Skoltech
Skolkovo Institute of Science and Technology

CAIT
Mobile Robotics Lab
Perception in Robotics Course

Lecture 10. GaphSLAM

Gonzalo Ferrer

20 February 2023

1 GraphSLAM as a full SLAM problem (almost)

GraphSLAM or SAM– Smoothing and Mapping;
Smoothing – the robot trajectory $x_{1:t}$;
Mapping – set of landmarks or any other representation.
- Alternative to filter-based (Lecture 8) SLAM;
- Keeping the full trajectory is beneficial (for non-linear systems)

2 GraphSLAM as a Bayes Network

Example of Bayes Network is presented in the Fig. 1

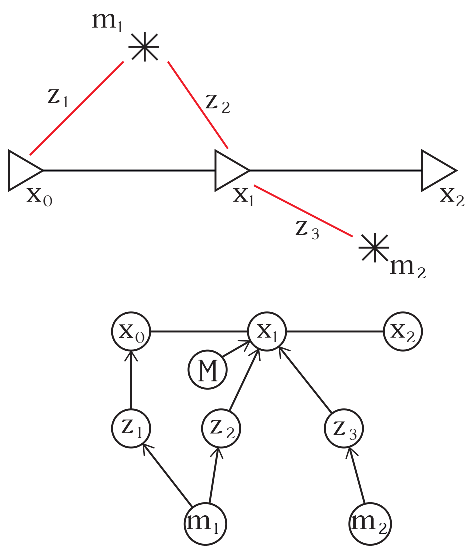

Figure 1: Example of Bayes Network. Graphical models are a powerful tool to describe SLAM. However, state variables and observations are all considered nodes.

$$\mathcal{P}(\mathcal{X},\mathcal{M},\mathcal{Z},\mathcal{U}) = p(x_0)\prod_{i=1}^{M} p(x_i|x_{i-1},\mu_i)\cdot\prod^{k} p(z_k|x_{i_k},m_{j_k}),$$

$p(x_0)$ – prior
$\prod_{i=1}^{\mathcal{M}} p(x_i|x_{i-1},\mu_i)$ – odometry
$\prod^{k} p(z_k|x_{i_k},m_{j_k})$ – observation

Objective: maximize the joint probability.

- Graph theory
- Linear Algebra

3 GraphSLAM as a Factor Graph

Factor Graph is a Bipartite Graph. Example of this graph is presented in the Fig. 2.

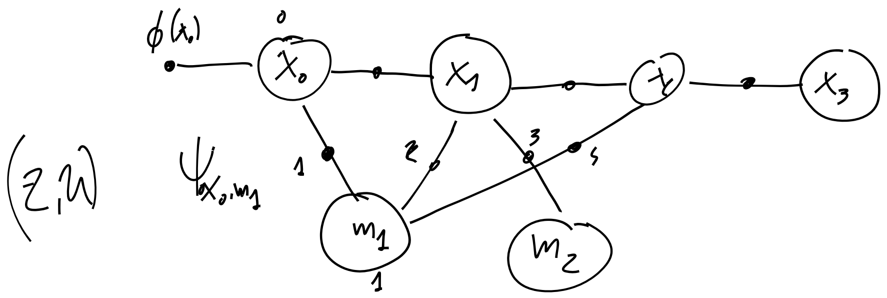

Figure 2: Example of Factor Graph. Factor Graphs are bipartite graphs, meaning that we express the relations from a set of vertices "nodes" which include our state variables through a set of vertices "factors", capturing the inherent distribution of the nodes variables due to observations. Bipartite is in the sense that edges of the graph are always from nodes to factors or vice versa.

- Bayesian networks are a natural way to express relations
- Factor Graphs have a tighter connection to optimization.

$\mathcal{Z}$,$\mathcal{U}$ – factors. Modulate distributions between variables $\mathcal{X},\mathcal{M}$.

$$\theta = \{\mathcal{X},\mathcal{M}\}, \qquad p(\theta) = \prod \phi_i(\theta_i) \cdot \prod \psi_{ij}(\theta_i\theta_j)]$$

$$\phi_0(x_0) \propto p(x_0)$$

$$\psi_{(i-1)i}(x_{i-1},x_i) \propto p(x_i|x_{i-1},u_i)$$

$$\psi_{i_k,j_k}(x_{i_k},m_{j_k}) \propto p(z_k|x_{i_{k1}},m_{j_k})$$

Same identical representation as Bayes Network if factors are defined this way.

