CPL-Sync: Efficient and Guaranteed Planar Pose Graph Optimization Using the Complex Number Representation

Taosha Fan   Hanlin Wang   Michael Rubenstein   Todd Murphey

***Abstract*— In this paper, we present CPL-Sync, a certifiably correct algorithm to solve planar pose graph optimization (PGO) using the complex number representation. We formulate planar PGO as the maximum likelihood estimation (MLE) on the product of unit complex numbers, and relax this nonconvex quadratic complex optimization problem to complex semidefinite programming (SDP). Furthermore, we simplify the corresponding semidefinite programming to Riemannian staircase optimization (RSO) on complex oblique manifolds that can be solved with the Riemannian trust region (RTR) method. In addition, we prove that the SDP relaxation and RSO simplification are tight as long as the noise magnitude is below a certain threshold. The efficacy of this work is validated through comparisons with existing methods as well as applications on planar PGO in simultaneous localization and mapping (SLAM), which indicates that the proposed algorithm is more efficient and capable of solving planar PGO certifiably. The C++ code for CPL-Sync is available at https://github.com/fantaosha/CPL-Sync.**

I. Introduction

Pose graph optimization (PGO) estimates poses from their relative noisy measurements, in which each unknown pose is associated with a vertex and each measurement is associated with an edge of the graph. In robotics, PGO has significant applications in simultaneous localization and mapping (SLAM) and has been extensively studied [1]–[3]. In the last decades, a number of methods have been developed to solve PGO [4]–[9] and verify the optimality of solutions to PGO [9]–[11].

In this paper, we consider the problem of planar PGO using the complex number representation. Our work is an extension of [9] that uses Riemannian staircase optimization to solve PGO efficiently and certifiably. In [9], PGO is formulated on $SE(d) \triangleq (\mathbb{R}^d, +) \rtimes SO(d)$ using the matrix representation and further simplified to quadratic programming on $SO(d)$, then it is relaxed to semidefinite programming and solved with the Riemannian staircase optimization (RSO) [12]. For planar PGO, we need to estimate poses on $SE(2) \triangleq (\mathbb{R}^2, +) \rtimes SO(2)$. In [9], the authors use a 2×2 real matrix to represent $SO(2)$, even though a unit 2×1 real vector or a unit complex number is sufficient. Therefore, it is redundant to use the matrix representation of $SE(2)$ to formulate planar PGO. Moreover, as shown in Section VII, the redundancy of representation induces extra computation to planar PGO, which greatly affects the overall efficiency.

In applied mathematics, it is common to use unit complex numbers to represent $SO(2)$ [13], and the complex number representation has been to used to formulate phase synchronization problems on $SO(2)$ [14], [15]. In particular, we want to mention that in [11], the authors have used complex numbers to represent $SO(2)$ and $SE(2)$ for optimality verification of planar PGO, in which the complex number representation makes the analysis much easier and clearer, whereas in this paper, in addition to optimality verification, we have also worked on how to solve planar PGO efficiently and certifiably using the complex number representation.

In this paper, we present, CPL-Sync, a certifiably correct algorithm to solve planar PGO using the complex number representation. Similar to SE-Sync in [9], CPL-Sync uses Riemannian staircase optimization [12] and provides an exact globally optimal solution to planar PGO as long as the noise magnitude is below a certain threshold. In contrast to SE-Sync, CPL-Sync has a relatively simple formulation and planar PGO is relaxed to complex oblique manifolds [16]. Most importantly, since the complex number representation reduces amounts of computation, CPL-Sync is several times faster than SE-Sync on all the tested 2D SLAM benchmark datasets in Section VII.

The rest of this paper is organized as follows. Section II introduces notations that are used throughout this paper. Section III reviews the complex number representation of $SO(2)$ and $SE(2)$. Section IV formulates planar PGO using the complex representation and Section V relaxes planar PGO to complex semidefinite programming. Section VI presents the CLP-Sync algorithm to solve planar PGO. Section VII presents and discusses comparisons of CPL-Sync with existing methods [7]–[9] on a suite of large 2D SLAM benchmark datasets. The conclusions are made in Section VIII.

II. Notation

$\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively; $\mathbb{R}^{m\times n}$ and $\mathbb{C}^{m\times n}$ denote the sets of $m \times n$ real and complex matrices, respectively; $\mathbb{R}^n$ and $\mathbb{C}^n$ denote the sets of $n \times 1$ real and complex vectors, respectively. $\mathbb{C}_1$ and $\mathbb{C}_1^n$ denote the sets of unit complex numbers and $n \times 1$ vectors over unit complex numbers, respectively. $\mathbb{Q}$ denotes the group of $(\mathbb{C}, +) \rtimes (\mathbb{C}_1, \cdot)$ and "$\rtimes$" denotes the semidirect product of groups [17]. $\mathbb{S}^n$ and $\mathbb{H}^n$ denote the sets of $n \times n$ real symmetric matrices and complex Hermitian matrices,

Taosha Fan and Todd Murphey are with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60201, USA. E-mail: taosha.fan@u.northwestern.edu, t-murphey@northwestern.edu

Hanlin Wang and Michael Rubenstein are with the Department of Electrical Engineering and Computer Science, Northwestern Univerity, Evanston, IL 60201, USA. E-mail: hanlinwang@u.northwestern.edu, rubenstein@northwestern.edu

respectively. The notation "$\mathbf{i}$" is reserved for the imaginary unit of complex numbers. The notation $|\cdot|$ denotes the absolute value of real and complex numbers, and the notation $\overline{(\cdot)}$ denote the conjugate of complex numbers. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose of a matrix, respectively. For a complex matrix W, $[W]_{ij}$ denotes its (i,j)-th entry; the notations $\Re(W)$ and $\Im(W)$ denote real matrices such that $W = \Re(W) + \Im(W)\mathbf{i}$; $W \succcurlyeq 0$ means that W is Hermitian and positive semidefinite; $\mathrm{trace}(W)$ denotes the trace of W; $\mathrm{diag}(W)$ extracts the diagonal of W into a vector and $\mathrm{ddiag}(W)$ sets all off-diagonal entries of W to zero; the notations $\|W\|_F$ and $\|W\|_2$ denote the Frobenius norm and the induced-2 norm, respectively. The notation $\langle\cdot,\cdot\rangle$ denotes the real inner product of matrices. For a vector v, the notation $[v]_i$ denotes its i-th entry; $\|v\|^2 = \|v\|_2^2 = \sqrt{\sum_i |[v]_i|^2} = \sqrt{v^H v}$; the notation $\mathrm{diag}(v)$ denotes the diagonal matrix with $\left[\mathrm{diag}(v)\right]_{ii} = v_i$. The notation $\mathbb{1} \in \mathbb{C}^n$ denotes the vector of all-ones. The notation $\mathbf{I} \in \mathbb{C}^{n\times n}$ denotes the identity matrix. For a hidden parameter x whose value we wish to infer, the notations $\underline{x}$, $\tilde{x}$ and $\hat{x}$ denote the true value of x, a noisy observation of $\underline{x}$ and an estimate of $\underline{x}$, respectively.

III. The Complex Number Representation of $SO(2)$ and $SE(2)$

In this section, we give a brief review of $SO(2)$ and $SE(2)$, and show that $SO(2)$ and $SE(2)$ can be represented using complex numbers.

