EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2021**

Lecture 27: SLAM 2

Scribes: Michael Wu

27.1 SLAM as an optimation problem

Recall that we can pose SLAM problem as the following minimization problem

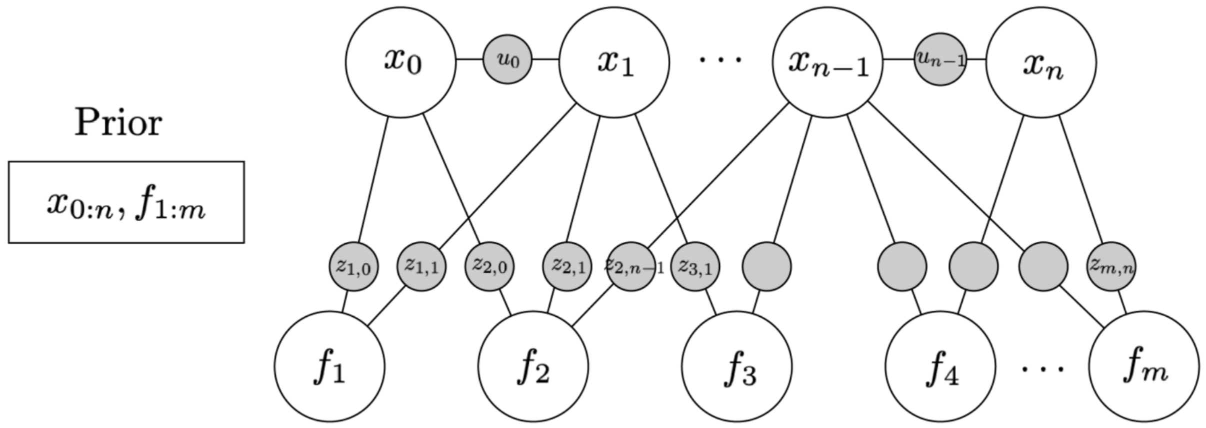
$$c(x) = \|x - \mu_n\|^2_{\Sigma_n^{-1}} + \sum_{t=1}^{n} \|x_t - g(x_{t-1}, u_{t-1})\|^2_{\Sigma_v^{-1}} + \sum_{j=1}^{n} \sum_{i \in S(x_j)} \|z_{ij} - h(f_i, x_j)\|^2_{\Sigma_w^{-1}} \tag{27.1}$$

where $S(x_j)$ is the set of indices i s.t. the ith feature is visible from pose x_j

We have two methods to solve this:

- Gauss-Newton Descent: A gradient based method to improve upon an initial estimate; converges to local minima.
- Marginalization: Used to remove variables from the cost function. Updates the prior mean and covariance (μ_n, $\sum_n$) to account for information loss.

27.2 Visualize SLAM as a graph problem

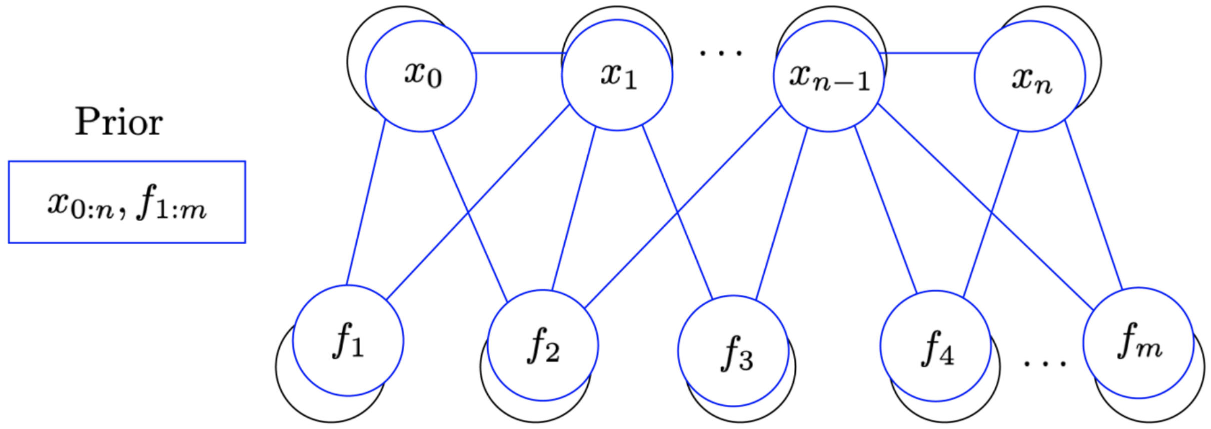


- Each unknown variable is a vertex
- Each edge represents a constraint (or cost) between two variable induced by a measurement.
- Each edge is like a "spring" tugging on the variables being connected, and the cost associated is the stored "energy"

- The objective is to find an assignment for the vertices that minimizes the total energy stored in the edges.

