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 capable of solving planar PGO certifiably, and is more efficient in numerical computation and more robust to measurement noises than existing state-of-the-art methods. 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]–[10] and verify the optimality of PGO [9]–[12].

In this paper, we consider the problem of planar PGO using the complex number representation. Our work is motivated by the work of Rosen et al. [9] that uses Riemannian staircase optimization to solve PGO efficiently and certifiably. In [9], PGO is formulated on $SE(d)$ 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) [13]. For planar PGO, we need to estimate poses on $SE(2)$ that consists of rotation on $SO(2)$ and translation on $\mathbb{R}^2$. Even though it is popular in robotics to represent $SO(2)$ with a 2×2 real matrix, a unit complex number is sufficient to capture the geometric structure of $SO(2)$. In a similar way, a complex number is equivalent to a 2×1 real vector on $\mathbb{R}^2$. Therefore, it is possible to represent $SE(2)$ using complex numbers, and the matrix representation of $SE(2)$ to formulate planar PGO is redundant in comparison with the complex number representation. Moreover, the redundancy of representation induces extra computation to planar PGO and renders the semidefinite relaxation of planar PGO more sensitive to noises, which affects both the numerical efficiency and theoretical robustness.

In applied mathematics, it is common to use unit complex numbers to represent $SO(2)$, and the complex number representation has been used to formulate phase synchronization problems on $SO(2)$ [14], [15]. In robotics, Carlone et al. [12] use complex numbers to represent $SO(2)$ and $SE(2)$ for the optimality verification of planar PGO, in which the complex number representation makes the analysis much easier and clearer. In contrast to the work of [12], we not only use the complex number representation to verify the optimality of planar PGO, but also to solve planar PGO certifiably, efficiently and robustly.

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 [13] 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 the resulting planar PGO is relaxed to complex oblique manifolds [16]. Most importantly, the complex number representation reduces amounts of computation and results in a much tighter semidefinite relaxation, and as is shown in Section VII, CPL-Sync is not only several times faster but also a lot more robust to measurement noises than SE-Sync on all the tested 2D SLAM datasets.

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 [6], [8], [9] on popular large 2D SLAM benchmark datasets and simulated 2D City datasets with high measurement noises. 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

All the authors are with the McCormick School of Engineering, Northwestern University, Evanston, IL 60201, USA. E-mail: {taosha.fan, hanlinwang}@u.northwestern.edu and {rubenstein, t-murphey}@northwestern.edu.

This material is based upon work supported by the National Science Foundation under award DCSD-1662233.

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. $\mathbb{P}$ denotes the group of $(\mathbb{C}, +) \rtimes (\mathbb{C}_1, \cdot)$ and "$\rtimes$" denotes the semidirect product of groups. $\mathbb{S}^n$ and $\mathbb{H}^n$ denote the sets of $n \times n$ real symmetric matrices and complex Hermitian matrices, respectively. The notation "**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 $\big[\mathrm{diag}(v)\big]_{ii} = v_i$. 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

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

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.,

$$\begin{aligned} b_1' &= a_1 b_1 - a_2 b_2 = b_1 \cos\theta - b_2 \sin\theta, \\ b_2' &= a_1 b_2 + a_2 b_1 = b_2 \cos\theta + b_1 \sin\theta, \end{aligned}$$

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

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

From group theory, $SE(2)$ is also 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, and for $g = (p, R)$, $g' = (p', R') \in SE(2)$, the group multiplication "$\circ$" and group inverse $(\cdot)^{-1}$ of $SE(2)$ using the matrix representation as Eq. (5) are defined to be

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

and

$$g^{-1} = (-R^\top p, R^\top), \tag{7}$$

respectively. Following the complex number representation of $SO(2)$ and $\mathbb{R}^2$ with Eqs. (2) and (4), the representation of $SE(2)$ as Eq. (5) is diffeomorphic and isomorphic to the semidirect product of $(\mathbb{C}, +)$ and $(\mathbb{C}_1, \cdot)$:

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

and from Eqs. (6) and (7), the associated group multiplication "$\odot$" and group inverse $(\cdot)^{-1}$ of $\mathbb{P}$ are defined to be

$$q \odot q' = (x \cdot \rho' + \rho, x \cdot x') \in \mathbb{P} \tag{8}$$

and

$$q^{-1} = (-\overline{x} \cdot \rho, \overline{x}) \in \mathbb{P}, \tag{9}$$

in which $q = (\rho, x)$, $q' = (\rho', x') \in \mathbb{P}$, and $x, x' \in \mathbb{C}_1$ and $\rho, \rho' \in \mathbb{C}$ are the complex number representation of $R, R' \in SO(2)$ and $p, p' \in \mathbb{R}^2$, respectively. In addition, note that the identity of $\mathbb{P}$ is $(0, 1) \in \mathbb{P}$. As a result, instead

of using the matrix representation, we might 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 $q = (\rho, x) \in \mathbb{P}$ is the complex number representation of $g \in SE(2)$, and β and β' are the complex number representation of b and b' as given in Eq. (4), respectively.

