3-D Inter-robot Relative Localization via Semidefinite Optimization

Ming Li[1] *Student Member, IEEE* and Tin Lun Lam[2] *Senior Member, IEEE* and Zhiyong Sun[1] *Member, IEEE*

***Abstract*—In this paper, the 3-D inter-robot relative localization problem is addressed using noise-corrupted odometric and distance measurements. Unlike the existing solutions, we are devoted to providing a relative localization method that has an "overall best performance", which means that the tradeoffs between the estimation accuracy (EA), the number of measurements (NoMs), and the computation efficiency (CE) are considered. We demonstrate that an existing formulation of the 3-D relative localization problem, the square distances weighted least square (SD-WLS), can be equivalently reformulated as a non-convex quadratic constrained quadratic programming (QCQP) problem. Further, to handle the non-convex nature of the QCQP problem, we adopt the semidefinite programming (SDP) relaxation approach, which drops the rank constraint and recovers the solution of the QCQP via an eigenvalue decomposition strategy. Finally, a refinement step is introduced to solve the problem that the quadratic constraints might not be satisfied due to the SDP relaxation. The simulation and experiment results show that, compared to existing methods, our method has the best overall performance when the three factors, i.e., EA, NoMs, and CE, are important for a relative localization application.**

***Index Terms*—3-D inter-robot relative localization, non-convex optimization, QCQP, SDP.**

I. Introduction

Multi-robot systems (MRSs) have received considerable attention in recent years due to their significant advantages in the aspects of robustness to failure, extensive coverage, and high estimation accuracy through data fusion. A variety of applications, such as search and rescue [1], environment surveillance [2], distributed power grids [3], have exhibited powerful capacity of MRSs. Undoubtedly, for the successful employment of these applications, expressing each robot's pose (translation and orientation) with respect to a common frame of reference is a prerequisite and hence an accurate localization system is required.

Manuscript received February 24, 2022; Revised May 31, 2022 ; Accepted July 2, 2022.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by a starting grant from Eindhoven Artificial Intelligence Systems Institute (EAISI), The Netherlands; in part by the National Natural Science Foundation of China (62073274). *(Corresponding author: Zhiyong Sun.)*

[1]Ming Li and Zhiyong Sun are with the Department of Electrical Engineering, Eindhoven University of Technology, and also with the Eindhoven Artificial Intelligence Systems Institute, PO Box 513, Eindhoven 5600 MB, The Netherlands. { m.li3, z.sun }@tue.nl

[2]Tin Lun Lam is with the School of Science and Engineering, Chinese University of Hong Kong, Shenzhen, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Chinese University of Hong Kong, Shenzhen 518172, China. tllam@cuhk.edu.cn

Digital Object Identifier: see top of this page.

Generally, existing localization systems can be divided into two major categories: absolute and relative localization systems. Global Positioning System (GPS) is a well-known absolute localization system, which determines all robots' locations in a global frame. However, GPS signal interference (e.g. jamming) may happen in some environments like underwater, underground, outer planets, and indoor space, which hinders its applications [4]. Other absolute localization systems, such as indoor motion capture systems and Ultra-WideBand (UWB) anchors with known constellation, are not suitable for an unknown environment because a pre-installation of the sensors is required. Compared to absolute localization, relative localization, which refers to determining relative locations between each pair of robots, is more prevalent for accomplishing tasks in an unknown environment.

Manually measuring the relative pose between robots is simple, but tedious, often with low accuracy, and not applicable to large robot teams. Alternatively, utilizing the absolute localization systems, such as GPS, motion capture system, and UWB anchors, is surely possible for relative localization but may encounter same problems in absolute localization systems. To resolve the disadvantages of the above solutions, many researchers prefer using proprioceptive sensors (e.g., camera, LiDAR, radar, UWB, IMU, wheel encoder, etc.) to achieve relative localization. For example, in [5], a laser-camera scheme is presented to solve the multi-robot map-alignment problem in simultaneous localization and mapping systems. A Visual-Inertial-UWB solution, which addresses the 3-D relative localization problem in an unmanned aerial vehicle collaboration application, is proposed in [6]. In our previous work [7] and some other related work [8], [9], the odometric (IMU and/or wheel encoder) information and UWB ranging measurements are combined to determine the relative pose between robots. Essentially, the above literature addresses the relative localization by combining odometric information and inter-robot measurements like bearing, distance, position, or some combination of them.

In this paper, we are interested in the distance-based solution for solving the 3-D relative localization problem. Unlike the existing solutions, we provide a relative localization method that has an "overall best performance". Specifically, for practical implementation of the relative localization algorithm, the following three factors are of vital importance: i) estimation accuracy (EA); ii) number of measurements (NoMs); and iii) computation efficiency (CE). Understanding the connections between those factors and finding out a tradeoff scheme that has good performances in all aspects are expected. However, there are few methods in the existing literature that address this consideration.

The main contributions of this paper are threefold:

1) We prove that an existing formulation of the 3-D relative localization problem using odometric and distance measurements, i.e., square distances weighted least square (SD-WLS), can be equivalently reformulated as a non-convex quadratically constrained quadratic programming (QCQP) problem.
2) A semidefinite programming (SDP) relaxation approach, which drops the rank constraint and recovers the solution of the QCQP via eigenvalue decomposition, is developed. In addition, to solve the problem that the quadratic constraints might not be satisfied due to the SDP relaxation, a refinement step is introduced, which seeks to find the closest quaternion of the estimate quaternion obtained from the SDP relaxation.
3) Simulation and experiment results (with real data) are both provided to demonstrate that our method has the best performance in terms of an overall consideration of EA, NoMs, and CE.

II. Related Work

Existing methods for 3-D relative localization using odometric information and distance measurements can be categorized into two types: algebraic-based and optimization-based methods. The algebraic-based methods, which rely purely on system geometry and do not consider measurement noises, are studied in [10]. However, as noise is unavoidable in measurements and has a severe impact on EA, optimization-based methods are preferable in practice. A typical example for the optimization-based methods is the WLS formulation and its improvement, i.e., SD-WLS. As reported in [11], to determine a unique solution of the relative pose of two robots, solving the WLS problem is computationally impracticable. With regard to the SD-WLS problem, the authors of [11] tried several approaches, including eigendecomposition, Sum-of-Squares-relaxation, and Lagrange-relaxation. Unsatisfactorily, even for the fastest method, i.e., the Lagrange-relaxation-based method, it will take 3-5s using the toolbox PHCpack [12]. As an alternative, formulating the relative localization problem as a QCQP problem and using SDP relaxation to deal with non-convexity is proven to be a promising solution [13], which only takes around 0.4s using the CVX toolbox in Matlab. This idea is quite similar to our previous paper [7], which solves a 2-D relative localization problem using odometry and UWB.

In a recent work [13], though the authors realized the importance of the EA, CE, and NoMs for a relative localization algorithm in real applications, they did not provide any specific attempts to minimize the NoMs. In particular, since there are 16 unknowns in their formulation, each robot needs to collect 16 measurements to solve a 3-D relative localization problem. Some quadratic constraints on the unknowns are introduced, but they are used to improve the accuracy of the solution. In contrast, we treat these constraints from a new perspective. In particular, inspired by the method in [14], we point out that the NoMs can be reduced by taking advantage of these constraints. In Section V, through simulations and experiments, we demonstrate that the EA of our method can achieve similar performance as the methods of [13] with almost the same CE but fewer NoMs.

