2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

ICS: Incremental Constrained Smoothing for State Estimation

Paloma Sodhi[1], Sanjiban Choudhury[2], Joshua G. Mangelson[1], and Michael Kaess[1]

***Abstract*— A robot operating in the world constantly receives information about its environment in the form of new measurements at every time step. Smoothing-based estimation methods seek to optimize for the most likely robot state estimate using all measurements up till the current time step. Existing methods solve for this smoothing objective efficiently by framing the problem as that of incremental unconstrained optimization. However, in many cases observed measurements and knowledge of the environment is better modeled as hard constraints derived from real-world physics or dynamics. A key challenge is that the new optimality conditions introduced by the hard constraints break the matrix structure needed for incremental factorization in these incremental optimization methods.**

Our key insight is that if we leverage primal-dual methods, we can recover a matrix structure amenable to incremental factorization. We propose a framework ICS that combines a primal-dual method like the Augmented Lagrangian with an incremental Gauss Newton approach that reuses previously computed matrix factorizations. We evaluate ICS on a set of simulated and real-world problems involving equality constraints like object contact and inequality constraints like collision avoidance.

I. Introduction

Localization and state estimation are increasingly formulated as smoothing problems since the use of such methods tends to lead to increased efficiency and accuracy [1]. Typically, the smoothing objective is formulated as a MAP inference problem over a graph whose nodes are the unknown state variables and edges encode sensor measurement information. For Gaussian models, MAP inference results in a sparse, unconstrained nonlinear least squares optimization [2]. Such problems can be solved *incrementally* [3, 4], i.e. as new measurements arrive at each time step, without having to re-solve from scratch. However, in many cases observed measurements and knowledge about the environment would be better modeled as *hard constraints* in the optimization. These can either be constraints imposed by physics, such as robots must not violate dynamics or teleport through objects, or these can be constraints imposed by certain sensors, such as contact measurements. We address the problem of satisfying such constraints within the framework of incremental optimization to allow for efficient and precise inference from measurements arriving at each time step.

[1]P. Sodhi, J. Mangelson and M. Kaess are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. {psodhi, jmangels, kaess}@cs.cmu.edu
[2] S. Choudhury is with School of Computer Science and Engineering, University of Washington, Seattle, WA, USA. sanjibac@cs.uw.edu
This work was partially supported by the Office of Naval Research under award N00014-16-1-2103.

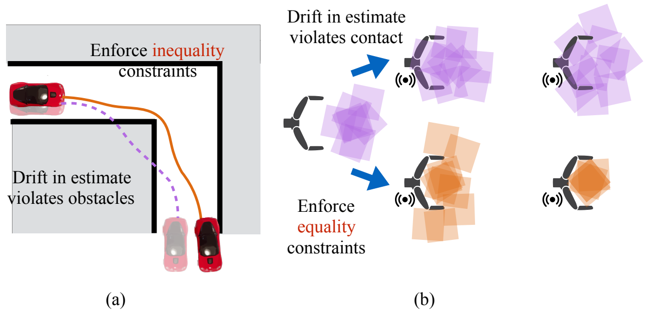

Enforce inequality constraints
Drift in estimate violates obstacles
Drift in estimate violates contact
Enforce equality constraints
(a)
(b)

Fig. 1. **(a)** Mobile robot navigating down a corridor: Enforcing inequality constraints would restrict the space of possible robot trajectory estimates to those not in collision. **(b)** Robot arm manipulating an object with uncertain pose: Enforcing equality constraints would restrict the space of possible poses to those in contact with robot gripper.

Consider the problem of a robot manipulating an object whose pose is uncertain (Fig. I(b)). There are two primary ways to sense object pose: one using a noisy and often occluded vision system, and another using a precise contact sensor placed on the arm. Under a typical unconstrained optimization framework, contact sensor measurements would be incorporated as high confidence cost priors, or an equivalent of penalty terms in the optimization [5]. However, this need not guarantee satisfaction of these terms, which depends on ratio of their weights relative to other cost terms in the objective. Too disproportionate a ratio can lead to ill-conditioning, causing numerical issues. Moreover, it is not straightforward to incorporate *inequality constraints* in the unconstrained framework—e.g. a robot moving down a corridor must lie within corridor boundaries even in presence of a drifting state estimate indicating otherwise (Fig. I(a)).

The unconstrained incremental optimization framework is an incremental Gauss-Newton method that relies on updating sparse matrix factorizations of the Jacobians efficiently, without needing to re-factorize at each time step [3]. These operations are no longer possible when constraints, i.e., Karush-Kuhn-Tucker (KKT) conditions [6] are introduced. Our key idea is to leverage primal-dual methods and work around this limitation by *splitting the optimization* in two separate steps. The primal step retains the structure of unconstrained nonlinear least squares, amenable to incremental Gauss Newton methods. The dual step is simply a gradient ascent step which is incremental in nature. As a result, constraints are precisely satisfied without having to recompute solutions from scratch every time step.