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

An immediate benefit of the complex number representation over the matrix representation is that only $\frac{1}{2}$ and $\frac{2}{3}$ of the storage space is needed for $SO(2)$ and $SE(2)$, respectively. Furthermore, as shown in the following sections, the complex number representation greatly simplifies the analysis for planar PGO, and most importantly, the semidefinite relaxation and the Riemannian optimization of planar PGO using the complex number representation is more simple and tighter, and thus requires less computational efforts and has greater robustness than those using the matrix representation in [9].

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 $\tilde{g}_{ij}$ of $g_{ij} \triangleq g_i^{-1} g_j \in SE(2)$. 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, $\cdots$, $q_n \in \mathbb{P}$ with m noisy relative measurements $\tilde{q}_{ij}$ of $q_{ij} \triangleq q_i^{-1} \odot q_j \in \mathbb{P}$, in which $q_{(\cdot)} = (\rho_{(\cdot)}, x_{(\cdot)}) \in \mathbb{P}$, 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 $\tilde{g}_{ij}$ or $\tilde{q}_{ij}$ 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 $\tilde{q}_{ij} = (\tilde{\rho}_{ij}, \tilde{x}_{ij})$ are random variables that satisfy

$$\tilde{\rho}_{ij} = \underline{\rho}_{ij} + \rho_{ij}^{\epsilon} \qquad \rho_{ij}^{\epsilon} \sim N(0, \tau_{ij}^{-1}), \tag{10a}$$

$$\tilde{x}_{ij} = \underline{x}_{ij} x_{ij}^{\epsilon} \qquad \tilde{x}_{ij}^{\epsilon} \sim \mathrm{vMF}(1, \kappa_{ij}), \tag{10b}$$

for all $(i,j) \in \overrightarrow{\mathcal{E}}$. In Eq. (10), $\underline{q}_{ij} = (\underline{\rho}_{ij}, \underline{x}_{ij})$ 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 [17]

$$f(x; x_0, \kappa) = \frac{1}{c_d(\kappa)} \exp\left(-\frac{\kappa}{2}|x - x_0|^2\right),$$

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

If $\tilde{\rho}_{ij}$ and $\tilde{x}_{ij}$ are independent, from Eqs. (3), (8) and (9), 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. (10a) and (10b). From Eqs. (1) and (2), it should be noted that $|x_i \tilde{x}_{ij} - x_j|^2 = \frac{1}{2}\|R_i \widetilde{R}_{ij} - R_j\|_F^2$ and $|\rho_j - \rho_i - x_i \tilde{\rho}_{ij}|^2 = \|p_j - p_i - R_i \tilde{p}_{ij}\|_F^2$, and as a result, (MLE) is equivalent to

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

which is almost the same as the formulation using the matrix representation in [9] except the weight factors.

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 $\widetilde{\Sigma}_i = \sum_{(i,k)\in\overrightarrow{\mathcal{E}}} \tau_{ik}|\tilde{\rho}_{ik}|^2$, 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 $\rho \triangleq \begin{bmatrix} \rho_1 & \cdots & \rho_n \end{bmatrix}^T \in \mathbb{C}^n$:

$$\min_{\rho\in\mathbb{C}^n} \rho^H L(W^\rho)\rho + 2\langle \rho, \widetilde{V}x\rangle + \underbrace{x^H L(\widetilde{G}^x)x + x^H\widetilde{\Sigma}x}_{\text{constant}}. \quad (11)$$

By definition, $L(W^\rho) \succeq 0$, and according to [18, Proposition 4.2] [1], the solution to Eq. (11) is

$$\rho = -L(W^\rho)^\dagger \widetilde{V}x. \quad (12)$$

Substituting Eq. (12) 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

$$\min_{x\in\mathbb{C}_1^n} x^H\widetilde{Q}x, \quad (13)$$

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

Furthermore, if we define $\Omega \triangleq \mathrm{diag}\{\tau_{e_1}, \cdots, \tau_{e_m}\} \in \mathbb{R}^{m\times m}$ 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. (10a), 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

$$\big[\widetilde{T}\big]_{ek} \triangleq \begin{cases} -\tilde{\rho}_{ik}, & e = (i, k) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \quad (14)$$

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

$$\big[A(\overrightarrow{G})\big]_{ke} \triangleq \begin{cases} 1, & e = (i, k) \in \overrightarrow{\mathcal{E}}, \\ -1, & e = (k, j) \in \overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

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

$$\widetilde{Q} = L(\widetilde{G}^x) + \widetilde{T}^H\Omega^{\frac{1}{2}}\Pi\Omega^{\frac{1}{2}}\widetilde{T} \quad (16)$$

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. (13) is equivalent to

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

For the detailed derivation of (QP), interested readers can refer to the full paper [19, Appendix A].