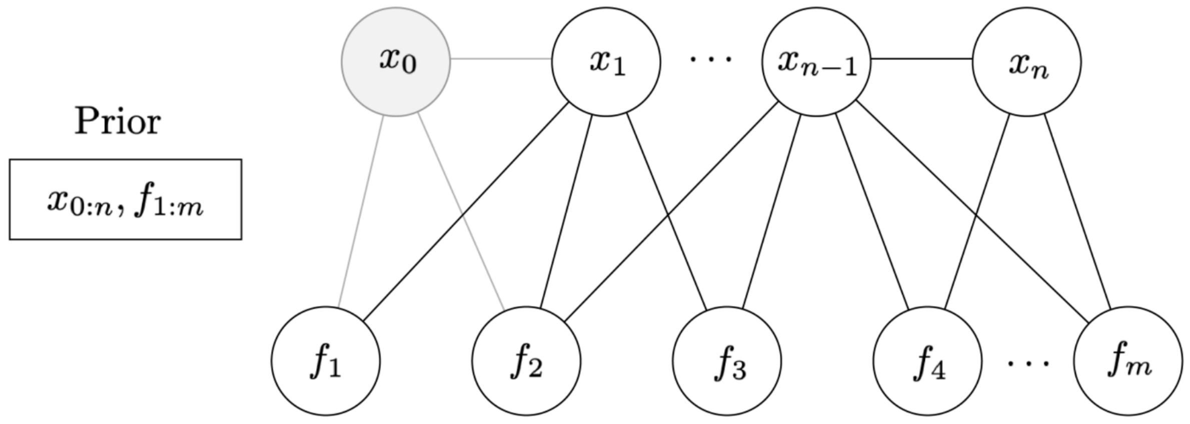
27.2.1 Gauss-Newton

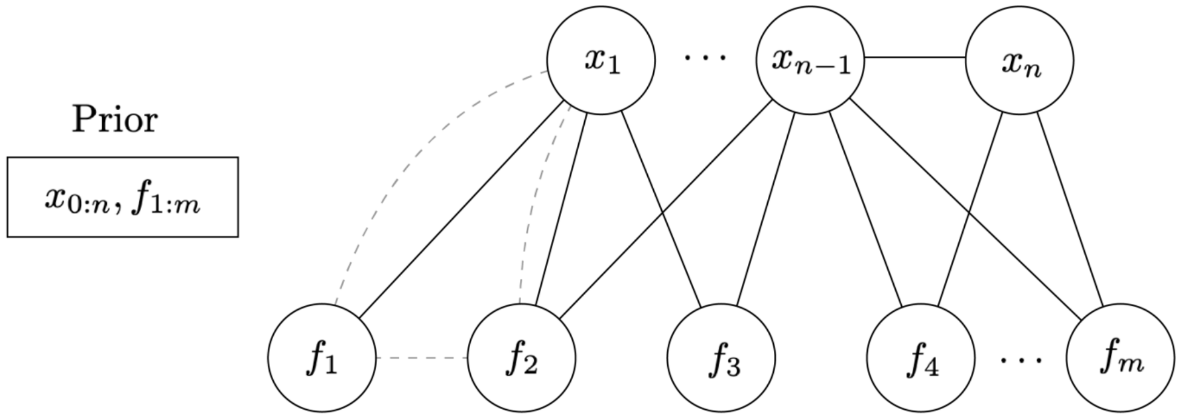
Re-position the vertices to reduce the total energy stored in the edges.

