

“© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Incremental SQP Method for Constrained Optimization Formulation in SLAM

Fang Bai*, Shoudong Huang*, Teresa Vidal-Calleja*, and Qingling Zhang[§]

*Faculty of Engineering and Information Technology
University of Technology Sydney, Ultimo, NSW 2007

[§]Institute of Systems Science
Northeastern University, Shenyang, Liaoning 110819, China

Email: Fang.Bai@student.uts.edu.au, Shoudong.Huang@uts.edu.au, Teresa.VidalCalleja@uts.edu.au, qlzhang@mail.neu.edu.cn

Abstract—The simultaneous localization and mapping (SLAM) problem has been a research focus for many years and have reached a mature state. However, more robust solutions to the SLAM problem are still required, especially in large noise level scenarios. Because of the strong non-linearity of the SLAM problem, it is vital to start from a good initial value to avoid being trapped in local minima. In this paper, we propose a new SLAM formulation transforming the unconstrained Least Squares formulation into a constrained optimization problem. Algorithms based on this new formulation can naturally start from good initial value. Different from other constrained optimization problem, this new formulation can be efficiently solved with Sequential Quadratic Programming (SQP) methods. Based on SQP, we propose an incremental SQP algorithm to solve SLAM, which shows great advantage over Gauss Newton (g2o implementation) when working in large noise level scenarios. Experimental results show the validity of the proposed approach.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a critical problem when a robot is navigating in an unknown environment without accurate information of its location. SLAM is fundamental when an external localization reference is not available and extremely important in the presence of noise which tends to accumulate during the robot's motion and observation processes. Since the solution to the SLAM problem enables robots to build a map without any prior knowledge of the environment and then possibly to navigate within the map, it has been a key research focus in robotics for two decades.

In essence, the most popular SLAM formulation aims to maximize the likelihood of the estimate of the robot's location and environment parameterization by solving a nonlinear optimization problem. A well-known approach arising from this is the state-of-the-art algorithm in which SLAM is formulated as nonlinear least squares (LS). Different algorithms have been developed to solve this optimization problem and to this extent, SLAM can be treated as a solved problem.

However some issues remain. One of the issues is how to find the global optimal solution to the problem. Until now, all methods developed are at the risk of getting trapped in a local minimum, in particular when the noise levels are large or the robot's trajectory is long. This is thorny in real applications because sometimes an incorrect SLAM solution (local minimum) can put human lives or the whole robotics system in danger. Therefore, more robust solutions to the SLAM problem are always in demand.

In recent years, a variety of approaches arised to address the convergence to local minima. The properties and factors that have impact on the local convergence of SLAM are investigated by Huang et al. [1], Liu et al. [8], Wang et al. [4], Carlone et al. [15] and Khosoussi et al. [9] [10], giving insights on how the rotation part contributes to the nonlinearity of the problem. Wang et al. [6] try to simplify the problem into a multi-dimensional optimization problem with respect to sensor's orientations and some properties are revealed for simple cases such as one-step SLAM under the assumption of spherical covariance matrices [5]. Besides the theoretical analysis on local convergence, some efforts to improve SLAM algorithms performance are made. Methods with larger basins of convergence have been proposed by Olson et al. [12], Grisetti et al. [11], and Rosen et al. [3], which can ensure globally optimal solutions in a certain local area of the state space. Other methods are based on bootstrapping [7], [13], [14] exploiting the theory that an iterative optimization algorithm (for example Gauss-Newton) is likely to converge to the global optimum if the algorithm starts from a good initial estimate. In addition, recent work by Carlone et al. [2] attempts to evaluate the optimality of the candidate SLAM solution using Lagrangian duality. While all these works give remarkable insights when dealing with local minima, more reliable algorithms to solve SLAM are still required.

In this paper, we firstly transform the unconstrained least squares SLAM formulation into a constrained optimization problem. With this new formulation, algorithms naturally start from reliable initial values which is the data gathered by the sensor. Based on this new formulation, we propose an incremental Sequential Quadratic Programming (SQP) [17] method to obtain more robust solutions for SLAM. Compared with other linearization based iterative algorithms such as Gauss Newton (g2o [16] implementation), our algorithm has two advantages: (1) The algorithm can always start from good initial value. (2) It can acquire better solutions than Gauss Newton in terms of the evaluation of the objective function, although the algorithm still cannot guarantee the global optimum.