We propose a framework ICS for doing incremental constrained smoothing. ICS combines a primal-dual method like the Augmented Lagrangian [7] with an incremental Gauss Newton approach that reuses previously computed matrix factorizations [3]. ICS works as follows: at each time step t, the robot receives new measurements and constraints, it then incrementally updates state estimates for times $1:t$ reusing matrix factorizations from previous step and performing alternating primal-dual updates to convergence at current step. Hence, at any given time, it maintains a solution consistent with all measurements received so far without having to re-solve for all constraints. Our main contributions are:

1) Introduce incremental constrained smoothing problem.
2) Propose a primal dual framework with incremental Gauss Newton steps for solving the problem.
3) Provide empirical evaluation against baselines for state estimation applications.

II. Related Work

Simultaneous Localization and Mapping (SLAM) problem has been well studied for several decades [1]. Early methods were based on the well-known extended Kalman Filter [8, 9]. However, such filtering-based methods had difficulty scaling due to the curse of dimensionality [10, 11]. Eventually it was recognized that by leveraging inherent sparsity, it was possible to solve SLAM as a smoothing problem tractably as a large but sparse nonlinear optimization over the entire robot trajectory [12, 13]. While this enables handling a large numbers of poses, solving it as a batch optimization requires factorizing a large Jacobian matrix composed of all previous measurements every new time step and is dependent on having a good linearization point. To address this, Kaess et al. [3] proposed incremental smoothing and mapping (iSAM) that solved this problem incrementally by updating the existing factorization with new measurements as opposed to factorizing from scratch, enabling real-time optimization.

Above methods formulate the problem as a MAP inference over a factor graph resulting in an unconstrained nonlinear optimization problem. Within this framework, measurements are incorporated as weighted terms in overall cost that penalizes deviation of predicted measurements from observed values [2]. In many cases, however, observed measurements or knowledge about the environment can be better modeled as hard constraints in the optimization problem. Some recent methods [14–17] have been proposed towards incorporating hard constraints into the factor graph framework. Cunningham et al. [14] perform hybrid elimination to solve a system of equations with mixed unconstrained and equality constraint factors. Ta et al. [15] too solve for mixed unconstrained and equality constraint factors by utilizing a special QR factorization within a SQP framework. Jimenez et al. [17] extend to inequality constraints by treating active inequalities as equality constraints in an active set. Choudhary et al. [16] solve for a distributed constrained objective using ADMM within a factor graph framework.

All these methods [14–17], however, are formulated for solving a batch optimization objective. It isn't clear how these can extend to incremental solutions wherein matrix factorizations from previous time steps are to be reused to avoid re-solving from scratch. Another set of related batch optimization techniques utilize Lagrangian duality for certifying optimality in SLAM solutions [18, 19]. A related work by Bai et al. [20] formulates incremental SLAM as a cycle/constraint selection problem finding an optimal set of consistent transformations. This differs from our use of incremental denoting reuse of matrix factorizations from previous time steps.

III. Problem Formulation

Maximum a posteriori (MAP) inference for localization (or SLAM) problems with Gaussian noise models is equivalent to solving a nonlinear least-squares problem [2]. For a factor graph, MAP inference involves maximizing product of all factor graph potentials, that is,

$$\hat{x} = \underset{x}{\operatorname{argmax}} \prod_{i=1}^{m} \phi_i(x) \tag{1}$$

Assuming that $\phi_i(x)$ are all Gaussian factors corrupted by zero-mean, normally distributed noise,

$$\begin{aligned} \phi_i(x) &\propto \exp\left\{-\frac{1}{2}||f_i(x) - z_i||^2_{\Sigma_i}\right\} \\ \Rightarrow \hat{x} &= \underset{x}{\operatorname{argmin}}\ \frac{1}{2}\sum_{i=1}^{m} ||f_i(x) - z_i||^2_{\Sigma_i} \end{aligned} \tag{2}$$

where, $f_i(x)$ is a likelihood function predicting the expected measurement given the state, z_i is the actual measurement, and $||\cdot||_{\Sigma_i}$ is the Mahalanobis distance with measurement covariance Σ_i.

A. Unconstrained nonlinear least-squares

The optimization objective in Eq. 2 is an unconstrained nonlinear least-squares minimization. For general nonlinear functions $f_i(x)$, this objective is non-convex. However, if we have a reasonable initial guess available, we can use nonlinear optimization methods like Gauss-Newton (GN) to converge to a global minimum where gradient of the optimization objective equals 0. GN proceeds by linearizing measurement functions $f_i(\cdot)$ at each iteration using a first-order Taylor expansion as,

$$f_i(x) = f_i(x^0 + \delta x) \approx f_i(x^0) + F_i \delta x \tag{3}$$

where, x^0 is the linearization point, $F_i = \left.\frac{\partial f_i(x)}{\partial x}\right|_{x^0}$ is the measurement Jacobian, and $\delta x = x - x^0$ the state update vector. Substituting the Taylor expansion back into Eq. 2,

$$\begin{aligned} \delta x^* &= \underset{\delta x}{\operatorname{argmin}} \sum_{i=1}^{m} ||F_i \delta x - \left(z_i - f_i(x^0)\right)||^2_{\Sigma_i} \\ &= \underset{\delta x}{\operatorname{argmin}}\ \frac{1}{2}\sum_{i=1}^{m} ||A_i \delta x - b_i||^2_2 \\ &= \underset{\delta x}{\operatorname{argmin}}\ \frac{1}{2}||A\delta x - b||^2_2 \end{aligned} \tag{4}$$

where,

$$A_i = \Sigma_i^{-1/2} F_i \ \ , \ b_i = \Sigma_i^{-1/2}\left(z_i - f_i(x_i^0)\right) \tag{5}$$

