Welcome to CSE 571
Robotics: Algorithms and Applications

Instructor: Tapomayukh “Tapo” Bhattacharjee
TAs: Aditya “AVK” Vamsikrishna, Brian Hou

Some slides adapted from Dieter Fox and probabilistic-robotics.org
High-level View on Robot Systems

World model

Control system

Sensor data

Actions
Sense - Plan - Act Paradigm

Sensing
Perceive world

Planning
Devise strategy

Control
Execute plan
Robotics Traditionally
Robotics Now
Current Research Trends / Topics

- Self-driving cars, drones
- Manipulation of everyday objects
- Complex household tasks
- Perception: object detection, 3D mapping, tracking, SLAM
- Human robot interaction
- Machine learning: Deep learning, Reinforcement learning
Course Objectives

- Learn about **fundamental algorithms** in robotics
- Obtain **hands-on** experience
- Understand and integrate subfields of robotics
  - state estimation, navigation, planning, controls, learning
- Critically evaluate current research and identify open problems
Course Details

- **Grading**
  - Homeworks (Theory + Programming): 60%
  - Team project: 30%
  - Participation: 10%

- **Office hours (to be scheduled)**

- **Prerequisites:** probability, linear algebra

- **Readings from Probabilistic Robotics, Reinforcement Learning**

- **Websites**
  - [https://courses.cs.washington.edu/courses/cse571/19wi/](https://courses.cs.washington.edu/courses/cse571/19wi/)
  - Piazza, Gradescope, Canvas
A little background

• Model-based:
  • Deterministic robot, full and accurate model available
  • From mid 70’s

*Motion planning (Reif ’79, Schwartz ‘87, Canny ‘87, Latombe ‘91, Kavraki ‘96 etc.) - Residual uncertainty using low-level controllers. With sensing, can react to unseen using Potential fields (Khatib ‘86), Navigation functions (Koditschek ‘87) etc.
A little background

- Model-free:
  - *Behavior-based robotics (no internal model)*
  - From mid 80’s

*Kaelbling and Rosenschien ‘91, Brooks ‘86, ‘90, Sensing is important, simple tasks (reactive), Hybrid architecture (Arkin ‘98)*
A little background

• **Probabilistic Robotics:**
  - *Both models and sensing present but incomplete / insufficient*
  - From **mid 90’s** but can be traced back to Kalman (60’s)

Key idea: Explicit representation of uncertainty

(using the calculus of probability theory)

*Smith and Cheeseman ‘86, Occupancy Grid Mapping (Elfes ‘87, Moravec ‘88), Partially Observable planning (Kaelbling ‘98), Particle filters (Dellaert ‘99), and it continues...*
Intro to Probability
Discrete Random Variables

• $X$ denotes a random variable.

• $X$ can take on a countable number of values in $\{x_1, x_2, ..., x_n\}$.

• $P(X=x_i)$, or $P(x_i)$, is the probability that the random variable $X$ takes on value $x_i$.

• $P(\cdot)$ is called probability mass function.

• E.g. $P(\text{Room}) = \langle 0.7, 0.2, 0.08, 0.02 \rangle$
Continuous Random Variables

- $X$ takes on values in the continuum.
- $p(X=x)$, or $p(x)$, is a probability density function.

$$
\Pr(x \in (a, b)) = \int_{a}^{b} p(x) \, dx
$$

- E.g.
Joint and Conditional Probability

• \( P(X=x \text{ and } Y=y) = P(x,y) \)

• If \( X \) and \( Y \) are independent then
  \[ P(x,y) = P(x) \cdot P(y) \]

• \( P(x \mid y) \) is the probability of \( x \) given \( y \)
  \[ P(x \mid y) = \frac{P(x,y)}{P(y)} \]
  \[ P(x,y) = P(x \mid y) \cdot P(y) \]

• If \( X \) and \( Y \) are independent then
  \[ P(x \mid y) = P(x) \]
Law of Total Probability, Marginals

Discrete case

\[ \sum_x P(x) = 1 \]

\[ P(x) = \sum_y P(x, y) \]

\[ P(x) = \sum_y P(x \mid y) P(y) \]

Continuous case

\[ \int p(x) \, dx = 1 \]

\[ p(x) = \int p(x, y) \, dy \]

\[ p(x) = \int p(x \mid y) p(y) \, dy \]
Events

- $P(+x, +y)$?
- $P(+x)$?
- $P(-y \text{ OR } +x)$?
- Independent?

$P(X, Y)$

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>+x</td>
<td>+y</td>
<td>0.2</td>
</tr>
<tr>
<td>+x</td>
<td>-y</td>
<td>0.3</td>
</tr>
<tr>
<td>-x</td>
<td>+y</td>
<td>0.4</td>
</tr>
<tr>
<td>-x</td>
<td>-y</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Marginal Distributions

\[ P(X, Y) \]

\[
\begin{array}{ccc}
   \text{X} & \text{Y} & \text{P} \\
   +x & +y & 0.2 \\
   +x & -y & 0.3 \\
   -x & +y & 0.4 \\
   -x & -y & 0.1 \\
\end{array}
\]

\[
P(x) = \sum_y P(x, y)
\]

\[
P(y) = \sum_x P(x, y)
\]

\[ P(X) \]

\[
\begin{array}{cc}
   \text{X} & \text{P} \\
   +x & \\
   -x & \\
\end{array}
\]

\[ P(Y) \]

\[
\begin{array}{cc}
   \text{Y} & \text{P} \\
   +y & \\
   -y & \\
\end{array}
\]
Conditional Probabilities

\[ P(X, Y) \]