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 [18, Proposition 4.2] was originally derived for real matrices, however, the results can be generalized to complex matrices.

V. The Semidefinite Relaxation

In a similar way to [12], [14], [15], it is straightforward to relax (QP) to

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

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 the conditions for the tight relaxation of (SDP), all the proofs of which can be found in [19].

From [20], 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 [19, Appendix B.1]. □

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 [19, Appendix B.2]. □

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 [19, Appendix B.3]. □

Lemma 3 gives sufficient conditions to verify whether (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 1. Let $\underline{Q} \in \mathbb{H}^n$ be the data matrix of the form Eq. (16) 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 [19, Appendix B.4]. □

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 rank-restricted semidefinite programing [13]:

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

in which $\mathrm{OB}(r,n) \triangleq \{Y \in \mathbb{C}^{n\times r} | \mathrm{diag}(YY^H) = \mathbb{1}\}$ is the complex oblique manifold [16]. Furthermore, (r-SDP) can be a tight relaxation of (SDP) if some conditions are met as stated in Propositions 2 and 3, whose proofs are immediate from [13, Theorem 2].

Proposition 2. If $\hat{Y} \in \mathrm{OB}(r,n)$ 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 3. If $r \geq \lceil \sqrt{n} \rceil$, then for almost all $\widetilde{Q} \in \mathbb{C}^{n\times n}$, every first-order critical $\hat{Y} \in \mathrm{OB}(r,n)$ for (r-SDP) is rank-deficient.

From Propositions 2 and 3, (SDP) is equivalent to successively solving (r-SDP) with the Riemannian trust region (RTR) method [21] 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) [13], [22]. In addition, it is known that the RTR method solves (r-SDP) locally in polynomial time [13, Proposition 3]. In contrast to interior point methods, the Riemannian staircase optimization is empirically orders of magnitude faster in solving large-scale semidefinite programming, and has been successfully implemented in [9], [14] 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 $\hat{x}_i$ to get $\hat{x} \in \mathbb{C}_1^n$. Moreover, note 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.

In particular, 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, which indicates that our formulation using the complex number representation results in semidefinite relaxations of smaller size. Furthermore, in contrast to the matrix representation in [9], the complex number representation significantly reduces the computational cost and, more importantly, results in a much tighter relaxation that is more robust to measurement noises, about which a detailed discussion is made in Section VII.

Algorithm 1 The Riemannian staircase optimization (RSO)

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

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

Input: An optimum $Y^* \in \mathrm{OB}(r,n)$ to (r-SDP)
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
for $i = 1 \to n$ **do**
  $\hat{x}_i = \hat{x}_i / |\hat{x}_i|$
end for
return $\hat{x} \in \mathbb{C}_1^n$

Algorithm 3 The CPL-Sync algorithm

Input: Integers $2 \leq r_0 < r_1 < \cdots < r_k \leq n+1$; an initial iterate $x_0 \in \mathbb{C}_1^n$
Compute an optimum $Y^* \in \mathrm{OB}(r,n)$ with Algorithm 1
Compute rotational states $\hat{x} \in \mathbb{C}_1^n$ with Algorithm 2
Compute translational states $\hat{\rho} \in \mathbb{C}^n$ with Eq. (12)
return $\hat{x} \in \mathbb{C}_1^n$ and $\hat{\rho} \in \mathbb{C}^n$

VII. The Results of Experiments

In this section, we implement CPL-Sync on a suite of large 2D SLAM benchmark datasets [9], [12] and simulated City datasets with high measurement noises. We also compare CPL-Sync with the state-of-the-art methods [6], [8], [9]. The chordal initialization [23] is used for initialization in