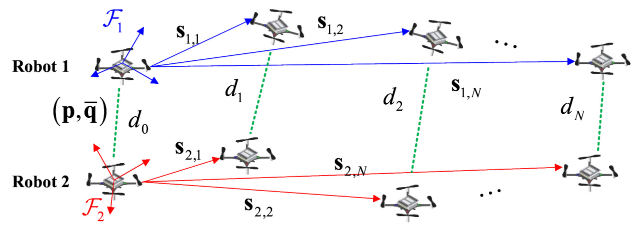

Fig. 1. An illustration of the relative localization: the blue arrow denotes the displacement of Robot 1, the red arrow denotes the displacement of Robot 2, and the green dashed line denotes the distance between Robots 1 and 2. Our goal is to determine $\mathbf{p}$ and $\bar{\mathbf{q}}$ using these displacement and distance measurements.

III. Problem Formulation

Consider two robots $\mathcal{R}_1$ and $\mathcal{R}_2$ moving in a 3-D space. Their initial poses are indicated by the frames of reference $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. As shown in Fig. 1, $\mathcal{R}_1$ and $\mathcal{R}_2$ are equipped with odometric sensors (e.g. IMU), thus the displacement estimates $\mathbf{s}_{1,l} \in \mathbb{R}^3, l = 1, ..., N$ and $\mathbf{s}_{2,l} \in \mathbb{R}^3, l = 1, ..., N$ of the two robots with respect to their own frames can be recorded. Along their trajectories, $N+1$ distance measurements, d_0 and $d_l \in \mathbb{R}, l = 1, ..., N$, are acquired with range sensors, such as UWB. In this paper, our goal is to determine the relative pose between $\mathcal{R}_1$ and $\mathcal{R}_2$ using the odometric and distance measurements. Without loss of generality, we only discuss the case of determining the relative pose of $\mathcal{R}_2$ with respect to $\mathcal{R}_1$.

Let $\mathbf{p} := [x, y, z]^\top$ and the rotation matrix $\mathbf{C}$ denote the relative translation and the relative orientation of frame $\mathcal{F}_1$ with respect to frame $\mathcal{F}_2$, respectively. To avoid the gimbal lock problem [15], we parameterize the rotation matrix $\mathbf{C}$ using a 4×1 unit quaternion rather than Euler angle, which is given as

$$\bar{\mathbf{q}} = [q_1, q_2, q_3, q_4]^\top = [\mathbf{q}, q_4]^\top, \qquad \bar{\mathbf{q}}^\top \bar{\mathbf{q}} = 1,$$

where $\mathbf{q} = [q_1, q_2, q_3]^\top$ and q_4 is the scalar term of a quaternion. The rotation matrix $\mathbf{C}$ can be expressed in terms of quaternion components as [16]:

$$\mathbf{C} = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}. \tag{1}$$

At each time instant l, we assume that the distance measurement d_l is corrupted by Gaussian noise v_l with zero mean and covariance σ_l^2, and produced according to

$$d_l = \sqrt{\mathbf{w}_l^\top \mathbf{w}_l} + v_l, \tag{2}$$

where $\mathbf{w}_l$ represents the relative translation at each time instant l and is defined by

$$\mathbf{w}_l := \mathbf{p} + \mathbf{C}\mathbf{s}_{2,l} - \mathbf{s}_{1,l}. \tag{3}$$

Since distance measurements $d_l, l=1,\cdots,N$ are recorded, we will have N nonlinear equations as of (3). By stacking those equations, it is easy to have

$$\mathbf{d}=\mathbf{h}+\mathbf{v}, \tag{4}$$

where $\mathbf{d}=[d_1,\cdots,d_N]^\top$, $\mathbf{h}=\left[\sqrt{\mathbf{w}_1^\top\mathbf{w}_1},\cdots,\sqrt{\mathbf{w}_N^\top\mathbf{w}_N}\right]^\top$ and $\mathbf{v}=[v_1,\cdots,v_N]^\top$. Let $\mathbf{v}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})$, then we know that $\boldsymbol{\Sigma}=\operatorname{diag}\left(\sigma_1^2,\cdots,\sigma_N^2\right)$, where $\operatorname{diag}(\cdot)$ denotes a diagonal matrix with the elements of the corresponding vector on the main diagonal.

Next, we will briefly introduce the SD-WLS method presented in [11]. Firstly, taking the square of both sides of (2), we have

$$d_l^2=\mathbf{w}_l^\top\mathbf{w}_l+2v_l\sqrt{\mathbf{w}_l^\top\mathbf{w}_l}+v_l^2. \tag{5}$$

Though the noise term $\hat{v}_l:=2v_l\sqrt{\mathbf{w}_l^\top\mathbf{w}_l}+v_l^2$ is not zero-mean Gaussian, the non-Gaussian random variable can be well approximated by a Gaussian density function with matching first and second order moments [11]. We assume that $\hat{v}_l\sim\mathcal{N}(\bar{v}_l,\bar{\sigma}_l^2)$, and thus (5) can be rewritten as:

$$\underline{d}_l=\mathbf{w}_l^\top\mathbf{w}_l+\underline{v}_l, \tag{6}$$

where $\bar{v}_l=\mathbf{E}[\hat{v}_l]$ and $\mathbf{E}[\cdot]$ denoting the expectation of a specific matrix, and we assume $\bar{v}_l$ is a constant and known. The variables $\underline{d}_l=d_l^2-\bar{v}_l$ and $\underline{v}_l=\hat{v}_l-\bar{v}_l$ follow a zero-mean Gaussian distribution, i.e., $\underline{v}_l\sim\mathcal{N}(0,\underline{\sigma}_l^2)$, where $\underline{\sigma}_l=\bar{\sigma}_l$ can be obtained by $\underline{\sigma}_l=\mathbf{E}[\underline{v}_l^2]$. Again, we stack those equations, which gives

$$\underline{\mathbf{d}}=\mathbf{h}_{\mathrm{sd}}+\underline{\mathbf{v}}, \tag{7}$$

where $\underline{\mathbf{d}}=[\underline{d}_1,\cdots,\underline{d}_N]^\top$, $\mathbf{h}_{\mathrm{sd}}=\left[\mathbf{w}_1^\top\mathbf{w}_1,\cdots,\mathbf{w}_N^\top\mathbf{w}_N\right]^\top$ and $\underline{\mathbf{v}}=[\underline{v}_1,\cdots,\underline{v}_N]^\top\sim\mathcal{N}(\mathbf{0},\underline{\boldsymbol{\Sigma}})$, $\underline{\boldsymbol{\Sigma}}=\operatorname{diag}(\underline{\sigma}_1^2,\cdots,\underline{\sigma}_N^2)$, and $\underline{\sigma}_1=\sigma_l(4\mathbf{w}_l^\top\mathbf{w}_l+2\sigma_l)$. Consequently, based on (7), the **3-D SD-WLS optimization problem** can be formulated as:

$$\begin{aligned}\min_{\mathbf{p},\bar{\mathbf{q}}}\quad & \frac{1}{2}(\underline{\mathbf{d}}-\mathbf{h}_{\mathrm{sd}})^\top\underline{\boldsymbol{\Sigma}}^{-1}(\underline{\mathbf{d}}-\mathbf{h}_{\mathrm{sd}}),\\ \text{s.t.}\quad & \bar{\mathbf{q}}^\top\bar{\mathbf{q}}=1.\end{aligned} \tag{8}$$

However, due to that the 3-D SD-WLS optimization problem is highly nonlinear, existing approaches are not computationally efficient [11]. In our previous work [7], the 2-D SD-WLS optimization problem is reformulated as a non-convex QCQP problem and addressed using an SDP strategy. In the next section, we propose to demonstrate that the 3-D SD-WLS can be addressed using a similar method.