The following sections are organized as follows. Section II reviews the unconstrained least squares SLAM formulation. Section III details the constrained SLAM formulation. Section IV describes a SQP method to solve the constrained SLAM problem. Based on general SQP, Section V describes the

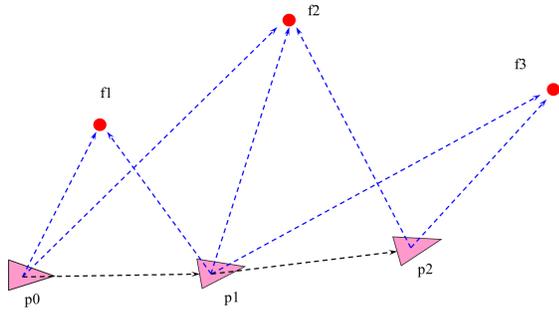


Fig. 1: An example of point-feature-based SLAM.

proposed incremental SQP method. Experimental results are provided to validate the effectiveness of the proposed method in Section VI. Section VII gives more insights on the proposed methods. Section VIII concludes the paper.

II. LEAST SQUARES SLAM FORMULATION

In a feature-based SLAM problem, all the variables that need to be optimized include:

- robot poses: a pose includes robot position and orientation
- feature positions

For a 2D SLAM problem, a robot pose p_i can be described by its position X_{p_i} and its orientation ϕ_{p_i} . The position of a feature f_j can be described by X_{f_j} . For example in Fig. 1 (a 3-pose-3-feature SLAM problem), the robot's position at pose p_1 can be described by $X_{p_1} = (x_{p_1}, y_{p_1})$, meanwhile the orientation is given by ϕ_1 . Similarly, the position of feature f_2 can be represented by $X_{f_2} = (x_{f_2}, y_{f_2})$.

A standard formulation of point feature-based SLAM is nonlinear least squares which is used to optimize the poses and features in a state vector

$$X_l^T = [\dots, X_{p_i}^T, \dots, \phi_{p_i}, \dots, X_{f_j}^T, \dots]$$

by the objective function

$$\sum_{i=1}^p \|\hat{O}_i^{i-1} - O_i^{i-1}(X_l)\|_{P_{O_i^{i-1}}}^2 + \sum_{i,j} \|\hat{Z}_j^i - Z_j^i(X_l)\|_{P_{Z_j^i}}^2.$$

where $\|X\|_P^2 = X^T P^{-1} X$ is the squared Mahalanobis distance with covariance matrix P .

Here \hat{O}_i^{i-1} represents the odometry data from pose p_{i-1} to pose p_i , and \hat{Z}_j^i represents the observation data from pose p_i to feature f_j . The mathematical models of odometry and observation are given by

$$O_i^{i-1}(X_l) = \begin{bmatrix} O_{i(xy)}^{i-1}(X_l) \\ O_{i(\phi)}^{i-1}(X_l) \end{bmatrix} = \begin{bmatrix} R(\phi_{p_{i-1}})^T (X_{p_i} - X_{p_{i-1}}) \\ \phi_{p_i} - \phi_{p_{i-1}} \end{bmatrix}$$

and

$$Z_j^i(X_l) = R(\phi_{p_i})^T (X_{f_j} - X_{p_i}),$$

where $R(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}$ is the rotation matrix with respect to ϕ .

III. CONSTRAINED FORMULATION

Because of the non-linearity of the SLAM problem, finding a good initial value for SLAM algorithms has always been a big issue. For LS (unconstrained) feature-based SLAM formulation, researchers have used bootstrapping methods to get a good initial value for the algorithm. However, due to the noises of the gathered data, bootstrapping methods cannot always build a good enough initial value.

This section presents a novel feature-based SLAM formulation by regarding the noise-free odometry (i.e. relative poses) and observations as new variables to be estimated. SLAM algorithms developed based on this formulation will always start from good initial value at that stage which is the data gathered by the sensor, and subsequently, SLAM becomes a constrained optimization problem.

In the constrained SLAM formulation, variables to be optimized are relative poses and observations. Although all the observations are assumed to be made independently, the observation value to a fixed feature are theoretically correlated. Constraints provide relationships among the correlated relative poses and observations. For example, in Fig. 2, feature f_2 is observed from pose p_0 , pose p_1 and pose p_2 . Due to the fact that the feature is static, there must be some relationships among these three observations Z_2^0 , Z_2^1 , Z_2^2 . This gives rise to the concept of constraints.

