16-662 – Robot Autonomy

Localization

Katharina Muelling
kmuelling@nrec.ri.cmu.edu
NSH 4521
Mid-term project reports are due today 27th.

For every day that you are late: you will lose 2 points
What we have learned so far?

• Need to perceive the state of our environment constantly with the help of sensors
• Sensor carry only partial information and are noisy
• Kalman Filters can be used to estimate the most likely state that we are interested in using linear motion models and Gaussian noise
• Addressed the problem of the linear motion model with Extended Kalman Filter
Today’s Lecture

Class Content

- Failure cases of the Kalman Filter
- Kidnapped robot problem
- Formal Probabilistic Filter
- Particle Filter
Localization Problem

• Every movement of the robot increases the uncertainty about its position, due to the uncertainty of the odometry sensors/motion model
• Autonomous robots need good estimates of their position
• Environmental sensors can give additional information about the state but are also quite noisy and might low feedback rates

Key idea of **Probabilistic Localization**:

• Combine position estimates and environmental sensor information to a single more accurate belief of the robots configuration/position.
State Estimation with Kalman Filter

Initial Guess

Move Forward

Prediction

Update

Measurement

\[ x_{t-1} \]

\[ x_t \]

\[ \hat{x}_t \]

\[ y_t \]
Assumption: Linear motion and sensor models and Gaussian noise

System state
\[ x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \]

Measurement
\[ y_t = H_t x_t + \delta_t \]

Process noise
\( \varepsilon_t \sim N(0, R) \)

Measurement noise
\( \delta_t \sim N(0, Q) \)
State Estimation with Kalman Filter

Linear transformation creates a Gaussian

Figure by S. Thrun, CS-223
Problem Cases

World is not linear

Mapping Gaussian function with non-linear function is not Gaussian

Kalman Filter cannot be applied anymore

Figure by S. Thrun, CS-223
Extended Kalman Filter

Solution: Linear approximation at state estimate

Extended Kalman Filter

Figure by S. Thrun, CS-223
Problem Cases

What else could be a problem?

Unimodal distribution

When could this be a problem?
Problem Cases
Problem Cases

**Global localization**: You don’t have a good initial distribution

Every place in the corridor has the same probability!
Problem Cases

Global localization: You don’t have a good initial distribution
Every place in the corridor has the same probability!
Problem Cases

What happens when we **measure** the environment and recognize a door?

Unimodal distribution cannot account for such a scenario!
Problem Cases

What happens when the robot moves?
Problem Cases

What happens next?

Measurement update!

Kidnapped robot problem
Probabilistic Localization

Given: Map of the environment, sensor model, odometry

Interested in: Probability distribution over all robot configurations

\[ P(x_t | u_{0:t-1}, y_{1:t}) \]

Control input/odometry

Sensor information

Posterior or belief

Assuming discrete state space and that \( y_t \) is independent of \( u_{0:t-1} \) and \( y_{1:t-1} \) if \( x_t \)

\[
P(x_t | u_{0:t-1}, y_{1:t}) = \eta_t P(y_t | x_t) \sum_{x_{t-1} \in X} (P(x_t | u_{t-1}, x_{t-1})P(x_{t-1} | u_{0:t-2}, y_{1:t-1}))
\]

Transition model

Observation model

Prior
Probabilistic Localization

\[
P(x_t | u_{0:t-1}, y_{1:t}) = \eta_t P(y_t | x_t) \sum_{x_{t-1} \in X} (P(x_t | u_{t-1}, x_{t-1}) P(x_{t-1} | u_{0:t-2}, y_{1:t-1}))
\]

Prediction

Update
Probabilistic Localization

How is the belief represented in the Kalman Filter?

\[ P(x_t | u_{0:t-1}, y_{1:t}) \]
\[ = \eta_t P(y_t | x_t) \sum_{x_{t-1} \in X} (P(x_t | u_{t-1}, x_{t-1})P(x_{t-1} | u_{0:t-2}, y_{1:t-1})) \]

How do we represent the posterior distribution to avoid the problems of the Kalman Filter?
Probabilistic Localization