IV. Semidefinite-Optimization-Based Method for 3-D Relative Localization

In this section, we prove that the 3-D SD-WLS optimization problem can be reformulated as an equivalent non-convex QCQP problem. The SDP relaxation technique is used to handle its non-convex nature, and a recovery strategy is provided to obtain the solution of the original 3-D SD-WLS optimization problem. Further, to solve the problem that the quadratic constraints might not be satisfied due to the SDP relaxation, a refinement step, which seeks to find the closest quaternion of the estimate quaternion obtained from the SDP relaxation, is introduced [1].

A. Non-convex QCQP

To formulate the non-convex QCQP problem, the following two lemmas are firstly presented.

Lemma 1. *By introducing two auxiliary variables $\mathbf{r}=\mathbf{C}^\top\mathbf{p}$ and $\tilde{\mathbf{q}}=[q_{11},q_{12},q_{13},q_{14},q_{22},q_{23},q_{24},q_{33},q_{34},q_{44}]^\top\in\mathbf{R}^{10\times1}$, where $q_{ij}=q_iq_j, i,j=1,\cdots,4, i\le j$, the cost function of (8) can be reformulated in a quadratic form, i.e.,*

$$\frac{1}{2}\mathbf{x}^\top\mathbf{M}_0\mathbf{x}, \tag{9}$$

where $\mathbf{x}=[\tilde{\mathbf{q}}^\top,\mathbf{r}^\top,\mathbf{p}^\top,1]^\top$, and $\mathbf{M}_0$ is a constant matrix, which is given in Appendix A.

Proof. See Appendix A. □

Remark 1. *The cost function in (8) can be written as a polynomial of the unknown variables $\mathbf{p}$ and $\bar{\mathbf{q}}$. The introduction of the two auxiliary variables $\mathbf{r}$ and $\tilde{\mathbf{q}}$ reduces the order of the polynomial function, i.e., the cost function of the optimization problem (8), from hexagonal to quadratic. In particular, as we will see in (15) in Appendix A, the computation of $\underline{\mathbf{d}}-\mathbf{h}_{\mathrm{sd}}$ produces a term $\mathbf{p}^\top\mathbf{C}$. Since the rotation matrix $\mathbf{C}$ can be written in a quadratic form of $\bar{\mathbf{q}}$ as given in (1), we know that $\underline{\mathbf{d}}-\mathbf{h}_{\mathrm{sd}}$ is cubic in terms of $\mathbf{p}$ and $\bar{\mathbf{q}}$. Then the order of (8) should be hexagonal. However, by introducing the two variables $\mathbf{r}$ and $\tilde{\mathbf{q}}$, the reformulated cost function given in (9) has a quadratic form in terms of $\mathbf{x}$.*

Lemma 2. *The introduction of variable $\mathbf{r}$ and $\tilde{\mathbf{q}}$ creates 6 and 20 different constraints for $\mathbf{x}$, respectively. Together with the constraints $\mathbf{p}^\top\mathbf{p}=d_0^2$ and $\bar{\mathbf{q}}^\top\bar{\mathbf{q}}=1$, the newly defined variable $\mathbf{x}$ in (9) has 28 constraints in total. In addition, all of those constraints can be written in a quadratic form, which is expressed as*

$$\mathbf{x}^\top\mathbf{M}_o\mathbf{x}=0,\quad o=1,...,28 \tag{10}$$

where $\mathbf{M}_o, o=1,...,28$ are all constant matrices, which will be introduced in detail in Appendix B.

Proof. See Appendix B. □

Based on Lemma 1 and Lemma 2, we obtain the following main theorem.

Theorem 1. *Assume that d_0, $\{\mathbf{s}_{1,l},\mathbf{s}_{2,l},d_l\}$, $l=1,\cdots,N$ are available from odometry and range sensors, and $\bar{v}_l$ is the mean value of the squared noise, which is assumed to be a known constant. Then the 3-D SD-WLS, i.e., the optimization problem (8), can be equivalently reformulated as a non-convex QCQP problem:*

$$\begin{aligned}f_{\mathrm{OPT}}^*=\min_{\mathbf{x}}\quad & \frac{1}{2}\mathbf{x}^\top\mathbf{M}_0\mathbf{x}\\ \text{s.t.}\quad & \mathbf{x}^\top\mathbf{M}_o\mathbf{x}=0,\quad o=1,...,28\end{aligned} \tag{11}$$

[1]To validate the correctness of our derivation in this section, Matlab testing code is provided and can be found in: https://github.com/lyric12345678/RAL_Derivations_of_Section_IV.

where f^*_{OPT} denotes the optimal value of (11).

Proof. Substituting the two results of Lemma 1 and Lemma 2, i.e., (9) and (10), back to the 3-D SD-WLS formulation (8), we can obtain (11). □

Generally, there are two major differences of the 3-D QCQP formulation between (11) and the formulation in [13]. Firstly, the unknown variables for the QCQP problem are different. This is because, in [13], the authors assume the elements of the rotation matrix are unknown variables, while we consider $\bar{\mathbf{q}}$ to be unknown in our formulation. In many scenarios, using unit quaternion to compute rotation matrix is more practical. For example, the transformation information of the IMU sensor is encoded by unit quaternion. In addition, compared to [13], the relation $\mathbf{p}^2 = d_0^2$ is used to reduce the number of unknowns in $\mathbf{x}$. Secondly, although both of our formulation and the QCQP formulation in [13] include 16 unknown variables, the number of constraints are different. In our formulation, we have 28 constraints in total, whereas 10 constraints are used in [13].