In Fig. 2(a), feature f_2 is firstly observed at pose p_0 , then after 2 time steps with relative poses O_1^0 , O_2^1 , it is observed again at pose p_2 . If feature f_2 is static, it is clear that these two observations Z_2^0 , Z_2^2 are correlated. The correlation is described by the constraint. In this example, the observation Z_2^0 can be expressed by observation Z_2^2 , and the relative poses O_1^0 , O_2^1 . The exact expression that describes this correlation is given by

$$Z_2^0 = O_{1(xy)}^0 + R(O_{1(\phi)}^0)O_{2(xy)}^1 + R(O_{1(\phi)}^0)R(O_{2(\phi)}^1)Z_2^2 \quad (1)$$

The other two constraints with regard to f_2 are illustrated in Fig. 2(b). These two constraints give the correlation between Z_2^0 and Z_2^1 , and that between Z_2^1 and Z_2^2 , which are described by two expressions below.

$$Z_2^0 = O_{1(xy)}^0 + R(O_{1(\phi)}^0)Z_2^1 \quad (2)$$

$$Z_2^1 = O_{2(xy)}^1 + R(O_{2(\phi)}^1)Z_2^2 \quad (3)$$

An interesting phenomenon arising from these expressions is that if we substitute Eq. (3) into Eq. (2), we get a new recursive expression

$$Z_2^0 = O_{1(xy)}^0 + R(O_{1(\phi)}^0)[O_{2(xy)}^1 + R(O_{2(\phi)}^1)Z_2^2] \quad (4)$$

for the correlation between Z_2^0 and Z_2^2 . By expanding Eq. (4), we get the same formula as Eq. (1), which means these three constraints are not independent. Moreover, this implicates that constraints with multiple relative poses like Eq. (1) can be built recursively using one-step relative poses by the virtue of sequentially substituting one constraint into another.

The more general scenario for constraint construction is given by the following theorem.

Theorem 1: If a feature f_k is observed from p_i and p_j ($j = i + m$), these two observations Z_k^i , Z_k^{i+m} and all the

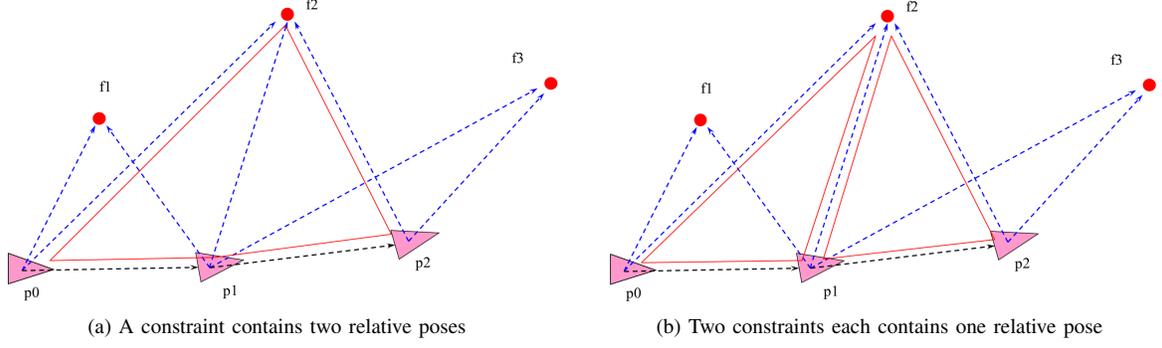


Fig. 2: Examples of constraints.

relative poses $O_{i+1}^i, O_{i+2}^{i+1}, \dots, O_{i+m}^{i+m-1}$ from pose p_i to pose p_{i+m} constitute a constraint

$$Z_k^i = \left[\prod_{l=0}^{m-1} R(O_{i+l+1(\phi)}^{i+l}) \right] Z_k^{i+m} + \sum_{t=0}^{m-1} \left[\prod_{l=0}^{t-1} R(O_{i+l+1(\phi)}^{i+l}) \right] O_{i+t+1(xy)}^{i+t} \quad (5)$$

when $t = 0$, $\prod_{l=0}^{t-1} R(O_{i+l+1(\phi)}^{i+l}) = I_2$.