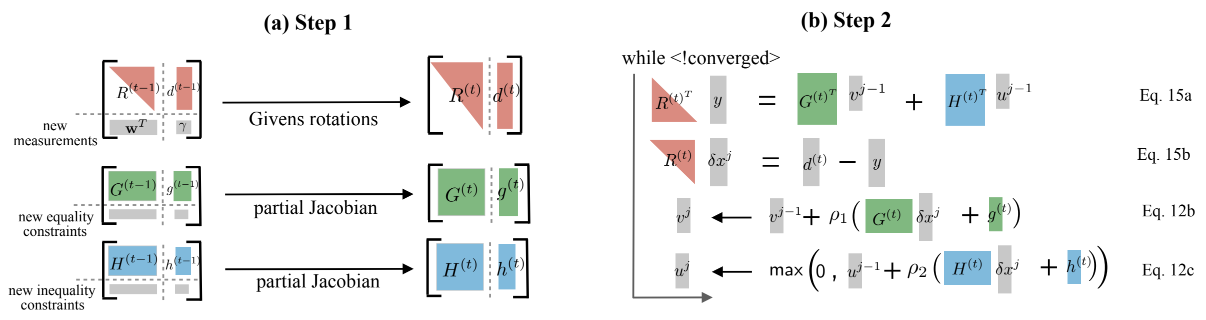


Fig. 2. Overview of set of operations involved at every new time step t of our ICS framework. **(a)** Step 1 updates to the most recent matrix factorization $R^{(t)}$ and Jacobians $G^{(t)}, H^{(t)}$ based on new measurements and new equality, inequality constraints arriving at time step t. **(b)** Step 2 performs the primal-dual updates iteratively until convergence to ensure that δx^* satisfies the equality, inequality constraints within a threshold.

and, A, b are constructed by collecting all A_i, b_i together into one giant matrix, vector respectively. Once the linear system is solved, x^0 is updated according to $x^0 \leftarrow x^0 + \delta x^*$ and we iterate until convergence criteria are met.

B. *Constrained nonlinear least-squares*

Consider now the same least-squares objective as in Eq. 2 but now subject to a set of nonlinear equality and inequality constraints,

$$\begin{aligned} \hat{x} =& \underset{x}{\text{argmin}}\ \frac{1}{2}\sum_{i=1}^{m} ||f_i(x) - z_i||^2_{\Sigma_i} \\ \text{s.t.,} \quad & g_j(x) = 0 \quad , \quad j = 1, \ \dots \ p \\ & h_k(x) \le 0 \quad , \quad k = 1, \ \dots \ q \end{aligned} \tag{6}$$

Linearizing measurement functions $f_i(\cdot)$, $g_j(\cdot)$, $h_k(\cdot)$, about a linearization point x^0 similar to Eq. 3 would result in a subproblem of form,

$$\begin{aligned} \delta x^* =& \underset{\delta x}{\text{argmin}}\ \frac{1}{2}\sum_{i=1}^{m} ||A\delta x - b||^2_2 \\ \text{s.t.,} \quad & G\delta x + g(x^0) = 0 \\ & H\delta x + h(x^0) \le 0 \end{aligned} \tag{7}$$

where, A, b are constructed by collecting all A_i, b_i together similar to the unconstrained case in Eqs. 4, 5. $G \in \mathbb{R}^{p\times n}$, $H \in \mathbb{R}^{q\times n}$ here are jacobians of equality, inequality constraints respectively, and $g(\cdot) \in \mathbb{R}^p$, $h(\cdot) \in \mathbb{R}^q$ are equality, inequality constraint evaluations at linearization point x^0.

IV. Approach

We want to solve these optimization problems *incrementally*, i.e. as measurements and constraints arrive. Section IV-A illustrates how to solve the unconstrained subproblem in Eq. 4 incrementally based on *Givens rotations* [3]. However, the constrained subproblem in Eq. 7 presents an additional challenge due to additional linearized equality/inequality constraints. It no longer suffices to set the gradient to 0, instead, the Karush–Kuhn–Tucker (KKT) conditions [6] need to be satisfied. In Section IV-B, we show how to do this in a manner that allows us to use Givens rotation to facilitate incremental updates.