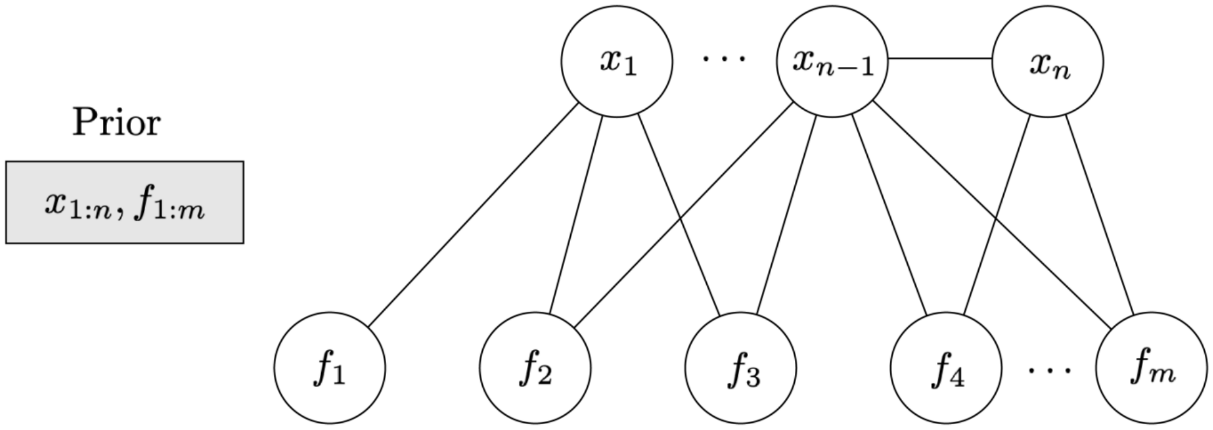


27.2.2 Marginalization

Delete a given set of variables from the graph.







27.3 Algorithm Template

Start off with an initial guess μ_0 and a prior covariance $\sum_0$ over the initial pose x_0 Set the cost function to $c(x) = \|x - \mu_0\|^2_{\Sigma_0^{-1}}$. Set t = 0

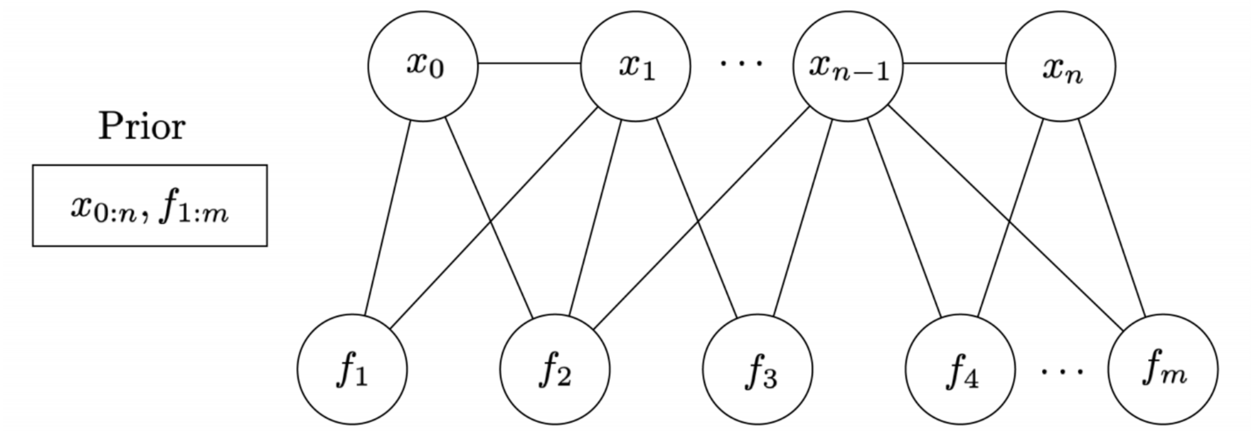
- Receive a new image measurement for time t + 1.
- Receive a new odometry measurement u_t . Add a term $| \; x_{t+1} - g\left(x_t, u_t\right) \|^2_{\Sigma_w^{-1}}$ to c. add initial guess $g(x_t, u_t)$ for x_{t+1} to estimate μ_t
- If needed, perform 1 or more Gauss-Newton steps to update μ_t or linearly approximate the cost function.
- Recieve landmark measurements. Add terms of the form $\|z - h(f, x)\|^2_{\Sigma_v^{-1}}$ to c. Add initial guesses for newly detected landmarks to μ_t supplied by the front-end.
- If needed, perform 1 or more Gauss-Newton steps to update μ_t or linearly approximate the cost function.