It is known that the set of unit complex numbers

$$\mathbb{C}_1 \triangleq \{a_1 + a_2\mathbf{i} \in \mathbb{C} | a_1^2 + a_2^2 = 1\}$$

forms a group under complex number multiplication "$\cdot$" for which the identity is 1 and the inverse is the conjugate, i.e., for $x, x' \in \mathbb{C}_1$, we obtain [13]

$$x \cdot x' \in \mathbb{C}_1, \quad 1 \cdot x = x \cdot 1 = x, \quad x \cdot \overline{x} = \overline{x} \cdot x = 1.$$

In addition, the group of unit complex numbers $(\mathbb{C}_1, \cdot)$ is diffeomorphic and isomorphic to the matrix Lie group $SO(2)$:

$$\begin{aligned} SO(2) &\triangleq \{\begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix} \in \mathbb{R}^{2\times 2} | a_1^2 + a_2^2 = 1\} \\ &\triangleq \{R \in \mathbb{R}^{2\times 2} | R^T R = \mathbf{I},\, \det(R) = 1\} \end{aligned}$$

under matrix multiplication. As a result, $SO(2)$ can be represented using unit complex numbers $\mathbb{C}_1$. More explicitly, if $R \in SO(2)$ is

$$R = \begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{1}$$

the corresponding unit complex number representation $x \in \mathbb{C}_1$ is

$$x = a_1 + a_2\mathbf{i} = e^{\mathbf{i}\theta} = \cos\theta + \sin\theta\mathbf{i} \tag{2}$$

in which $e^{\mathbf{i}\theta} = \cos\theta + \sin\theta\mathbf{i}$. Furthermore, if $b' = \begin{bmatrix} b_1' & b_2' \end{bmatrix}^T \in \mathbb{R}^2$ is rotated by $R \in SO(2)$ in Eq. (1) from $b = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^T \in \mathbb{R}^2$, i.e.,

$$b_1' = a_1 b_1 - a_2 b_2 = b_1\cos\theta - b_2\sin\theta,$$

$$b_2' = a_1 b_2 + a_2 b_1 = b_2\cos\theta + b_1\sin\theta,$$

we obtain

$$\beta' = x \cdot \beta = \underbrace{a_1 b_1 - a_2 b_2}_{b_1'} + (\underbrace{a_1 b_2 + a_2 b_1}_{b_2'})\mathbf{i}, \tag{3a}$$

or equivalently,

$$\begin{aligned} \beta' &= x \cdot \beta = e^{\mathbf{i}\theta} \cdot \beta \\ &= \underbrace{b_1\cos\theta - b_2\sin\theta}_{b_1'} + (\underbrace{b_2\cos\theta + b_1\sin\theta}_{b_2'})\mathbf{i}, \end{aligned} \tag{3b}$$

in which x is a unit complex number as that given in Eq. (2), and

$$\beta = b_1 + b_2\mathbf{i} \quad \text{and} \quad \beta' = b_1' + b_2'\mathbf{i} \tag{4}$$

are the complex number representation of b and b', respectively. As a result, rotating a vector can also be described using the complex number representation.

In general, the special Euclidean group $SE(2)$ is the matrix Lie group

$$SE(2) \triangleq \{\begin{bmatrix} R & p \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} | R \in SO(2),\, p \in \mathbb{R}^2\}, \tag{5}$$

whose group multiplication is matrix multiplication. In terms of group theory, $SE(2)$ is also represented as the semidirect product of $(\mathbb{R}^2, +)$ and $SO(2)$:

$$SE(2) \triangleq (\mathbb{R}^2, +) \rtimes SO(2),$$

in which "$\rtimes$" denotes the semidirect product of groups [17]. If "$\circ$" is the group multiplication of $SE(2)$ using the matrix representation as Eq. (5), then for $g = (p, R)$, $g' = (p', R') \in SE(2)$, we obtain

$$g \circ g' = (Rp' + p, RR') \tag{6}$$

Following the complex number representation of $SO(2)$ and $\mathbb{R}^2$, the representation of $SE(2)$ as Eq. (5) is diffeomorphic and isomorphic to the semidirect product of $(\mathbb{C}, +)$ and $(\mathbb{C}_1, \cdot)$:

$$\mathbb{Q} \triangleq (\mathbb{C}, +) \rtimes (\mathbb{C}_1, \cdot).$$

If "$\odot$" denotes the group multiplication of $\mathbb{Q}$, from Eqs. (2) and (3), the multiplication of $g \circ g'$ in Eq. (6) is equivalent to

$$q \odot q' = (x \cdot \rho' + \rho, x \cdot x') \in \mathbb{Q}, \tag{7}$$

in which $q = (\rho, x)$, $q' = (\rho', x') \in \mathbb{Q}$. In Eq. (7), x, $x' \in \mathbb{C}_1$ and ρ, $\rho' \in \mathbb{C}$ are the complex number representation of R, $R' \in SO(2)$ and p, $p' \in \mathbb{R}^2$, respectively, which follow the same representation as that in Eqs. (2) and (4). In addition, the identity of $\mathbb{Q}$ is $(0, 1) \in \mathbb{Q}$ and the inverse of $q = (\rho, x) \in \mathbb{Q}$ is

$$q^{-1} = (-\overline{x} \cdot \rho, \overline{x}) \in \mathbb{Q}. \tag{8}$$

As a result, instead of using the matrix representation, we represent $SE(2)$ with a 2-tuple of complex numbers. Furthermore, if $b' \in \mathbb{R}^2$ is transformed by $g \in SE(2)$ from $b \in \mathbb{R}^2$, we obtain

$$\beta' = x \cdot \beta + \rho,$$

in which $q = (\rho, x) \in \mathbb{Q}$ is the complex number representation of $g \in SE(2)$, and β and β' are the complex number representation of b and b', respectively.

For notational convenience, in the rest of paper, we will omit the complex number multiplication "$\cdot$" if there is no ambiguity.

In terms of the computation of group multiplication and transformation only, the complex number representation of $SO(2)$ and $SE(2)$ has the same complexity as the matrix representation. In spite of this, as shown in the following sections, the complex number representation greatly simplifies the analysis for planar PGO, and most importantly, the semidefinite relaxation and Riemannian optimization of planar PGO using the complex number representation is tighter, more simple and requires less computationa than that using the matrix representation in [9].

In the following sections, we will use the complex number representation of $SO(2)$ and $SE(2)$ to formulate and solve planar PGO.

IV. Problem Formulation and Simplification

In this section, we formulate planar PGO as maximum likelihood estimation, and further simplify it to complex quadratic programming on the product of unit complex numbers.

A. Problem Formulation

Planar PGO consists of estimating n unknown poses g_1, g_2, $\cdots$, $g_n \in SE(2)$ with m noisy relative measurements $g_{ij} \triangleq g_i^{-1} g_j \in SE(2)$. Following the matrix representation of $SE(2)$, we assume that each $g_{(\cdot)} \in SE(2)$ is described as $g_{(\cdot)} = (p_{(\cdot)}, R_{(\cdot)})$, in which $p_{(\cdot)} \in \mathbb{R}^2$ and $R_{(\cdot)} \in SO(2)$. According to Section III, the problem is equivalent to estimating n 2-tuples of complex numbers q_1, q_2, $\cdots$, $q_n \in \mathbb{Q}$ with m relative measurements $q_{ij} \triangleq q_i^{-1} \odot q_j \in \mathbb{Q}$, in which $q_{(\cdot)} = (\rho_{(\cdot)}, x_{(\cdot)}) \in \mathbb{Q}$, and $\rho_{(\cdot)} \in \mathbb{C}$ and $x_{(\cdot)} \in \mathbb{C}_1$ are the complex number representation of $p_{(\cdot)} \in \mathbb{R}^2$ and $R_{(\cdot)} \in SO(2)$, respectively. The n unknown poses and m relative measurements can be described with a directed graph $\overrightarrow{G} = (\mathcal{V}, \overrightarrow{\mathcal{E}})$ in which $i \in \mathcal{V} \triangleq \{1, \cdots, n\}$ is associated with g_i or q_i, and $(i,j) \in \overrightarrow{\mathcal{E}} \subset \mathcal{V} \times \mathcal{V}$ if and only if the relative measurement g_{ij} or q_{ij} exists. If the orientation of edges in $\overrightarrow{\mathcal{E}}$ is ignored, we obtain the undirected graph of $\overrightarrow{G}$ that is denoted as $G = (\mathcal{V}, \mathcal{E})$. In the rest of this paper, we assume that $\overrightarrow{G}$ is weakly connected and G is (equivalently) connected. In addition, we assume that the m noisy relative measurements $q_{ij} = (\rho_{ij}, x_{ij})$ are random variables that satisfy

$$\tilde{\rho}_{ij} = \underline{\rho}_{ij} + \rho_{ij}^{\epsilon} \qquad \rho_{ij}^{\epsilon} \sim N(0, \tau_{ij}^{-1}), \tag{9a}$$

$$\tilde{x}_{ij} = \underline{x}_{ij} x_{ij}^{\epsilon} \qquad \tilde{x}_{ij}^{\epsilon} \sim \mathrm{vMF}(1, \kappa_{ij}), \tag{9b}$$

for all $(i,j) \in \overrightarrow{\mathcal{E}}$. In Eq. (9), $\underline{q}_{ij} = (\underline{\rho}_{ij}, \underline{x}_{ij})$ is the true (latent) value of q_{ij}, $N(\mu, \Sigma)$ denotes the complex normal distribution with mean $\mu \in \mathbb{C}$ and covariance $\Sigma \succcurlyeq 0$, and $\mathrm{vMF}(x_0, \kappa)$ denotes the von Mises-Fisher distribution on $\mathbb{C}_1$ with mode $x_0 \in \mathbb{C}_1$, concentration number $\kappa \geq 0$ and the probability density function of $\mathrm{vMF}(x_0, \kappa)$ is [18]

$$f(x; x_0, \kappa) = \frac{1}{c_d(\kappa)} \exp\left(\kappa(\overline{x}_0 x + x_0 \overline{x})\right),$$

in which $c_d(\kappa)$ is a function of κ.