3.1 Transition model

$$p(x_i|x_{i-1},u_i) = \mathcal{N}\big(x_i; g_i(x_{i-1},u_i),\Sigma_{u_i}\big) = \eta \cdot \exp\left\{-\frac{1}{2}\|g_i(x_{i-1},u_i) - x_i\|^2_{\Sigma_i}\right\},$$

where the covariance matrix Σ_{u_i}, for simplicity Σ_i has been previously defined in class as R_i.

Example:

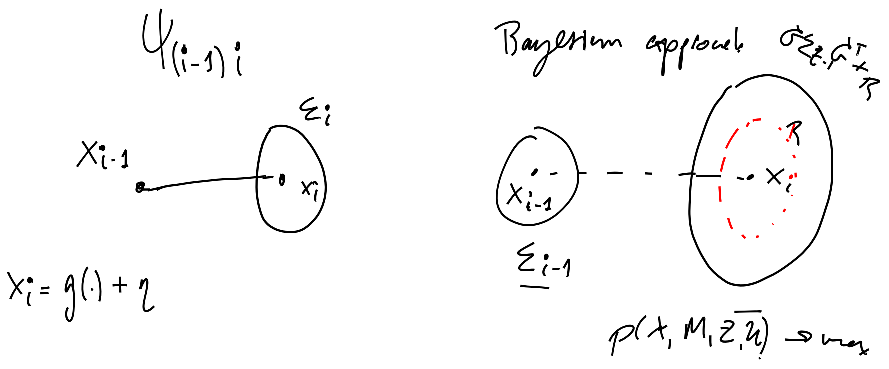

Figure 3: Example of transition model. On the left, factor considering only covariance of action Σ_i and deterministic prior state x_{i-1}. On the right, the KF derivation, where the joint distribution of state and action is calculated. We will see later the advantages of the FG approach.

The function $g()$ could be any of the propagation models described in L05: odometry, unicycle, etc.

3.2 Smoothing: optimization of the chain trajectory

The *smoothing* approach, using FGs, calculates the relative relations between $i-1$ and i. As a result, the trajectory resembles a chain and covariances do not get propagated over the state variables (see fig. 4).

The main difference is that full SLAM estimates x_i joint distributions of all considered random variables.

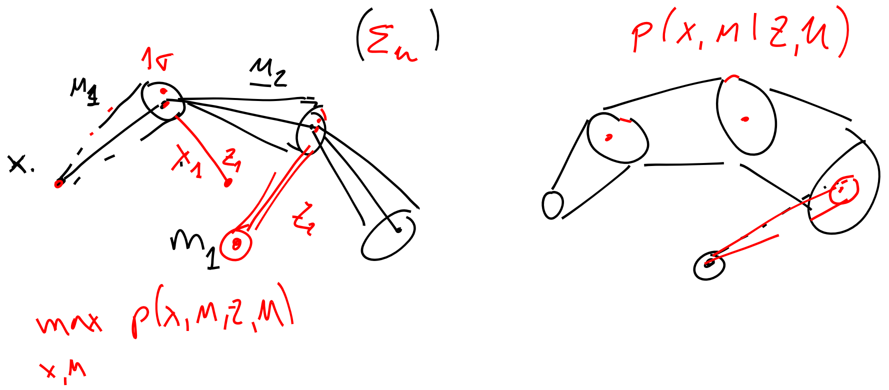

Figure 4: Chain Trajectory. On the left, smoothing of the trajectory using FGs. On the right, the augmented KF approach, showing the joint distribution for state and action, accumulating over time.

3.3 Observation model

$$p(z_k|x_{i_k}, m_{j_k}) = \mathcal{N}\big(z_k|h_k(x_{i_k}, m_{j_k}), \Sigma_k\big) = \eta \exp\left\{-\frac{1}{2}\|h_k(x_{i_k}, m_{j_k}) - z_k\|^2_{\Sigma_k}\right\}$$

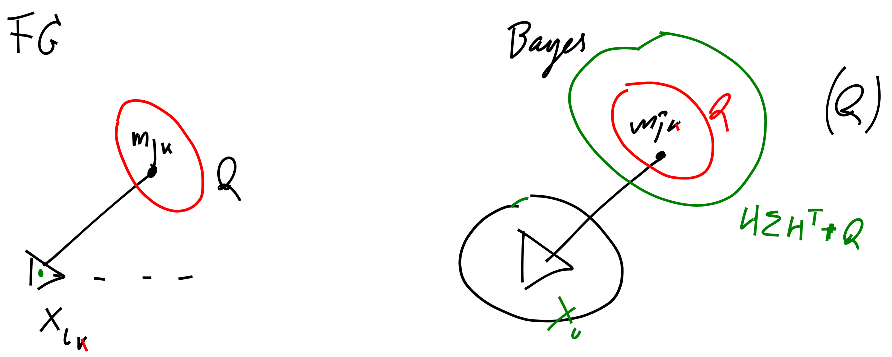

Figure 5: Observation model. On the left, FG consider the state as "fixed" to calculate the observation model. On the right, the joint approach considering directly all sources of uncertainty.