Proof: By the definition of observations, we have

$$\begin{aligned} Z_k^i(X_l) &= R(\phi_{p_i})^T (X_{f_k} - X_{p_i}) \\ &= R(\phi_{p_i})^T \left[X_{f_k} + \sum_{t=1}^m (X_{p_{i+t}} - X_{p_{i+t-1}}) - X_{p_i} \right] \\ &= R(\phi_{p_i})^T \left[X_{f_k} - X_{p_{i+m}} + \sum_{t=0}^{m-1} (X_{p_{i+t+1}} - X_{p_{i+t}}) \right] \\ &= R(\phi_{p_i})^T (X_{f_k} - X_{p_{i+m}}) + \sum_{t=0}^{m-1} R(\phi_{p_i})^T (X_{p_{i+t+1}} - X_{p_{i+t}}). \end{aligned}$$

Noting that $R(-\theta) = R(\theta)^T$, $R(\theta_1 + \theta_2) = R(\theta_1)R(\theta_2)$, we have

$$\begin{aligned} R(\phi_{p_i})^T &= R(\phi_{p_i} + \sum_{l=1}^m (\phi_{p_{i+l}} - \phi_{p_{i+l-1}}))^T \\ &= R(\phi_{p_{i+m}} + \sum_{l=0}^{m-1} (\phi_{p_{i+l}} - \phi_{p_{i+l+1}}))^T \\ &= \left[\prod_{l=0}^{m-1} R(\phi_{p_{i+l+1}} - \phi_{p_{i+l}}) \right] R(\phi_{p_{i+m}})^T \\ &= \left[\prod_{l=0}^{m-1} R(O_{i+l+1(\phi)}^{i+l}(X_l)) \right] R(\phi_{p_{i+m}})^T. \end{aligned}$$

Similarly,

$$R(\phi_{p_i})^T = \left[\prod_{l=0}^{t-1} R(O_{i+l+1(\phi)}^{i+l}(X_l)) \right] R(\phi_{p_{i+t}})^T$$

when $t = 0$, $\prod_{l=0}^{t-1} R(O_{i+l+1(\phi)}^{i+l}) = I_2$. Hence,

$$\begin{aligned} Z_k^i(X_l) &= \left[\prod_{l=0}^{m-1} R(O_{i+l+1(\phi)}^{i+l}(X_l)) \right] R(\phi_{p_{i+m}})^T (X_{f_k} - X_{p_{i+m}}) \\ &\quad + \sum_{t=0}^{m-1} \left[\prod_{l=0}^{t-1} R(O_{i+l+1(\phi)}^{i+l}(X_l)) \right] R(\phi_{p_{i+t}})^T (X_{p_{i+t+1}} - X_{p_{i+t}}) \\ &= \left[\prod_{l=0}^{m-1} R(O_{i+l+1(\phi)}^{i+l}(X_l)) \right] Z_k^{i+m}(X_l) \\ &\quad + \sum_{t=0}^{m-1} \left[\prod_{l=0}^{t-1} R(O_{i+l+1(\phi)}^{i+l}(X_l)) \right] O_{i+t+1(xy)}^{i+t}(X_l) \end{aligned}$$

This completes the proof. \blacksquare

To summarize, the state vector of constrained SLAM formulation is

$$X^T = [\dots, (O_{i+1}^i)^T, \dots, (Z_k^i)^T, \dots].$$

The objective becomes

$$\sum_{i=1}^p \|\hat{O}_i^{i-1} - O_i^{i-1}\|_{P_{O_i^{i-1}}}^2 + \sum_{i,j} \|\hat{Z}_j^i - Z_j^i\|_{P_{Z_j^i}}^2 = \|\hat{X} - X\|_P^2$$

where $P = \text{diag}(\dots, P_{O_{i+1}^i}, \dots, P_{Z_k^i}, \dots)$.

The problem can be explicitly written as

$$\begin{aligned} \min \quad & \|\hat{X} - X\|_P^2 \\ \text{s.t.} \quad & C_i(X) = 0 \quad (i \in D) \end{aligned} \quad (6)$$

where $C_i(X) = 0$ describes the i th nonlinear constraint and D is the set of all constraints indices.

IV. SQP METHOD

In the constrained optimization based SLAM formulation, the objective function is quadratic. If the constraints can be linearized appropriately, the problem can be solved by SQP. This section will describe the approach to linearize the constraints and the implementation of the SQP method to this particular problem.

A. Linearization

The nonlinear constraint is a sum of product terms in the form of $[\prod_{i=1}^n R(\phi_i)]Y$. Hence, if term $[\prod_{i=1}^n R(\phi_i)]Y$ can be linearized, the whole constraint can be linearized by gradually linearizing all the product terms and then summing up the results.

Theorem 2: The linearization of $[\prod_{i=1}^n R(\phi_i)]Y$ based on the first-order Taylor expansion is

$$\begin{aligned} & [\prod_{i=1}^n R(\phi_i)]Y \\ &= [\prod_{j=1}^n R(\phi_j^0)]Y + \\ & \sum_{i=1}^n R'(\phi_i^0)R(\phi_i^0)^T [\prod_{j=1}^n R(\phi_j^0)]Y^0 \Delta\phi_i + o \end{aligned} \quad (7)$$

where $\phi_1^0, \phi_2^0, \dots, \phi_n^0, Y^0$ is the initial linearization point, and o is the remainder in Taylor expansion.