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>+x</td>
<td>+y</td>
<td>0.2</td>
</tr>
<tr>
<td>+x</td>
<td>-y</td>
<td>0.3</td>
</tr>
<tr>
<td>-x</td>
<td>+y</td>
<td>0.4</td>
</tr>
<tr>
<td>-x</td>
<td>-y</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- \( P(+x \mid +y) \)?
- \( P(-x \mid +y) \)?
- \( P(-y \mid +x) \)?
Bayes Formula

\[ P(x, y) = P(x \mid y)P(y) = P(y \mid x)P(x) \]

\[ \Rightarrow \]

\[ P(x \mid y) = \frac{P(y \mid x) P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}} \]

• Often causal knowledge is easier to obtain than diagnostic knowledge.
• Bayes rule allows us to use causal knowledge.
Conditioning

- Bayes rule and background knowledge:

\[ P(x \mid y, z) = \frac{P(y \mid x, z) \ P(x \mid z)}{P(y \mid z)} \]

\[ P(x \mid y) = \int P(x \mid y, z) \ P(z \mid y) \ dz \]
Conditional Independence

\[ P(x, y \mid z) = P(x \mid z)P(y \mid z) \]

• Equivalent to

\[ P(x \mid z) = P(x \mid z, y) \]

and

\[ P(y \mid z) = P(y \mid z, x) \]
Bayes Filtering
Combining Evidence

• Often the world is *dynamic* since
  • actions carried out by the robot,
  • actions carried out by other agents,
  • or just the *time* passing by

change the world

• How can we *combine the evidence* or *incorporate* such actions?
Typical Actions

- The robot *turns its wheels* to move
- The robot *uses its manipulator* to grasp an object
- Plants grow over *time*...

- Actions are *never carried out with absolute certainty*.
- In contrast to measurements, *actions generally increase the uncertainty*. 
Modeling Belief

- Robot’s internal knowledge about the state

\[ Bel(x_t) = p(x_t | z_{1:t}, u_{1:t}) \]
Modeling Motion

• To incorporate the outcome of an action $u$ into the current “belief”, we use the conditional pdf

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

• This term specifies the pdf that executing $u$ changes the state from $x_{t-1}$ to $x_t$. 
Modeling Sensor

- Generative model of the sensor given the state:

\[ p(z_t | x_t) \]
Bayes Filters: Framework

- **Given:**
  - Initial belief of the system state \( Bel(x_0) \)
  - Sensor model \( p(z_t | x_t) \)
  - Motion model \( p(x_t | x_{t-1}, u_t) \)

- **Wanted:**
  - Estimate of the state \( X \) of a dynamical system.
  - The belief of the state is also called posterior:

\[
Bel(x_t) = p(x_t | z_{1:t}, u_{1:t})
\]
Markov Assumption

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors
Bayes Filters (See notes)

\[ \text{Bel}(x_t) = P(x_t \mid u_1, z_1, \ldots, u_t, z_t) \]

Bayes

\[ = \eta \ P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \ P(x_t \mid u_1, z_1, \ldots, u_t) \]

Markov

\[ = \eta \ P(z_t \mid x_t) \ P(x_t \mid u_1, z_1, \ldots, u_t) \]

Total prob.

\[ = \eta \ P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \]

\[ \ P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \ dx_{t-1} \]

Markov

\[ = \eta \ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \ P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \ dx_{t-1} \]

Markov

\[ = \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \ P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \ dx_{t-1} \]

\[ = \eta \ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \ \text{Bel}(x_{t-1}) \ dx_{t-1} \]
\[
Bel(x_i) = \eta \ P(z_t \mid x_i) \int P(x_t \mid u_t, x_{t-1}) \ Bel(x_{t-1}) \ dx_{t-1}
\]

1. Algorithm **Bayes_filter** \((Bel(x_{t-1}), u_t, z_t)\):

2. for all \(x_t\) do

3. \(\bar{Bel}(x_t) = \int p(x_t \mid x_{t-1}, u_t) Bel(x_{t-1}) dx_{t-1}\)

4. \(Bel(x_t) = \eta p(z_t \mid x_t) \bar{Bel}(x_t)\)

5. endfor

6. return \(Bel(x_t)\)

**Prediction - Correction Cycle**
Bayes Filters: Framework

• **Initial belief** of the system state $Bel(x_0)$
• **Sensor model** $p(z_t \mid x_t)$
• **Motion model** $p(x_t \mid x_{t-1}, u_t)$
• **Belief representation**
  • Gaussians - Kalman Filters
  • Particles - Particle Filter
Representations for Bayesian Robot Localization

Discrete approaches ('95)
- Topological representation ('95)
  - uncertainty handling (POMDPs)
  - occas. global localization, recovery
- Grid-based, metric representation ('96)
  - global localization, recovery

Kalman filters (late-80s)
- Gaussians, unimodal
- approximately linear models
- position tracking

Particle filters ('99)
- sample-based representation
- global localization, recovery

Multi-hypothesis ('00)
- multiple Kalman filters
- global localization, recovery

AI

Robotics
Summary

• Bayes rule allows us to compute probabilities that are hard to assess otherwise.

• Under the Markov assumption, recursive Bayesian updating can be used to efficiently combine evidence.

• Bayes filters are a probabilistic tool for estimating the state of dynamic systems.