4 Solving GraphSLAM

The solution is the result of maximizing the joint probability:

$$\begin{aligned}
\theta^* &= \arg\max_\theta \mathcal{P}(\mathcal{X},\mathcal{M}|\mathcal{Z},\mathcal{U}) = \arg\max_\theta \mathcal{P}(\mathcal{X},\mathcal{M},\mathcal{Z},\mathcal{U}) \\
&= \arg\min_\theta \{-\log \mathcal{P}(\mathcal{X},\mathcal{M},\mathcal{Z},\mathcal{U})\} = \\
&= \arg\min_\theta \left\{ \sum_i^M \|g_i(x_{i-1},u_i) - x_i\|^2_{\Sigma_i} + \sum_{k=1}^K \|h_k(x_{i_k},m_{j_k}) - z_k\|^2_{\Sigma_k} \right\}. \qquad (1)
\end{aligned}$$

The problem has become a Non-linear least squares problem (NLLSQ).
This solution is exactly equivalent to *EKF* of the augmented state $x_{0:1}$ for linear systems. Recall that we obtained a solution that was based on the Bayes filter after marginalizing and conditioning a joint Gaussian (L04). The reason why the probabilistic approach provided a solution is because the maximum of this probability, for Gaussian distributions, corresponds to the mean (also mode and median). Therefore, it was irrelevant if our solution was obtained by optimization or by expanding the Bayes filter, the solution was the same.

In the non-linear case, the mode (maximum) and the mean in general do not coincide. For that reason the maximization proposed here is an alternative to simply linearization and Bayes filtering.

4.1 Linearize the NLLSQ

$$g_i(x_{i-1},u_i) - x_i \simeq \left[g_i(x^0_{i-1},u_i) + G_i^{i-1}\delta x_{i-1}\right] - \left[x_i^0 + \delta x_i\right] = (G_i^{i-1}\delta x_{i-1} - \delta x_i) - a_i, \qquad (2)$$

where a_i is the residual, the state variable $x_i = x_i^0 + \delta x_i$ is composed of the current estimate x_i^0 and an increment value δx_i.

Jacobian $G_i^{i-1} = \left.\dfrac{\partial g_i(x_i,u_i)}{\partial x_{i-1}}\right|_{x^0_{i-1}}$.

Note that the variables to optimize are x_i, but it will more convenient to update values of the perturbed values δx_i when calculating the solution of the NNLSQ.

The observation function:

$$\begin{aligned}
h_k(x_{i_k},m_{j_k}) - z_k &\simeq h_k(x^0_{i_k},m^0_{j_k}) + H_k^{i_k}\delta x_{i_k} + J_k^{i_k}\delta m_{j_k} - z_k = \\
&= (H_k^{i_k}\delta x_{i_k} + J_k^{i_k}\delta m_{j_k}) - c_k, \qquad (3)
\end{aligned}$$

where $c_k = h_k(x^0_{i_k} - z_k)$.

Jacobians:

$$H_k^{i_k} = \left.\frac{\partial h}{\partial x_{i_k}}\right|_{(x^0_{i_k},m^0_{j_k})}$$

$$J_k^{i_k} = \left.\frac{\partial h}{\partial m_{j_k}}\right|_{(x^0_{i_k}, m^0_{j_k})}$$

4.2 Linearized LSQ

$$\delta^* = \arg\min_{\delta} \left\{ \sum_{i=1}^{M} \|G_i^{i-1}\delta x_{i-1} - I\delta x_i - a_i\|^2_{\Sigma_i} + \sum_{k=1}^{K} \|H_k^{i_k}\delta x_{i_k} + J_k^{i_k}\delta m_{j_k} - c_k\|^2_{\Sigma_k} \right\}$$

How to simplify this expression?

$$\Sigma^{-1} = \Lambda = LL^T = \sqrt{\Sigma^{-1}}\sqrt{\Sigma^{-1}}^T = \Sigma^{-\frac{1}{2}} \cdot \Sigma^{-\frac{T}{2}}$$

$$\|e\|^2_\Sigma = e^T\Sigma^{-1}e = (\Sigma^{-\frac{T}{2}}e)^T(\Sigma^{-\frac{T}{2}}e) = \|\Sigma^{-\frac{T}{2}}e\|^2_2$$

The process followed is 1) Invert 2) Cholesky factorization 3) Group $\rightarrow$ transposed squared root inverse. Mahalanobis distance become Euclidean distance by pre-multiplying.

$$\delta^* = \arg\min_{\delta} \left\{ \sum_{i=1}^{M} \|\Sigma_i^{-\frac{T}{2}}(G_i^{i-1}\delta x_{i-1} - I\delta x_i) - \Sigma_i^{-\frac{T}{2}}a_i\|^2_2 + \sum_{k=1}^{K} \|\Sigma_k^{-\frac{T}{2}}(H_k^{i_k}\delta x_{i_k} + J_k^{i_k}\delta m_{j_k}) - \Sigma_k^{-\frac{T}{2}}c_k\|^2_2 \right\}$$

We have obtained a LLSQ which is compactly written as: $\delta^* = \arg\min_{\delta} \|A\delta - b\|^2_2$.

