TSRT14: Sensor Fusion
Lecture 9
— Simultaneous localization and mapping (SLAM)
Gustaf Hendeby
gustaf.hendeby@liu.se

Le 9: simultaneous localization and mapping (SLAM)

Whiteboard:
- SLAM problem formulation
- Framework for EKF-SLAM and fastSLAM (with PF and MPF)

Slides:
- Algorithms
- Properties
- Examples and illustrations

Lecture 8: summary

**SIS PF Algorithm**
Choose the number of particles \( N \), a proposal density \( q(x_k^{(i)}|x_{0:k-1}^{(i)},y_{1:k}) \), and a threshold \( N_{th} \) (for instance \( N_{th} = \frac{3}{2} N \)).

- **Initialization**: Generate \( x_0^{(i)} \sim p_{x_0}, i = 1, \ldots, N \).

Iterate for \( k = 1, 2, \ldots \):

1. **Measurement update**: For \( i = 1, 2, \ldots, N \):
   \[ w_k^{(i)} \propto w_k^{(i)} p(y_k|x_k^{(i)}), \text{ and normalize } w_k^{(i)}. \]
2. **Estimation**: MMSE \( \hat{x}_k \approx \sum_{i=1}^{N} w_k^{(i)} x_k^{(i)} \).
3. **Resampling**: Resample with replacement when \( N_{eff} = \frac{1}{\sum_i (w_k^{(i)})^2} < N_{th} \).
4. **Prediction**: Generate samples \( x_{k+1}^{(i)} \sim q(x_k^{(i)}|x_{k-1}^{(i)},y_k) \),
   update the weights \( w_{k+1}^{(i)} \propto w_k^{(i)} \frac{p(x_k^{(i)}|x_{k-1}^{(i)},y_k)}{q(x_k^{(i)}|x_{k-1}^{(i)},y_k)}, \text{ and normalize } w_{k+1}^{(i)} \).
SLAM: problem formulations

**Localization** concerns the estimation of pose from known landmarks.

**Navigation** concerns estimation of pose, velocities and other states from known landmarks.

**Mapping** concerns the estimation of landmark positions from known values of pose.

**SLAM** concerns the joint estimation of pose and landmark positions.

- Variations on the same theme: *Simultaneous navigation and mapping SNAM?!* and *Simultaneous tracking and mapping STAM?!

Original SLAM Application

- Assume a ground robot with three states: \( x = (X, Y, \psi)^T \).
- Robot measures speed and turn rate: \( u = (v, \hat{\psi})^T \).
- Simple dynamics.
- Sensor:
  1. Ranging sensor (sonar, laser scanner, radar) measures distance to obstacles (walls, furniture); tends to hundreds of landmarks.
  2. Vision (camera, Kinect, stereo camera) provides detections (corners, markers, patterns) as potential landmarks; thousands or tens of thousands of landmarks.
- \( P_{k|k}^{xx} \) small matrix, \( P_{k|k}^{mx} \) thin matrix and \( P_{k|k}^{mm} \) large matrix.

Approach?

Both EKF and PF apply to the problem, but how to handle the large dimensions in the best way? Start with studying the basic EKF.

EKF SLAM: model

- Assume a linear(-ized) model
  \[
  x_{k+1} = F x_k + G u_k \quad \text{cov}(v_k) = Q
  \]
  \[
  \hat{m}_{k+1} = \hat{m}_k
  \]
  \[
  y_k = H_k^x x_k + H_k^m (c_{k|k}^{1:m}) \hat{m}_k + e_k, \quad \text{cov}(e_k) = R.
  \]
- The map is represented by \( \hat{m}_k \).
- The index \( c_{k|k}^{1:m} \) relate the observed landmark \( i \) to a map landmark \( j_i \), which affects the measurement model.
- Association is crucial for some sensors (laser, radar, etc.), but less of a problem some applications (camera using image features, microphones using designed pings).
- The state and its covariance matrix
  \[
  \hat{z}_{k|k} = \begin{pmatrix} \hat{x}_{k|k} \\ \hat{m}_{k|k} \end{pmatrix}, \quad P_{k|k} = \begin{pmatrix} P_{k|k}^{xx} & P_{k|k}^{xm} \\ P_{k|k}^{mx} & P_{k|k}^{mm} \end{pmatrix}
  \]

EKF SLAM: basic KF steps

Time update:

\[
\hat{z}_{k|k-1} = \begin{pmatrix} F \\ 0 \end{pmatrix} \hat{z}_{k-1|k-1},
\]

\[
P_{k|k-1} = \begin{pmatrix} F_k P_{k-1|k-1}^{xx} F_k^T + G_k Q_k G_k^T & F_k P_{k-1|k-1}^{xm} \\ P_{k-1|k-1}^{mx} & P_{k-1|k-1}^{mm} \end{pmatrix}^{-1}
\]

