Introduction to Mobile Robotics

Probabilistic Motion Models

Wolfram Burgard, Cyrill Stachniss,
Maren Bennewitz, Kai Arras
Robot Motion

- Robot motion is inherently uncertain.
- How can we model this uncertainty?
Dynamic Bayesian Network for Controls, States, and Sensations
Probabilistic Motion Models

- To implement the Bayes Filter, we need the transition model \( p(x_t \mid x_{t-1}, u_t) \)

- The term \( p(x_t \mid x_{t-1}, u_t) \) specifies a posterior probability, that action \( u \) carries the robot from \( x_{t-1} \) to \( x_t \).

- In this section we will specify, how \( p(x_t \mid x_{t-1}, u_t) \) can be modeled based on the motion equations.
Coordinate Systems

- The configuration of a typical wheeled robot in 3D can be described by six parameters.
- Three-dimensional Cartesian coordinates plus three Euler angles roll, pitch, and yaw.
- Throughout this section, we consider robots operating on a planar surface.
- The state space of such systems is three-dimensional \((x, y, \theta)\).
Typical Motion Models

- In practice, one often finds two types of motion models:
  - Odometry-based
  - Velocity-based (dead reckoning)

- Odometry-based models are used when systems are equipped with wheel encoders.
- Velocity-based models have to be applied when no wheel encoders are given.
- They calculate the new pose based on the velocities and the time elapsed.
Example Wheel Encoders

These modules provide +5V output when they "see" white, and a 0V output when they "see" black.

These disks are manufactured out of high quality laminated color plastic to offer a very crisp black to white transition. This enables a wheel encoder sensor to easily see the transitions.

Source: http://www.active-robots.com/
Dead Reckoning

- Derived from “deduced reckoning.”
- Mathematical procedure for determining the present location of a vehicle.
- Achieved by calculating the current pose of the vehicle based on its velocities and the time elapsed.
- Historically used to log the position of ships.

[Image source: Wikipedia, LoKiLeCh]
Reasons for Motion Errors

- ideal case
- bump
- different wheel diameters
- carpet

and many more ...
Odometry Model

- Robot moves from \(\langle \bar{x}, \bar{y}, \bar{\theta} \rangle\) to \(\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle\).
- Odometry information \(u = \langle \delta_{\text{rot}1}, \delta_{\text{rot}2}, \delta_{\text{trans}} \rangle\).

\[
\begin{align*}
\delta_{\text{trans}} &= \sqrt{(x' - x)^2 + (y' - y)^2} \\
\delta_{\text{rot}1} &= \text{atan2}(y' - y, x' - x) - \bar{\theta} \\
\delta_{\text{rot}2} &= \bar{\theta}' - \bar{\theta} - \delta_{\text{rot}1}
\end{align*}
\]
The atan2 Function

- Extends the inverse tangent and correctly copes with the signs of x and y.

\[
\text{atan2}(y, x) = \begin{cases} 
\text{atan}(y/x) & \text{if } x > 0 \\
\text{sign}(y) \left( \pi - \text{atan}(|y/x|) \right) & \text{if } x < 0 \\
0 & \text{if } x = y = 0 \\
\text{sign}(y) \pi/2 & \text{if } x = 0, y \neq 0
\end{cases}
\]
Noise Model for Odometry

- The measured motion is given by the true motion corrupted with noise.

\[
\begin{align*}
\hat{\delta}_{\text{rot}1} &= \delta_{\text{rot}1} + \varepsilon_{\alpha_1 |\delta_{\text{rot}1}| + \alpha_2 |\delta_{\text{trans}}|} \\
\hat{\delta}_{\text{trans}} &= \delta_{\text{trans}} + \varepsilon_{\alpha_3 |\delta_{\text{trans}}| + \alpha_4 |\delta_{\text{rot}1} + \delta_{\text{rot}2}|} \\
\hat{\delta}_{\text{rot}2} &= \delta_{\text{rot}2} + \varepsilon_{\alpha_1 |\delta_{\text{rot}2}| + \alpha_2 |\delta_{\text{trans}}|}
\end{align*}
\]
Typical Distributions for Probabilistic Motion Models

