Joint Estimation and Control using Graphical Models with Constraints (#16)

Carnegie Mellon University

Jerry Hsiung, Paloma Sodhi

Motivation

Simultaneous localization and mapping (SLAM) problems are increasingly formulated as probabilistic inference in graphical models. A commonly employed class of graphical models is a factor graph that is capable of representing factorization of probability distribution functions.

SLAM problems using factor graphs are, however, traditionally formulated as unconstrained optimizations. In this project, we would like to extend the **factor graph** formulation to solve **a SLAM optimization problem with nonlinear robot dynamics constraints**. Our results demonstrate the capabilities of factor graphs for general optimization problems.

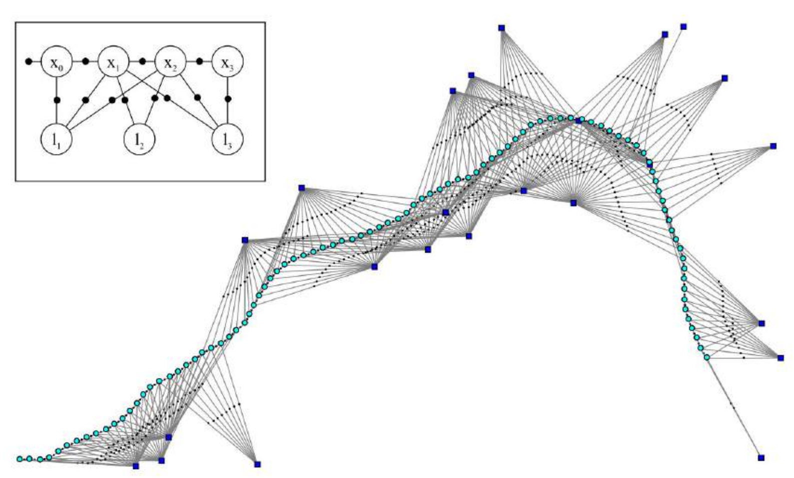

Figure 1. Large-scale SLAM problem and its factor graph representation [1]

Problem Formulation

Below we formulate the robot state estimation and control problem as that of a constrained optimization using factor graphs,

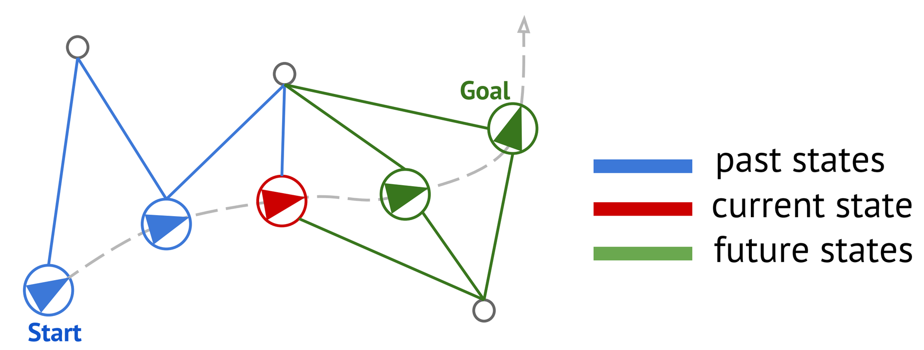

Figure 2. A sample control and estimation problem of robot moving from a start position to the goal position

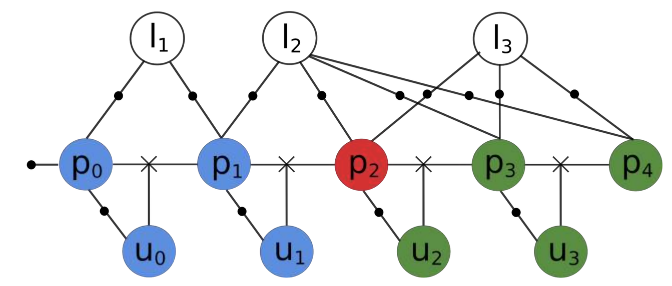

Figure 3. The factor graph representing control and estimation problem

Complete process is mathematically expressed as,

$$\dot{p}(t) = f(p(t), u(t), t) \qquad \text{(dynamics constraints)}$$

$$z_{ij} = h_{ij}(p(t_i), l_j) + v_{ij} \qquad \text{(measurements)}$$

$$u(t) \in \mathcal{U},\ p(t) \in \mathcal{G} \qquad \text{(domain constraints)}$$