TABLE I: Results of the 2D SLAM Benchmark datasets

Dataset	n	m	f^*	PDL-GN [6], [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	3.2×10^0	2.6×10^0	2.7×10^0	1.0×10^0	1.2×10^0
city10000	10000	20687	6.386×10^2	1.8×10^0	8.6×10^{-1}	1.2×10^0	5.2×10^{-1}	5.4×10^{-1}
CSAIL	1045	1172	3.170×10^1	2.6×10^{-2}	5.0×10^{-3}	1.4×10^{-2}	1.0×10^{-3}	5.0×10^{-3}
M3500	3500	5453	1.939×10^2	3.3×10^{-1}	1.5×10^{-1}	2.2×10^{-1}	7.4×10^{-2}	9.8×10^{-2}
M3500-a	3500	5453	1.598×10^3	4.1×10^{-1}	1.6×10^{-1}	2.3×10^{-1}	8.0×10^{-2}	1.0×10^{-1}
M3500-b	3500	5453	3.676×10^3	1.6×10^0	5.3×10^{-1}	5.9×10^{-1}	2.6×10^{-1}	2.8×10^{-1}
M3500-c	3500	5453	4.574×10^3	2.4×10^0	7.5×10^{-1}	8.2×10^{-1}	3.7×10^{-1}	4.0×10^{-1}
manhattan	3500	5453	6.432×10^3	1.7×10^{-1}	4.4×10^{-2}	1.1×10^{-1}	1.9×10^{-2}	4.2×10^{-2}
intel	1728	2512	5.236×10^1	1.3×10^{-1}	3.8×10^{-2}	6.1×10^{-2}	1.7×10^{-2}	2.6×10^{-2}
KITTI_00	4541	4677	1.257×10^2	2.6×10^{-1}	7.6×10^{-2}	1.1×10^{-1}	2.7×10^{-2}	3.8×10^{-2}
KITTI_02	4661	4703	1.084×10^2	2.5×10^{-1}	5.3×10^{-2}	8.5×10^{-2}	1.8×10^{-2}	2.9×10^{-2}
KITTI_05	2761	2826	2.765×10^2	7.4×10^{-2}	2.3×10^{-2}	3.2×10^{-2}	8.0×10^{-3}	1.5×10^{-2}
KITTI_06	1101	1150	3.533×10^1	2.7×10^{-2}	5.0×10^{-3}	1.3×10^{-2}	1.0×10^{-3}	4.0×10^{-3}
KITTI_07	1101	1106	2.393×10^1	2.8×10^{-2}	6.0×10^{-3}	1.4×10^{-2}	2.0×10^{-3}	5.0×10^{-3}
KITTI_09	1591	1592	6.131×10^1	9.5×10^{-2}	1.8×10^{-2}	2.9×10^{-2}	6.0×10^{-3}	1.0×10^{-2}

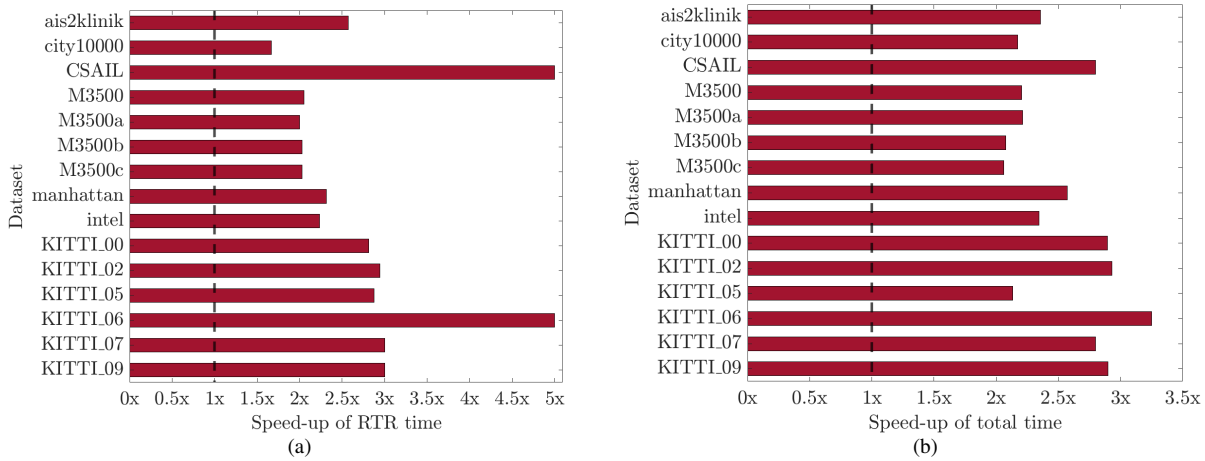


Fig. 1: The speed-up of CPL-Sync over SE-Sync on 2D SLAM benchmark datasets. The results are (a) speed-up of RTR time of CP-Sync over SE-Sync and (b) speed-up of total time of CPL-Sync over SE-Sync. CPL-Sync is on average 2.78 and 2.51 times faster than SE-Sync for RTR time and total time, respectively.

all the experiments. The C++ code of CPL-Sync is available at https://github.com/fantaosha/CPL-Sync.