If $\tilde{\rho}_{ij}$ and $\tilde{x}_{ij}$ are independent, from Eqs. (3), (7) and (8), a straightforward algebraic manipulation indicates that the maximum likelihood estimation (MLE) is a least square problem as follows

$$\min_{\substack{x_i \in \mathbb{C}_1,\\ \rho_i \in \mathbb{C}}} \sum_{(i,j) \in \overrightarrow{\mathcal{E}}} \Big[\kappa_{ij} |x_i \tilde{x}_{ij} - x_j|^2 + \tau_{ij} |\rho_j - \rho_i - x_i \tilde{\rho}_{ij}|^2 \Big] \tag{MLE}$$

in which κ_{ij} and τ_{ij} are as given in Eqs. (9a) and (9b). Furthermore, as Proposition 1 states, (MLE) is equivalent to the formulation using the matrix representation in [9].

Proposition 1. The maximum likelihood estimation (MLE) is equivalent to

$$\min_{\substack{R_i \in SO(2),\\ p_i \in \mathbb{R}^2}} \sum_{(i,j) \in \overrightarrow{\mathcal{E}}} \Big[\frac{\kappa_{ij}}{2} \|R_i \widetilde{R}_{ij} - R_j\|_F^2 + \tau_{ij} \|p_j - p_i - R_i \tilde{p}_{ij}\|_F^2 \Big].$$

In the next subsection, we will simplify (MLE) to quadratic programming on the product of unit complex numbers $\mathbb{C}_1^n$.

B. Problem Simplification

The simplification of (MLE) is similar to that of [9, Appendix B], the difference of which is that we use the complex number representation while Rosen *et. al* in [9] use the matrix representation to formulate planar PGO.

For notational convenience, we define $x_{ji} = \overline{x}_{ij}$, $\kappa_{ji} = \kappa_{ij}$ and $\tau_{ji} = \tau_{ij}$, and (MLE) can be reformulated as

$$\min_{q \in \mathbb{C}^n \times \mathbb{C}_1^n} q^H \begin{bmatrix} L(W^\rho) & \widetilde{V} \\ \widetilde{V}^H & L(\widetilde{G}^x) + \widetilde{\Sigma} \end{bmatrix} q \tag{P}$$

in which $q \triangleq \begin{bmatrix} \rho_1 & \cdots & \rho_n & x_1 & \cdots & x_n \end{bmatrix}^T \in \mathbb{C}^n \times \mathbb{C}_1^n$. In (P), we define $L(W^\rho) \in \mathbb{R}^{n\times n}$, $\widetilde{V} \in \mathbb{C}^{n\times n}$, $L(\widetilde{G}^x) \in \mathbb{C}^{n\times n}$ and $\widetilde{\Sigma} \in \mathbb{R}^{n\times n}$ to be

$$[L(W^\rho)]_{ij} \triangleq \begin{cases} \sum\limits_{(i,k)\in\mathcal{E}} \tau_{ik}, & i = j, \\ -\tau_{ij}, & (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[\widetilde{V}]_{ij} \triangleq \begin{cases} \sum\limits_{(i,k)\in\overrightarrow{\mathcal{E}}} \tau_{ik} \tilde{\rho}_{ik}, & i = j, \\ -\tau_{ij} \tilde{\rho}_{ji}, & (j,i) \in \overrightarrow{\mathcal{E}}, \\ 0 & \text{otherwise}, \end{cases}$$

$$[L(\widetilde{G}^x)]_{ij} \triangleq \begin{cases} \sum\limits_{(i,k)\in\mathcal{E}} \kappa_{ik}, & i = j, \\ -\kappa_{ij} \tilde{x}_{ji}, & (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise}, \end{cases}$$

and $\widetilde{\Sigma} \triangleq \mathrm{diag}\{\widetilde{\Sigma}_1, \cdots, \widetilde{\Sigma}_n\}$ with $\widetilde{\Sigma}_i = \sum\limits_{(i,k)\in\overrightarrow{\mathcal{E}}} \tau_{ik}|\widetilde{\rho}_{ik}|^2$, respectively.

If rotational states $x \triangleq \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T \in \mathbb{C}_1^n$ are known, (P) is reduced to unconstrained complex quadratic programming on translational states $\rho \triangleq \begin{bmatrix} \rho_1 & \cdots & \rho_n \end{bmatrix}^T \in \mathbb{C}^n$:

$$\min_{\rho\in\mathbb{C}^n} \ \rho^H L(W^\rho)\rho + 2\langle \rho, \widetilde{V}x\rangle + \underbrace{x^H L(\widetilde{G}^x)x + x^H\widetilde{\Sigma}x}_{\text{constant}}. \quad (10)$$

By definition, $L(W^\rho) \succeq 0$, and according to [19, Proposition 4.2] [1], the solution to Eq. (10) is

$$\rho = -L(W^\rho)^\dagger \widetilde{V}x. \quad (11)$$

Substituting Eq. (11) into (P) and simplifying the resulting equation, we obtain complex quadratic programming on the product of unit complex numbers $\mathbb{C}_1^n$ as follows

$$\min_{x\in\mathbb{C}_1^n} x^H\widetilde{Q}x, \quad (12)$$

in which $\widetilde{Q} = L(\widetilde{G}^x) + \widetilde{\Sigma} - \widetilde{V}^H L(W^\rho)^\dagger \widetilde{V} \succeq 0$.

Furthermore, if we define $\Omega = \mathrm{diag}\{\tau_{e_1}, \cdots, \tau_{e_m}\} \in \mathbb{R}^{m\times m}$ to be the diagonal matrix whose diagonal elements are indexed by the directed edges $e \in \overrightarrow{\mathcal{E}}$ and in which $\tau_e \in \mathbb{R}$ is the precision of translational observations as given in Eq. (9a), and $\widetilde{T} \in \mathbb{C}^{m\times n}$ to be the matrix indexed by $e \in \overrightarrow{\mathcal{E}}$ and $k \in \mathcal{V}$ whose $(e,\,k)$-element is given by

$$[\widetilde{T}]_{ek} \triangleq \begin{cases} -\widetilde{t}_{ik}, & e = (i,\,k) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \quad (13)$$

and $A(\overrightarrow{G}) \in \mathbb{R}^{n\times m}$ to be the matrix indexed by $k \in \mathcal{V}$ and $e \in \overrightarrow{\mathcal{E}}$ whose $(k,\,e)$-element is given by

$$[A(\overrightarrow{G})]_{ke} = \begin{cases} 1, & e = (i,\,k) \in \overrightarrow{\mathcal{E}}, \\ -1, & e = (k,\,j) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \quad (14)$$

then $\widetilde{Q} = L(\widetilde{G}^x) + \widetilde{\Sigma} - \widetilde{V}^H L(W^\rho)^\dagger \widetilde{V}$ can be rewritten as

$$\widetilde{Q} = L(\widetilde{G}^x) + \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T} \quad (15)$$

in which $\Pi \in \mathbb{R}^{m\times m}$ is the matrix of the orthogonal projection operator $\pi : \mathbb{C}^m \to \ker(A(\overrightarrow{G})\Omega^{\frac{1}{2}})$ onto the kernel of $A(\overrightarrow{G})\Omega^{\frac{1}{2}}$. As a result, Eq. (12) is equivalent to

$$\begin{gathered} \min_{x\in\mathbb{C}_1^n} \mathrm{trace}(\widetilde{Q}xx^H), \\ \widetilde{Q} = L(\widetilde{G}^x) + \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T}. \end{gathered} \quad \text{(QP)}$$

For the detailed derivation of (QP), interested readers can refer to the full paper [20].