A. *Incremental unconstrained optimization*

The unconstrained objective in Eq. 4 is solved as,

$$\begin{aligned} & \nabla_{\delta x}\ \frac{1}{2}||A\delta x - b||^2 = 0 \\ \Rightarrow\ & A^T A\delta x = A^T b \end{aligned} \tag{8}$$

Eq. 8 is referred to as the *normal equations*. Since A is typically sparse for localization (or SLAM) problems, we can solve these normal equations efficiently using sparse matrix factorization like QR decomposition [2, 3]. That is, factoring $m\times n$ matrix A with $m \ge n$ as $Q\begin{bmatrix}R & 0\end{bmatrix}^T$, and substituting this factorization into the normal equations in Eq. 8,

$$\begin{aligned} & \begin{bmatrix}R^T & 0\end{bmatrix} Q^T Q \begin{bmatrix}R \\ 0\end{bmatrix} \delta x = \begin{bmatrix}R^T & 0\end{bmatrix} Q^T b \\ \Rightarrow \quad & R^T R\delta x = \begin{bmatrix}R^T & 0\end{bmatrix} \begin{bmatrix}d \\ e\end{bmatrix} \\ \Rightarrow \quad & R\delta x = d \end{aligned} \tag{9}$$

where Q is a $m\times m$ orthogonal matrix with $Q^TQ = I$, R is a $n \times n$ upper triangular matrix, and $d \in \mathbb{R}^n$, $e \in \mathbb{R}^{(m-n)}$.

To solve this incrementally, when a new measurement at time step t arrives, it is more efficient to modify the previous time step factorization $R^{(t-1)}$ directly by *QR-updating* instead of updating and refactoring entire matrix A again [3]. Adding a new measurement row $\mathbf{w}^T$ and corresponding RHS γ at time t yields a new system that is not yet in the correct factorized form,

$$A^{(t)} \leftarrow \begin{bmatrix} R^{(t-1)} \\ \mathbf{w}^T \end{bmatrix}, \quad b^{(t)} \leftarrow \begin{bmatrix} d^{(t-1)} \\ \gamma \end{bmatrix} \tag{10}$$

To convert $A^{(t)}$ into correct factorized form $R^{(t)}$, Givens rotations [3] are applied to zero out the new rows $\mathbf{w}^T$ corresponding to new measurements at time step t. RHS of Eq. 10 is simultaneously updated with the same rotations to obtain $d^{(t)}$. This Givens rotations based updating of the previous time step factorization $R^{(t-1)}$ provides the basis of efficient incremental solutions for the unconstrained problem [3]. In general, the maximum number of Givens rotations needed for adding a new measurement row is n. However, since both $R^{(t-1)}$, $\mathbf{w}^T$ are sparse, only a constant number of Givens rotations are needed.

B. Incremental constrained optimization

For solving the constrained nonlinear least-squares objective in Eq. 7 it is no longer sufficient to only satisfy gradient of the quadratic objective becoming zero (Eq. 8). Instead we must construct the Lagrangian and ensure the KKT conditions are satisfied at optimality. However, directly solving KKT equations does not retain the form in Eq. 9 amenable to incremental factorization. Instead, we rely on primal-dual methods that essentially split the optimization in two: a primal step and a dual step. Specifically, we use the augmented Lagrangian/method of multipliers [7]. We start by writing out the Lagrangian for Eq. 7,

$$\begin{aligned}\mathcal{L}(\delta x, v, u) = \frac{1}{2}\ ||A\delta x - b||_2^2\ + v^T\left(G\delta x + g(x^0)\right) + \\ u^T\left(H\delta x + h(x^0)\right)\end{aligned} \tag{11}$$

where, $v \in \mathbb{R}^p$, $u \in \mathbb{R}^q$ are the dual variables associated with equality, inequality constraints respectively.

The augmented Lagrangian also includes additional *augmentation* terms (for improved robustness) in the Lagrangian of the form $\frac{\rho_1}{2}||G\delta x + g(x^0)||_2^2$ and $\frac{\rho_2}{2}||H\delta x + h(x^0)||_2^2$ corresponding to equality and inequality constraints respectively. For simplicity of notations, we will absorb these additional quadratic augmentation terms within $||A\delta x - b||_2^2$ itself. That is, similar to the derivation in Eq. 5, the overall A, b matrix, vector will now additionally include rows A_i, b_i with $A_i = \sqrt{\frac{\rho_1}{2}}G$, $b_i = -\sqrt{\frac{\rho_1}{2}}g(x^0)$ and $A_i = \sqrt{\frac{\rho_2}{2}}H$, $b_i = -\sqrt{\frac{\rho_2}{2}}h(x^0)$.