Remark 2. *Since each entry of the rotation matrix should be within the range of $[-1, 1]$, more constraints on $\mathbf{x}$ can be introduced. In other words, we are able to provide more constraints to* (11) *by introducing the reformulation-linearization-technique constraints [17]. Due to the limited paper length, we will not present more discussions on this topic.*

B. SDP Relaxation

The SD-WLS is reformulated as a typical non-convex QCQPs problem in (11), which is hard to solve since it comprises many NP-hard problems. However, it is well known that this kind of problems can be relaxed to a convex SDP. Here, a crucial first step in deriving an SDP of problem (11) is to observe that

$$\mathbf{x}^\top \mathbf{M}_o \mathbf{x} = \mathbf{Tr}(\mathbf{x}^\top \mathbf{M}_o \mathbf{x}) = \mathbf{Tr}(\mathbf{M}_o \mathbf{X}), o = 1, ..., 28$$

where $\mathbf{Tr}(\cdot)$ denotes the trace of a given matrix, and we define $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$. Then we obtain the following equivalent formulation of problem (11)

$$\begin{aligned} \min_{\mathbf{X}} \quad & \mathbf{Tr}(\mathbf{M}_0 \mathbf{X}) \\ \text{s.t.} \quad & \mathbf{Tr}(\mathbf{M}_o \mathbf{X}) = 0 \quad l = 1, \cdots 28, \\ & \mathbf{X} \succeq 0, \qquad \operatorname{rank}(\mathbf{X}) = 1. \end{aligned}$$

Here, for the symmetric matrix $\mathbf{X}$, $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is positive semidefinite. As we see, we obtain an additional property that the objective and constraints are affine in $\mathbf{X}$, besides the last constraint $\operatorname{rank}(\mathbf{X}) = 1$, which is non-convex. By relaxing the rank constraint, we obtain an SDP relaxation:

$$\begin{aligned} f^*_{\mathrm{SDP}} = \min_{\mathbf{X}} \quad & \mathbf{Tr}(\mathbf{M}_0 \mathbf{X}) \\ \text{s.t.} \quad & \mathbf{Tr}(\mathbf{M}_o \mathbf{X}) = 0 \quad o = 1, \cdots 28, \\ & \mathbf{X} \succeq 0. \end{aligned} \tag{12}$$

where f^*_{SDP} denotes the solution of (12). Globally optimal solution to the above equations (12) can be found by available numerical algorithms in polynomial time (often by interior-point methods), and many solvers can achieve this, such as CVX, SEDUMI and SDPT3.

Lemma 3. *If the measurements, d_0, $\{\mathbf{s}_{1,l}, \mathbf{s}_{2,l}, d_l\}$ for all $l = 1, \cdots, N$ are noise free or corrupted by small noise,* [2]*, the global optimum of the original QCQP can be found by optimizing its SDP relaxation, which means that $f^*_{\mathrm{Opt}} = f^*_{\mathrm{SDP}}$.*

Proof. See Appendix B in [7]. □

C. Recovery of Original Problem

We have introduced the method of computing the solution of QCQP via SDP relaxation above. However, after obtaining the solution $\mathbf{X}^*$ in (12), we still encounter another difficulty: the SDP solution $\mathbf{X}^*$ may not fulfill the condition $\mathbf{rank}(\mathbf{X}^*) = 1$. To address this problem, we may recover $\mathbf{x}^*$ from the low rank decomposition $\mathbf{X}^* = \mathbf{x}^*(\mathbf{x}^*)^\top$ [18]. Specifically, let $R = \operatorname{rank}(\mathbf{X}^*)$ and $\mathbf{X}^*$ is expressed as

$$\mathbf{X}^* = \sum_{r=1}^{R} \mu_r \mathbf{u}_r^* (\mathbf{u}_r^*)^\top$$

where $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_R > 0$ are the eigenvalues and $\mathbf{u}_r, r = 1, \cdots R$ are the respective eigenvectors. We write the rank-one approximation to be

$$\bar{\mathbf{X}}^* = \mu_1 \mathbf{u}_1^* (\mathbf{u}_1^*)^\top$$

To this end, the relative pose can be simply extracted by $\bar{\mathbf{x}}^* = \sqrt{\mu_1}\mathbf{u}_1^*$, where $\bar{\mathbf{x}}^* = [(\tilde{\mathbf{q}}^*)^\top, (\mathbf{r}^*)^\top, (\mathbf{p}^*)^\top, 1]^\top$.

D. Determining Relative Pose

After completing the SDP relaxation and the solution recovery steps, we are capable to obtain $\bar{\mathbf{x}}^*$, and hence the relative pose can be acquired. Apparently, the translation $\mathbf{p}^*$ can be read directly from $\bar{\mathbf{x}}^*$, while the elements of $\bar{\mathbf{q}}^*$ have to be computed from $\tilde{\mathbf{q}}^*$. This can be simply achieved by using the following relationship $q_i^* = \pm\sqrt{q_{ii}^*}$. To determine the sign for q_i^*, we can simply use the convention $q_4^* \geq 0$ and the remaining elemenents of $\bar{\mathbf{q}}^*$ follow $\operatorname{sgn}(q_i) = \operatorname{sgn}(q_{i4})$.

E. Relative Pose Refinement

In practice, due to the SDP relaxation, the constraints given in (10) might not be satisfied anymore. It is not very important to ensure all those constraints remain to be satisfied. For example, for the equation $\mathbf{r}^* = \mathbf{C}^\top \mathbf{p}^*$, $\mathbf{r}^*$ is an auxiliary variable, hence it does not cause problem for relative pose estimation. The key constraint is the quaternion constraint $(\bar{\mathbf{q}}^*)^\top \bar{\mathbf{q}}^* = 1$, i.e., the last constraint in (10). To guarantee that the estimate $\bar{\mathbf{q}}^*$ remains to be a unit quaternion, a further optimization refinement strategy is provided. In particular, we seek to find the closet quaternion $\bar{\mathbf{q}}^*_{\mathrm{CL}}$ to $\bar{\mathbf{q}}^*$, which is formulated as

$$\begin{aligned} \bar{\mathbf{q}}^*_{\mathrm{OPT}} = \operatorname*{argmin}_{\bar{\mathbf{q}}^*_{\mathrm{CL}}} \quad & \frac{1}{2}\|\bar{\mathbf{q}}^*_{\mathrm{CL}} - \bar{\mathbf{q}}^*\|^2 \\ \text{s.t.} \quad & (\bar{\mathbf{q}}^*_{\mathrm{CL}})^\top \bar{\mathbf{q}}^*_{\mathrm{CL}} = 1. \end{aligned} \tag{13}$$

[2]The term "small noise" is defined by an evaluation of the signal-to-noise (SNR) ratio on the measurements: we define the measurement noise to be "small" if $\mathrm{SNR} \geq 20\mathrm{dB}$ otherwise it is "large".

This is again a QCQP problem, which can be solved using a gradient descent method [19].

Remark 3. *In practice, one can introduce the WLS refinement step given in [13] to further improve EA, where the relative pose estimates obtained by our method can be used as an initial guess. Although the WLS step can lead to an improvement of the EA, we did not implement this step in this paper because we propose to eliminate the influence of WLS for the comparison of the three methods to be considered in Section V. More precisely, for a comparison of the three methods, the overall performance (i.e., EA, NoMs and CE) gaps can be reduced if the WLS refinement step is included.*

A summary of our SDP relaxation algorithm is given in Algorithm 1. Note that once the relative pose is obtained, additional distance measurements can be processed with an extended Kalman filter or particle filter [20].

Algorithm 1: The SDP Relaxation Algorithm.

Input: Measurements $\{\mathbf{s}_{1,l}, \mathbf{s}_{2,l}, d_l, d_0\}$. Mean value and covariance of the squared noise, i.e., $\bar{v}_l$ and $\bar{\sigma}_l$ for $l = 1, \cdots N$.

Output: $\mathbf{p}^*$ and $\bar{\mathbf{q}}$.