In the next section, we will relax (QP) to complex semidefinite programming and show that the semidefinite relaxation is tight as long as the noise magnitude is below a certain threshold.

[1]It should be noted that [19, Proposition 4.2] was originally derived for real matrices, however, the results can be generalized to complex matrices as well.

V. The Semidefinite Relaxation

In a similar way to [11], [14], [15], it is straightforward to relax (QP) to

$$\begin{gathered} \min_{X\in\mathbb{H}^n} \langle \widetilde{Q}, X\rangle \\ \text{s.t.} \quad X \succeq 0, \quad \mathrm{diag}(X) = \mathbb{1}. \end{gathered} \quad \text{(SDP)}$$

It should be noted that if $\hat{X} \in \mathbb{H}^n$ has rank one and solves (SDP), then a solution $\hat{x} \in \mathbb{C}_1^n$ to (QP) can be exactly recovered from $\hat{X}$ through singular value decomposition with which $\hat{X} = \hat{x}\hat{x}^H$.

In the rest of section, we will analyze and derive the conditions for the optimality of (QP) and (SDP), and conditions for the tight relaxation of (SDP), all the proofs of which can be found in [20].

From [21], the necessary conditions for the local optimality of (QP) can be well characterized in terms of the Riemannian gradients and Hessians.

Lemma 1. If $\hat{x} \in \mathbb{C}_1^n$ is a local optimum of (QP), then there exists a real diagonal matrix $\hat{\Lambda} \triangleq \Re\{\mathrm{ddiag}(\widetilde{Q}\hat{x}\hat{x}^H)\} \in \mathbb{R}^{n\times n}$ such that $\hat{S} \triangleq \widetilde{Q} - \hat{\Lambda} \in \mathbb{H}^n$ satisfies the following conditions:

(1) $\hat{S}\hat{x} = \mathbf{0}$;
(2) $\langle \dot{x}, \hat{S}\dot{x}\rangle \geq 0$ for all $\dot{x} \in T_{\hat{x}}\mathbb{C}_1^n$.

If $\hat{x}$ satisfies (1), it is a first-order critical point, and if $\hat{x}$ satisfies (1) and (2), it is a second-order critical point.

Proof. See [20]. □

Since (SDP) is convex and the identity matrix $\mathbf{I} \in \mathbb{C}^{n\times n}$ is strictly feasible, the sufficient and necessary conditions for the global optimality of (SDP) can be derived in terms of the Karush-Kuhn-Tucker (KKT) conditions.

Lemma 2. A Hermitian matrix $\hat{X} \in \mathbb{H}^n$ is a global optimum of (SDP) if and only if there exists $\hat{S} \in \mathbb{H}^n$ such that the following conditions hold:

(1) $\mathrm{diag}(\hat{X}) = \mathbb{1}$;
(2) $\hat{X} \succeq 0$;
(3) $\hat{S}\hat{X} = \mathbf{0}$;
(4) $\widetilde{Q} - \hat{S}$ is real diagonal;
(5) $\hat{S} \succeq 0$.

Furthermore, if $\mathrm{rank}(\hat{S}) = n - 1$, then $\hat{X}$ has rank one and is the unique global optimum of (SDP).

Proof. See [20]. □

As a result of Lemmas 1 and 2, we obtain the sufficient conditions for the exact recovery of (QP) from (SDP).

Lemma 3. If $\hat{x} \in \mathbb{C}_1^n$ is a first-order critical point of (QP) and $\hat{S} = \widetilde{Q} - \hat{\Lambda} \succeq 0$ in which $\hat{\Lambda} = \Re\{\mathrm{ddiag}(\widetilde{Q}\hat{x}\hat{x}^H)\}$, then $\hat{x}$ is a global optimum of (QP) and $\hat{X} = \hat{x}\hat{x}^H$ is a global optimum of (SDP). Moreover, if $\mathrm{rank}(\hat{S}) = n - 1$, then $\hat{X}$ is the unique optimum of (SDP).

Proof. See [20]. □

Lemma 3 gives conditions that (SDP) is a tight relaxation of (QP). As a matter of fact, if the noises of measurements are not too large, it is guaranteed that (SDP) is always a tight relaxation of (QP) as the following proposition states.

Proposition 2. Let $\underline{Q} \in \mathbb{H}^n$ be the data matrix of the form Eq. (15) that is constructed with the true (latent) relative measurements $\underline{q}_{ij} = (\underline{\rho}_{ij}, \underline{x}_{ij})$, then there exists a constant $\gamma = \gamma(\underline{Q}) > 0$ such that if $\|\widetilde{Q} - \underline{Q}\|_2 < \gamma$, then (SDP) has the unique global optimum $\hat{X} = \hat{x}\hat{x}^H \in \mathbb{H}^n$, in which $\hat{x} \in \mathbb{C}_1^n$ is a global optimum of (QP).

Proof. See [20]. □

VI. The CPL-Sync Algorithm

In general, interior point methods to solve (SDP) take polynomial time, which is intractable if n is large. Instead of solving (SDP) directly, Boumal *et. al* found that (SDP) can be relaxed to a series of rank-restricted semidefinite programing [12]:

$$\min_{Y \in \mathrm{OB}(r,n)} \operatorname{trace}(\widetilde{Q}YY^H) \qquad (r\text{-SDP})$$

in which

$$\mathrm{OB}(r,n) \triangleq \{Y \in \mathbb{C}^{n\times r} | \operatorname{diag}(YY^H) = \mathbb{1}\}$$

is the complex oblique manifold [16]. Furthermore, (r-SDP) can be a tight relaxation of (SDP) if some conditions are satisfied as stated in Propositions 3 and 4, whose proofs are immediate from [12, Theorem 2].

Proposition 3. If $\hat{Y} \in \mathrm{OB}(p,n)$ is rank-deficient and second-order critical for (r-SDP), then it is globally optimal for (r-SDP) and $\hat{X} = \hat{Y}\hat{Y}^H \in \mathbb{H}^n$ is globally optimal for (SDP).

Proposition 4. If $p \geq \lceil \sqrt{n}\, \rceil$, then for almost all $\widetilde{Q} \in \mathbb{C}^{n\times n}$, every first-order critical $\hat{Y} \in \mathrm{OB}(p,n)$ for (r-SDP) is rank-deficient.

From Propositions 3 and 4, (SDP) is equivalent to successively solving (r-SDP) with the Riemannian trust region (RTR) method [22] for $2 \leq r_1 < r_2 < \cdots < r_k \leq n+1$ until a rank-deficient second-order critical point is found, and such a method is referred as the Riemannian staircase optimization (Algorithm 1) [12], [23]. In addition, it is known that the RTR method solves (r-SDP) locally in polynomial time [12, Proposition 3]. In contrast to interior point methods to solve (SDP), the Riemannian staircase optimization is empirically orders of magnitude faster in solving large-scale semidefinite programming, and has been successfully implemented in [9], [14], [23] to solve semidefinite relaxations of synchronization problems.

As shown in Algorithm 2, the solution rounding of an optimum of $Y^* \in \mathrm{OB}(r,n)$ of (r-SDP) is simply to assign $\hat{x} = \begin{bmatrix}\hat{x}_1 & \cdots & \hat{x}_n\end{bmatrix} \in \mathbb{C}^n$ to be the left-singular vector of Y^* that is associated with the greatest singular value, and then normalize each x_i to get $\hat{x} \in \mathbb{C}_1^n$. Moreover, it should be noted that the solution rounding algorithm can recover the global optimum $\hat{x} \in \mathbb{C}_1^n$ from Y^* as long as the exactness of (SDP) holds.

From algorithms of Riemannian staircase optimization (Algorithm 1) and solution rounding (Algorithm 2), the proposed CPL-Sync algorithm for planar PGO is as shown in Algorithm 3.