Proof: Let

$$\begin{aligned} W &= \{\phi_1, \phi_2, \dots, \phi_n, Y\} \\ W_0 &= \{\phi_1^0, \phi_2^0, \dots, \phi_n^0, Y^0\} \end{aligned}$$

$$F(\phi_1, \phi_2, \dots, \phi_n, Y) = F(W) = [\prod_{i=1}^n R(\phi_i)]Y$$

The first-order Taylor expansion of $F(W)$ at point W_0 is

$$\begin{aligned} F(W) &= F(W_0) + \frac{\partial F(W_0)}{\partial Y} \Delta Y + \sum_{i=1}^n \frac{\partial F(W_0)}{\partial \phi_i} \Delta \phi_i + o \\ &= R(\phi_1^0)R(\phi_2^0) \cdots R(\phi_n^0)Y^0 + \\ & R(\phi_1^0)R(\phi_2^0) \cdots R(\phi_n^0)\Delta Y + \\ & \sum_{i=1}^n R(\phi_1^0) \cdots R'(\phi_i^0) \cdots R(\phi_n^0)\Delta \phi_i + o \\ &= R(\phi_1^0)R(\phi_2^0) \cdots R(\phi_n^0)Y + \\ & \sum_{i=1}^n R(\phi_1^0) \cdots R'(\phi_i^0) \cdots R(\phi_n^0)\Delta \phi_i + o \end{aligned}$$

Combing the fact that

$$R(\phi_1^0) \cdots R'(\phi_i^0) \cdots R(\phi_n^0) = R'(\phi_i^0)R^T(\phi_i^0) \prod_{j=1}^n R(\phi_j^0)$$

the proof is straightforward. This completes the proof. \blacksquare

After linearizing all the constraints, the results can be generalized into the linear form $AX = b$. In the following discussion, we will use this form to represent the linear constraint.

B. SQP Algorithm

SQP is carried out by linearizing the constraints and computing the solution of a quadratic programming problem

$$\begin{aligned} \min \quad & \|\hat{X} - X\|_P^2 \\ \text{s.t.} \quad & AX = b \end{aligned} \quad (8)$$

Algorithm 1 SQP Algorithm

- 1: Set initial value $X_0 = \hat{X}$
- 2: **repeat**
- 3: For every $i \in D$, at X_k , linearize constraint

$$C_i(X) = 0 \Rightarrow A_k^i X = b_k^i$$

- 4: Get A_k, b_k by putting A_k^i, b_k^i ($i \in D$) together
- 5: Compute the solution and objective function of the quadratic programming problem by

$$X_{k+1} = \hat{X} - PA_k^T(A_k PA_k^T)^{-1}(A_k \hat{X} - b_k)$$

$$F_{k+1} = \|\hat{X} - X_{k+1}\|_P^2$$

- 6: **until** $C_i(X) = 0$ ($i \in D$) **and** X_{k+1} converge **and** F_{k+1} converge
-

sequentially. The solution to the quadratic programming problem Eq. (8) can be obtained conveniently by Lemma 1.

Lemma 1: [18] The solution to the quadratic programming problem in Eq. (8) is

$$X^* = \hat{X} - PA^T(APA^T)^{-1}(A\hat{X} - b).$$

In the first iteration of the SQP algorithm, SQP uses the data \hat{X} as initial value X_0 to linearize the constraint $C_i(X) = 0$ ($i \in D$) and get linear constraints $A_0 X = b_0$. After linearization, the solution to the quadratic programming problem can be directly obtained by Lemma 1, denoted as X_1 . Then by linearizing the constraint $C_i(X) = 0$ ($i \in D$) at point X_1 and computing the solution to the quadratic programming problem again, we get solution X_2 . The final solution of SQP is acquired by iterating this process until it converges. The SQP method is described in Algorithm 1.

V. INCREMENTAL SQP METHOD

In SQP algorithm, all the constraints are considered together. Since a constraint whose value $C_i(X)$ is far from zero at the linearization point, may result in bad linearization quality and a leap of linearization points between two iterations. Thus, in some cases it is difficult for SQP to converge to the correct solution. To improve the performance of the SQP method, this section presents a new way to organize and add the constraints rather than including them all together into the algorithm from the start.