Measurement update:

\[
S_k = H_k^x P_{k|x}^{xx} H_k^x + H_k^m P_{k|m}^{xx} H_k^m + H_k^m P_{k|m}^{mx} H_k^m + H_k^m P_{k|m}^{mm} H_k^m + R_k
\]

\[
K_k^x = H_k^x P_{k|x}^{xx} H_k^x + H_k^m P_{k|m}^{xx} H_k^m + H_k^m P_{k|m}^{mx} H_k^m + H_k^m P_{k|m}^{mm} H_k^m + R_k
\]

\[
K_k^m = H_k^m P_{k|m}^{xx} H_k^m + H_k^m P_{k|m}^{mx} H_k^m + H_k^m P_{k|m}^{mm} H_k^m
\]

\[
\varepsilon_k = y_k - H_k^x \hat{z}_{k|k-1} - H_k^m \hat{m}_{k|k-1}
\]

\[
\hat{z}_{k|k} = \hat{z}_{k|k-1} + (K_k^x) \varepsilon_k
\]

\[
P_{k|k} = P_{k|k-1} - \left( \frac{K_k^x}{K_k^m} \right) S_k^{-1} \left( \frac{K_k^x}{K_k^m} \right)^T
\]
EKF SLAM: KF problems

- All elements in $P_{mm}^{k|k}$ are affected by the measurement update.
- It turns out that the cross correlations are essential for performance.
- No simple turn-around.

\[ \hat I_{k|l} = I_{k|l} - \hat P_{k|l}^{-1} \hat H_{k}^{T} R_{k}^{-1} y_{k} \]
\[ I_{k|l} = I_{k|l-1} + H_{k}^{T} R_{k}^{-1} H_{k} \]

Note:
The update is sparse!!!

EKF SLAM: information form

- Focus on sufficient statistics and information matrix

\[ I_{k|l} = I_{k|l} - \hat P_{k|l}^{-1} \hat H_{k}^{T} R_{k}^{-1} y_{k} \]
\[ I_{k|l} = I_{k|l-1} + H_{k}^{T} R_{k}^{-1} H_{k} \]

Note:
The update is sparse!!!

EKF SLAM: information filter algorithm (1/4)

Initialization:

\[ i_{x}^{1|0} = 0_{3 \times 1} \]
\[ i_{m}^{1|0} = 0_{0 \times 0} \]
\[ I_{xx}^{1|0} = I_{3 \times 3} \]
\[ I_{mm}^{1|0} = 0_{0 \times 0} \]

Note:
The information form allows for representing no prior knowledge with zero information (infinite covariance).

EKF SLAM: information filter algorithm (2/4)

1. **Associate** a map landmark $j = c_{k}$ to each observed landmark $j$, and construct the matrix $H_{k}^{m}$. This step includes data gating for outlier rejection and track handling to start and end landmark tracks.

2. **Measurement update:**

\[ i_{x}^{k|k} = i_{x}^{k|k-1} + H_{k}^{xT} R_{k}^{-1} y_{k} \]
\[ i_{m}^{k|k} = i_{m}^{k|k-1} + H_{k}^{mT} R_{k}^{-1} y_{k} \]
\[ I_{xx}^{k|k} = I_{xx}^{k|k-1} + H_{k}^{xT} R_{k}^{-1} H_{k}^{x} \]
\[ I_{mm}^{k|k} = I_{mm}^{k|k-1} + H_{k}^{mT} R_{k}^{-1} H_{k}^{m} \]

Note:
$H_{k}^{m}$ is very thick, but contains mostly zeros.
The low-rank sparse corrections influencing only a fraction of the matrix elements.
EKF SLAM: summary

- EKF SLAM scales well in state dimension.
- EKF SLAM scales badly in landmark dimension, though natural approximations exist for the information form.
- EKF SLAM is not robust to incorrect associations.

FastSLAM: idea

Basic factorization idea:

\[ p(x_{1:k}, m|y_{1:k}) = p(m|x_{1:k}, y_{1:k})p(x_{1:k}|y_{1:k}). \]

- The first factor corresponds to a classical mapping problem, and is solved by the (E)KF.
- The second factor is approximated by the PF.
- Leads to a marginalized PF (MPF) where each particle is a pose trajectory with an attached map corresponding to mean and covariance of each landmark, but no cross-correlations.
FastSLAM: mapping solution (1/3)

Assume observation model linear(ized) in landmark position

\[ 0 = h^0(y^m_k, x_k) + h^1(y^m_k, x_k)m^l_k + e^m_k, \quad \text{cov}(e^m_k) = R^m_k. \]

This formulation covers:
- First order Taylor expansions.
- Bearing and range measurements, where \( h^i(y^m_k, x_k) \) has two rows per landmark in 2D SLAM.
- Bearing-only measurements coming from a camera detection.

FastSLAM: mapping solution (2/3)

Likelihood in the Gaussian case:

\[ p(y^m_k | y_{1:k-1}, x_{1:k}) = \mathcal{N}(h^0(y^m_k, x_k) + h^1(y^m_k, x_k)m^l_{k-1}, R^m_k + h^1(y^m_k, x_k)(I^l_k)^{-1}h^1(y^m_k, x_k)). \]

FastSLAM: the algorithm (1/2)

1. **Initialize** the particles

   \[ x_1^{(i)} \sim p_0(x), \]

   where \( N \) denotes the number of particles.

