TSRT14: Sensor Fusion
Lecture 8
Simultaneous localization and mapping (SLAM)

Gustaf Hendeby

Spring 2015
Whiteboard:
- SLAM problem formulation
- Framework for EKF-SLAM and fastSLAM (with PF and MPF)

Slides:
- Algorithms
- Properties
- Examples and illustrations
Basic SIR PF algorithm

Choose the number of particles \( N \).

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

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

1. **Measurement update:** For \( k = 1, 2, \ldots \),
   \[
   w_k^{(i)} = w_{k-1}^{(i)} p(y_k | x_k^{(i)})
   \]

2. **Normalize:** \( w_k^{(i)} := w_k^{(i)} / \sum_i w_k^{(i)} \)

3. **Estimation:** MMSE \( \hat{x}_k \approx \sum_{i=1}^N w_k^i x_k^i \) or MAP

4. **Resampling:** Bayesian bootstrap: Take \( N \) samples with replacement from the set \( \{x_k^{(i)}\}_{i=1}^N \) where the probability to take sample \( i \) is \( w_k^{(i)} \). Let \( w_k^{(i)} = 1/N \).

5. **Prediction:** Generate random process noise samples
   \[
   v_k^{(i)} \sim p_{v_k}, \quad x_{k+1}^{(i)} = f(x_k^{(i)}, v_k^{(i)})
   \]
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)!!*
EKF SLAM: model

- Assume a linear(-ized) model
  \[ x_{k+1} = Fx_k + Gv_k \quad \text{cov}(v_k) = Q \]
  \[ m_{k+1} = m_k \]
  \[ y_k = H^x_k x_k + H^m_k (c^1_{k} : l_k) m_k + e_k, \quad \text{cov}(e_k) = R \]

- The map is represented by \( m_k \)
- The index \( c^1_{k} : l_k \) 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
  \[ \hat{x}_{k|k} = \begin{pmatrix} \hat{x}_{k|k} \\ \hat{m}_{k|k} \end{pmatrix}, \quad P_{k|k} = \begin{pmatrix} P^{xx}_{k|k} & P^{xm}_{k|k} \\ P^{mx}_{k|k} & P^{mm}_{k|k} \end{pmatrix} \]
Original SLAM Application

- Assume a ground robot with three states: $x = (X, Y, \psi)^T$
- Robot measures speed and turn rate: $u = (v, \dot{\psi})^T$
- Simple dynamics
- Sensor:
  1. Ranging sensor (sonar, laser scanner, radar) measures distance to obstacles (walls, furniture); tens 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
- $P_{xx|k}$ small matrix, $P_{mx|k}$ thin matrix and $P_{mm|k}$ 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: basic KF steps

Time update:

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

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

Measurement update:

\[
S_k = H_k^x P_{k|k-1}^{xx} H_k^{xT} + H_k^m P_{k|k-1}^{mm} H_k^{fT} + H_k^m P_{k|k-1}^{mx} H_k^{xT} + H_k^x P_{k|k-1}^{xm} H_k^{mT} + R_k
\]

\[
K_k^x = \left( H_k^x P_{k|k-1}^{xx} + H_k^m P_{k|k-1}^{mx} \right) S_k^{-1}
\]

\[
K_k^m = \left( H_k^x P_{k|k-1}^{xm} + H_k^m P_{k|k-1}^{mm} \right) S_k^{-1}
\]

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

\[
\hat{z}_{k|k} = \hat{z}_{k|k-1} + \begin{pmatrix} K_k^x \\ K_k^m \end{pmatrix} \varepsilon_k
\]

\[
P_{k|k} = P_{k|k-1} - \begin{pmatrix} K_k^x \\ K_k^m \end{pmatrix} S_k^{-1} \begin{pmatrix} K_k^x \\ K_k^m \end{pmatrix}^T
\]
All elements in $P_{k|k}^{mm}$ are affected by the measurement update.

It turns out that the cross correlations are essential for performance.

No simple turn-around.
Focus on sufficient statistics and information matrix

$$\mathbf{\kappa}_k|l = \mathcal{I}_k|l \hat{\mathbf{x}}_k|l$$

$$\mathcal{I}_k|l = P_{k|l}^{-1} = \begin{pmatrix} P_{xx|k}^{|l} & P_{xm|k}^{|l} \\ P_{mx|k}^{|l} & P_{mm|k}^{|l} \end{pmatrix}^{-1} = \begin{pmatrix} \mathcal{I}_{xx|l}^{|k} & \mathcal{I}_{xm|l}^{|k} \\ \mathcal{I}_{mx|l}^{|k} & \mathcal{I}_{mm|l}^{|k} \end{pmatrix}$$

Measurement update trivial

$$\mathbf{\kappa}_k|k = \mathbf{\kappa}_k|k-1 + H_k^T R_k^{-1} y_k$$

$$\mathcal{I}_k|k = \mathcal{I}_k|k-1 + H_k^T R_k^{-1} H_k$$

Note:
The update is sparse!!!
Initialization:

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

Note:

The information form allows for representing no prior knowledge with zero information (infinite covariance).
1. **Associate** a map landmark \( j = c_k^i \) to each observed landmark \( j \), and construct the matrix \( H_k^f \). This step includes data gating for outlier rejection and track handling to start and end landmark tracks.