Normal distribution

Triangular distribution

\[ \mathcal{E}_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} x^2} \]

\[ \mathcal{E}_{\sigma^2}(x) = \begin{cases} 
0 & \text{if } |x| > \sqrt{6\sigma^2} \\
\frac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} & \text{if } |x| \leq \sqrt{6\sigma^2}
\end{cases} \]
Calculating the Probability Density (zero-centered)

- For a normal distribution
  1. Algorithm `prob_normal_distribution`(a,b):
  2. return \( \frac{1}{\sqrt{2\pi} \cdot b^2} \exp \left\{ -\frac{1}{2} \frac{a^2}{b^2} \right\} \)

- For a triangular distribution
  1. Algorithm `prob_triangular_distribution`(a,b):
  2. return \( \max \left\{ 0, \frac{1}{\sqrt{6} \cdot b} - \frac{|a|}{6 \cdot b^2} \right\} \)
Calculating the Posterior Given x, x', and Odometry

1. Algorithm \textbf{motion\_model\_odometry}(x, x', \overline{x}, \overline{x}')

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

3. \( \delta_{\text{rot1}} = \text{atan2}(y' - \overline{y}, x' - \overline{x}) - \overline{\theta} \)

4. \( \delta_{\text{rot2}} = \theta' - \overline{\theta} - \delta_{\text{rot1}} \)

5. \( \hat{\delta}_{\text{trans}} = \sqrt{(x' - x)^2 + (y' - y)^2} \)

6. \( \hat{\delta}_{\text{rot1}} = \text{atan2}(y' - y, x' - x) - \overline{\theta} \)

7. \( \hat{\delta}_{\text{rot2}} = \theta' - \overline{\theta} - \hat{\delta}_{\text{rot1}} \)

8. \( p_1 = \text{prob}(\delta_{\text{rot1}} - \hat{\delta}_{\text{rot1}}, \alpha_1 \mid \hat{\delta}_{\text{rot1}} \mid + \alpha_2 \hat{\delta}_{\text{trans}}) \)

9. \( p_2 = \text{prob}(\delta_{\text{trans}} - \hat{\delta}_{\text{trans}}, \alpha_3 \hat{\delta}_{\text{trans}} + \alpha_4 (| \hat{\delta}_{\text{rot1}} | + | \hat{\delta}_{\text{rot2}} |)) \)

10. \( p_3 = \text{prob}(\delta_{\text{rot2}} - \hat{\delta}_{\text{rot2}}, \alpha_1 \mid \hat{\delta}_{\text{rot2}} \mid + \alpha_2 \hat{\delta}_{\text{trans}}) \)

11. return \( p_1 \cdot p_2 \cdot p_3 \)
Application

- Repeated application of the sensor model for short movements.
- Typical banana-shaped distributions obtained for the 2d-projection of the 3d posterior.

\[ p(x' | u, x) \]
Sample-Based Density Representation
Sample-Based Density Representation
How to Sample from Normal Distributions?

- Sampling from a normal distribution

1. Algorithm `sample_normal_distribution(b)`:

2. return \( \frac{1}{2} \sum_{i=1}^{12} \text{rand}(-b, b) \)
Normally Distributed Samples

10^6 samples
How to Sample from Normal or Triangular Distributions?