- If needed, marginalize away some subset of variables. This will update both μ_t and $\sum_t$ (the current prior term in the cost function). Drop some other subset of variables.
- Set $\mu_{t+1} \leftarrow \mu_t, \Sigma_{t+1} \leftarrow \Sigma_t, t \leftarrow t+1,$ and repeat.

27.4 An overview of SLAM algorithms

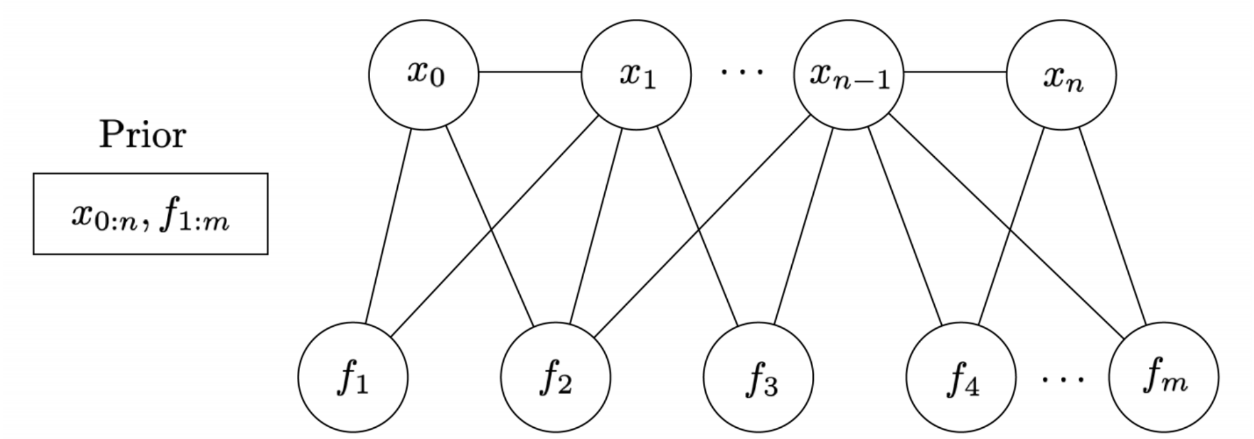
- Graph SLAM
- Bundle adjustment
- Pose Graph SLAM
- Fixed-lag smoother
- Extended Kalman Filter (EKF)
- Iterated Extended Kalman Filter (iEKF)

27.4.1 Graph SLAM



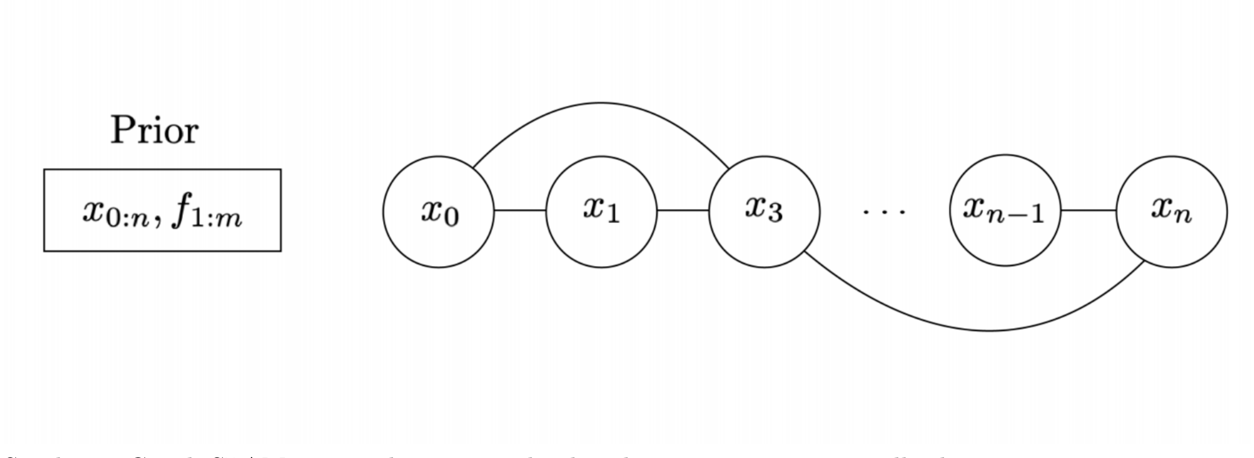
- Construct the whole graph of the problem when all data is available.
- Then optimize the whole problem at once.
- Solves the full SLAM problem.