2. **Measurement update:**

\[
\begin{align*}
\xi_{x|k} &= \xi_{x|k-1} + H_k^{xT} R_k^{-1} y_k \\
\xi_{m|k} &= \xi_{m|k-1} + H_k^{mT} R_k^{-1} y_k \\
I_{x|k} &= I_{x|k-1} + H_k^{xT} R_k^{-1} H_k^x \\
I_{x|k} &= I_{x|k-1} + H_k^{xT} R_k^{-1} H_k^x \\
I_{m|k} &= I_{m|k-1} + H_k^{mT} R_k^{-1} H_k^m \\
I_{m|k} &= I_{m|k-1} + H_k^{mT} R_k^{-1} H_k^m
\end{align*}
\]

**Note:**

\( H_k^m \) is very thick, but contains mostly zeros.

The low-rank sparse corrections influencing only a fraction of the matrix elements.
3. Time update:

\[
\bar{I}_{k|k-1}^{xx} = F_k^{-1} I_{k-1|k-1}^{xx} F_k^{-T}
\]

\[
\bar{I}_{k|k-1}^{xm} = F_k^{-1} I_{k-1|k-1}^{xm}
\]

\[
M_k = G_k \left( G_k^T F_k^{-1} I_{k-1|k-1}^{xx} F_k^{-T} + Q_k^{-1} \right)^{-1} G_k^T,
\]

\[
\bar{I}_{k|k-1}^{xx} = \bar{I}_{k|k-1}^{xx} - \bar{I}_{k|k-1}^{xx} M_k \bar{I}_{k|k-1}^{xx}
\]

\[
\bar{I}_{k|k-1}^{xm} = \bar{I}_{k|k-1}^{xm} - \bar{I}_{k|k-1}^{xx} M_k \bar{I}_{k|k-1}^{xm}
\]

\[
\bar{I}_{k|k-1}^{mm} = \bar{I}_{k|k-1}^{mm} - \bar{I}_{k|k-1}^{mx} M_k G_k^T \bar{I}_{k|k-1}^{xm}
\]

\[
\bar{\imath}_{k|k-1}^{x} = (I - \bar{I}_{k|k-1}^{xx} G_k Q_k G_k^T F_k^T) \bar{\imath}_{k-1|k-1}^{x}
\]

\[
\bar{\imath}_{k|k-1}^{m} = \bar{\imath}_{k-1|k-1}^{m} - \bar{I}_{k|k-1}^{mx} G_k Q_k G_k^T F_k^T \bar{\imath}_{k|k-1}^{x}
\]

Note:

Now, \( \bar{I}_{k|k-1}^{mm} \) is corrected with the inner product of \( \bar{I}_{k|k-1}^{xm} \) which gives a full matrix. Many of the elements in \( \bar{I}_{k|k-1}^{xm} \) are close to zero and may be truncated!
4. Estimation:

\[
P_{k|k} = \mathcal{I}_{k|k}^{-1}, \\
\hat{x}_{k|k} = P_{x|x}^{k|k} \hat{x}_{k|k} + P_{x|m}^{k|k} \hat{m}_{k|k}, \\
\hat{m}_{k|k} = P_{m|x}^{k|k} \hat{x}_{k|k} + P_{m|m}^{k|k} \hat{m}_{k|k}. 
\]

Here is another catch, the information matrix needs to be inverted! The pose is needed at all times for linearization and data gating. How to proceed?

Idea:
Solve

\[
\varphi_{k|l} = \mathcal{I}_{k|l} \hat{z}_{k|l},
\]

directly using a gradient search algorithm initialized at previous estimate.
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
Assume observation model linear(-ized) in landmark position

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

This formulation covers:

- First order Taylor expansions
- Bearing and range measurements, where \( h^i(y^n_k, x_k) \) has two rows per landmark in 2D SLAM
- Bearing-only measurements coming from a camera detection
Linear estimation theory applies.
WLS estimate:

$$
\hat{m}^l = \left( \sum_{k=1}^{N} h^1^T (y_{k}^{n_i}, x_k)(R_k^{n_i})^{-1} h^1 (y_{k}^{n_i}, x_k) \right)^{-1} \mathcal{I}_N^l \\
\sum_{k=1}^{N} -h^1^T (y_{k}^{n_i}, x_k)(R_k^{n_i})^{-1} h^0 (y_{k}^{n_i}, x_k) = (\mathcal{I}_N^l)^{-1} \iota_N^l.
$$

In this sum, $h^i$ is an empty matrix if the map landmark $n$ does not get an associated observation landmark at time $k$.