Algorithm 1 The Riemannian staircase optimization (RSO)

```
1: Input: Integers $2 \leq r_0 < r_1 < \cdots < r_k \leq n+1$; an initial iterate $x_0 \in \mathbb{C}_1^n$
2: $Y_0 = \begin{bmatrix}\hat{x}_0 & \mathbf{0}\end{bmatrix} \in \mathrm{OB}(r_0, n)$
3: for $i = 1 \to k$ do
4:     Implement the Riemannian optimization to solve
           $Y_i^* = \arg\min_{Y \in \mathrm{OB}(r_i,n)} \operatorname{trace}(\widetilde{Q}YY^H)$
       locally with $Y_i$ as an initial guess
5:     if $\operatorname{rank}(Y_i^*) < p_i$ then
6:         return $Y_i^* \in \mathrm{OB}(r_i, n)$
7:     else
8:         $Y_{i+1} = \begin{bmatrix}\hat{Y}_i & \mathbf{0}\end{bmatrix} \in \mathrm{OB}(r_{i+1}, n)$
9:     end if
10: end for
11: return $Y_i^* \in \mathrm{OB}(r_k, n)$
```

Algorithm 2 The rounding procedure for solutions of (r-SDP)

```
1: Input: An optimum $Y^* \in \mathrm{OB}(r,n)$ to ($r$-SDP)
2: Assign $\hat{x} = \begin{bmatrix}\hat{x}_1 & \cdots & \hat{x}_n\end{bmatrix}^T \in \mathbb{C}^n$ to be the left-singular vector of $Y^*$ that is associated with the greatest singular value
3: for $i = 1 \to n$ do
4:     $\hat{x}_i = \frac{\hat{x}_i}{|\hat{x}_i|}$
5: end for
6: return $\hat{x} \in \mathbb{C}_1^n$
```

Algorithm 3 The CPL-Sync algorithm

```
1: Input: Integers $2 \leq r_0 < r_1 < \cdots < r_k \leq n+1$; an initial iterate $x_0 \in \mathbb{C}_1^n$
2: Implement Algorithm 1 to compute an optimum $Y^* \in \mathrm{OB}(r,n)$
3: Implement Algorithm 2 to compute rotational states $\hat{x} \in \mathbb{C}_1^n$
4: Implement Eq. (11) to compute translational states $\hat{\rho} \in \mathbb{C}^n$
5: return $\hat{x} \in \mathbb{C}_1^n$ and $\hat{\rho} \in \mathbb{C}^n$
```

In particular, we remind the reader that our work is different from [9], [11], [14]. It should be noted that the $n \times n$ complex positive semidefinite matrix $X \in \mathbb{H}^n$ in our semidefinite relaxation can be parameterized with $n^2 - n$ real numbers, whereas the semidefinite relaxation in [9] using the matrix representation needs $2n^2 - 3n$ real numbers to parameterize the $2n \times 2n$ real positive semidefinite matrix,

TABLE I: Results of the 2D SLAM datasets

Dataset	n	m	f^*	PDL-GN [7], [8]	SE-Sync [9]		CPL-Sync [ours]	
				Total time (s)	RTR time (s)	Total time (s)	RTR time (s)	Total time (s)
ais2klinik	15115	16727	1.885×10^2	10.20	2.60	2.71	1.01	1.15
city10000	10000	20687	6.386×10^2	2.0	0.86	1.17	0.515	0.538
CSAIL	1045	1172	3.170×10^1	0.08	0.005	0.014	0.001	0.005
M3500	3500	5453	1.939×10^2	0.32	0.152	0.216	0.074	0.098
M3500-a	3500	5453	1.598×10^3	0.506	0.16	0.228	0.08	0.103
M3500-b	3500	5453	3.676×10^3	2.85	0.527	0.59	0.26	0.28
M3500-c	3500	5453	4.574×10^3	3.2	0.75	0.82	0.37	0.40
manhattan	3500	5453	6.432×10^3	1.88	0.044	0.108	0.019	0.042
intel	1728	2512	5.236×10^1	0.14	0.038	0.061	0.017	0.026
KITTI_00	4541	4677	1.257×10^2	0.41	0.076	0.11	0.027	0.038
KITTI_02	4661	4703	1.084×10^2	0.29	0.053	0.085	0.018	0.029
KITTI_05	2761	2826	2.765×10^2	0.21	0.023	0.032	0.008	0.015
KITTI_06	1101	1150	3.533×10^1	0.076	0.005	0.013	0.001	0.004
KITTI_07	1101	1106	2.393×10^1	0.039	0.006	0.014	0.002	0.005
KITTI_09	1591	1592	6.131×10^1	0.059	0.018	0.029	0.006	0.01

which indicates that our formulation using the complex number representation results in semidefinite relaxations of smaller size. For the semidefinite relaxation using the matrix representation, the authors in [9] actually relax the constraint of the special orthogonal group $SO(2) = \{R \in \mathbb{R}^{2\times 2} | R^T R = \mathbf{I}, \det(R) = 1\}$ to that of the orthogonal group $O(2) = \{A \in \mathbb{R}^{2\times 2} | A^T A = \mathbf{I}\}$, which might induce ambiguities (gauge symmetry) to recover a solution from the semidefinite relaxation even if a rank-two solution is obtained, whereas our method always recovers a solution in $\mathbb{C}_1^n$ from a rank-one solution to (SDP). As a result, our semidefinite relaxation using the complex number representation should be a tighter relaxation of planar PGO than that of [9] using the matrix representation. Moreover, in contrast to the matrix representation in [9], the complex number representation significantly reduces the computational cost, about which a detailed discussion is made in Section VII. Even though the semidefinite relaxation in [11] also uses the complex number representation, our semidefinite relaxation is more simple, which uses $n \times n$ complex matrices and only depends on rotational states $x \in \mathbb{C}_1^n$, whereas [11] uses $2n \times 2n$ complex matrices and depends on both translational and rotational states $\rho \in \mathbb{C}^n$ and $x \in \mathbb{C}_1^n$. Moreover, [11] mainly focuses on the optimality verification of planar PGO, in comparison, we not only work on optimality verification, but also present algorithms to solve planar PGO. As mentioned before, the problem solved in [14] is phase synchronization on $SO(2)$, whereas ours is pose synchronization on $SE(2)$.

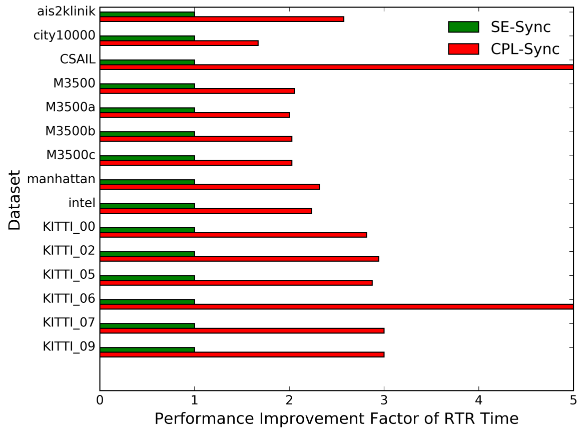
VII. The Results of Experiments

In this section, we implement CPL-Sync on a suite of large 2D SLAM benchmark datasets [9], [11], [24] and make comparisons with the popular methods Powells Dog-Leg method (PDL-GN) [7], [8] and SE-Sync [9]. The chordal initialization [25] is used to generate an initial guess for all the three methods. The C++ code of CPL-Sync is available at https://github.com/fantaosha/CPL-Sync.

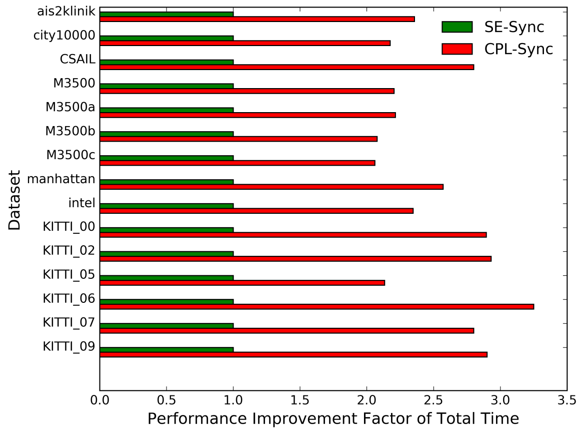
All the experiments have been performed on a laptop with an Intel i7-8750H CPU and 32GB of RAM running Ubuntu 18.04 and using `g++ 7.8` as C++ compiler. We have done the computation on a single core of CPU. For all the datasets, we have used a fixed rank $r_{\text{SE}} = 3$ and $r_{\text{CPL}} = 2$ for SE-Sync and CPL-Sync, respectively, since we find that $r_{\text{SE}} = 3$ and $r_{\text{CPL}} = 2$ are good enough for SE-Sync and CPL-Sync to solve planar PGO given the noises in robotics and computer vision applications.