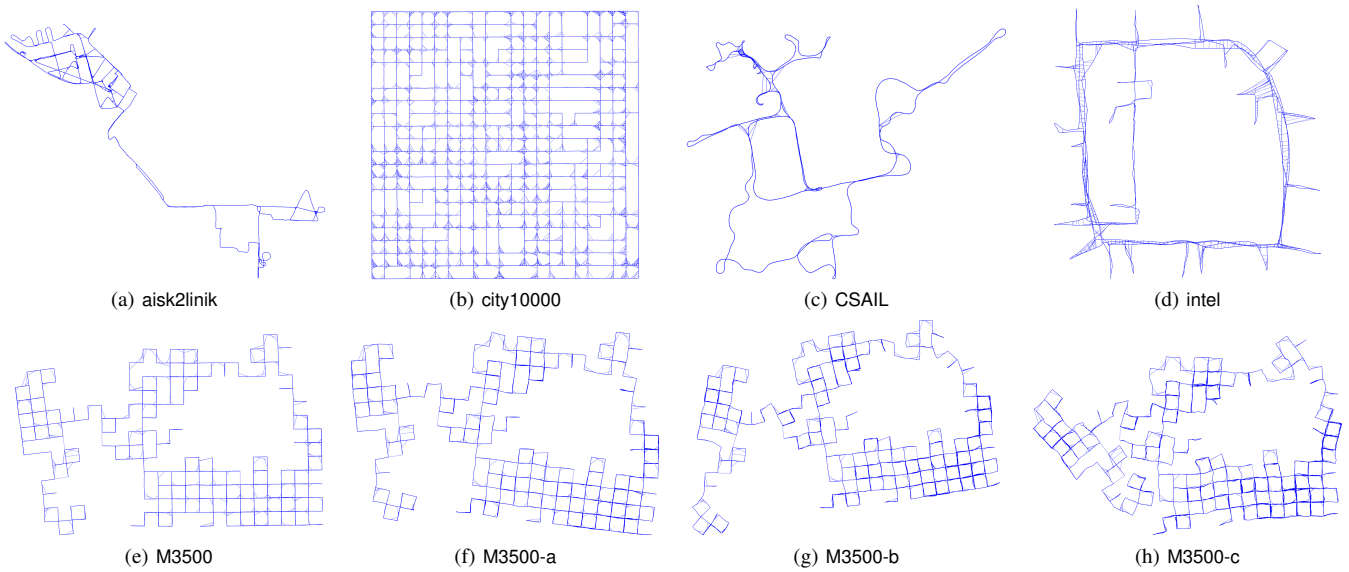
All the experiments have been performed on a laptop with an Intel i7-8750H CPU and 32GB of RAM running Ubuntu 18.04 and using `g++ 7.8` as C++ compiler. We have done the computation on a single core of CPU. For all the datasets, we start with ranks $r_{\mathrm{SE}} = 3$ and $r_{\mathrm{CPL}} = 2$ for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

A. 2D SLAM Benchmark Datasets

In this subsection, we test CPL-Sync on a number of large 2D SLAM benchmark datasets [9], [12] and make comparisons with Powell's Dog-Leg method (PDL-GN) [6], [8] and SE-Sync [9].

All the three methods converge to the globally optimal solution for all the datasets with chordal initialization. The results are in Table I, in which n is the number of unknown poses, m is the number of 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 speed up of CPL-Sync over SE-Sync in terms of RTR and total computational time are also shown in Fig. 1. From Table I and Fig. 1, it can be seen that CLP-Sync is several times faster than both Powell's Dog-Leg and SE-Sync for all datasets. In particular, 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 globally optimal results of CPL-Sync on some 2D SLAM benchmark datasets are in Fig. 2. It should be noted that M3500-a, M3500-b and M3500-c in Fig. 2f to Fig. 2h respectively have extra Gaussian noises with standard deviation 0.1rad, 0.2rad and 0.3rad added to the rotational measurements of M3500 in Fig. 2e [12], which indicates that CPL-Sync can tolerate noisy measurements that are orders



(a) aisk2linik (b) city10000 (c) CSAIL (d) intel

(e) M3500 (f) M3500-a (g) M3500-b (h) 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 in (f)-(h), which has large extra Gaussian noises added to the rotational measurements of M3500 in (e).

of magnitude greater than real-world SLAM applications.

The greater computational efficiency 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 $r_{\mathrm{SE}} = 3$ and $r_{\mathrm{CPL}} = 2$, 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 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.

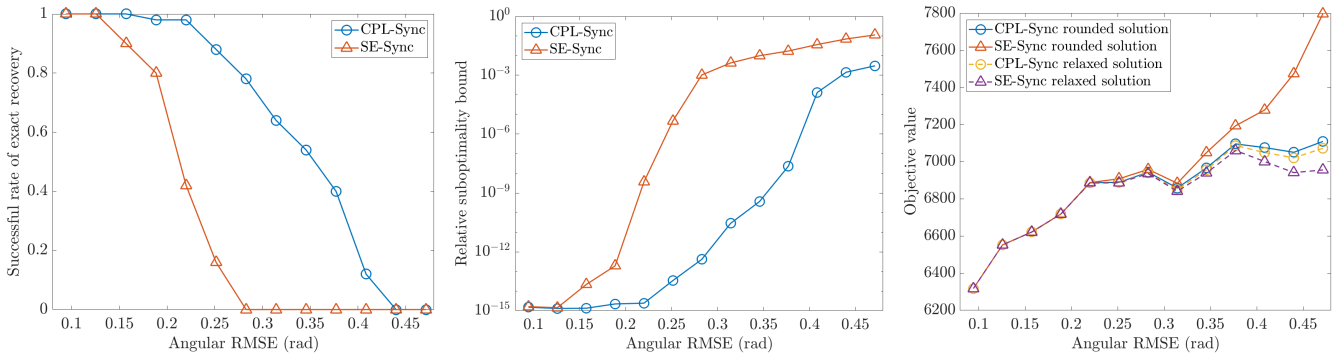
B. City Datasets with High Measurement Noises