1: **Construct** $\mathbf{M}_0$ and $\mathbf{M}_o, o = 1, \cdots 28$ and formulate the QCQP problem (11). $\mathbf{M}_0$ can be constructed via $\mathbf{A}^\top \underline{\mathbf{\Sigma}}^{-1} \mathbf{A}$, where $\mathbf{A}$ is given in Appendix A. In addition, $\mathbf{M}_o, o = 1, \cdots 28$ can be obtained by utilizing the strategies given in Appendix B.

2: **Obtain** $\mathbf{X}^*$ by solving the SDP problem (12) via CVX or other optimization tools.

3: **Recover** the solution of the original problem $\mathbf{x}^*$ by using $\bar{\mathbf{x}}^* = \sqrt{\mu_1}\mathbf{u}_1^*$.

4: **Determine** $\mathbf{p}^*$ and $\bar{\mathbf{q}}^*$ and provide its refinement using the strategies given in Section IV. E.

V. Simulation and Experiment Results

In this section, extensive simulation and experiment (with real data) results are exhibited to demonstrate the conclusion that our method has the best overall performance when the three factors, i.e., EA, NoMs, and CE, are all important. We include the algebraic-based method in [10] and the QCQP-SDP method in [13], and our method for comparison.

A. Implementation Details

In order to evaluate the performance of our localization algorithm and provide a fair enough comparison with the algebraic-based and QCQP-SDP methods, the following descriptions of the implementation details are provided.

1) The simulated data and real data are both processed on a 64 bit Intel core i7-9750H with a 2.6-GHz processor in Matlab 2019a with a CVX solver for SDP methods.
2) The CE is evaluated by the computation time of each solver for finding the optimal solution, while excludes other data processing steps, such as constructing $\mathbf{M}_0$ and $\mathbf{M}_o, o = 1, \cdots, 28$ in our formulation.
3) We separate the quaternion and translation errors to evaluate the relative localization error of each method by using the following performance matrices,

$$\mathbf{I}_{\bar{\mathbf{q}},e} = \|\bar{\mathbf{q}}^*_{\mathrm{OPT}} - \bar{\mathbf{q}}\|, \mathbf{I}_{\mathbf{p},e} = \|\mathbf{p}^* - \mathbf{p}\|, \tag{14}$$

where $\bar{\mathbf{q}}$ and $\mathbf{p}$ are corresponding to the true values of the relative pose generated in the simulations or reading from the motion capture in the experiments.
4) For the results shown in the simulation part, the trajectories and distance measurements are generated as follows: i) The two drones are moving in a 3-D space with $x \in [0\mathrm{m}, 20\mathrm{m}]$, $y \in [0\mathrm{m}, 20\mathrm{m}]$ and $z \in [0\mathrm{m}, 20\mathrm{m}]$; ii) The two drones start at initial positions $1\mathrm{m}$ apart from each other and record their first distance measurement; iii) each drone moves randomly with the velocity bound $[-0.6\mathrm{m/s}, 0.6\mathrm{m/s}]$; and iv) the drones record a distance measurement at their new positions, and repeat steps iii) and iv). In total 20 measurements are generated for the analysis of the influence of the NoMs (we found that 20 measurements are sufficient to support the comparisons and analysis).
5) In the simulation, the odometry and distance measurements are generated from the noise-free data by adding Gaussian noise. In addition, both odometry and distance noises follow zero-mean Gaussian distributions. The corresponding covariances are $\mathbf{Q} = \sigma_p^2\mathbf{I}$ and $\mathbf{R} = \sigma_l^2\mathbf{I}$, respectively.
6) For the experiment, the real data are collected in the Control Systems Group laboratory of TU/e. The setup of the experiment is shown in Fig. 4, which consists of two drones moving in a $3\mathrm{m} \times 3\mathrm{m} \times 3\mathrm{m}$ region with an approximate speed of $1.5\mathrm{m/s}$. The initial distance of the two drones is $0.795\mathrm{m}$. We use the motion capture system to obtain the relative pose, and regard the data as the true quantity. The docking board is for safe landing of the two drones. Similar to step 4), 20 measurements are recorded (with a unified sampling rate of $10\mathrm{Hz}$, around $2\mathrm{s}$) for comparison.
7) Each drone estimates its position from the IMU measurements with a frequency of $500\mathrm{Hz}$, which has a standard deviation of $\sigma_p = 0.08\mathrm{m}$. To obtain the displacement estimates, i.e., $s_{1,l}$ and $s_{2,l}$, $l = 1, ..., N$, we adopt the IMU preintegration strategy in [21]. It can be found that the IMU data is quite accurate by estimating the change of positions over short periods of time, where the errors from IMU grow to around $0.04\mathrm{m}$ in $2\mathrm{s}$ in our experiment. In addition, the distance measurements are generated by the motion capture system (see Fig. 4) via adding a zero-mean Gaussian noise with $\sigma_l = 0.05\mathrm{m}$, and the sampling rate is $10\mathrm{Hz}$.
8) When applying our method, one could face a similar singularity problem as noted in [13]. For example, in the special scenario that the distance between two drones remains constant at all time, our method will be invalid. This problem can be avoided by requiring that the drones' trajectories contain small oscillations.

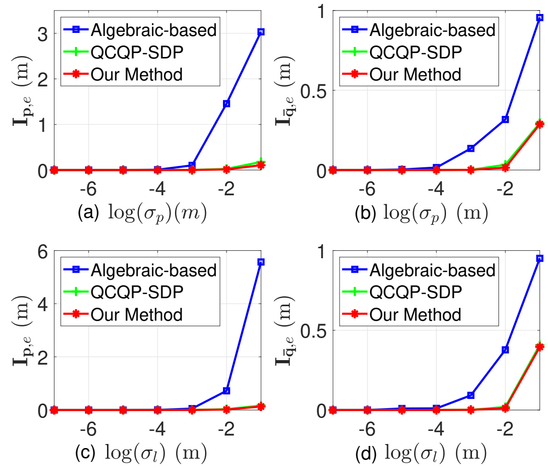

Fig. 2. A comparison of the three methods with different noise level: $\mathbf{I}_{\mathbf{p},e}$ and $\mathbf{I}_{\bar{\mathbf{q}},e}$ denote the estimation error of translation and quaternion, respectively. (a) $\log(\sigma_p)$ versus $\mathbf{I}_{\mathbf{p},e}$. (b) $\log(\sigma_p)$ versus $\mathbf{I}_{\bar{\mathbf{q}},e}$. (c) $\log(\sigma_l)$ versus $\mathbf{I}_{\mathbf{p},e}$. (d) $\log(\sigma_l)$ versus $\mathbf{I}_{\bar{\mathbf{q}},e}$.

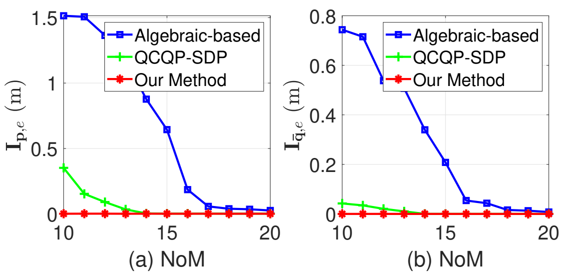

Fig. 3. A comparison of the three methods with different NoMs: $\mathbf{I}_{\mathbf{p},e}$ and $\mathbf{I}_{\bar{\mathbf{q}},e}$ denote the estimation error of translation and quaternion, respectively. (a) NoMs versus $\mathbf{I}_{\mathbf{p},e}$. (b) NoMs versus $\mathbf{I}_{\bar{\mathbf{q}},e}$.

