SENSOR FUSION

Simultaneous Localization and Mapping (SLAM): FastSLAM

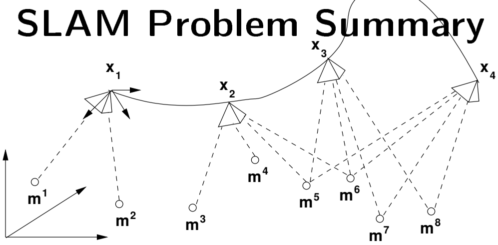
Sensor Fusion

Fredrik Gustafsson
fredrik.gustafsson@liu.se

Gustaf Hendeby
gustaf.hendeby@liu.se

Linköping University

SLAM Problem Summary



- *Simultaneous localization and mapping* (SLAM) is the problem of finding ones position, x_k, in a map, $\mathbf{m}$, while the map is built. Both parts must be considered simultaneous.
- Model:

$$z_k = \begin{pmatrix} x_{k+1} \\ \mathbf{m}_{k+1} \end{pmatrix} = \begin{pmatrix} f(x_k, v_k) \\ \mathbf{m}_k \end{pmatrix}, \qquad \mathrm{Cov}(v_k) = Q$$

$$y_k^i = h(x_k, \mathbf{m}_k^{c_k^i}) + e_k^i, \qquad \mathrm{Cov}(e_k^i) = R, \quad i = 1, \dots, I_k.$$

- Solve using essentially a marginalized particle filter yields the FastSLAM 1.0/FastSLAM 2.0 algorithm.

Idea: factorize the posterior as in the MPF

Basic factorization idea:

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

- 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.

More General Measurement Model

Assume observation model linear(-ized) in landmark position

$$0 = h^0(y_k^i, x_k) + h^1(y_k^i, x_k)\mathbf{m}_k^{c_k^i} + e_k^i, \qquad \mathrm{Cov}(e_k^i) = R_k^i.$$

The special case $y_k^i = h(x_k, \mathbf{m}_k^{c_k^i}) + e_k^i$ yields

$$h^0(y_k^i, x_k) = h(x_k, \hat{\mathbf{m}}_k^{c_k^i}) - h'_{\mathbf{m}}(x_k, \hat{\mathbf{m}}_k^{c_k^i})\hat{\mathbf{m}}_k^{c_k^i} - y_k^i$$
$$h^1(y_k^i, x_k) = h'_{\mathbf{m}}(x_k, \hat{\mathbf{m}}_k^{c_k^i}).$$

This formulation covers:

- First order Taylor expansions.
- Bearing and range measurements, where $h^i(y_k^n, x_k)$ has two rows per landmark in 2D SLAM.
- Bearing-only measurements coming from a camera detection.

Estimating the Mapping: WLS

Linear estimation theory applies. The WLS estimate:

$$\hat{\mathbf{m}}^j = \Bigg(\underbrace{\sum_{k=1}^{N} \big(h^1(y_k^{\bar{c}_k^j}, x_k)\big)^T (R_k^{\bar{c}_k^j})^{-1} h^1(y_k^{\bar{c}_k^j}, x_k)}_{\mathcal{I}_N^j}\Bigg)^{-1}$$

$$\underbrace{\sum_{k=1}^{N} -\big(h^1(y_k^{\bar{c}_k^j}, x_k)\big)^T (R_k^{\bar{c}_k^j})^{-1} h^0(y_k^{\bar{c}_k^j}, x_k)}_{\imath_N^j} = (\mathcal{I}_N^j)^{-1} \imath_N^j,$$

where $i = \bar{c}_k^j$ is the inverse mapping from landmark j to measurement i. In this sum, the terms where the map landmark j does not get an associated observation landmark at time k are dropped.

Under a Gaussian noise assumption, the posterior distribution is Gaussian

$$(\mathbf{m}^j | y_{1:N}, x_{1:N}) \sim \mathcal{N}\big((\mathcal{I}_N^j)^{-1} \imath_N^j, (\mathcal{I}_N^j)^{-1}\big).$$

Mapping Solution: information filter

Recursive estimation of the map using information filter form

$$\imath_k^j = \imath_{k-1}^j + \left(h^1(y_k^{\bar{c}_k^j}, x_k)\right)^T R_k^{-1} h^0(y_k^{\bar{c}_k^j}, x_k),$$
$$\mathcal{I}_k^j = \mathcal{I}_{k-1}^j + \left(h^1(y_k^{\bar{c}_k^j}, x_k)\right)^T R_k^{-1} h^1(y_k^{\bar{c}_k^j}, x_k),$$
$$\hat{\mathbf{m}}^j = (\mathcal{I}_k^j)^{-1} \imath_k^j.$$

