference between $\bar{x}_{t-1}$ and $\bar{x}_t$ is a good estimator of the difference between poses x_{t-1} and x_t.

Format of control information

- The motion u_t information is given by the pair

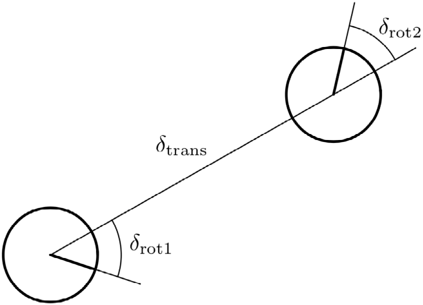
$$u_t = \begin{pmatrix} \overline{x}_{t-1} \\ \overline{x}_t \end{pmatrix}$$

- To extract relative odometry, u_t is transformed into a sequence of three steps:
  1. Rotation (initial turn) δ_{rot1}. Followed by
  2. Straight line motion (translation) δ_{trans}. Followed by
  3. Second rotation δ_{rot2}.

Fact

Each pair of positions $(\bar{s}\ \bar{s}')$ has a unique parameter vector $\begin{pmatrix} \delta_{rot1} \\ \delta_{trans} \\ \delta_{rot2} \end{pmatrix}$. These three parameters assumed to be corrupted by independent noise(s).

Format of control information



Motion-Model-Odometry(x_t, u_t, x_{t-1}), page 134

1. Algorithm Motion-Model-Odometry(x_t, u_t, x_{t-1}):
2. $\delta_{\text{rot1}} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
3. $\delta_{\text{trans}} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$
4. $\delta_{\text{rot2}} = \bar{\theta}' - \bar{\theta} - \delta_{\text{rot1}}$
5. $\hat{\delta}_{\text{rot1}} = \text{atan2}(y' - y, x' - x) - \theta$
6. $\hat{\delta}_{\text{trans}} = \sqrt{(x - x')^2 + (y - y')^2}$
7. $\hat{\delta}_{\text{rot2}} = \theta' - \theta - \hat{\delta}_{\text{rot1}}$
8. $p_1 = \textbf{prob}(\delta_{\text{rot1}} - \hat{\delta}_{\text{rot1}}, \alpha_1 \hat{\delta}^2_{\text{rot1}} + \alpha_2 \hat{\delta}^2_{\text{trans}})$
9. $p_2 = \textbf{prob}(\delta_{\text{trans}} - \hat{\delta}_{\text{trans}}, \alpha_3 \hat{\delta}^2_{\text{trans}} + \alpha_4 \hat{\delta}^2_{\text{rot1}} + \alpha_4 \hat{\delta}^2_{\text{rot2}})$
10. $p_3 = \textbf{prob}(\delta_{\text{rot2}} - \hat{\delta}_{\text{rot2}}, \alpha_1 \hat{\delta}^2_{\text{rot2}} + \alpha_2 \hat{\delta}^2_{\text{trans}})$
11. return: $p_1 \cdot p_2 \cdot p_3$

Motion and Maps §5.4

Definition

A *map* m of the environment is a *list of objects* in the environment and their *locations*. $\boxed{m = \{m_1, m_2, m_3, \ldots, m_N\}}$.

- Feature based. Constructs the shapes in the environment from sensor data.
- Location based. *Occupancy grid* assigns $\{0, 1\}$ to each (x, y).

1. The motion model thus far: $p(x_t|u_t, x_{t-1})$ assumes that robot is moving on an empty plane.
2. Occupancy maps m, distinguish traversable (free) terrain from an occupied one.
3. Map based motion model: $p(x_t|u_t, x_{t-1}, m)$, computes the likelihood that robot placed in a world with map m, will arrive at pose x_t upon executing action u_t at pose x_{t-1}.

Motion and Maps

Problem

No guarantee that unoccupied path exists between x_{t-1} and x_t (after executing u_t).

Solution

If the distance between x_{t-1} and x_t is small (compared to diameter of the robot), then can approximate

$$p(x_t|u_t, x_{t-1}, m) \approx \eta \frac{p(x_t|u_t, x_{t-1})p(x_t|m)}{p(x_t)}$$

Here $p(x_t|m)$ expresses the "consistency" of pose x_t with map m.

Apply Bayes rule $p(A|B, C, D) = \frac{p(B|A,C,D)p(A,C,D)}{p(B|C,D)}$ twice

$$
\begin{aligned}
p(x_t|u_t, x_{t-1}, m) &= \frac{\overbrace{p(m|u_t, x_{t-1}, x_t)}^{\approx p(m|x_t)}\, p(x_t|u_t, x_{t-1})}{p(m|u_t, x_{t-1})} \\
&\approx \eta p(m|x_t) p(x_t|u_t, x_{t-1}) \\
&= \not{\eta} \frac{p(x_t|m)\not{p(m)}}{p(x_t)} p(x_t|u_t, x_{t-1}) \\
&= \eta \frac{p(x_t|u_t, x_{t-1}) p(x_t|m)}{p(x_t)}
\end{aligned}
$$