Under a Gaussian noise assumption, the posterior distribution is Gaussian

$$
(m^l|y_{1:N}, x_{1:N}) \sim \mathcal{N}\left( (\mathcal{I}_N^l)^{-1}\iota_N^l, (\mathcal{I}_N^l)^{-1} \right).
$$
Kalman filter for mapping on information form

\[ i_k^I = i_{k-1}^I + h_1^T(y_k^{nl}, x_k)R_k^{-1}h_0(y_k^{nl}, x_k), \]

\[ I_k^I = I_{k-1}^I + h_1^T(y_k^{nl}, x_k)R_k^{-1}h_1(y_k^{nl}, x_k), \]

\[ \hat{m}^I = (I_k^I)^{-1}i_k^I. \]

Note the problem of inverting a large matrix to compute the landmark positions.

Likelihood in the Gaussian case:

\[
p(y_k^{nl}|y_{1:k-1}, x_{1:k}) = \mathcal{N}(h_0(y_k^{nl}, x_k) + h_1(y_k^{nl}, x_k)\hat{m}_{k-1}^I, R_k^{nl} + h_1(y_k^{nl}, x_k)(I_k^I)^{-1}h_1^T(y_k^{nl}, x_k))
\]
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**

   \[
   w_k^{(i)} = \prod_l \mathcal{N}(h^0(y^l_k, x_k) + h^1(y^l_k, x_k) \hat{m}^{n_l}_{k-1}, R^l_k + h^1(y^l_k, x_k)(I_k^{n_l})^{-1}h^1^T(y^l_k, x_k)).
   \]

   where the product is taken over all observed landmarks \( l \), and normalize

   \[
   \bar{w}_k^{(i)} = w_k^{(i)} / \sum_{j=1}^{N} w_k^{(j)}.
   \]

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

   \[
   \Pr(x_k^{(i)} = x_k^{(j)}) = \bar{w}_k^{(j)}, \quad j = 1, \ldots, N.
   \]
5. Map measurement update:

\[ p(m^{(i)}|x_{1:k}^{(i)}, y_{1:k}) = \mathcal{N}((I_k^{(i)})^{-1} \iota_k^{(i)}, (I_k^{(i)})^{-1}), \]

\[ \iota_k = \iota_{k-1} + h^1^T(y_k, x_k)R_k^{-1}h^0(y_k, x_k), \]

\[ I_k = I_{k-1} + h^1^T(y_k, x_k)R_k^{-1}h^1(y_k, x_k). \]

6. Pose time update:

fastSLAM 1.0 (SIR PF)

\[ x_{k+1}^{(i)} \sim p(x_{k+1}|x_{1:k}^{(i)}). \]

fastSLAM 2.0 (PF with optimal proposal)

\[ x_{k+1}^{(i)} \sim p(x_{k+1}|x_{1:k}^{(i)}, y_{1:k+1}) \]

\[ \propto p(x_{k+1}|x_{1:k}^{(i)})p(y_{k+1}|x_{k+1}, x_{1:k}^{(i)}, y_{1:k}). \]
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!
Factorization in three factors

\[ p(x_{1:k}^P, x_k^j, m_k|y_{1:k}) = p(m_k|x_k^j, x_{1:k}^P, y_{1:k})p(x_k^j|x_{1:k}^P, y_{1:k})p(x_{1:k}^P|y_{1:k}) \]

corresponding to:

- one low-dimensional PF
- one (E)KF attached to each particle
- one WLS estimate to each particle and each map landmark
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.
Prize winning MSc thesis at FOI 2014

Foot-mounted IMU for odometry

Opportunistic radio signals for fingerprinting

![RSS at 95500 kHz](image)
Estimate the error in the odometry
Gaussian process to represent the map
Particle filter solution
LEGO Mindstorm ground robot with sonar
http://youtu.be/7K8dZwqBSSA

Indoor mapping by UAV with laser scanner
http://youtu.be/IMSozUpFFkU

Indoor mapping using hand-held stereo camera with IMU at FOI
http://youtu.be/7f-nqXmo1qE

Intelligent vacuum cleaner using ceiling vision
http://youtu.be/bq5HZzGF3vQ

App *Ball Invasion* uses augmented reality based on SLAM (KTH student did the SLAM implementation)
http://youtu.be/WHGtvdxTVZk
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 $z_k = (x_k^T, \theta^T)^T$