- Sampling from a normal distribution
  
  1. Algorithm `sample_normal_distribution(b)`:
  
        2. return \( \frac{1}{2} \sum_{i=1}^{12} \text{rand}(-b, b) \)

- Sampling from a triangular distribution
  
  1. Algorithm `sample_triangular_distribution(b)`:
  
        2. return \( \frac{\sqrt{6}}{2} [\text{rand}(-b, b) + \text{rand}(-b, b)] \)
For Triangular Distribution

10³ samples

10⁴ samples

10⁵ samples

10⁶ samples
How to Obtain Sample from Arbitrary Functions?
Rejection Sampling

- Sampling from arbitrary distributions
- Sample $x$ from a uniform distribution from $[-b,b]$
- Sample $c$ from $[0, \max f]$
- if $f(x) > c$ keep the sample
  otherwise reject the sample
Rejection Sampling

- Sampling from arbitrary distributions

1. Algorithm \texttt{sample\_distribution}(f,b):
2. repeat
3. \hspace{1cm} \( x = \text{rand}(-b, b) \)
4. \hspace{1cm} \( y = \text{rand}(0, \max\{f(x) \mid x \in [-b, b]\}) \)
5. until \( y \leq f(x) \)
6. return \( x \)
Example

- Sampling from

\[ f(x) = \begin{cases} 
\text{abs}(x) & x \in [-1; 1] \\
0 & \text{otherwise}
\end{cases} \]
Sample Odometry Motion Model

1. Algorithm `sample_motion_model(u, x)`:
   \[
   u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, \quad x = \langle x, y, \theta \rangle
   \]
   1. \[\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 | \delta_{rot1} | + \alpha_2 \delta_{trans})\]
   2. \[\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_2 \delta_{trans} + \alpha_4 (| \delta_{rot1} | + | \delta_{rot2} |))\]
   3. \[\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 | \delta_{rot2} | + \alpha_2 \delta_{trans})\]
   4. \[x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})\]
   5. \[y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})\]
   6. \[\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}\]
   7. Return \(\langle x', y', \theta' \rangle\)
Examples (Odometry-Based)
Sampling from Our Motion Model
Velocity-Based Model

\[ u = (v, \omega)^T \]
Equation for the Velocity Model

\[ x_{t-1} = (x, y, \theta)^T \]
\[ x_t = (x', y', \theta')^T \]

Center of circle:

\[
\begin{pmatrix}
  x^* \\
  y^*
\end{pmatrix}
= \begin{pmatrix}
  x \\
  y
\end{pmatrix} + \begin{pmatrix}
  -\lambda \sin \theta \\
  \lambda \cos \theta
\end{pmatrix}
\]

some constant

(center of circle is orthogonal to the initial heading)
Equation for the Velocity Model

\[ x_{t-1} = (x, y, \theta)^T \]
\[ x_t = (x', y', \theta')^T \]

Center of circle:

\[
\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin \theta \\ \lambda \cos \theta \end{pmatrix} = \begin{pmatrix} \frac{x + x'}{2} + \mu(y - y') \\ \frac{y + y'}{2} + \mu(x' - x) \end{pmatrix}
\]

some constant (circle’s center lies on a ray half way between \(x\) and \(x'\) and is orthogonal to the line between \(x\) and \(x'\)).
Equation for the Velocity Model

\[
x_{t-1} = (x, y, \theta)^T
\]
\[
x_t = (x', y', \theta')^T
\]

Center of circle:

\[
\begin{pmatrix}
x^* \\
y^*
\end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin \theta \\ \lambda \cos \theta \end{pmatrix} = \begin{pmatrix} \frac{x + x'}{2} + \mu (y - y') \\ \frac{y + y'}{2} + \mu (x' - x) \end{pmatrix}
\]

Allows us to solve the equations to:

\[
\mu = \frac{1}{2} \frac{(x - x') \cos \theta + (y - y') \sin \theta}{(y - y') \cos \theta - (x - x') \sin \theta}
\]
Equation for the Velocity Model

\[
x_{t-1} = (x, y, \theta)^T \\
x_t = (x', y', \theta')^T
\]

\[
\begin{pmatrix}
x^* \\
y^*
\end{pmatrix} = \begin{pmatrix}
\frac{x + x'}{2} + \mu(y - y') \\
\frac{y + y'}{2} + \mu(x' - x)
\end{pmatrix}
\]