Considering the case that none of the constraints is added into the problem, the solution to the problem is the data itself and the objective function of the problem is exactly zero. Every time we add some constraints into the problem, the solution may change a little bit and the objective function value will grow. It is evident that if the constraints added into the problem naturally hold for the existing solution, the solution at that step will not change.

The insight here is that if we reorganize the constraints by the value of $\|C_i(X)\|$, and at each iteration the constraint with the smallest value $\|C_i(X)\|$ is added, the objective will grow slightly. Finally, when all the constraints have been added into the problem, we will get a solution with a relatively small objective. Details on how to implement incremental SQP method are presented in Algorithm 2. Note that in the

Algorithm 2 Incremental SQP Algorithm

- 1: Initialize $X_0 = \hat{X}$, $D_s = 0$
 - 2: **repeat**
 - 3: Sort $v_i = \|C_i(X_0)\|$ ($i \in \{D - D_s\}$) in the ascending order
 - 4: Add i with the smallest value v_i into D_s {Add constraint $C_i(X) = 0$ into the sub-problem}
 - 5: Formulate a sub-problem s

$$\begin{aligned} \min \quad & \|\hat{X} - X\|_P^2 \\ \text{s.t.} \quad & C_i(X) = 0 \quad (i \in D_s) \end{aligned}$$
 - 6: Solve the sub-problem by Algorithm 1. Denote the solution to the sub-problem by X^*
 - 7: Set initial value $X_0 = X^*$
 - 8: **until** $D_s = D$ {All constraints are added}
-

algorithm, we use D_s to denote the subset of constraints indices added to the sub-problem s .

It is worth noting that the initial value of incremental SQP method is based on the data, then it is reasonable to assume that the method only works well at low noise levels. However, our experiments show that the incremental SQP method also outperforms SQP and Gauss Newton (g2o implementation) when the noise is large. The simulation results in the next section show the robustness of the incremental SQP method.

VI. EXPERIMENTAL RESULTS

This section presents experimental results of SQP algorithm (SQP), incremental SQP algorithm (iSQP) and Gauss Newton (GN) using simulation data and the DLR dataset [19].

In our simulation experiments, data are generated via different noise levels and the solutions to the simulation data are computed using iSQP, SQP and GN accordingly. The ground truth of the robot's trajectory and features are plotted in Fig. 3, in which the black circles represent the robot poses and blue dots represent the features. The robot starts from the origin of the coordinate frame and moves in a square trajectory, with 1m relative poses each step until it returns to the origin of the

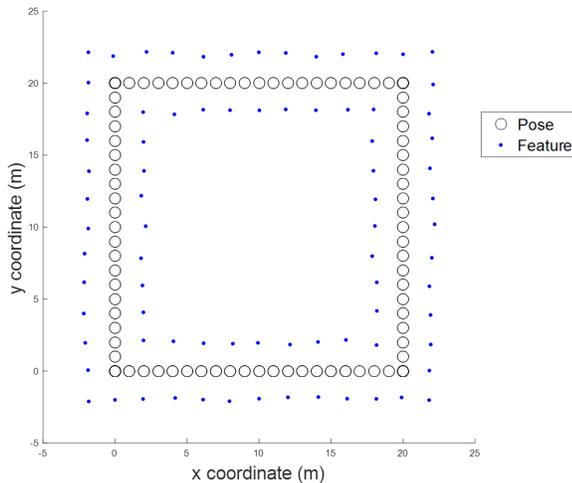


Fig. 3: Robot's trajectory and features in the simulation.

TABLE I: Performance of iSQP, SQP and GN

Noise Level		Same Out of	Difference Out of 500 Trials		
N_{xy}	N_ϕ	500 Trials	iSQP	SQP	GN
0.1	10	465	35/12	2/0	21/0
0.1	15	424	76/47	4/0	25/0
0.2	10	473	27/6	1/0	20/0
0.2	15	438	62/33	6/0	23/0
0.3	10	484	16/6	2/0	8/0
0.3	15	422	78/39	7/0	32/0

coordinate frame. In the experiments, the robot can observe features in the range of 6m from its current pose with 360° field-of-view.

To compare the robustness of the proposed method with GN, the experiments are done with different noise levels in a total number of 500 trials for each noise level. The noise is added linearly to the odometry and observations, where N_{xy} represents the standard deviation of the position noise and N_ϕ is the standard deviation of the angle noise. For example $N_{xy} = 0.1$, $N_\phi = 10$ means the standard deviations of positions and angles are 0.1m and 10 degree respectively as per relative poses.