2. **Data association** that assigns a map landmark \( n_l \) to each observed landmark \( l \). Initialize new map landmarks if necessary.

3. **Importance weights**

   \[ w^{(i)}_k = \prod_l \mathcal{N}(h^0(y^l_k, x_k) + h^1(y^l_k, x_k)m^m_{k-1}, R^m_k + h^1(y^l_k, x_k)(I^m_k)^{-1}h^1(y^l_k, x_k)). \]

   where the product is taken over all observed landmarks \( l \), and normalize \( \tilde{w}^{(i)}_k = w^{(i)}_k / \sum_{j=1}^N w^{(j)}_k \).

4. **Resampling** a new set of particles with replacement

   \[ \Pr(x_k^{(i)} = x_k^{(j)}) = \tilde{w}^{(j)}_k, \quad j = 1, \ldots, N. \]
FastSLAM: the algorithm (2/2)

5. Map measurement update:
\[ p(m^{(i)}|x^{(i)}_{1:k}, y_{1:k}) = \mathcal{N}(\varepsilon m^{(i)}_{k|k-1}, \varepsilon m^{(i)}_{k|k-1}), \varepsilon \mathcal{I}_k - 1) \]
\[ \varepsilon m_k = \varepsilon m_{k-1} + h^T(y_k, x_k)R_k^{-1}h(y_k, x_k), \]
\[ \mathcal{I}_k = \mathcal{I}_{k-1} + h^T(y_k, x_k)R_k^{-1}h(y_k, x_k). \]

6. Pose time update:
fastSLAM 1.0 (SIR PF)
\[ x^{(i)}_{k+1} \sim p(x_{k+1}|x^{(i)}_{1:k}). \]
fastSLAM 2.0 (PF with optimal proposal)
\[ x^{(i)}_{k+1} \sim p(x_{k+1}|x^{(i)}_{1:k}, y_{1:k+1}) \propto p(x_{k+1}|x^{(i)}_{1:k}) p(y_{k+1}|x_{k+1}, x^{(i)}_{1:k}, y_{1:k}). \]

MFastSLAM: idea

Factorization in three factors
\[ p(x^P_{1:k}, x^j_k, m_k|y_{1:k}) = p(m_k|x^j_k, x^P_{1:k}, y_{1:k})p(x^j_k|x^{(i)}_{1:k}, y_{1:k}) p(x^P_{1:k}|y_{1:k}) \]
corresponding to:
- one low-dimensional PF,
- one (E)KF attached to each particle,
- one WLS estimate to each particle and each map landmark.

FastSLAM: summary

FastSLAM is ideal for a ground robot with three states and vision sensors providing thousands of landmarks.
- FastSLAM scales linearly in landmark dimension.
- As the standard PF, FastSLAM scales badly in the state dimension.
- FastSLAM is relatively robust to incorrect associations, since associations are local for each particle and not global as in EKF-SLAM.

MFastSLAM combines all the good landmarks of the EKF SLAM and fastSLAM!

SLAM Illustration

- Airborne simultaneous localization and mapping (SLAM) using UAV with camera.
- Research collaboration with IDA.
- General idea: augment state vector with parameters representing the map.

Comparison of EKF and FastSLAM on same dataset.
Ex: RSS SLAM (1/2)

- Prize winning MSc thesis at FOI 2014.
- Foot-mounted IMU for odometry.
- Opportunistic radio signals for fingerprinting.

Ex: RSS SLAM (2/2)

- Estimate the error in the odometry.
- Gaussian process to represent the map.
- Particle filter solution.

SLAM Examples on Youtube

- LEGO Mindstorm ground robot with sonar  http://youtu.be/7K8dZwqBSSA
- Indoor mapping by UAV with laser scanner  http://youtu.be/1MSozUpFFkU
- Indoor mapping using hand-held stereo camera with IMU at FOI  http://youtu.be/7f-9X001qE
- Intelligent vacuum cleaner using ceiling vision  http://youtu.be/bq5H7zGF3vQ
- App Ball Invasion uses augmented reality based on SLAM (KTH student did the SLAM implementation)  http://youtu.be/WHGtvdxTVZk

Summary
Lecture 9: summary

Simultaneous Localization And Mapping (SLAM)

- Joint estimation of trajectory $x_{1:k}$ and map parameters $\theta$ in sensor model $y_k = h(x_k; \theta) + e_k$.

Algorithms:
- NLS SLAM: iterate between filtering and mapping.
- EKF-SLAM: EKF (information form) on augmented state vector $z_k = (x_k^T, \theta^T)^T$.
- FastSLAM: MPF on augmented state vector $z_k = (x_k^T, \theta^T)^T$.

http://youtu.be/VQlaCHl3Yc4
http://youtu.be/hA_NZeuoy9Y