A Very Brief Introduction to the Localization and SLAM Problems

Part 2

Alberto Pretto

Thanks to Wolfram Burgard, Giorgio Grisetti, Davide Scaramuzza and Cyrill Stachniss for some slides!
Bayes Filter Recap

- Represent the state in a probabilist way (e.g., in a discrete space, a probability value for each possible state realization)
  \[ \text{Bel} (x_i) = p(x_i) \]

- Update (i.e., \~"move") the probability density using the actions (e.g., motion)
  \[ \text{Bel}'(x_{i+1}) = \int P(x_{i+1} | u, x_i) \text{Bel}(x_i) \, dx_i \]

- Update (i.e., \~"re-weight", or \~"correct") the probability density using the observations (sensory measurements)
  \[ \text{Bel}(x_{i+1}) = P(z \mid x_{i+1}) \text{Bel}'(x_{i+1}) \]
How to represent the state?

• So far: discrete way (i.e., "grid based")
  - Non parametric

• Why not in an analytical way?

Prediction

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

Correction

\[
\text{bel}(x_t) = \eta \, p(z_t \mid x_t) \, \text{bel}(x_t)
\]
Gaussian Densities

Parametric, unimodal densities with special properties

- **Univariate** \( p(x) \sim N(\mu, \sigma^2) \):

  \[
p(x) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
  \]

- **Multivariate** \( p(x) \sim N(\mu, \Sigma) \):

  \[
p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}
  \]
Properties of Gaussians

- **Intuition**: products and linear combinations of Gaussian densities are still Gaussian densities

\[
X \sim N(\mu, \Sigma) \\
Y = AX + B
\] \implies \ Y \sim N(A\mu + B, A\Sigma A^T)

\[
X_1 \sim N(\mu_1, \Sigma_1) \\
X_2 \sim N(\mu_2, \Sigma_2)
\] \implies \ p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}}\right)
Kalman Filter

Estimates the state $x$ of a discrete-time controlled process that is governed by a linear (or linearized) stochastic equation:

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

with measurements:

$$z_t = C_t x_t + \delta_t$$
Kalman Filter: Prediction

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

\[ p(x_t \mid u_t, x_{t-1}) = N(x_t; A_t x_{t-1} + B_t u_t, R_t) \]

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

\[ \downarrow \]

\[ \sim N(x_t; A_t x_{t-1} + B_t u_t, R_t) \]

\[ \downarrow \]

\[ \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}) \]
Kalman Filter: Correction

\[ z_t = C_t x_t + \delta_t \]

\[ p(z_t \mid x_t) = N(z_t; C_t x_t, Q_t) \]

\[ \text{bel}(x_t) = \eta \ p(z_t \mid x_t) \]

\[ \Downarrow \]

\[ \sim N(z_t; C_t x_t, Q_t) \]

\[ \overline{\text{bel}}(x_t) \]

\[ \Downarrow \]

\[ \sim N(x_t; \overline{\mu}_t, \overline{\Sigma}_t) \]
Kalman Filter Algorithm

1. Algorithm **Kalman_filter**( \( \mu_{t-1}, \Sigma_{t-1}, u_t, z_t \)):

2. Prediction:
3. \( \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \)
4. \( \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \)

5. Correction:
6. \( K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \)
7. \( \mu_t = \bar{\mu}_t + K_t (z_t - C_t \mu_t) \)
8. \( \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \)

9. Return \( \mu_t, \Sigma_t \)
With linear models
Without linear models: linearize!
Particle Filters

Represent belief by random samples

Estimation of non-Gaussian, nonlinear processes

"More the samples (particles) are dense in a neighborhood, more the probability is high in such neighborhood"
Localization with Particle Filters

Particle are **pose hypotheses**

- Update each pose hypotheses with the more recent motion, "weight" each pose hypotheses with the more recent sensor reading.
- Poses with high weight (i.e., likelihood) will be easily duplicated, vice versa will be easily deleted.
Localization with Particle Filters
Bayes Filter solutions: recap

- Grid based approaches

- Particle Filters

- Parametric in continuous space
SLAM

Estimate the robot’s path and the map

\[ p(x_{0:T}, m \mid z_{1:T}, u_{1:T}) \]

- distribution
- path
- map
- given
- observations
- controls
Full SLAM Graphical Model

\[ p(x_{0:T}, m \mid z_{1:T}, u_{1:T}) \]
Online SLAM Graphical Model

\[ p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1}) = \]
\[ \int_{x_0} \cdots \int_{x_t} p(x_{0:t+1}, m \mid z_{1:t+1}, u_{1:t+1}) \, dx_t \cdots dx_0 \]
Why is SLAM a hard problem?

Robot path and map are both unknown
Why is SLAM a hard problem?

The mapping between observations and the map is unknown.

Picking **wrong data associations** can have catastrophic consequences.
Motion and Observation Model

"Motion model"

"Observation model"
Three Main Paradigms

Kalman filter
Particle filter
Graph-based
Landmark Based SLAM with KF
Landmark Based SLAM with KF
SLAM with Particle Filters

Particle are **full trajectories hypoteses**

- Update each trajectory with the more recent motion, "weight" each trajectory with the more recent sensor reading.
- Trajectories with high weight (likelihood) will be easily duplicated, vice versa will be easily deleted.