The results are shown in Table I. In Table I, "Same Out of 500 Trials" means the times that all three algorithms converge to the same minima (global or local). "Difference Out of 500 Trials" is the times that at least one algorithm converges to a better solution. " n_1/n_2 " means the algorithm converges to a better solution together with another algorithm for n_1 times, and this algorithm solely converges to a better solution for n_2 times. For example, in the first row, the standard deviation of the position and angle noise are 0.1m and 10 degree. Under this noise level, the three algorithms converge to the same solutions for 465 times out of 500 trials. 35 times of trials show different results. Within these 35 trials, iSQP get the best solution for 35 times (every time), while the times for SQP and GN to also achieve the best solution are only 2 and 21 respectively.

From Table I, we can see that in the trials which give

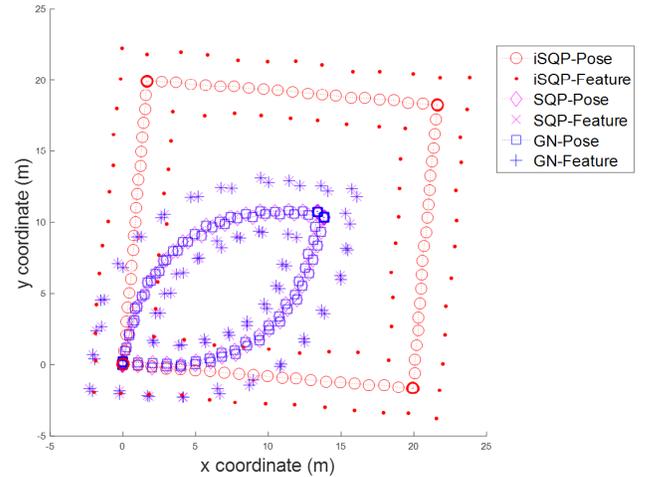


Fig. 4: An example of solutions obtained by iSQP, SQP and GN (the results of SQP and GN are identical).

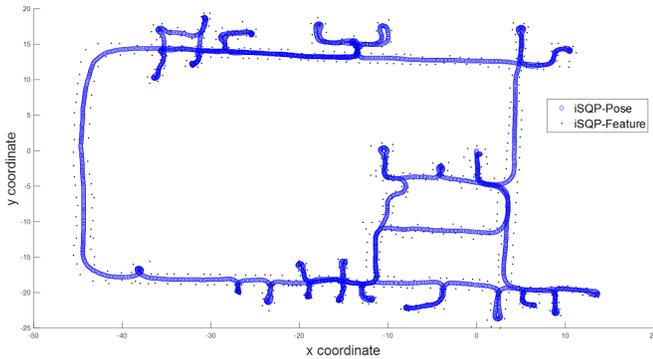


Fig. 5: DLR dataset computed by iSQP algorithm.

different results, the proposed iSQP algorithm can always converge to the best solution, while SQP and GN may fail sometimes. For some cases, only iSQP gets good solutions. An example that SQP and GN converge to a poor solution and iSQP converges to a better solution is given in Fig. 4.

The proposed iSQP algorithm is also tested in a real experimental dataset. The solution of the iSQP algorithm on the DLR dataset [19] is depicted in Fig. 5. The result is the same as that obtained from GN or SQP.

VII. DISCUSSIONS

Based on the analysis and results, we would like to highlight the following points:

- The constrained optimization based SLAM formulation has extracted the nonlinear part of SLAM into constraints which give the relationship among related relative poses and observations. The value of the constraints describes the consistency of the related relative poses and observations.
- In the new constrained SLAM formulation, we can always use the whole data gathered as the initial value to start algorithms. This kind of initial values contain all the information gathered by the sensor and the error only depends on the sensor noises, in contrast to initial values in LS formulation which only use part of information gathered and the error for robot poses and features far away from origin accumulates due to the robot motion.
- For SQP and iSQP, the algorithms only optimize the data relative to each other without paying attention to the actual global poses and features. Therefore the method can work without specifying the coordinate frame. The coordinate frame is only assigned when computing the global poses and features.
- In the proposed iSQP algorithm, by adding constraints each time with the value $\|C_i(X)\|$ close to zero, the algorithm can acquire better linearization quality than SQP, which makes iSQP more robust. However, there might be other better approaches to add the constraints incrementally. The proposed incremental SQP algorithm is one approach which already shows the advantages.