In this subsection, we evaluate the tightness of CPL-Sync on a series of simulated City datasets that are similar to city10000 (Fig. 2b) but with high measurement noises. As a basis for comparison, we also evaluate the tightness of SE-Sync using the matrix representation [9]. In general, CPL-Sync and SE-Sync are said to be tight if the globally optimal solution is exactly recovered from the semidefinite relaxation, or equivalently, there is no suboptimality gap between the rounded solution and the relaxed solution.

In our experiments, each City dataset consists of 25×25 square grids with side length of 1 m, a robot trajectory of $n = 3000$ poses along the rectilinear path of the grid, odometric measurements that are available between sequential poses along the robot trajectory, and loop-closure measurements that are available at random between non-sequential poses with a probability $p_C = 0.1$. The odometric and loop-closure measurements are generated from noise models of Eq. (10) with fixed $\tau = 88.89$ that corresponds to an expected translational root-mean-squared error (RMSE) of 0.15 m and varying κ that corresponds to different angular RMSEs.

The results of CPL-Sync and SE-Sync on the simulated City datasets with high measurement noises are in Fig. 3. For each angular RMSE, we calculate the successful rates of exact recovery from the semidefinite relaxation (Fig. 3a), relative suboptimality bounds between rounded and relaxed solutions (Fig. 3b), and objective values of rounded and relaxed solutions (Fig. 3c) statistically from 50 randomly generated City datasets, in which we assume the globally optimal solution is exactly recovered if the relative suboptimality bound is less than 1×10^{-6}. From Fig. 3, it can be seen that when the angular RMSE is small, i.e., approximately less than 0.15 rad, both CPL-Sync and SE-Sync exactly recover the globally optimal solution from the semidefinite relaxation. As angular RMSE increases and is greater than 0.15 rad, CPL-Sync and SE-Sync begin to fail. In spite of this, we find that CPL-Sync has a much higher successful rate of exact recovery from the semidefinite relaxation (Fig. 3a) and orders of magnitude smaller relative suboptimality bounds (Fig. 3b). Furthermore, for the objective value, CPL-Sync has greater lower bound from the relaxed solution but lower upper bound from the rounded solution in scenarios of high measurement noises (Fig. 3c). All of these results indicate that CPL-Sync has a tighter semidefinite relaxation using the complex number representation than SE-Sync using the matrix representation, and thus, is more robust to measurement noises.

We argue that the improved tightness and robustness of CPL-Sync over SE-Sync in planar PGO are associated with the more compact representation of complex numbers over matrices in the semidefinite relaxation. SE-Sync drops the determinant-one constraint of $SO(2)$ in the semidefinite relaxation using the matrix representation, whereas ours using the complex number representation still keeps determinant-



(a) (b) (c)

Fig. 3: The comparisons of CPL-Sync and SE-Sync on City datasets with high measurement noises with $n = 3000$, $p_C = 0.1$, $\tau = 88.89$ corresponding to translational RMSE = 0.15 m and varying κ corresponding to different angular RMSEs. The results are (a) successful rates of exact recovery from the semidefinite relaxation, (b) relative suboptimality bounds between rounded and relaxed solutions, and (c) objective values of rounded and relaxed solutions.

one constraint. Moreover, the semidefinite matrix resulting from the solution to planar PGO using the matrix representation should take the form as $X_R = \left[X_{R_{ij}}\right] \in \mathbb{R}^{2n\times 2n}$ in which each (i,j)-th block $X_{R_{ij}}$ has the algebraic structure $X_{R_{ij}} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \in \mathbb{R}^{2\times 2}$, however, SE-Sync drops such an algebraic structure in the semidefinite relaxation. In comparison, CPL-Sync preserves the algebraic structure of $X_{R_{ij}}$ as complex numbers in the semidefinite relaxation with no extra constraints introduced. Therefore, it can be concluded that the semidefinite relaxation in CPL-Sync using the complex number representation is tighter than that in SE-Sync using the matrix representation, which further suggests that CPL-Sync is more robust to measurement noises than SE-Sync.

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 [6], [8] and SE-Sync [9] on 2D SLAM benchmark datasets and simulated 2D City datasets with high measurement noises. The results of experiments indicate that CPL-Sync is capable of solving planar PGO certifiably, and is more efficient in numerical computation and more robust to measurement noises than existing state-of-the-art methods.

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*, 2014.

[6] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[7] 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*, 2014.