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

and

\[
r^* = \sqrt{(x' - x)^2 + (y' - y)^2}
\]

\[
\Delta \theta = \text{atan2}(y' - y^*, x' - x^*) - \text{atan2}(y - y^*, x - x^*)
\]
Equation for the Velocity Model

- The parameters of the circle:
  \[ r^* = \sqrt{(x' - x)^2 + (y' - y)^2} \]
  \[ \Delta \theta = \text{atan2}(y' - y^*, x' - x^*) - \text{atan2}(y - y^*, x - x^*) \]

- allow for computing the velocities as

  \[ v = \frac{\Delta \theta}{\Delta t} r^* \]
  \[ \omega = \frac{\Delta \theta}{\Delta t} \]
Noise Model for the Velocity-Based Model

- The measured motion is given by the true motion corrupted with noise.

\[
\hat{v} = v + \varepsilon \alpha_1 |v| + \alpha_2 |\omega|
\]

\[
\hat{\omega} = \omega + \varepsilon \alpha_3 |v| + \alpha_4 |\omega|
\]

- Question: What is the disadvantage of this noise model?
Noise Model for the Velocity-Based Model

- The circle constrains the final orientation
- 2D manifold in a 3D space
- Better approach:

\[
\hat{v} = v + \epsilon \alpha_1 |v| + \alpha_2 |\omega|
\]

\[
\hat{\omega} = \omega + \epsilon \alpha_3 |v| + \alpha_4 |\omega|
\]

\[
\hat{\gamma} = \epsilon \alpha_5 |v| + \alpha_6 |\omega|
\]

Term to account for the final rotation
Motion Including $3^{\text{rd}}$ Parameter

\[
x' = x - \frac{\dot{v}}{\omega} \sin \theta + \frac{\dot{v}}{\omega} \sin(\theta + \dot{\omega} \Delta t)
\]
\[
y' = y + \frac{\dot{v}}{\omega} \cos \theta - \frac{\dot{v}}{\omega} \cos(\theta + \dot{\omega} \Delta t)
\]
\[
\theta' = \theta + \dot{\omega} \Delta t + \dot{\gamma} \Delta t
\]

Term to account for the final rotation
Posterior Probability for Velocity Model

1: Algorithm motion_model_velocity($x_t, u_t, x_{t-1}$): $p(x_t \mid x_{t-1}, u_t)$

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

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

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

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

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

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

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

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

10: return $\text{prob}(v - \hat{v}, \alpha_1|v| + \alpha_2|\omega|) \cdot \text{prob}(\omega - \hat{\omega}, \alpha_3|v| + \alpha_4|\omega|) \cdot \text{prob}(\hat{\gamma}, \alpha_5|v| + \alpha_6|\omega|)$
Sampling from Velocity Model

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

2: \[ \hat{v} = v + \text{sample}(\alpha_1 |v| + \alpha_2 |\omega|) \]

3: \[ \hat{\omega} = \omega + \text{sample}(\alpha_3 |v| + \alpha_4 |\omega|) \]

4: \[ \hat{\gamma} = \text{sample}(\alpha_5 |v| + \alpha_6 |\omega|) \]

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: \[ \text{return } x_t = (x', y', \theta')^T \]
Examples (Velocity-Based)
Map-Consistent Motion Model

\[ p(x' | u, x) \neq p(x' | u, x, m) \]

Approximation: \[ p(x' | u, x, m) = \eta p(x' | m) p(x' | u, x) \]
Summary

- We discussed motion models for odometry-based and velocity-based systems.
- We discussed ways to calculate the posterior probability $p(x'|x,u)$.
- We also described how to sample from $p(x'|x,u)$.
- Typically the calculations are done in fixed time intervals $\Delta t$.
- In practice, the parameters of the models have to be learned.
- We also discussed an extended motion model that takes the map into account.