Robot Mapping

A Short Introduction to the Bayes Filter and Related Models

Cyrill Stachniss

State Estimation

- Estimate the state $x$ of a system given observations $z$ and controls $u$

Goal:

$$p(x \mid z, u)$$

Recursive Bayes Filter 1

$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$$

Definition of the belief

Recursive Bayes Filter 2

$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) = \eta p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \cdot p(x_t \mid z_{1:t-1}, u_{1:t})$$

Bayes’ rule
Recursive Bayes Filter 3

\[ \text{bel}(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t, z_{1:t-1}, u_{1:t}) \ p(\dot{x}_t \mid z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ p(\dot{x}_t \mid z_{1:t-1}, u_{1:t}) \]

Markov assumption

Recursive Bayes Filter 4

\[ \text{bel}(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t, z_{1:t-1}, u_{1:t}) \ p(\dot{x}_t \mid z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \ dx_{t-1} \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \ dx_{t-1} \]

Law of total probability

Recursive Bayes Filter 5

\[ \text{bel}(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t, z_{1:t-1}, u_{1:t}) \ p(\dot{x}_t \mid z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ p(\dot{x}_t \mid z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \ dx_{t-1} \]

Markov assumption

Recursive Bayes Filter 6

\[ \text{bel}(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t, z_{1:t-1}, u_{1:t}) \ p(\dot{x}_t \mid z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \ dx_{t-1} \]
\[ = \eta \ p(\dot{z}_t \mid x_t) \ \int_{x_{t-1}} \ p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \ dx_{t-1} \]

Markov assumption
Recursive Bayes Filter

\[
\text{bel}(x_t) = p(x_t \mid z_{1:t}, u_{1:t})
\]

\[
= \eta p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})
\]

\[
= \eta p(z_t \mid x_t) p(x_t \mid z_{1:t-1}, u_{1:t})
\]

\[
= \eta p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1}
\]

\[
= \eta p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1}
\]

\[
= \eta p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1}
\]

\[
= \eta p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) \text{bel}(x_{t-1}) \, dx_{t-1}
\]

Recursive term

Prediction and Correction Step

- Bayes filter can be written as a two step process
- **Prediction step**
  \[
  \text{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \text{bel}(x_{t-1}) \, dx_{t-1}
  \]
- **Correction step**
  \[
  \text{bel}(x_t) = \eta p(z_t \mid x_t) \text{bel}(x_{t-1})
  \]

Motion and Observation Model

- **Prediction step**
  \[
  \text{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \text{bel}(x_{t-1}) \, dx_{t-1}
  \]

  motion model

- **Correction step**
  \[
  \text{bel}(x_t) = \eta p(z_t \mid x_t) \text{bel}(x_{t-1})
  \]

  sensor or observation model

Different Realizations

- The Bayes filter is a **framework** for recursive state estimation
- There are **different realizations**
- **Different properties**
  - Linear vs. non-linear models for motion and observation models
  - Gaussian distributions only?
  - Parametric vs. non-parametric filters
  - ...
In this Course
- Kalman filter & friends
  - Gaussians
  - Linear or linearized models
- Particle filter
  - Non-parametric
  - Arbitrary models (sampling required)

Motion Model
\[
\tilde{\text{bel}}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \text{bel}(x_{t-1}) \, dx_{t-1}
\]

Robot Motion Models
- Robot motion is inherently uncertain
- How can we model this uncertainty?

Probabilistic Motion Models
- Specifies a posterior probability that action \( u \) carries the robot from \( x \) to \( x' \).
\[
p(x_t \mid u_t, x_{t-1})
\]
Typical Motion Models

- In practice, one often finds two types of motion models:
  - **Odometry-based**
  - **Velocity-based**
- Odometry-based models for systems that are equipped with wheel encoders
- Velocity-based when no wheel encoders are available

Probability Distribution

- Noise in odometry $u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$
- Example: Gaussian noise
  $$u \sim \mathcal{N}(0, \Sigma)$$

Odometry Model

- Robot moves from $(\bar{x}, \bar{y}, \bar{\theta})$ to $(\bar{x}', \bar{y}', \bar{\theta}')$
- Odometry information $u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$
  
  \[
  \begin{align*}
  \delta_{trans} &= \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2} \\
  \delta_{rot1} &= \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta} \\
  \delta_{rot2} &= \bar{\theta}' - \bar{\theta} - \delta_{rot1}
  \end{align*}
  \]

Examples (Odometry-Based)
**Velocity-Based Model**

\[ u = (v, \omega)^T \]

**Motion Equation**

- Robot moves from \((x, y, \theta)\) to \((x', y', \theta')\)
- Velocity information \(u = (v, \omega)\)

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

**Problem of the Velocity-Based Model**

- Robot moves on a circle
- The circle constrains the final orientation
- **Fix:** introduce an additional noise term on the final orientation

**Motion Including 3rd Parameter**

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

Term to account for the final rotation
Examples (Velocity-Based)

Sensor Model

\[ \text{bel}(x_t) = \eta \frac{p(z_t \mid x_t) \text{bel}(x_{t-1})}{\int p(z_t \mid x_t) \text{bel}(x_{t-1}) \, dx_t} \]

Model for Laser Scanners

- Scan \( z \) consists of \( K \) measurements.

\[ z_t = \{ z_t^1, \ldots, z_t^K \} \]

- Individual measurements are independent given the robot position

\[ p(z_t \mid x_t, m) = \prod_{i=1}^{K} p(z_t^i \mid x_t, m) \]

Beam-Endpoint Model
Beam-Endpoint Model

Ray-cast Model
- Ray-cast model considers the first obstacle along the line of sight
- Mixture of four models

Feature-based Model for Range-Bearing Sensors
- Range-bearing $z_i^j = (r_i^i, \phi_i^i)^T$
- Robot’s pose $(x, y, \theta)^T$
- Observation of feature $j$ at location $(m_{j,x}, m_{j,y})^T$

$$
\begin{pmatrix}
    r_i^i \\
    \phi_i^i
\end{pmatrix} = 
\begin{pmatrix}
    \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\
    \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta
\end{pmatrix} + Q_i
$$

Summary
- Bayes filter is a framework for state estimation
- Motion and sensor model are the central models in the Bayes filter
- Standard models for robot motion and laser-based range sensing
Literature

On the Bayes filter

- Thrun et al. “Probabilistic Robotics”, Chapter 2
- Course: Introduction to Mobile Robotics, Chapter 5

On motion and observation models

- Thrun et al. “Probabilistic Robotics”, Chapters 5 & 6
- Course: Introduction to Mobile Robotics, Chapters 6 & 7