27.4.2 Bundle Adjustment



- Same as Graph SLAM, except there are no motion constraints available.
- This is the standard approach to structure-from-motion, where a scene needs to be reconstructed given only a set of image observations.
- May introduce priors over poses and some features to deal with scale and global pose ambiguities.

27.4.3 Pose-graph SLAM



Similar to Graph SLAM except there are no landmarks; measurements are all relative pose measurements.

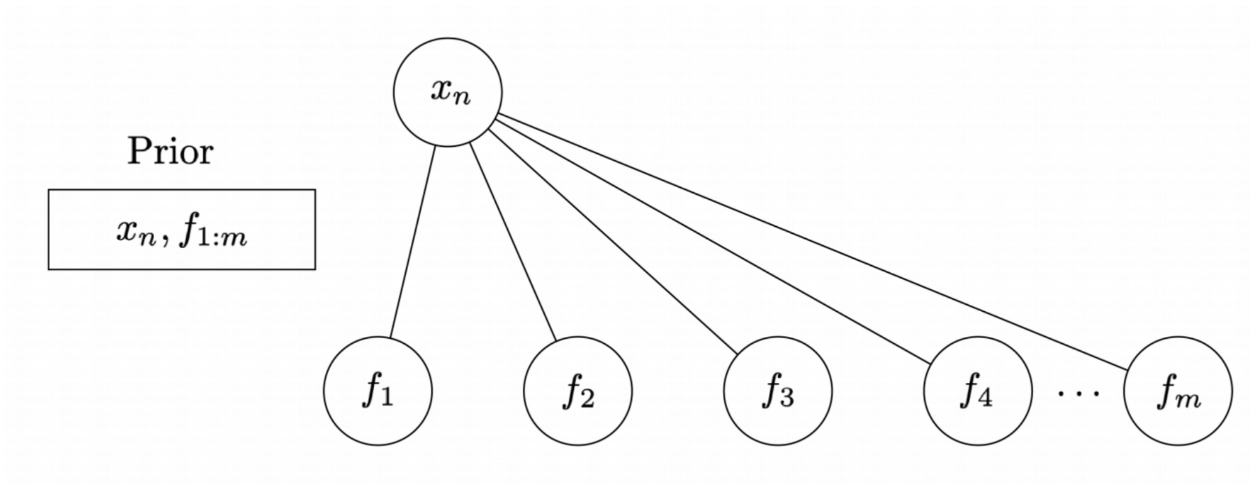
$$\|z_{ij} \Theta h\left(x_i, x_j\right)\|^2_{\Sigma_v^{-1}} \tag{27.2}$$

For example, $z_{ij} \in SE(3)$ and $h\left(x_i, x_j\right) = x_i^{-1} x_j$

Maintain a "sliding window" of poses; only keep the last k poses. Marginalize oldest pose.

Optionally, marginalize features as soon as they are no longer visible from any "active" pose.

27.4.4 Extended Kalman Filter



- The popular Extended Kalman Filter (EKF) is just a sliding window feature with window size k=1, where we only take 1 Gauss-Newton step at each iteration rather than repeating till convergence.