VIII. CONCLUSION

In conclusion, this paper presents a new formulation for the SLAM problem by regarding relative poses and observations as variables to be estimated and therefore transforming LS into

a constrained optimization problem. Algorithms based on this new formulation can naturally start with a good initial value. A SQP algorithm can be used to solve the constrained optimization problem. Moreover, an incremental SQP algorithm, a more robust SQP algorithm which adds the constraints incrementally and can achieve solutions with improved quality, is proposed. The robustness of the proposed algorithm is validated by the simulation and experimental results. The future work includes extending iSQP to pose-graph SLAM and 3D SLAM problems, and reducing the computational complexity of the algorithm.

REFERENCES

- [1] Shoudong Huang, Yingwu Lai, Udo Frese, and Gamini Dissanayake. "How far is SLAM from a linear least squares problem?" In *Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference on, pp. 3011-3016. IEEE, 2010.
- [2] Luca Carlone, David M. Rosen, Giuseppe Calafiore, John J. Leonard, and Frank Dellaert. "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions." In *Intelligent Robots and Systems (IROS)*, 2015 IEEE/RSJ International Conference on, pp. 125-132. IEEE, 2015.
- [3] David M. Rosen, Michael Kaess, and John J. Leonard. "An incremental trust-region method for robust online sparse least-squares estimation." In *Robotics and Automation (ICRA)*, 2012 IEEE International Conference on, pp. 1262-1269. IEEE, 2012.
- [4] Heng Wang, Gibson Hu, Shoudong Huang, and Gamini Dissanayake. "On the Structure of Nonlinearities in Pose Graph SLAM." In *Robotics: Science and Systems*. 2012.
- [5] Heng Wang, Shoudong Huang, Udo Frese, and Gamini Dissanayake. "The nonlinearity structure of point feature SLAM problems with spherical covariance matrices." *Automatica* 49, no. 10 (2013): 3112-3119.
- [6] Heng Wang, Shoudong Huang, Kasra Khosoussi, Udo Frese, Gamini Dissanayake, and Bingbing Liu. "Dimensionality reduction for point feature SLAM problems with spherical covariance matrices." *Automatica* 51 (2015): 149-157.
- [7] Gibson Hu, Kasra Khosoussi, and Shoudong Huang. "Towards a reliable SLAM back-end." In *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference on, pp. 37-43. IEEE, 2013.
- [8] Minjie Liu, Shoudong Huang, Gamini Dissanayake, and Heng Wang. "A convex optimization based approach for pose slam problems." In *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, pp. 1898-1903. IEEE, 2012.
- [9] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. "Novel insights into the impact of graph structure on SLAM." In *Intelligent Robots and Systems (IROS)*, IEEE/RSJ International Conference on, pp. 2707-2714. IEEE, 2014.
- [10] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. "Exploiting the Separable Structure of SLAM." *Proceedings of Robotics: Science and Systems (RSS)*, 2015.
- [11] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. "Nonlinear constraint network optimization for efficient map learning." *Intelligent Transportation Systems*, IEEE Transactions on 10, no. 3 (2009): 428-439.
- [12] Edwin Olson, John Leonard, and Seth Teller. "Fast iterative alignment of pose graphs with poor initial estimates." In *Robotics and Automation (ICRA)*, Proceedings IEEE International Conference on, pp. 2262-2269. IEEE, 2006.
- [13] Luca Carlone, Rosario Aragues, Jos A. Castellanos, and Basilio Bona. "A linear approximation for graph-based simultaneous localization and mapping." *Robotics: Science and Systems VII*. Cambridge, USA: MIT Press (2012): 41-48.
- [14] David M. Rosen, Charles DuHadway, and John J. Leonard. "A convex relaxation for approximate global optimization in simultaneous localization and mapping." In *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on, pp. 5822-5829. IEEE, 2015.
- [15] Luca Carlone. "A convergence analysis for pose graph optimization via gauss-newton methods." In *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, pp. 965-972. IEEE, 2013.
- [16] Giorgio Grisetti, Rainer Kummerle, Hauke Strasdat, and Kurt Konolige. "g2o: A general framework for graph optimization." *IEEE International Conference on Robotics and Automation*. 2011.
- [17] Paul T. Boggs, and Jon W. Tolle. "Sequential quadratic programming." *Acta numerica* 4 (1995): 1-51.
- [18] Michele Benzi. "Solution of equality-constrained quadratic programming problems by a projection iterative method." *Rend. Mat. Appl.*(7) 13 (1993): 275-296.
- [19] Jorg Kurlbaum, and Udo Frese. "A benchmark data set for data association." *Univ. Bremen, Bremen, Germany, SFB/TR 8* (2009): 017-02.