To satisfy KKT conditions at optimality, augmented Lagrangian solves a *maximin* dual problem of the form $\max_{u\geq 0, v}\min_{\delta x}\mathcal{L}(\delta x, v, u)$. It solves this problem iteratively [7]: first, solving for δx by minimizing the Lagrangian treating v, u as constants, followed by maximizing resultant dual function treating δx as constant. This iterative update process at each iteration j can be expressed as,

$$\delta x^j = \text{argmin}_{\delta x}\ \mathcal{L}(\delta x, v^{j-1}, u^{j-1}) \tag{12a}$$

$$v^j = v^{j-1} + \rho_1(G\delta x^j + g(x^0)) \tag{12b}$$

$$u^j = \max\left(0, u^{j-1} + \rho_2(H\delta x^j + h(x^0))\right) \tag{12c}$$

where, Eq. 12a is the primal update and Eqs. 12b, 12c are the dual updates. The dual updates are simply projected gradient ascent steps for maximizing the dual function $\max_{u\geq 0, v} g(v, u) := \max_{u\geq 0, v}\mathcal{L}(\delta x^j, v, u)$. This step is incremental in nature. ρ_1, ρ_2 are the ascent step sizes that appeared earlier as coefficients of the augmentation terms. Note that the dual function $g(v, u)$ requires inequality duals to be $u \geq 0$. This ensures that $g(v, u)$ always lower bounds the primal objective, and that maximizing $g(v, u)$ approaches the optimal primal objective [6].

Having split the optimization in Eq. 12, the primal update can now be computed using Givens rotations similar to Section IV-A. Expanding Eq. 12a we have

$$\begin{aligned}&\nabla_{\delta x}\mathcal{L}(\delta x, v^{j-1}, u^{j-1}) = 0\\ \Rightarrow\ &A^TA\delta x^j - A^Tb + G^Tv^{j-1} + H^Tu^{j-1} = 0\\ \Rightarrow\ &A^TA\delta x^j = A^Tb - G^Tv^{j-1} - H^Tu^{j-1}\end{aligned} \tag{13}$$

Algorithm 1 ICS: Incremental Constrained Smoothing

```
for $t$ = 1 to $T$ do
  $\mathcal{Z}_{1:t} \leftarrow \mathcal{Z}_{1:t-1} \cup \mathcal{Z}_t$
  if !(batch) then
    $[R^{(t)}, d^{(t)}] = \text{givensUpdate}(x^{(t-1)}, \mathcal{Z}_t, R^{(t-1)}, d^{(t-1)})$
    $[G^{(t)}, g^{(t)}] = \text{partialJacobian}(x^{(t-1)}, \mathcal{Z}_t, G^{(t-1)}, g^{(t-1)})$
    $[H^{(t)}, h^{(t)}] = \text{partialJacobian}(x^{(t-1)}, \mathcal{Z}_t, H^{(t-1)}, h^{(t-1)})$
    $[\delta x^*, v^*, u^*] = \text{primalDual}(R^{(t)}, d^{(t)}, G^{(t)}, g^{(t)}, H^{(t)}, h^{(t)}, v^{(t\text{-}1)}, u^{(t\text{-}1)})$
    $x^{(t)} \leftarrow x^{(t-1)} + \delta x^*$
    $v^{(t)} \leftarrow v^*, u^{(t)} \leftarrow u^*$
  else
    while <!converged> do
      $[R^{(t)}, d^{(t)}] = \text{fullQR}(x^{(t-1)}, \mathcal{Z}_{1:t})$
      $[G^{(t)}, g^{(t)}] = \text{fullJacobian}(x^{(t-1)}, \mathcal{Z}_{1:t})$
      $[H^{(t)}, h^{(t)}] = \text{fullJacobian}(x^{(t-1)}, \mathcal{Z}_{1:t})$
      $[\delta x^*, v^*, u^*] = \text{primalDual}(R^{(t)}, d^{(t)}, G^{(t)}, g^{(t)}, H^{(t)}, h^{(t)}, v^{(t\text{-}1)}, u^{(t\text{-}1)})$
      $x^{(t)} \leftarrow x^{(t-1)} + \delta x^*$
      $v^{(t)} \leftarrow v^*, u^{(t)} \leftarrow u^*$
    end while
  end if
end for
```

Function: primalDual($R^{(t)}, d^{(t)}, G^{(t)}, g^{(t)}, H^{(t)}, h^{(t)}, v^{(t-1)}, u^{(t-1)}$)

```
Initialize: $j \leftarrow 1$, $v^0 \leftarrow v^{(t-1)}$, $u^0 \leftarrow u^{(t-1)}$
while <!converged> do
  ${R^{(t)}}^T y = {G^{(t)}}^T v^{j-1} + {H^{(t)}}^T u^{j-1}$
  $R^{(t)}\delta x^j = d^{(t)} - y$
  $v^j \leftarrow v^{j-1} + \rho_1(G^{(t)}\delta x^j + g^{(t)})$
  $u^j \leftarrow \max\left(0,\ u^{j-1} + \rho_2(H^{(t)}\delta x^j + h^{(t)})\right)$
end while
return $\delta x^j, v^j, u^j$
```

This form is similar to the *normal equations* in Eq. 8. Since A is sparse, we can apply QR factorization to solve Eq. 13. That is, factoring $m\times n$ matrix A with $m \geq n$ as $Q\begin{bmatrix}R & 0\end{bmatrix}^T$, and substituting this factorization in Eq. 13,

$$\begin{aligned}&[R^T\quad 0]Q^TQ\begin{bmatrix}R\\0\end{bmatrix}\delta x^j = [R^T\quad 0]\,Q^Tb - G^Tv^{j\text{-}1} - H^Tu^{j\text{-}1}\\ &\Rightarrow\ R^TR\delta x^j = \begin{bmatrix}R^T & 0\end{bmatrix}\begin{bmatrix}d\\e\end{bmatrix} - G^Tv^{j-1} - H^Tu^{j-1}\\ &\Rightarrow\ R\delta x^j = d - R^{-T}(G^Tv^{j-1} + H^Tu^{j-1})\end{aligned} \tag{14}$$