All the three methods converge to the same globally optimal solution for all the datasets. The computational time is as shown in Table I, in which n is the number of unknown poses, m is the number of noisy measurements, f^* is the globally optimal objective value, the total time accounts for all the time taken to solve PGO, and the RTR time only accounts for the time taken by the RTR method to solve Riemannian staircase optimization. The comparisons of SE-Sync and CPL-Sync are as shown in Fig. 1, in which the performance improvement factor of CPL-Sync over SE-Sync is the computational time of SE-Sync divided by that of CPL-Sync. From Table I and Fig. 1, it can be seen that CLP-Sync is faster than both Powell's Dog-Leg and SE-Sync for all datasets. In addition, CPL-Sync outperforms SE-Sync by a factor of 2.78 on average for the computation of the RTR method, and by a factor 2.51 on average for the overall computation of planar PGO.

The improvements of CPL-Sync over SE-Sync [9] in planar PGO can be explained from three perspectives. First, CPL-Sync is more efficient for the objective and gradient evaluation, e.g., if the rank is $r_{\text{SE}} = 3$ and $r_{\text{CPL}} = 2$, CPL-Sync only needs $\frac{1}{2} \sim \frac{2}{3}$ and $\frac{1}{4} \sim \frac{2}{3}$ operations of SE-Sync to evaluate the objective and gradient, respectively. Second, CPL-Sync is more efficient for the projection or retraction onto the manifold – the projection map of CPL-Sync is just to normalize n vectors, whereas that of SE-Sync has to compute n singular value decompositions, which is much more time

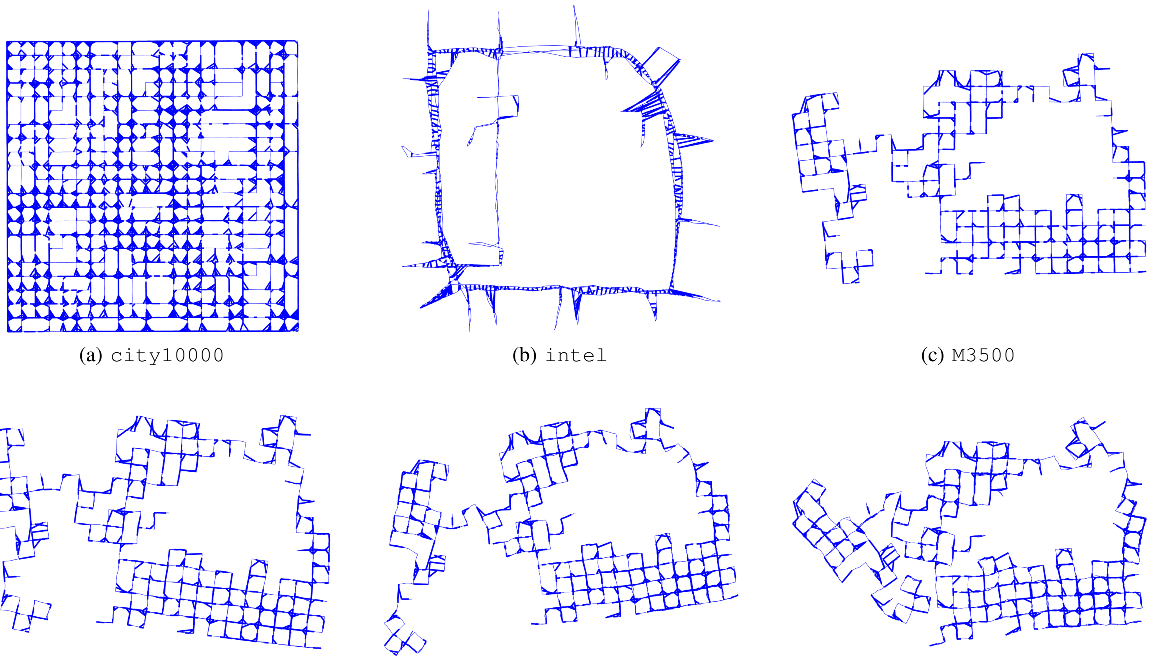


(a)



(b)

Fig. 1: The comparison of SE-Sync and CPL-Sync. The results are (a) performance improvement factor of RTR time of CP-Sync over SE-Sync and (b) performance improvement factor of total time of CPL-Sync over SE-Sync. The performance improvement factor of CPL-Sync over SE-Sync is the computational time of SE-Sync divided by that of CPL-Sync.



(a) `city10000` (b) `intel` (c) `M3500`

(d) `M3500-a` (e) `M3500-b` (f) `M3500-c`

Fig. 2: The globally optimal results of CPL-Sync on some 2D SLAM benchmark datasets. It should be noted that CPL-Sync still obtains global optima on `M3500-a`, `M3500-b` and `M3500-c`, which has large extra noises added to the rotational measurements of `M3500`.

consuming. Third, CPL-Sync is more efficient for chordal initialization and solution rounding. As a result, CPL-Sync should be theoretically more efficient than SE-Sync, which is further confirmed by the results of the experiments.

The globally optimal results of CPL-Sync on some 2D SLAM benchmark datasets are as shown in Fig. 2. It should be noted that `M3500-a`, `M3500-b` and `M3500-c` respectively have extra Gaussian noises with standard deviation 0.1rad, 0.2rad and 0.3rad added to the rotational measurements of `M3500` [11], which indicates that CPL-Sync can tolerate noisy measurements that are orders of magnitude greater than real-world SLAM applications.

VIII. Conclusion

In this paper, we present CPL-Sync for planar PGO using the complex number representation, and prove that CPL-Sync exactly solves planar PGO as long as the noise magnitude is below a certain threshold. The proposed CPL-Sync is compared against Powell's Dog-Leg [7], [8] and SE-Sync [9] on 2D SLAM benchmark datasets, and the results of experiments indicate that CPL-Sync works reliably and is several times faster than the other two methods.

Acknowledgment

This material is based upon work supported by the National Science Foundation under award DCSD-1662233. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.
[2] 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 Transactions on Robotics*, 2016.
[3] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, 2010.
[4] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^{2}o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*.
[5] 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.
[6] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 475–492, 2014.
[7] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.
[8] D. M. Rosen, M. Kaess, and J. J. Leonard, "Rise: An incremental trust-region method for robust online sparse least-squares estimation," *IEEE Transactions on Robotics*, 2014.
[9] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *arXiv preprint arXiv:1612.07386*, 2016.
[10] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
[11] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, 2016.
[12] N. Boumal, V. Voroninski, and A. Bandeira, "The non-convex Burer-Monteiro approach works on smooth semidefinite programs," in *Advances in Neural Information Processing Systems*, 2016.
[13] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.
[14] A. S. Bandeira, N. Boumal, and A. Singer, "Tightness of the maximum likelihood semidefinite relaxation for angular synchronization," *Mathematical Programming*, vol. 163, no. 1-2, pp. 145–167, 2017.
[15] N. Boumal, "Nonconvex phase synchronization," *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2355–2377, 2016.
[16] P.-A. Absil and K. A. Gallivan, "Joint diagonalization on the oblique manifold for independent component analysis," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.
[17] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.
[18] C. Khatri and K. Mardia, "The von Mises-Fisher matrix distribution in orientation statistics," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 95–106, 1977.
[19] J. Gallier, "The schur complement and symmetric positive semidefinite (and definite) matrices," 2010.
[20] T. Fan, H. Wang, M. Rubenstein, and T. Murphey, "CPL-Sync: Efficient and guaranteed planar pose graph optimization using the complex number representation," 2019. [Online]. Available: https://northwestern.box.com/s/eb7w389ajypdx7p60bdjugcx8alg4i7h
[21] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
[22] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on riemannian manifolds," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.
[23] N. Boumal, "A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints," *arXiv preprint arXiv:1506.00575*, 2015.
[24] Y. Latif, C. Cadena, and J. Neira, "Robust graph slam back-ends: A comparative analysis."
[25] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
[26] C. D. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2000, vol. 71.
[27] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, "Complementarity and nondegeneracy in semidefinite programming," *Mathematical programming*, 1997.