Example:

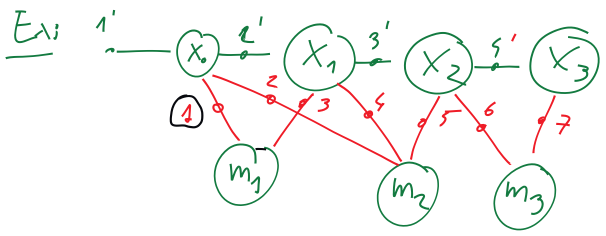

Figure 6: Factor graph of a given SLAM problem.

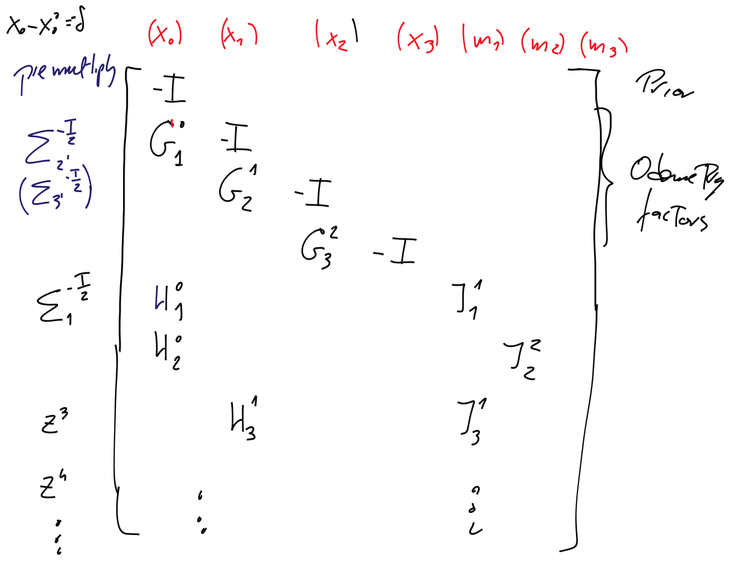

Figure 7: Adjacency matrix A expanded by block elements corresponding to the Jacobians. This is a sparse matrix so efficient Linear Algebra algorithms could be applied.

4.3 Solve the LLSQ

$$b = \begin{bmatrix} \Sigma_1^{-\frac{T}{2}} a_i \\ \vdots \\ \Sigma_k^{-\frac{T}{2}} c_k \end{bmatrix}$$

$$\min_\delta \|A\delta - b\|_2^2 \xrightarrow{\frac{\partial}{\partial\delta}} \frac{1}{2}(A\delta - b) \cdot A = 0 \Leftrightarrow A\delta = b$$

A is not square. It is an over-constrained problem

$$\begin{aligned} A\delta &= b \\ A^T A\delta &= A^T b \\ \delta &= (A^T A)^{-1} A^T b \end{aligned}$$

4.4 Algorithm raw GrapSLAM

```
1: while True do
2:     Calculate $A$, $b$ around $\{\mathcal{X}^0, \mathcal{M}^0\} = \Theta^0$;
3:     $\delta^* := \arg\min_\delta \|A\delta - b\|_2^2$;
4:     Update $\mathcal{X}^0, \mathcal{M}^0$,    by $\Theta^0 := \Theta^0 + \delta^0$
5:     if convergence then
6:         return $\Theta$
```

5 Summary

- SLAM as a factor graph (previously we used Bayes network)
- $\theta^* = \arg\max_\theta \mathcal{P}(\mathcal{X}, \mathcal{M} | \mathcal{Z}, \mathcal{U}) = \arg\max_\theta \{\sum || \cdot ||_{\Sigma_i}^2\}$
- After linearizing and other simplifications, $\delta^* = \arg\min_\delta \|A\delta - b\|_2^2$
    1) Taylor 1st order
    2) $\|e\|_\Sigma^2 = \|\Sigma^{-\frac{T}{2}} e\|_2^2$.
- Solve the LSQ $A\delta = b$ until convergence.