Eq. 14 can hence be solved as an iterative two-step process, a forward substitution followed by a back substitution, i.e.,

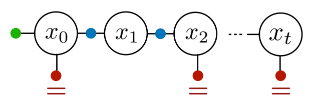
$$R^Ty = (G^Tv^{j-1} + H^Tu^{j-1}) \tag{15a}$$

$$R\delta x^j = d - y \tag{15b}$$

This differs from the original normal equations solution that involved only a single back substitution (Eq. 9).

Figure 2 provides an overview of the set of operations involved at every new time step t of ICS. We need to update the factorization $R^{(t)}$ and Jacobians $G^{(t)}, H^{(t)}$ once every time step (Step 1), followed by solving for the primal, dual updates iteratively until convergence (Step 2).

Algorithm 1 summarizes the overall incremental constrained smoothing method. It also shows periodic batch steps that computes matrix factorization from scratch. These steps may be triggered whenever we want to re-linearize past states or update variable ordering to retain sparsity in the factorization [3]. For simplicity, dual ascent step sizes ρ_1, ρ_2 are shown fixed in the algorithm. In practice, we use adaptive step sizes for improved convergence. We make use of simple



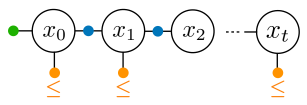


Fig. 4. Factor graphs for **(a)** Simulated Contact Dataset **(b)** Simulated/Real-world Navigation Dataset. Factors shown are prior (green), odometry (blue), equality constraints (red) and inequality constraints (orange).

adaptation scheme where: (a) $\rho^j = \rho^{j-1} \times g$ if constraint violation decreases by less than a minimum change factor, or (b) $\rho^j = \rho^{j-1} \times 1/r$ if constraint violation increases.

V. Results and Evaluation

We evaluate ICS qualitatively and quantitatively on metrics like constraint violations, estimation accuracies and runtime performance against an incremental unconstrained optimization approach like iSAM [3]. We implemented ICS within the iSAM C++ library. All evaluations are done on a laptop with an Intel Core i7-7820HQ 2.9GHz processor.

A. Datasets and Baselines

We evaluate our approach on the following three datasets: (1) Simulated Contact, (2) Simulated Navigation, and (3) Real-world Navigation. For the simulated contact dataset we enforce equality constraints like object contact, while for simulated and real-world navigation datasets we enforce inequality constraints like collision avoidance.

Fig. 4(a),(b) shows factor graphs for Simulated Contact and Simulated/Real-world Navigation datasets respectively. A factor graph is a bipartite graph with: variable nodes $x \in \mathcal{V}$ (empty circles) and factor nodes $\phi \in \mathcal{U}$ (filled circles) [13]. In Fig. 4(a) state variables $x_t = \begin{bmatrix} p_x^t & p_y^t & p_\theta^t \end{bmatrix}^T$ model the 2D pose of a circular object at time t being pushed by a circular gripper. There are two measurements: object odometry and contact sensor on the gripper. We model contact measurements as *equality constraint factors* that constrain the squared distance between object and gripper.

In Fig. 4(b) state variables $x_t = \begin{bmatrix} p_x^t & p_y^t & p_\theta^t \end{bmatrix}^T$ models the 2D pose a mobile robot navigating inside a building. There is one measurement: the robot odometry. We additionally have *inequality constraint factors* that are piece-wise linear constraints corresponding to boundaries of the corridor that the robot is in currently. For now, we extract these corridor constraints manually using the floor plan for both simulated and real-world navigation datasets, and assume that we have access to these constraints on the back-end.

We compare performance of ICS against an incremental unconstrained optimization framework like iSAM [3]. For comparison, equality/inequality constraint factors in ICS are incorporated as regular unconstrained cost factors in iSAM. We don't include loop closure factors in any of the datasets for simplicity of analysis. However, these can be incorporated in both frameworks as additional binary factors to the graph.

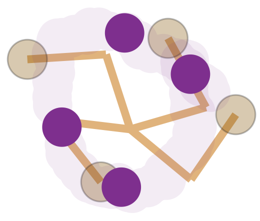
Gaussian noise with $\sigma_{odom} = 0.008, 0.294, 0.942$ are added to odometry measurements for datasets 1, 2, 3 respectively. The convergence criteria for alternating primal dual updates is set to $||G\delta x + g||_2 \leq \epsilon_1$ for equality constraints and $||H\delta x + h||_2 \leq \epsilon_2$ for inequality constraints that are not satisfied (i.e. > 0). ϵ_1 is set as 10^{-6} (dataset 1), ϵ_2 as 10^{-4} (datasets 2,3). Maximum allowed primal-dual iterations are set to 100. $\rho_1 = 500$ for dataset 1, and $\rho_2 = 20, 1$ for datasets 2, 3 respectively. Adaptive dual ascent step size parameters $g, 1/r$ are set to $5, 0.1$ respectively.