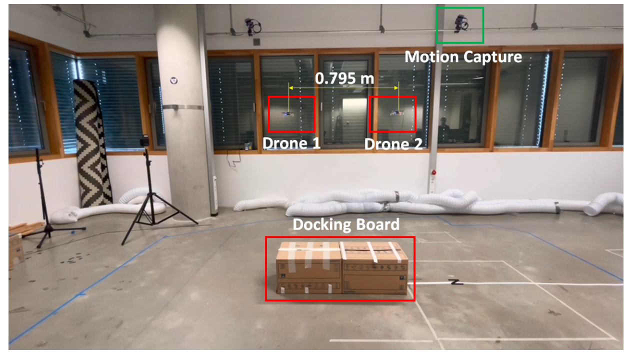

Fig. 4. The setups of the experiment: the initial distance of two drones is 0.795m and the distance measurements are obtained by the motion capture system via adding a zero-mean Gaussian noise with $\sigma_l = 0.05\text{m}$. The docking board is used to ensure safe landing of the two drones.

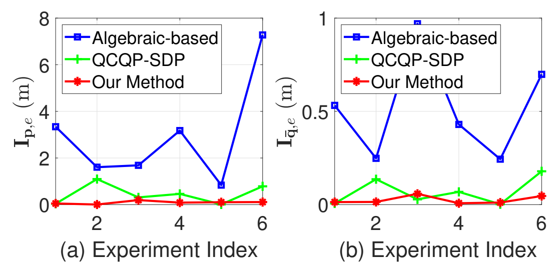

Fig. 5. A comparison of the three methods in terms of EA: $\mathbf{I}_{\mathbf{p},e}$ and $\mathbf{I}_{\bar{\mathbf{q}},e}$ denote the estimation error of translation and quaternion, respectively. (a) Experiment Index versus $\mathbf{I}_{\mathbf{p},e}$. (b) Experiment Index versus $\mathbf{I}_{\bar{\mathbf{q}},e}$.

B. Simulations

In this subsection, we propose to evaluate our method from the aspects of EA, NoMs, and CE, and then draw the conclusion that it has the best overall performance compared to the algebraic-based and QCQP-SDP methods.

1) Estimation Accuracy: In this subsection, we propose to show the influence of measurement noise on different methods. Meantime, it is demonstrated that our method has a much better EA than the algebraic-based method, and it has almost the same EA as the QCQP-SDP method even if fewer measurements are used.

Firstly, we set the NoMs of the algebraic-based, QCQP-SDP, and our method to be 10, 16, and 10, respectively. Then, we change the parameters σ_p and σ_l to show the influence of the measurement noise on the EA. In particular, each factor is analyzed separately, which means that we fix $\sigma_p = 0$ when the influence of σ_l is discussed, and vice versa. As shown in Fig. 2, σ_p and σ_l vary from 10^{-7}m to 10^{-1}m, and a combination of the two parameters σ_p and σ_l with the two estimation errors $\mathbf{I}_{\bar{\mathbf{q}},e}$ and $\mathbf{I}_{\mathbf{p},e}$ is given. We first examine the performance of the different algorithms when the measurement noise is relatively small, i.e., the error is less than 10^{-4}m. As shown in Fig. 2, the curve of the QCQP-SDP and our method is consistently lower than the algebraic-based method, which means that both of the two methods have a higher EA than the algebraic-based method. Because the EA of the algebraic-based method is too small when σ_p or σ_l is greater than 10^{-4}m, we conclude that the algebraic-based method is not suitable for the scenarios that the measurements are corrupted by large noise. In addition, we can find that the estimation errors of $\mathbf{p}$ and $\bar{\mathbf{q}}$ of our method are both close to zero, which supports our conclusion in Lemma 3. Further, we also find that the estimation error can be limited to a quite small level even though the measurement noise deviation is set to be sufficiently large. In particular, for our method, $\mathbf{I}_{\mathbf{p},e} = 0.22\text{m}$ when $\sigma_l = 10^{-1}\text{m}$. Additionally, as shown in Fig. 2, the EA of the QCQP-SDP and our method is almost the same, we conclude that our method has almost the same EA as the QCQP-SDP method even though 6 fewer measurements are used.

2) Number of Measurements: To eliminate the influence of the measurement noise, we set the noise of odometry and distance measurement to be $\sigma_p = 10^{-4}\text{m}$ and $\sigma_l = 10^{-4}\text{m}$. As shown in Fig. 3, when the NoMs is less than 14, our method has much better EA than the algebraic-based and the QCQP-SDP method, which demonstrates that our method is preferable when measurements are hard to be acquired. However, when $(N+1) \geq 14$, we may find that the QCQP-SDP has almost the same EA as our method.

3) Computation Efficiency: In this subsection, the computation time of each method is evaluated. Generally, two

TABLE I
EXPERIMENT INDICES WITH DIFFERENT INITIAL POSES

Index	1	2	3	4	5	6
$\mathbf{p}^\top$	$[0.7, 0.37, 0.07]$	$[0.6, 0.51, 0.05]$	$[0.51, 0.61, 0.02]$	$[0.41, 0.68, 0.03]$	$[0.29, 0.74, 0.01]$	$[0.2, 0.77, 0.01]$
$\bar{\mathbf{q}}^\top$	$[0.99, 0.2, 0.1, 0.1]$	$[-0.2, 0.7, -0.36, 0.6]$	$[0.3, 0.8, 0.43, 0.3]$	$[0.36, -0.7, 0.5, -0.43]$	$[0.76, 0.17, 0.6, 0.24]$	$[0.4, 0.57, 0.4, -0.6]$

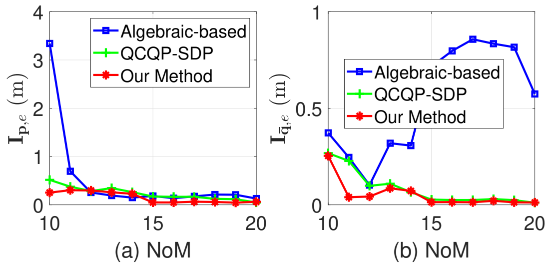

Fig. 6. A comparison of the three methods with different NoMs: $\mathbf{I}_{\mathbf{p},e}$ and $\mathbf{I}_{\bar{\mathbf{q}},e}$ denote the estimation error of translation and quaternion, respectively. (a) NoMs versus $\mathbf{I}_{\mathbf{p},e}$. (b) NoMs versus $\mathbf{I}_{\bar{\mathbf{q}},e}$.

conclusions can be obtained: i) the algebraic-based method is demonstrated to be the fastest method, which uses approximately 0.1s to obtain a unique solution of the relative pose. Our method has almost the same computation speed as the QCQP-SDP method, which takes around 0.4s; and ii) The complexity of our method and the QCQP-SDP method is not influenced by the NoMs. This is because the number of the constraints in our formulation, i.e., equation (11) is not changed with the increase of the NoMs.