Cost functions for estimation and control objectives are,

$$J_{est} = \quad ||p_o - \hat{p}_o||^2_{\Sigma_{p_o}} + \sum_{ij} ||h(p_i, l_j) - z_{ij}||^2_{\Sigma_{v_{ij}}}$$

$$J_{mpc} = \quad \phi(p_N) + \sum_{i=c}^{N-1} L_i(p_i, u_i, l_{1:K})$$

Approach

We solve the problem using a factor graph version of a Sequential Quadratic Programming (SQP) objective with X_i, $X_j \subset X$ as sets of primal variables and λ_j the dual variables, that is,

$$\min_{\Delta X_i} \sum_i \left(\frac{1}{2} \Delta X_i^T \nabla^2_{X_i X_i} L_i^{(k)} \Delta X_i + \Delta X_i^T \nabla_{X_i} L_i^{(k)} \right) \quad + \sum_j \tfrac{1}{2} \lambda_j^{(k)} \Delta X_j^T \nabla^2_{X_j X_j} F_j^{(k)} \Delta X_j$$

$$\text{such that,} \quad \nabla_{X_j} F_j^{(k)T} \Delta X_j + F_j^{(k)} = 0 \qquad \forall \quad j = 1 \dots n_c$$

where, $\sum_i L(X_i) = J_{est}(X) + J_{mpc}(X)$ is the factorized objective cost function and $F = p(t^+) - p(t) \exp \frac{h}{2} \left(F(\widehat{p(t), u}(t)) + F(\widehat{p(t^+), u}(t)) \right)$ is the dynamics equality constraint derived on Lie-Manifold.

At k-th iteration, the Newton update on the KKT condition is:

$$\begin{bmatrix} \sum_i \nabla^2_{X_i X_i} L_i^{(k)} + \sum_j \lambda_j \nabla^2_{X_j X_j} F_j^{(k)} & \nabla_{X_j} F_j^{(k)} \\ \nabla_{X_j} F_j^{(k)T} & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \sum_i \nabla_{X_i} L_i^{(k)} + \sum_j \lambda_j \nabla_{X_j} F_j^{(k)} \\ F_j^{(k)} \end{bmatrix}$$

We solve the above update in two stages, by representing the problem as a *primal* and a *dual factor graph*. The solutions of each graph are used to update the primal and dual variables respectively.

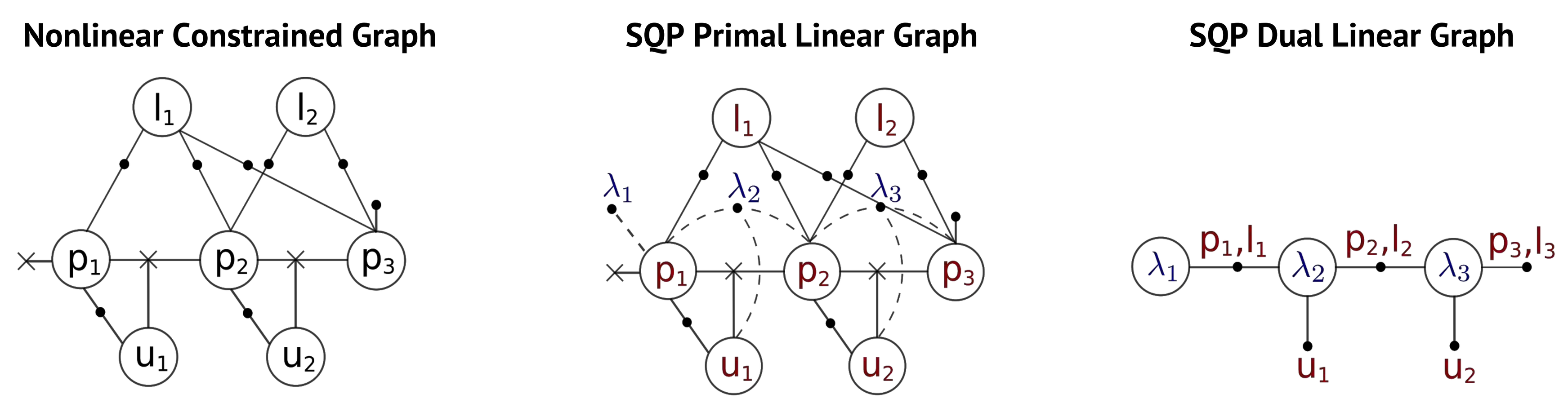

Figure 4. Original nonlinear constrained factor graph (left) is solved by decomposing it as SQP primal linear graph (middle) and dual linear graph (right)