[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] J. G. Mangelson, J. Liu, R. M. Eustice, and R. Vasudevan, "Guaranteed globally optimal planar pose graph and landmark SLAM via sparse-bounded sums-of-squares programming," *arXiv preprint arXiv:1809.07744*, 2018.

[11] 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.

[12] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, 2016.

[13] 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.

[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. Gallivan, "Joint diagonalization on the oblique manifold for independent component analysis," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[17] 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.

[18] J. Gallier, "The Schur complement and symmetric positive semidefinite (and definite) matrices," *Penn Engineering*, 2010.

[19] 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

[20] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[21] P.-A. Absil, C. Baker, and K. Gallivan, "Trust-region methods on Riemannian manifolds," *Foundations of Computational Mathematics*, 2007.

[22] N. Boumal, "A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints," *arXiv preprint arXiv:1506.00575*, 2015.

[23] 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.

[24] C. D. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2000, vol. 71.
[25] 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

$$\min_{x\in\mathbb{C}_1^n,\,\rho\in\mathbb{C}^n} \left\| B \begin{bmatrix} \rho \\ x \end{bmatrix} \right\|_2^2 \tag{17}$$

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

$$[B_1]_{ek} = \begin{cases} -\sqrt{\tau_{kj}}, & e=(k,\,j)\in\overrightarrow{\mathcal{E}}, \\ \sqrt{\tau_{ik}}, & e=(i,\,k)\in\overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \tag{18a}$$

$$[B_2]_{ek} = \begin{cases} -\sqrt{\tau_{kj}}\tilde{\rho}_{kj}, & e=(k,\,j)\in\overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}, \end{cases} \tag{18b}$$

and

$$[B_3]_{ek} = \begin{cases} -\sqrt{\kappa_{kj}}\tilde{x}_{kj}, & e=(k,\,j)\in\overrightarrow{\mathcal{E}}, \\ \sqrt{\kappa_{ik}}, & e=(i,\,k)\in\overrightarrow{\mathcal{E}}, \\ 0, & \text{otherwise}. \end{cases} \tag{18c}$$

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

$$B_1^H B_1 = L(W^\rho), \tag{19a}$$
$$B_1^H B_2 = \widetilde{V}, \tag{19b}$$
$$B_2^H B_2 = \widetilde{\Sigma}, \tag{19c}$$
$$B_3^H B_3 = L(\widetilde{G}^x) \tag{19d}$$

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. (19a) to (19d), we obtain

$$\begin{aligned} \widetilde{Q}^\sigma &= B_2^H B_2 - B_2^H B_1 (B_1^H B_1)^\dagger B_1^H B_2 \\ &= B_2^H \left(\mathbf{I} - B_1 (B_1^H B_1)^\dagger B_1^H\right) B_2, \end{aligned} \tag{20}$$

in which B_1, B_2 and B_3 are defined as Eqs. (18a) to (18c). It should be noted that we might rewrite B_1 and B_2 as

$$B_1 = \Omega^{\frac{1}{2}} A^T, \qquad B_2 = \Omega^{\frac{1}{2}} \widetilde{T}, \tag{21}$$

in which $A \triangleq A(\overrightarrow{G})$ and $\widetilde{T}$ are given by Eq. (15) and Eq. (14), respectively. Substituting Eq. (21) into Eq. (20), we obtain

$$\begin{aligned} \widetilde{Q}^\sigma &= B_2^H \left(\mathbf{I} - B_1 (B_1^H B_1)^\dagger B_1^H\right) B_2 \\ &= \widetilde{T}^H \Omega^{\frac{1}{2}} \Pi \Omega^{\frac{1}{2}} \widetilde{T}, \end{aligned} \tag{22}$$

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 $X^T(XX^T)^\dagger = X^\dagger$ for any matrix X, then we further obtain

$$\begin{aligned} \Pi &= \mathbf{I} - \Omega^{\frac{1}{2}} A^T \left(A\Omega A^T\right)^\dagger A\Omega^{\frac{1}{2}} \\ &= \mathbf{I} - \left(A\Omega^{\frac{1}{2}}\right)^\dagger A\Omega^{\frac{1}{2}}, \end{aligned} \tag{23}$$

which according to [24, 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 [20] 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 [20, 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 [25, 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 [25, 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 $\operatorname{rank}(\hat{S}) = n - 1$.

B.4. Proof of Proposition 1

In order to prove Proposition 1, we need Propositions 4 and 5 as follows.