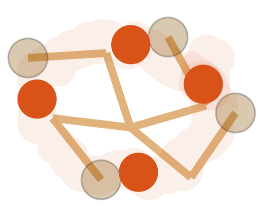
B. Performance accuracies

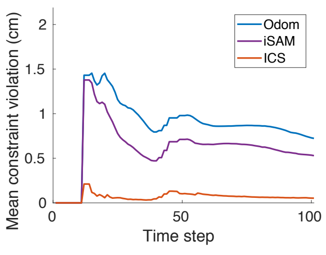
Fig. 3 shows performance accuracies on the Simulated Contact dataset. Fig. 3(a), (b) qualitatively show estimated object poses using ICS, iSAM respectively at four intermediate time steps where the gripper was in contact with object. Incorporating contact as an equality constraint in ICS is able to satisfy constraints more precisely (i.e. object in contact with gripper) than incorporating it as an unconstrained factor as in the case of iSAM. This can also be seen quantitatively where lower mean constraint violations in Fig. 3(c) leads to improved state estimates with lower RMSE absolute trajectory errors (ATE) in Fig. 3(d).

Fig. 5 shows performance accuracies on the Simulated Navigation dataset. Fig. 5(a) qualitatively show estimated robot trajectory using ICS, iSAM at the final time step. Incorporating corridor constraints as inequality constraints is able to satisfy the constraint more precisely (i.e. collision-free trajectories) than incorporating it as an unconstrained factor. This can also be seen quantitatively where lower mean constraint violations in Fig. 5(b) leads to improved state estimates with lower RMSE ATE in Fig. 5(c).

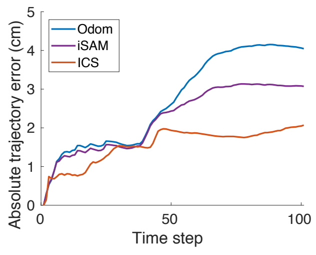
Fig. 6 shows similar set of results as Fig. 5 but now on the real-world MIT CSAIL 2D navigation dataset [21]. For our evaluations, we treat the trajectory with loop closures as the ground truth, and disable loop closures for all other runs. As can be seen qualitatively in Fig. 6(a) ICS is able to satisfy the constraint more precisely (i.e. collision-free trajectories). This is also seen quantitatively in Fig. 6(b),(c) where lower mean constraint violations leads to lower RMSE ATE values.


(a)


(b)

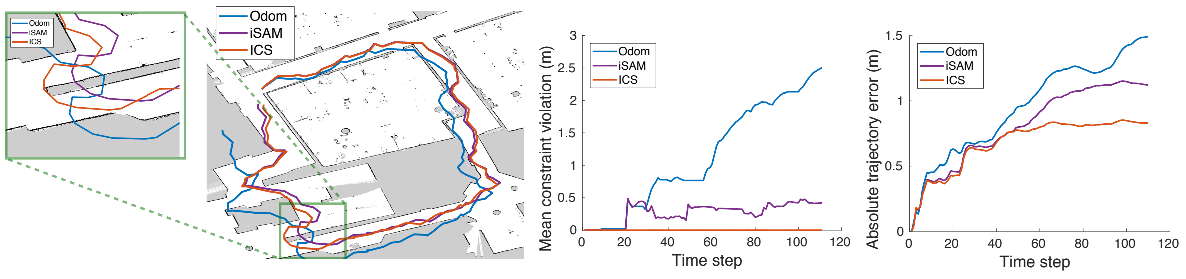


(c)



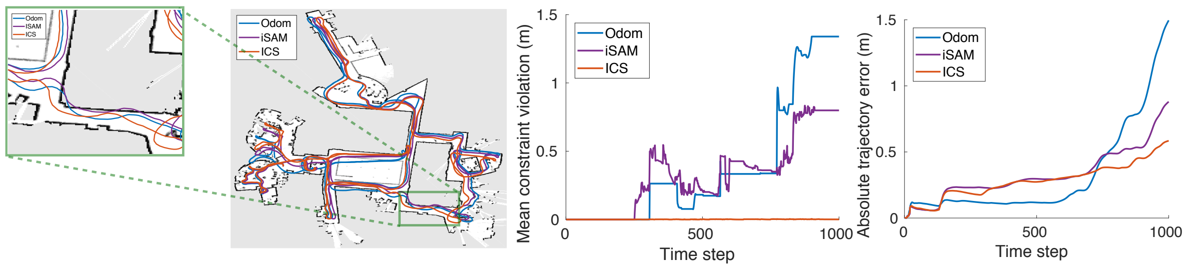
(d)

Fig. 3. Dataset 1: Simulated Contact Performance Accuracies



(a) (b) (c)