27.5 Global consistency

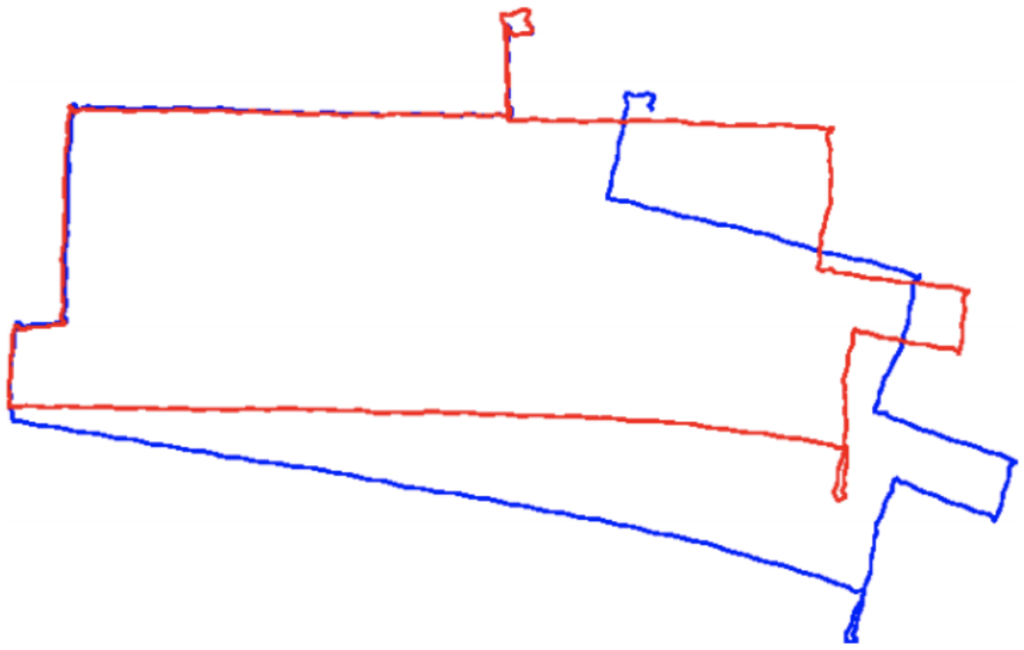
- So far, we have discussed how to pose the SLAM problem in the form. of a real-time optimization.

- Only the most recent poses/landmarks are maintained in the optimization problem. It is local.

- This prevents us from revisiting estimates for old poses even when new information is available that may let us improve those estimates.

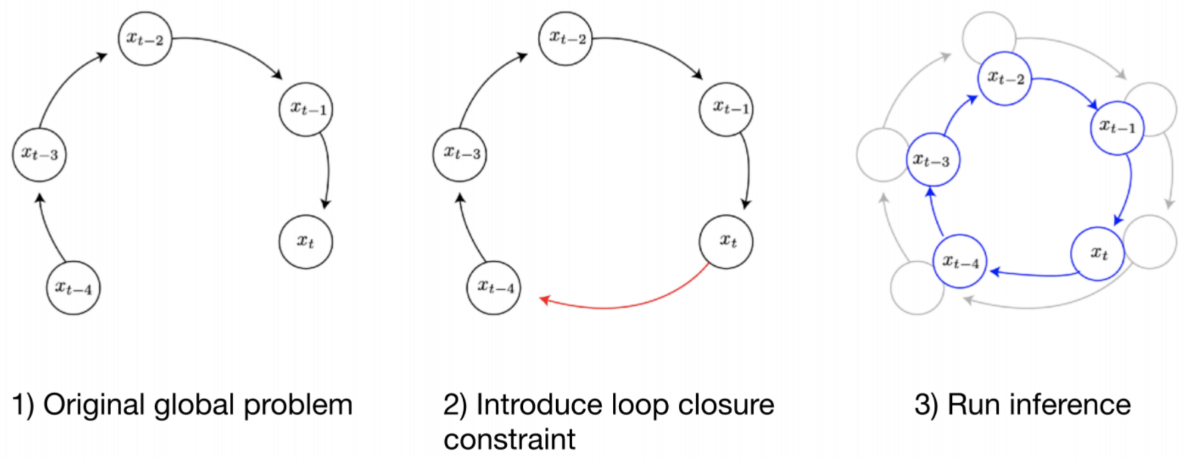
27.5.1 Loop Closure

When we revisit a map we have seen before, we can register new measurements of the same features, thus allowing us to correct for any drift that has occurred in the meantime. This is called loop closure.



In addition to incremental pose constraints, we also introduce "loop closure" constraints which are activated when the robot re-visits a part of the map it has seen before.

Inference (optimization) is done each time a new loop closure is registered.


x_{t-2}
x_{t-1}
x_{t-3}
x_t
x_{t-4}
1) Original global problem
2) Introduce loop closure constraint
3) Run inference

27.5.2 Detecting loop closures

- Bag-of-words approach: assigns a binary descriptor to each image encoding the presence or absence of certain features.
-