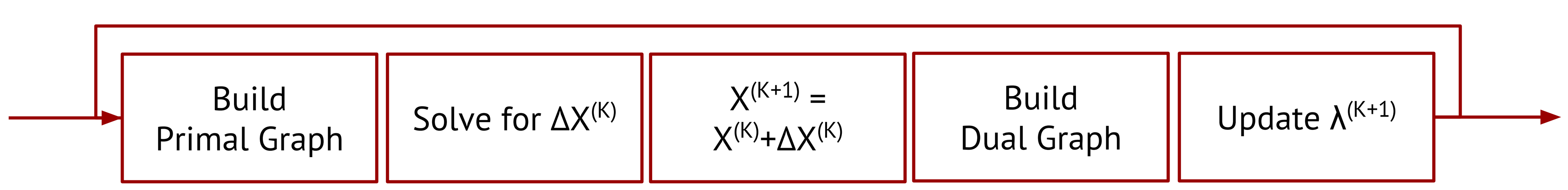

Figure 5. Implementation pipeline for solving an SQP iteration update

Experimental Setup

Our experimental setup constitutes a simulated 2D world with a forward moving non-holonomic robot system whose states are defined as,

$$p_i = \begin{bmatrix} x_i & y_i & \theta_i & v_{x_i} & v_{y_i} \end{bmatrix}^T \in \mathbb{R}^5 \quad i = 1 \dots n$$

$$u_i = \begin{bmatrix} T_i & \omega_i \end{bmatrix}^T \in \mathbb{R}^2 \quad i = 1 \dots n$$

$$l_j = \begin{bmatrix} x_j & y_j \end{bmatrix}^T \in \mathbb{R}^2 \quad j = 1 \dots m$$

$$X = \begin{bmatrix} p_1 & \dots & p_n & u_1 & \dots & u_n & l_1 & \dots & l_m \end{bmatrix}^T$$