Appendix A. The Derivation of (QP)

In this section, we derive (QP) following a similar procedure of [9, Appendix B] even though ours uses the complex number representation.

It is straightforward to rewrite (MLE) as

$$\min_{x\in\mathbb{C}_1^n,\,\rho\in\mathbb{C}^n} \left\| B\begin{bmatrix}\rho\\ x\end{bmatrix}\right\|_2^2 \tag{16}$$

in which

$$B \triangleq \begin{bmatrix} B_1 & B_2 \\ \mathbf{0} & B_3\end{bmatrix} \in \mathbb{C}^{2m\times 2n}.$$

Here $B_1 \in \mathbb{R}^{m\times n}$, $B_2 \in \mathbb{C}^{m\times n}$ and $B_3 \in \mathbb{C}^{m\times n}$ are respectively given by

$$[B_1]_{ek} = \begin{cases} -\sqrt{\tau_{kj}}, & e=(k,\,j)\in\overrightarrow{\mathcal{E}},\\ \sqrt{\tau_{ik}}, & e=(i,\,k)\in\overrightarrow{\mathcal{E}},\\ 0, & \text{otherwise},\end{cases} \tag{17a}$$

$$[B_2]_{ek} = \begin{cases} -\sqrt{\tau_{kj}}\tilde{\rho}_{kj}, & e=(k,\,j)\in\overrightarrow{\mathcal{E}},\\ 0, & \text{otherwise},\end{cases} \tag{17b}$$

and

$$[B_3]_{ek} = \begin{cases} -\sqrt{\kappa_{kj}}\tilde{x}_{kj}, & e=(k,\,j)\in\overrightarrow{\mathcal{E}},\\ \sqrt{\kappa_{ik}}, & e=(i,\,k)\in\overrightarrow{\mathcal{E}},\\ 0, & \text{otherwise}.\end{cases} \tag{17c}$$

Since (P) is also equivalent to (MLE), it can be concluded that

$$B_1^H B_1 = L(W^\rho), \tag{18a}$$
$$B_1^H B_2 = \widetilde{V}, \tag{18b}$$
$$B_2^H B_2 = \widetilde{\Sigma}, \tag{18c}$$
$$B_3^H B_3 = L(\widetilde{G}^x) \tag{18d}$$

in which $L(W^\rho)$, $\widetilde{V}$, $\widetilde{\Sigma}$ and $L(\widetilde{G}^x)$ are as defined in (P). If we let $\widetilde{Q}^\sigma \triangleq \widetilde{\Sigma} - \widetilde{V}^H L(W^\rho)^\dagger \widetilde{V}$, then from Eqs. (18a) to (18d), we obtain

$$\begin{aligned}\widetilde{Q}^\sigma &= B_2^H B_2 - B_2^H B_1 (B_1^H B_1)^\dagger B_1^H B_2 \\ &= B_2^H \left(\mathbf{I} - B_1 (B_1^H B_1)^\dagger B_1^H\right) B_2,\end{aligned} \tag{19}$$

in which B_1, B_2 and B_3 are defined as Eqs. (17a) to (17c). It should be noted that we might rewrite B_1 and B_2 as

$$B_1 = \Omega^{\frac{1}{2}} A^T, \qquad B_2 = \Omega^{\frac{1}{2}} \widetilde{T}, \tag{20}$$

in which $A \triangleq A(\overrightarrow{G})$ and $\widetilde{T}$ are given by Eq. (14) and Eq. (13), respectively. Substituting Eq. (20) into Eq. (19), we obtain

$$\begin{aligned}\widetilde{Q}^\sigma &= B_2^H \left(\mathbf{I} - B_1 (B_1^H B_1)^\dagger B_1^H\right) B_2 \\ &= \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T},\end{aligned} \tag{21}$$

in which $\Pi = \mathbf{I} - \Omega^{\frac{1}{2}} A^T \left(A \Omega A^T\right)^\dagger A \Omega^{\frac{1}{2}} \in \mathbb{R}^{m\times m}$. It should be noted that $X^T (X X^T)^\dagger = X^\dagger$ for any matrix X, then we further obtain

$$\begin{aligned}\Pi &= \mathbf{I} - \Omega^{\frac{1}{2}} A^T \left(A \Omega A^T\right)^\dagger A \Omega^{\frac{1}{2}} \\ &= \mathbf{I} - \left(A \Omega^{\frac{1}{2}}\right)^\dagger A \Omega^{\frac{1}{2}},\end{aligned} \tag{22}$$

which according to [26, Chapter 5.13] is the matrix of orthogonal projection operator $\pi : \mathbb{C}^m \to \ker(A(\overrightarrow{G})\Omega^{\frac{1}{2}})$ onto the kernel space of $A\Omega^{\frac{1}{2}}$. Moreover, it is possible to decompose Π in terms of sparse matrices and their inverse for efficient computation, which is similar to that in [9, Appendix B.4].

Appendix B. Proofs of the Lemmas and Propositions in Section V

In this section, we present proofs of the lemmas and propositions in Section V. These proofs draw heavily on [21] and are similar to that of [9, Appendix C] and [14, Section 4.3].

B.1. Proof of Lemma 1

It is known that the unconstrained Euclidean gradient of $F \triangleq x^H \widetilde{Q} x$ is $\nabla F(x) = 2\widetilde{Q}x$, and thus, if we let $S(x) \triangleq Q - \Re\{\mathrm{ddiag}(Qxx^H)\}$, the Riemannian gradient is

$$\begin{aligned}\mathrm{grad}\, F(x) &= \mathrm{proj}_x(\nabla F(x)) \\ &= 2(Q - \Re\{\mathrm{ddiag}(Qxx^H)\})x \\ &= 2S(x)x.\end{aligned}$$

In addition, the Riemannian Hessian is

$$\mathrm{Hess}\, F(x)[\dot{x}] = \mathrm{proj}_x \mathrm{D}\, \mathrm{grad}\, F(x)[\dot{x}] = \mathrm{proj}_x 2S(x)\dot{x},$$

from which we obtain

$$\langle \mathrm{Hess}\, F(x)[\dot{x}], \dot{x}\rangle = 2\langle S(x)\dot{x}, \dot{x}\rangle.$$

Moreover, according to [21, Chapter 5], if $\exp_x : T_x\mathbb{C}_1^n \to \mathbb{C}_1^n$ is the exponential map at $x \in \mathbb{C}_1^n$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} F \circ \exp_x(t\dot{x})\Big|_{t=0} = \langle \mathrm{grad}\, F(x), \dot{x}\rangle$$

and

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} F \circ \exp_x(t\dot{x})\Big|_{t=0} = \langle \mathrm{Hess}\, F(x)[\dot{x}], \dot{x}\rangle.$$

Therefore, if $\hat{x} \in \mathbb{C}_1^n$ is a local optimum for Eq. (QP) and $\hat{S} = S(\hat{x})$, it is required that $\hat{S}\hat{x} = 0$ and $\langle \dot{x}, \hat{S}\dot{x}\rangle \geq 0$ for all $\dot{x} \in T_x\mathbb{C}_1^n$, which completes the proof.

B.2. Proof of Lemma 2

It should be noted that (1) to (5) in Lemma 2 are KKT conditions of (SDP), which proves the necessity. Since the identity matrix $\mathbf{I} \in \mathbb{C}^{n\times n}$ is strictly feasible to Lemma 2, the Slater's condition is satisfied, which proves the sufficiency. In addition, it should be noted that the Slater's condition also holds for the dual of (SDP). If $\mathrm{rank}(\hat{S}) = n-1$, according to [27, Theorem 6], $\hat{S}$ is dual nondegenerate. Moreover, by complementary slackness, $\hat{S}$ is also optimal for the dual of (SDP), which, as a result of [27, Theorem 10], implies that $\hat{X}$ is unique. Since $\hat{S}\hat{X} = \mathbf{0}$, it can be concluded that $\hat{X}$ has rank one.

B.3. Proof of Lemma 3

Since $\hat{x} \in \mathbb{C}_1^n$ is a first-order critical point and $\hat{S} \succeq 0$, we conclude that $\hat{x}$ is a second-order critical point from Lemma 1. Also it can be checked that $\hat{X} = \hat{x}^H\hat{x} \in \mathbb{H}^n$ satisfies (1) to (5) in Lemma 2, thus, $\hat{x}$ solves (QP), and $\hat{X}$ solves (SDP) and is the unique global optimum for (SDP) if $\mathrm{rank}(\hat{S}) = n-1$.