Proposition 4. If $\underline{Q} \in \mathbb{H}^n$ is data matrix of the form Eq. (16) 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 $\Theta \triangleq \operatorname{diag}\{\underline{x}_1, \cdots, \underline{x}_n\} \in \mathbb{C}^{n\times n}$ 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. (19a) to (19c) and (20), it can be concluded that $\underline{Q}^\sigma$ is the Schur complement of

$$\begin{bmatrix} B_1^H B_1 & B_1^H B_2 \\ B_2^H B_1 & B_2^H B_2 \end{bmatrix} = \begin{bmatrix} B_1^H \\ B_2^H \end{bmatrix} \begin{bmatrix} B_1 & B_2 \end{bmatrix} \succeq 0,$$

which suggests that $\underline{Q}^\sigma \succeq 0$ and $\lambda_1(\underline{Q}^\sigma) \geq 0$. As a result, we obtain

$$\lambda_2(\underline{Q}) \geq \lambda_2(L(\underline{G}^x)) + \lambda_1(\underline{Q}^\sigma) > 0,$$

which completes the proof. □

Proposition 5. 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 $d(\underline{x}, \hat{x}) \triangleq \min\limits_{\theta\in\mathbb{R}} \|\hat{x} - e^{\mathbf{i}\theta}\underline{x}\|$, then we obtain

$$d(\underline{x}, \hat{x}) \leq 2\sqrt{\frac{n\|\widetilde{Q} - \underline{Q}\|_2}{\lambda_2(\underline{Q})}} \tag{24}$$

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

$$\underline{x}^H\widetilde{Q}\underline{x} = \underline{x}^H\underline{Q}\underline{x} + \underline{x}^H\Delta Q\underline{x} = \underline{x}^H\Delta Q\underline{x} \leq n\|\Delta Q\|_2, \tag{25}$$

in which, according to Proposition 4, $\underline{x}^H\underline{Q}\underline{x} = 0$. In addition, it should be noted that

$$\underline{x}^H\widetilde{Q}\underline{x} \geq \hat{x}^H\widetilde{Q}\hat{x} \tag{26}$$

and

$$\hat{x}^H\widetilde{Q}\hat{x} = \hat{x}^H\underline{Q}\hat{x} + \hat{x}^H\Delta Q\hat{x} \geq \hat{x}^H\underline{Q}\hat{x} - n\|\Delta Q\|_2. \tag{27}$$

From Eqs. (25) to (27), we obtain

$$2n\|\Delta Q\|_2 \geq \hat{x}^H\underline{Q}\hat{x} \tag{28}$$

As a result of Proposition 4, we obtain $\underline{Q}\underline{x} = \mathbf{0}$ and $\lambda_2(\underline{Q}) > 0$, and furthermore,

$$\begin{aligned} \hat{x}^H\underline{Q}\hat{x} &\geq (\hat{x} - \frac{1}{n}\underline{x}^H\hat{x}\underline{x})^H\underline{Q}(\hat{x} - \frac{1}{n}\underline{x}^H\hat{x}\underline{x}) \\ &\geq \frac{1}{n}\lambda_2(\underline{Q})(n^2 - |\underline{x}^Hx|^2) \\ &\geq \lambda_2(\underline{Q})(n - |\underline{x}^Hx|) \end{aligned} \tag{29}$$

Substituting Eq. (29) into Eq. (28) and simplifying the resulting equation, we obtain

$$n - |\underline{x}^Hx| \leq \frac{2n\|\Delta Q\|_2}{\lambda_2(\underline{Q})}. \tag{30}$$

In addition, it is straightforward to show $d(\underline{x}, \hat{x}) = \sqrt{2n - 2|\hat{x}^H\underline{x}|}$, and then from Eq. (30), we complete the proof. □

To prove Proposition 1, we first decompose $\hat{S} = \widetilde{Q} - \Re(\operatorname{ddiag}(\widetilde{Q}\hat{x}^H\hat{x}))$ as follows.

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

in which $\underline{x} \in \mathbb{C}_1^n$ is the true (latent) value of $x \in \mathbb{C}_1^n$ such that $\underline{Q}\underline{x} = \mathbf{0}$, $\hat{x}$ solves (QP), and $\Delta x \triangleq \hat{x} - \underline{x}$. In addition, we assume $\|\hat{x} - \underline{x}\| = d(\hat{x}, \underline{x}) \triangleq \min\limits_{\theta\in\mathbb{R}} \|\hat{x} - e^{\mathbf{i}\theta}\underline{x}\|$. 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 5, we obtain $\|\Delta x\| \to 0$ as long as $\|\Delta Q\|_2 \to 0$. As a result, from continuity, there exists some $\gamma > 0$ such that $\|\Delta S\|_2 < \lambda_2(\underline{Q})$ as long as $\|\Delta Q\|_2 < \gamma$. Then we obtain

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

for all $i \geq 2$, which implies that $\hat{S}$ has at least $n-1$ 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 $\operatorname{rank}(\hat{S}) = n - 1$. Furthermore, Lemma 3 guarantees that $\hat{X} = \hat{x}\hat{x}^H \in \mathbb{H}^n$ is the optimum of (SDP) since $\hat{S} \succeq 0$ and $\operatorname{rank}(\hat{S}) = n - 1$.