and dynamics F(x,u) are defined as,

$$\dot{x} = v_x \ , \ \dot{y} = v_y$$

$$\dot{R} = R(\omega^b)^\vee$$

$$\dot{v} = \tfrac{T}{m} R \begin{bmatrix} 1 & 0 \end{bmatrix}^T$$

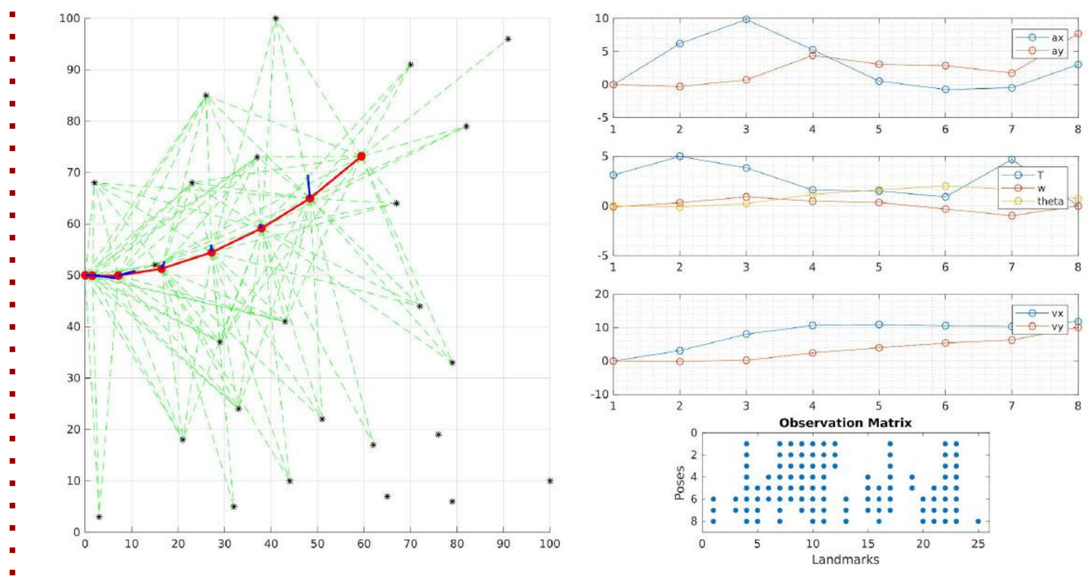

Figure 6. Simulated 2D world with robot states and landmark locations visualized

Results

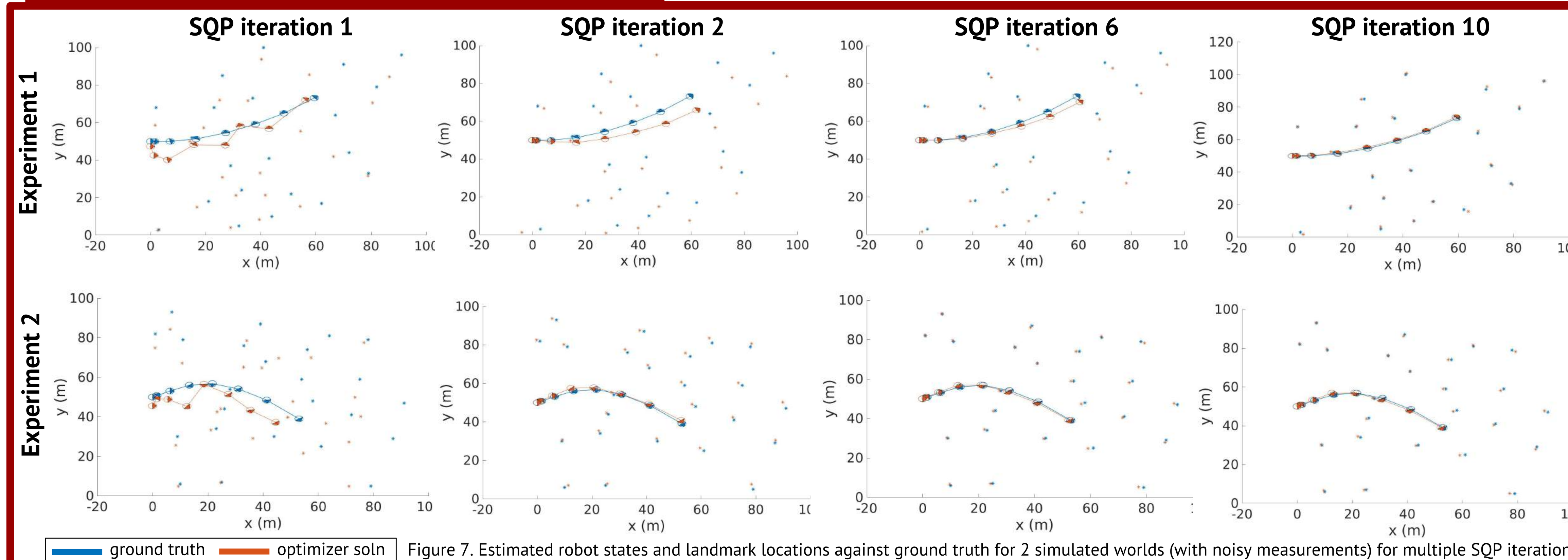

Figure 7. Estimated robot states and landmark locations against ground truth for 2 simulated worlds (with noisy measurements) for multiple SQP iterations

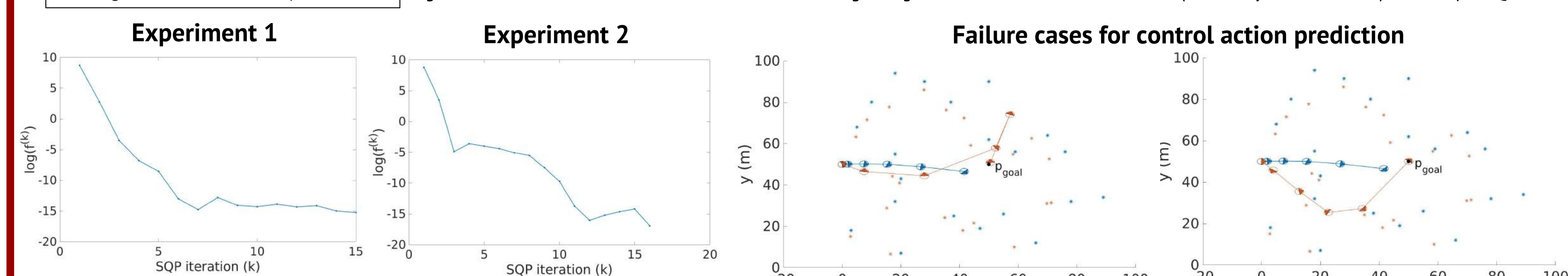

Figure 8. Variation of Primal Objective Function with SQP Iterations

Discussions

In our experiments, we initialize states not very far from the groundtruth. This assumption is valid because this is intended to be an online estimation method. Therefore, at the start of each iteration, the previously estimated solution should be close to the minimum. Currently, our results do not include the model predictive control estimation. Our preliminary testing shows that with prediction states added, our simulation cannot initialize the controls close enough to the groundtruth. The results tend to diverge as the function becomes highly non-convex as shown in above failure cases. .

Future directions include adding appropriate obstacle avoidance costs and adding constraints to the controls so the trajectory is feasible.

References

[1] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. Foundations and Trends in Robotics, 6(1-2):1–139, 2017.
[2] Mustafa Mukadam, Jing Dong, Frank Dellaert, and Byron Boots. Simultaneous trajectory estimation and planning via probabilistic inference. In Robotics Science and Systems, 2017.
[3] Duy-Nguyen Ta, Marin Kobilarov, and Frank Dellaert. A factor graph approach to estimation and model predictive control on unmanned aerial vehicles In Unmanned Aircraft Systems (ICUAS), 2014 International Conference on, pages 181–188. IEEE, 2014.
[4] Jorge Nocedal and Stephen J Wright. Sequential quadratic programming. Springer, 2006.
[5] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Georgia Institute of Technology, 2012