Fig. 5. Dataset 2: Simulated Navigation Performance Accuracies



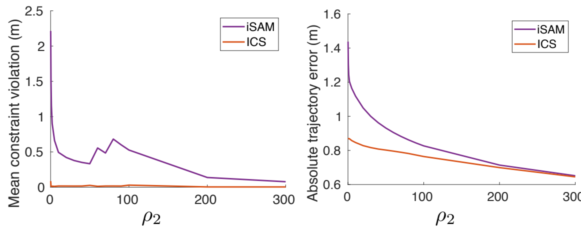
(a) (b) (c)

Fig. 6. Dataset 3: Real-world Navigation Performance Accuracies

The jumps in constraint violations as seen with iSAM in Figs. 5(b), 6(b) happen as iSAM is correcting pure odometry trajectory to minimize the overall unconstrained cost: a combination of trajectory and constraint violation costs.

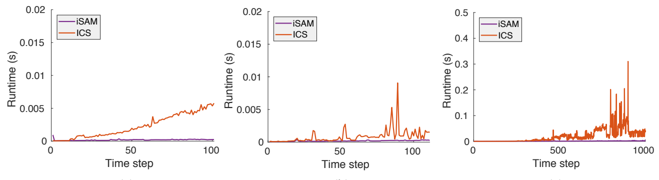
C. *Effect of varying cost weights*

Since constraints within iSAM are incorporated as costs or *soft constraints*, the degree to which they are satisfied would depend on their weights relative to other terms in the cost function. Fig. 7 shows variation of mean constraint violations and RMSE ATE values for iSAM, ICS with varying values of ρ_2 for Dataset 2. ρ_2 is the weighting coefficient of the augmentation term $\frac{\rho_2}{2}||H\delta x + h||_2^2$ associated with the inequality constraint factor. It can be seen that accuracy of an unconstrained framework like iSAM is much more sensitive to changes in this weighting. ICS ensures that constraint violations lie under a set threshold irrespective of ρ_2.



(a) (b)

Fig. 7. Performance accuracies with varying cost weights (Dataset 2)



(a) (b) (c)

Fig. 8. Runtime every time step for **(a)** Dataset 1 (equality constraints) **(b)** Dataset 2 (inequality constraints) **(c)** Dataset 3 (inequality constraints)

D. *Runtime performance*

Fig. 8 shows runtime of an incremental step in ICS/iSAM against number of time steps. ICS is more computationally expensive than iSAM, since iterative primal-dual iterations need to be done every incremental step. However, in comparison to a batch constrained solver (eg. active set method) that solves for the KKT conditions directly, incrementally solving for constraints will be more efficient for large matrices. This is since a direct solution to the KKT conditions involves the expensive operation of factorizing a matrix with $(n+p+q)$ rows/columns from scratch every time step, where n is the dimension of A^TA, and p, q are number of equality, active inequality constraints respectively.

Note that ICS runtimes for dataset 1 (Fig. 8a) increases more steadily compared to datasets 2, 3 (Figs. 8b,c). This happens since equality constraints are always active for any optimal point that is feasible, while inequality constraints may or may not be active at optimal points that are feasible.

VI. Discussions

We presented a framework ICS for incremental constrained smoothing. By combining a primal-dual method like Augmented Lagrangian with an incremental Gauss Newton reusing previously computed matrix factorizations, we were able to satisfy hard constraints precisely within a threshold without having to recompute solutions from scratch.

A current limitation is that of a fixed linearization point for older states (similar to iSAM) making it unsuitable for highly nonlinear problems. Future work would be to explore the proposed approach within iSAM2 [4] that allows for fluid relinearizations. Runtime performance of ICS can also be further improved by exploiting sparsity in constraint jacobians, locality in dual updates, and better step size adaptation techniques.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[3] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[4] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, pp. 216–235, Feb. 2012.

[5] K.-T. Yu and A. Rodriguez, "Realtime state estimation with tactile and visual sensing. application to planar manipulation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 7778–7785, IEEE, 2018.

[6] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[8] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium* (G. Lakemeyer and B. Nebel, eds.), Morgan Kaufmann, 2002.

[9] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[11] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[12] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.

[13] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[14] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed SLAM using constrained factor graphs," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3025–3030, IEEE, 2010.

[15] D.-N. Ta, M. Kobilarov, and F. Dellaert, "A factor graph approach to estimation and model predictive control on unmanned aerial vehicles," in *Intl. Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 181–188, IEEE, 2014.

[16] S. Choudhary, L. Carlone, H. I. Christensen, and F. Dellaert, "Exactly sparse memory efficient SLAM using the multi-block alternating direction method of multipliers," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1349–1356, IEEE, 2015.

[17] I. D. D. Jimenez Rodriguez, "A factor graph approach to constrained optimization," 2017.

[18] L. Carlone, G. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Trans. Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[19] L. Carlone, D. Rosen, G. Calafiore, J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 125–132, 2015.

[20] F. Bai, T. Vidal-Calleja, and S. Huang, "Robust incremental slam under constrained optimization formulation," *IEEE Robotics and Automation Letters*, no. 2, pp. 1207–1214, 2018.

[21] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014.