**Goal**: Dealing with arbitrary distributions
Particle Filter

**Goal**: Dealing with arbitrary distributions

**Idea**: Don’t use parametric representation of the distribution. Instead: Use multiple samples
**Goal:** Dealing with arbitrary distributions

**Idea:** Don’t use parametric representation of the distribution. Instead: Use multiple samples
**Goal**: Dealing with arbitrary distributions

**Idea**: Don’t use parametric representation of the distribution. Instead: Use multiple samples
Particle Filter

**Goal:** Dealing with arbitrary distributions

**Idea:** Don’t use parametric representation of the distribution. Instead: Use multiple samples
Particle Filter

Represent distribution by a set of $N$ samples:

$$X = \{(x^i, w^i)\}_{i=1:N}$$

State hypothesis  Importance weight

Main steps:
1. Sample from the proposal distribution
2. Compute the importance weights
3. Resample the particles with respect to their weights
Particle Filter

Hypotheses of state

Samples

Probability
Particle Filter

Measurement: laser scan indicates opening to the side
Particle Filter

Measurement: laser scan indicates opening to the side
Particle Filter

Measurement: laser scan indicates opening to the side
Particle Filter

Measurement: laser scan indicates opening to the side
Particle Filter

Measurement: laser scan indicates opening to the side
Particle Filter

Move

Samples
Particle Filter

Measurement

Samples
Particle Filter

Measurement

Samples
Particle Filter

Resample

Samples
Particle Filter

Move

Samples
Particle Filter

Particle Filter

Input: $X, u_{t-1}, y_t$

$$X' = \emptyset$$

For $i = 1: N$ do

$$x^i \sim P(x^i | u_{t-1}, x^i)$$

$$w^i = P(y_t | x^i)$$

$$\eta = \eta + w^i$$

End

For $i = 1: N$ do

$$w^i = \frac{1}{\eta}w^i$$

$$X' = X' \cup \{x^i, w^i\}$$

End

$$X = \text{resample}(X')$$
Particle Filter

\[
P(x_t|u_{0:t-1}, y_{1:t}) = \eta_t P(y_t|x_t) \sum_{x_{t-1} \in X} (P(x_t|u_{t-1}, x_{t-1})P(x_{t-1}|u_{0:t-2}, y_{1:t-1}))
\]

Sample new particle from motion

\[x^i \sim P(x^i|u_{t-1}, x^i)\]

Resample

\[X = \text{resample}(X')\]

Importance weight

\[
w_i = \frac{\eta_t P(y_t|x_t) \sum_{x_{t-1} \in X} (P(x_t|u_{t-1}, x_{t-1})P(x_{t-1}|u_{0:t-2}, y_{1:t-1}))}{\sum_{x_{t-1} \in X} (P(x_t|u_{t-1}, x_{t-1})P(x_{t-1}|u_{0:t-2}, y_{1:t-1}))} \propto P(y_t|x^i)
\]
Particle Filter

Resampling \( X = \text{resample}(X') \)

Goal: draw sample \( i \) with probability \( w^i \)

Why is it important to resample?
Limit the number of particle in less likely regions and have many in more likely areas

How can we implement this?
Particle Filter

Resampling \[ X = \text{resample}(X') \]

Goal: draw sample \( i \) with probability \( w^i \)

Stochastic universal sampling

\[
X = \emptyset \\
\Delta = \text{rand}(0, 1/N) \\
c = w_1 \\
j = 0 \\
\text{For } i = 0: N - 1 \\
\quad u = \Delta + \frac{i}{N} \\
\quad \text{while } u > c \\
\quad \quad j = j + 1 \\
\quad \quad c = c + w^j \\
\quad \text{end} \\
X = X \cup \left( x^j, \frac{1}{N} \right) \\
\text{end}
\]
Summary

Problems of Kalman Filter:
• Transforming the distribution of using a nonlinear function
• Assumes that all distributions are Gaussian

Particle Filter
• Recursive Bayesian Filter
• Represent the posterior by a set of weighted samples
• Main steps
  • The particles are propagated according to the motion model
  • They are weighted according to the likelihood of the observations
  • New particles are drawn with a probability proportional to their weights