Pose Solution: particle filter

Given $x_{1:k}$ and $y_{1:k}$, $\mathbf{m}$ is obtainable with WLS, then likelihood in the Gaussian case becomes:

$$p(y_k^{\bar{c}_k^j} | y_{1:k-1}, x_{1:k})$$
$$= \mathcal{N}\Big(h^0(y_k^{\bar{c}_k^j}, x_k) + h^1(y_k^{\bar{c}_k^j}, x_k)\hat{\mathbf{m}}_{k-1}^j, R_k^{\bar{c}_k^j} + h^1(y_k^{\bar{c}_k^j}, x_k)(\mathcal{I}_k^j)^{-1}\big(h^1(y_k^{\bar{c}_k^j}, x_k)\big)^T\Big).$$

This can be used as measurement equation in the **measurement update** in a particle filter.

The proposal distribution in the **time update** can be the SIR or the optimal:

FastSLAM 1.0: $x_{k+1}^{(i)} \sim p(x_{k+1} | x_k^{(j)})$

FastSLAM 2.0: $x_{k+1}^{(i)} \sim p(x_{k+1} | x_{1:k}^{(i)}, y_{1:k+1}) \propto p(x_{k+1} | x_k^{(j)}) p(y_{k+1} | x_{k+1})$

FastSLAM Algorithm (1/2)

1. **Initialize** the particles
$$x_1^{(n)} \sim p_0(x),$$
where N denotes the number of particles.
2. **Data association** that assigns a map landmark c_k^i to each observed landmark i. Initialize new map landmarks if necessary.
3. **Pose measurement update**
$$\omega_k^{(n)} \propto \prod_i \mathcal{N}\Big(h^0(y_k^i, x_k) + h^1(y_k^i, x_k)\hat{m}_{k-1}^{c_k^i}, R_k^i + h^1(y_k^i, x_k)(\mathcal{I}_k^{c_k^i})^{-1}\big(h^1(y_k^i, x_k)\big)^T\Big).$$
where the product is taken over all observed landmarks i, and normalize such that $\sum_n \omega_k^{(n)} = 1$.
4. **Resampe** Draw a new set of particles with replacement based on the particle weights.

FastSLAM Algorithm (2/2)

5. **Map measurement update:**

$$p(\mathbf{m}^{(n)}|x_{1:k}^{(n)}, y_{1:k}) = \mathcal{N}\left((\mathcal{I}_k^{(n)})^{-1}\imath_k^{(n)}, (\mathcal{I}_k^{(n)})^{-1}\right),$$

$$\imath_k^j = \imath_{k-1}^j + \left(h^1(y_k^{\bar{c}_k^j}, x_k^{(n)})\right)^T R_k^{-1} h^0(y_k^{c_k^j}, x_k^{(n)}),$$

$$\mathcal{I}_k^j = \mathcal{I}_{k-1}^j + \left(h^1(y_k^{\bar{c}_k^j}, x_k^{(n)})\right)^T R_k^{-1} h^1(y_k^{\bar{c}_k^j}, x_k^{(n)}).$$

6. **Pose time update:**

FastSLAM 1.0 (SIR PF)

$$x_{k+1}^{(n)} \sim p(x_{k+1}|x_{1:k}^{(n)}).$$

FastSLAM 2.0 (SIS PF with optimal proposal)

$$\begin{aligned} x_{k+1}^{(n)} &\sim p(x_{k+1}|x_{1:k}^{(n)}, y_{1:k+1}) \\ &\propto p(x_{k+1}|x_{1:k}^{(n)})p(y_{k+1}|x_{k+1}, x_{1:k}^{(n)}, y_{1:k}). \end{aligned}$$

Properties

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.
- Loop closure can be problematic due to particle depletion.

FastSLAM Illustration

- Airborne simultanous localization and mapping (SLAM) using a UAV with camera producing image features.
- Research collaboration with IDA.

http://youtu.be/hA_NZeuoy9Y

FastSLAM Illustration

- Airborne simultanous localization and mapping (SLAM) using a UAV with camera producing image features.
- Research collaboration with IDA.

http://youtu.be/hA_NZeuoy9Y

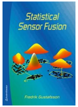
Summary

The simultaneous localization and mapping (SLAM) problem has been solved using a marginalized particle filter:

- FastSLAM 1.0.
- FastSLAM 2.0.

Properties:

- Scales well with the number of landmarks, but poorly with state dimension.
- Landmark not extremely associations critical.
- Loop closure is nontrivial.



Section 11.3