In terms of the comparison among the three methods in the three aspects, we conclude that our method is the fastest solution for the scenario where the measurement noise cannot be ignored and only a limited NoMs can be obtained.

C. Experiments

In addition to simulations, we have tested our method with real-world experiments. Also, we evaluate the performance of our method from the three factors: EA, NoMs, and CE. The experiment results strongly support the conclusion given in the simulation part.

1) Estimation Accuracy: We keep the setup the same as the simulation part and conduct 6 experiments to test the three methods with different initial poses, which are listed in TABLE I. The EA is evaluated by (14), and a comparison of the three methods is given in Fig. 5. The algebraic-based method is incapacitated when dealing with real data, while both our method and the QCQP-SDP have a relatively small estimation error on translation and quaternion. our method has a better performance on EA than the QCQP-SDP for practical implementations. The evaluations of CE are given in Fig. 5 and Fig. 6, which are in accord with those conclusions in the simulation part.

VI. CONCLUSION

In this paper, an extension of the 2-D relative localization method in [7] to a 3-D scenario is provided. In particular, the SD-WLS is reformulated as an equivalent non-convex QCQP problem and solved by an SDP relaxation approach. Compared to the SD-WLS, our method has a much better CE, which only requires 0.4s to obtain the relative pose using the CVX toolbox in Matlab. Moreover, compared to the QCQP-SDP method in [13], fewer NoMs are required for our method. Generally, 10 measurements are sufficient to determine the relative pose. The simulation and experiment results demonstrate that, compared to the existing algebraic-based and QCQP-SDP methods, our method has the overall best performance in terms of EA, NoMs, and CE.

APPENDIX A

According to (3) and (6), it is easy to show that

$$\begin{aligned}\underline{d}_l - \mathbf{w}_l^\top \mathbf{w}_l &= \underline{d}_l - (\mathbf{p} + \mathbf{C}\mathbf{s}_{2,l} - \mathbf{s}_{1,l})^\top (\mathbf{p} + \mathbf{C}\mathbf{s}_{2,l} - \mathbf{s}_{1,l}) \\ &= \varepsilon_l + 2(\mathbf{s}_{1,l} - \mathbf{p})^\top \mathbf{C}\mathbf{s}_{2,l} + 2\mathbf{s}_{1,l}^\top \mathbf{p},\end{aligned} \tag{15}$$

where $\varepsilon_l = d_l^2 - \mathbf{p}^2 - \mathbf{s}_{1,l}^\top \mathbf{s}_{1,l} - \mathbf{s}_{2,l}^\top \mathbf{s}_{2,l} - \bar{v}_l$, $\mathbf{p}^2 = d_0^2$. Clearly, ε_l is a constant since $\mathbf{s}_{1,l}$, $\mathbf{s}_{2,l}$ d_0, $d_l, l = 1, \cdots, N$ can be obtained from odometry and range sensors and $\bar{v}_l$ is assumed to be known. We adopt the idea of the algebraic-based method given in [10], where a new variable $\mathbf{r} = \mathbf{C}^\top \mathbf{p}$ is introduced. Then (15) can be simplified to the form of

$$\begin{aligned}\underline{d}_l - \mathbf{w}_l^\top \mathbf{w}_l &= \varepsilon_l + 2\mathbf{s}_{1,l}^\top \mathbf{C}\mathbf{s}_{2,l} - 2\mathbf{r}^\top \mathbf{s}_{2,l} + 2\mathbf{s}_{1,l}^\top \mathbf{p} \\ &= \varepsilon_l + 2\mathbf{e}_l^\top \mathcal{C} + 2[-\mathbf{s}_{2,l}^\top, \mathbf{s}_{1,l}^\top] \begin{bmatrix} \mathbf{r} \\ \mathbf{p} \end{bmatrix},\end{aligned} \tag{16}$$

where $\mathbf{e}_l = \mathrm{vec}(\mathbf{s}_{1,l} \cdot \mathbf{s}_{2,l}^\top)$ and $\mathcal{C} = \mathrm{vec}(\mathbf{C})$, $\mathrm{vec}(\cdot)$ is the "vec"-operator that stacks one column of a matrix underneath the other. In addition, the rotation matrix can be written as

$$\mathcal{C} = \mathbf{G}\tilde{\mathbf{q}}. \tag{17}$$

Since $\mathcal{C}$ is composed of the concatenation of the elements in (1) and $\tilde{\mathbf{q}} = [q_{11}, q_{12}, q_{13}, q_{14}, q_{22}, q_{23}, q_{24}, q_{33}, q_{34}, q_{44}]^\top \in \mathbf{R}^{10\times 1}$, the matrix $\mathbf{G}$ is a constant matrix, which takes a linear combination of the elements in $\tilde{\mathbf{q}}$ and produces $\mathcal{C}$. To write $\mathbf{G}$ in a compact form, the Matlab notation $\mathrm{sparse}(\underline{\mathbf{i}}, \underline{\mathbf{j}}, \underline{\mathbf{s}}, m, n)$ is borrowed, where a $m \times n$ sparse matrix is generated from the *vectors* $\underline{\mathbf{i}}$, $\underline{\mathbf{j}}$ and $\underline{\mathbf{s}}$. Specifically, we define K to be the number of non-zero elements of $\mathbf{G}$, then $\underline{\mathbf{i}} = [\underline{i}_1, \cdots, \underline{i}_K]^\top$ is the *vector* of row index, $\underline{\mathbf{j}} = [\underline{j}_1, \cdots, \underline{j}_K]^\top$ is the *vector* of column index, and $\underline{s}_k \in \underline{\mathbf{s}}, k = 1, \cdots K$ is the value of $\mathbf{G}$ at index $(\underline{i}_k, \underline{j}_k)$. As a result, we express $\mathbf{G}$ in that $\mathbf{G} = \mathrm{sparse}(\underline{\mathbf{i}}_{\mathbf{G}}, \underline{\mathbf{j}}_{\mathbf{G}}, \underline{\mathbf{s}}_{\mathbf{G}}, 9, 10)$, where $\underline{\mathbf{i}}_{\mathbf{G}} = [1, 5, 9, 2, 4, 3, 7, 6, 8, 1, 5, 9, 6, 8, 3, 7, 1, 5, 9, 2, 4, 1, 5, 9]^\top$, $\underline{\mathbf{j}}_{\mathbf{G}} = [1, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 5, 6, 6, 7, 7, 8, 8, 8, 9, 9, 10, 10, 10]^\top$ and $\underline{\mathbf{s}}_{\mathbf{G}} = [1, -1, -1, 2, 2, 2, 2, 2, 2, -2, -1, 1, -1, 2, 2, -2, 2, -1, -1, 1, 2, -2, 1, 1, 1]^\top$.