B.4. Proof of Proposition 2

In order to prove Proposition 2, we need Propositions 5 and 6 as follows.

Proposition 5. If $\underline{Q} \in \mathbb{H}^n$ is data matrix of the form Eq. (15) that is constructed with the true (latent) relative measurements, and $\underline{x} \in \mathbb{C}_1^n$ is the true (latent) value of rotational states x, then $\underline{Q}\underline{x} = \mathbf{0}$ and $\lambda_2(\underline{Q}) > 0$.

Proof. For consistency, we assume that (P) and (QP) are formulated with the true (latent) relative measurements. Let $\underline{\rho} \in \mathbb{C}^n$ be the true (latent) value of translational states ρ, then $\underline{q} = \begin{bmatrix}\underline{\rho}^T & \underline{x}^T\end{bmatrix}^T \in \mathbb{C}^n \times \mathbb{C}_1^n$ solves (P), and the optimal objective value is 0. Since (QP) is equivalent to (P), it can be concluded that $\underline{x} \in \mathbb{C}_1^n$ solves (QP), and the optimal objective value of (QP) is 0 as well. Furthermore, since $\underline{Q} \succeq 0$, we obtain $\underline{Q}\underline{x} = \mathbf{0}$. Let $\Theta \triangleq \mathrm{diag}\{\underline{x}_1, \cdots, \underline{x}_n\} \in \mathbb{C}^{n\times n}$ and $L(W^x) \in \mathbb{R}^{n\times n}$ be the Laplacian such that

$$[L(W^x)]_{ij} \triangleq \begin{cases}\sum\limits_{(i,k)\in\mathcal{E}} \kappa_{ik}, & i = j, \\ -\kappa_{ij}, & (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise},\end{cases}$$

we obtain $L(\underline{G}^x) = \Theta L(W^x)\Theta^H$. It should be noted that G is assumed to be connected, as a result, $\lambda_2(L(\underline{G}^x)) > 0$ and $L(\underline{G}^x)\underline{x} = \mathbf{0}$. In addition, by definition, we have $\underline{Q} = L(\underline{G}^x) + \underline{Q}^\sigma$, in which $\underline{Q}^\sigma = \underline{\Sigma} - \underline{V}^H L(W^\rho)^\dagger \underline{V}$, and from Eqs. (18a) to (18c), it can be concluded that $\underline{Q}^\sigma \succeq 0$. As a result, we obtain $\lambda_2(\underline{Q}) \geq \lambda_2(L(\underline{G}^x)) > 0$. □

Proposition 6. If $\underline{x} \in \mathbb{C}_1^n$ is the true (latent) value of $x \in \mathbb{C}_1^n$, and $\hat{x}$ solves (QP), and $d(\underline{x},\, \hat{x}) \triangleq \min_{\theta\in\mathbb{R}} \|\hat{x} - e^{\mathrm{i}\theta}\underline{x}\|$, then we obtain

$$d(\underline{x},\, \hat{x}) \leq 2\sqrt{\frac{n\|\widetilde{Q} - \underline{Q}\|_2}{\lambda_2(\underline{Q})}} \tag{23}$$

Proof. If we define $\Delta Q \triangleq \widetilde{Q} - \underline{Q} \in \mathbb{H}^n$ to be the perturbation matrix, then

$$\underline{x}^H\widetilde{Q}\underline{x} = \underline{x}^H\underline{Q}\underline{x} + \underline{x}^H\Delta Q\underline{x} = \underline{x}^H\Delta Q\underline{x} \leq n\|\Delta Q\|_2, \tag{24}$$

in which, according to Proposition 5, $\underline{x}^H\underline{Q}\underline{x} = 0$. In addition, it should be noted that

$$\underline{x}^H\widetilde{Q}\underline{x} \geq \hat{x}^H\widetilde{Q}\hat{x} \tag{25}$$

and

$$\hat{x}^H\widetilde{Q}\hat{x} = \hat{x}^H\underline{Q}\hat{x} + \hat{x}^H\Delta Q\hat{x} \geq \hat{x}^H\underline{Q}\hat{x} - n\|\Delta Q\|_2. \tag{26}$$

From Eqs. (24) to (26), we obtain

$$2n\|\Delta Q\|_2 \geq \hat{x}^H\underline{Q}\hat{x} \tag{27}$$

As a result of Proposition 5, we obtain $\underline{Q}\underline{x} = \mathbf{0}$ and $\lambda_2(\underline{Q}) > 0$, and furthermore,

$$\begin{aligned}\hat{x}^H\underline{Q}\hat{x} &\geq (\hat{x} - \frac{1}{n}\underline{x}^H\hat{x}\underline{x})^H\underline{Q}(\hat{x} - \frac{1}{n}\underline{x}^H\hat{x}\underline{x}) \\ &\geq \frac{1}{n}\lambda_2(\underline{Q})(n^2 - |\underline{x}^Hx|^2) \\ &\geq \lambda_2(\underline{Q})(n - |\underline{x}^Hx|)\end{aligned} \tag{28}$$

Substituting Eq. (28) into Eq. (27) and simplifying the resulting equation, we obtain

$$n - |\underline{x}^Hx| \leq \frac{2n\|\Delta Q\|_2}{\lambda_2(\underline{Q})}. \tag{29}$$

In addition, it is straightforward to show $d(\underline{x},\, \hat{x}) = \sqrt{2n - 2|\hat{x}^H\underline{x}|}$, and then from Eq. (29), we complete the proof. $\square$

To prove Proposition 2, we first decompose $\hat{S} = \widetilde{Q} - \Re(\mathrm{ddiag}(\widetilde{Q}\hat{x}^H\hat{x}))$ as follows.

$$\begin{aligned}\hat{S} =& \widetilde{Q} - \Re(\mathrm{ddiag}(\widetilde{Q}\hat{x}^H\hat{x})) \\ =& \underline{Q} + \Delta Q - \\ & \qquad \Re\left\{\mathrm{ddiag}\left((\underline{Q} + \Delta Q)(\underline{x} + \Delta x)(\underline{x} + \Delta x)^H\right)\right\} \\ =& \underline{Q} + \Delta Q - \Re\{\mathrm{ddiag}(\underbrace{\underline{Q}\Delta x x^H + \mathrm{ddiag}(\underline{Q}\Delta x\Delta x^H + \Delta Q(\underline{x} + \Delta x)(\underline{x} + \Delta x)^H)\}}_{\Delta S},\end{aligned}$$

in which $\underline{x} \in \mathbb{C}_1^n$ is the true (latent) value of $x \in \mathbb{C}_1^n$, $\hat{x}$ solves (QP), and $\Delta x \mathrel{=\!\triangleq} \hat{x} - \underline{x}$. In addition, we assume $\|\hat{x} - \underline{x}\| = d(\hat{x},\, \underline{x}) \triangleq \min_{\theta\in\mathbb{R}} \|\hat{x} - e^{\mathrm{i}\theta}\underline{x}\|$. It is obvious that $\|\Delta S\|_2 \to 0$ as long as $\|\Delta Q\|_2 \to 0$ and $\|\Delta x\| \to 0$, and by Proposition 6, $\|\Delta x\| \to 0$ as long as $\|\Delta Q\|_2 \to 0$. As a result, from continuity, there exists some $\gamma > 0$ such that $\|\Delta S\|_2 \leq \lambda_2(\underline{Q})$ as long as $\|\Delta Q\|_2 < \gamma$. Then we obtain

$$\lambda_i(\hat{S}) \geq \lambda_i(\underline{Q}) - \|\Delta S\|_2 > \lambda_i(\underline{Q}) - \lambda_2(\underline{Q}) \geq 0,$$

which implies that $\hat{S}$ has $n-1$ positive eigenvalues. In addition, by Lemma 1, we obtain $\hat{S}\hat{x} = \mathbf{0}$, from which it can be concluded that $\hat{S} \succeq 0$ and $\mathrm{rank}(\hat{S}) = n-1$. Furthermore, Lemma 3 guarantees that $\hat{X} = \hat{x}\hat{x}^H \in \mathbb{H}^n$ is the optimum of (SDP).