Substituting (17) back to (16), we have

$$\begin{aligned}\underline{d}_l - \mathbf{w}_l^\top \mathbf{w}_l &= \varepsilon_l + 2\mathbf{e}_l^\top \mathbf{G}\tilde{\mathbf{q}} + 2[-\mathbf{s}_{2,l}^\top, \mathbf{s}_{1,l}^\top] \begin{bmatrix} \mathbf{r} \\ \mathbf{p} \end{bmatrix} \\ &= \mathbf{a}_l^\top \mathbf{x},\end{aligned} \tag{18}$$

where $\mathbf{a}_l = [2\mathbf{e}_l^\top \mathbf{G}, -2\mathbf{s}_{2,l}^\top, 2\mathbf{s}_{1,l}^\top, \varepsilon_l]^\top$, $\mathbf{x} = [\tilde{\mathbf{q}}^\top, \mathbf{r}^\top, \mathbf{p}^\top, 1]^\top$. Afterwards, we stack the equations (18) for $l = 1, ...N$, which gives a compact equation $\underline{\mathbf{d}} - \mathbf{h}_{\mathrm{sd}} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]^\top$. Next, the cost function of (8) can be expressed as

$$\frac{1}{2}\left[(\underline{\mathbf{d}} - \mathbf{h}_{\mathrm{sd}})^\top \underline{\mathbf{\Sigma}}^{-1} (\underline{\mathbf{d}} - \mathbf{h}_{\mathrm{sd}})\right] = \frac{1}{2}\mathbf{x}^\top \mathbf{M}_0 \mathbf{x},$$

where $\mathbf{M}_0 = \mathbf{A}^\top \underline{\mathbf{\Sigma}}^{-1} \mathbf{A}$.

APPENDIX B

With the introduction of the variable $\tilde{\mathbf{q}} = [q_{11}, q_{12}, ..., q_{44}]^\top \in \mathbf{R}^{10\times 1}$, there are 20 constraints formulated as below.

$$\begin{array}{lll} q_{11}q_{44} = q_{14}^2 & q_{14}q_{24} = q_{12}q_{44} & q_{23}q_{34} = q_{33}q_{24} \\ q_{22}q_{44} = q_{24}^2 & q_{24}q_{34} = q_{23}q_{44} & q_{13}q_{34} = q_{33}q_{14} \\ q_{33}q_{44} = q_{34}^2 & q_{34}q_{14} = q_{13}q_{44} & q_{23}q_{24} = q_{22}q_{34} \\ q_{11}q_{22} = q_{12}^2 & q_{12}q_{13} = q_{11}q_{23} & q_{12}q_{24} = q_{22}q_{14} \\ q_{22}q_{33} = q_{23}^2 & q_{12}q_{23} = q_{22}q_{13} & q_{13}q_{14} = q_{11}q_{34} \\ q_{33}q_{11} = q_{13}^2 & q_{13}q_{23} = q_{33}q_{12} & q_{12}q_{14} = q_{11}q_{24} \\ & q_{12}q_{34} = q_{13}q_{24} & q_{12}q_{34} = q_{14}q_{23} \end{array} \tag{19}$$

All constraints in (19) can be written in a quadratic form with respect to $\mathbf{x}$, which is given in (10). Note that $\mathbf{M}_o, o = 1, \cdots 28$ are all constant matrices. For the above 20 constraints, taking $q_{11}q_{44} = q_{14}^2$ as an example, we know that

$$q_{11}q_{44} = q_{14}^2 \Leftrightarrow \mathbf{x}^\top \mathbf{M}_1 \mathbf{x} = 0$$

where $\mathbf{M}_1 = \mathrm{sparse}([1, 4], [10, 4], [1, -1], 17, 17)$. Apparently, the remaining 19 constraints can also be written in quadratic forms by following the same rule. Next, using the fact that $\mathbf{r} = \mathbf{C}^\top \mathbf{p}$ and $\mathbf{p} = \mathbf{C}\mathbf{r}$, we have other 6 constraints. Consider the first constraint

$$r_1 - (\mathbf{G}(1:3,:)\tilde{\mathbf{q}})^\top \mathbf{p} = 0,$$

where $\mathbf{G}(1:3,:)$ denotes the first three rows and all columns of the matrix $\mathbf{G}$, and $\mathbf{r} = [r_1, r_2, r_3]^\top$. Then we know that $\mathbf{M}_{21} = \mathrm{sparse}(\underline{\mathbf{i}}_{21}, \underline{\mathbf{j}}_{21}, \underline{\mathbf{s}}_{21}, 17, 17)$, where $\underline{\mathbf{i}}_{21} = [1, 2, 3, 5, 7, 8, 9, 10, 11]^\top$, $\underline{\mathbf{j}}_{21} = [14, 15, 16, 14, 16, 14, 15, 14, 17]^\top$ and $\underline{\mathbf{s}}_{21} = [1, 2, 2, -1, -2, -1, 2, 1 - 1]^\top$. The constraint $\mathbf{p}^\top \mathbf{p} = d_0^2$ gives $\mathbf{M}_{27} = \mathrm{sparse}(\underline{\mathbf{i}}_{27}, \underline{\mathbf{j}}_{27}, \underline{\mathbf{s}}_{27}, 17, 17)$, where $\underline{\mathbf{i}}_{27} = [14, 15, 16, 17]^\top$, $\underline{\mathbf{j}}_{27} = [14, 15, 16, 17]^\top$ and $\underline{\mathbf{s}}_{27} = [1, 1, 1, -d_0^2]^\top$. For the last constraint, we first write it as $(\bar{\mathbf{q}}^\top \bar{\mathbf{q}} - 1)^2 = (q_{11} + q_{22} + q_{33} + q_{44} - 1)^2 = 0$. Then we know that $\mathbf{M}_{28} = \mathrm{sparse}(\underline{\mathbf{i}}_{28}, \underline{\mathbf{j}}_{28}, \underline{\mathbf{s}}_{28}, 17, 17)$, where $\underline{\mathbf{i}}_{28} = [1, 1, 1, 1, 1, 5, 5, 5, 5, 8, 8, 8, 10, 10, 17]^\top$, $\underline{\mathbf{j}}_{28} = [1, 5, 8, 10, 17, 5, 8, 10, 17, 8, 10, 17, 10, 17, 17]^\top$ and $\underline{\mathbf{s}}_{28} = [1, 2, 2, 2, -2, 1, 2, 2, -2, 1, 2, -2, 1, -2, 1]^\top$.

REFERENCES

[1] Z. F. Yang and S. J. Shen, "Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 39–51, 2016.

[2] A. G. Jain and N. Saunier, "Autocamera calibration for traffic surveillance cameras with wide angle lenses," *arXiv preprint arXiv:2001.07243*, 2020.

[3] Y. Zong, D. Kullmann, A. Thavlov, O. Gehrke, and H. W. Bindner, "Application of model predictive control for active load management in a distributed power system with high wind penetration," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 1055–1062, 2012.

[4] A. Jaimes, S. Kota, and J. Gomez, "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles UAV (s)," in *2008 IEEE International Conference on System of Systems Engineering*. IEEE, 2008, pp. 1–6.

[5] X. S. Zhou and S. I. Roumeliotis, "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case," in *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2006, pp. 1785–1792.

[6] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, "Decentralized visual-inertial-UWB fusion for relative state estimation of aerial swarm," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 8776–8782.

[7] M. Li, G. Liang, H. Luo, H. Qian, and T. L. Lam, "Robot-to-robot relative pose estimation based on semidefinite relaxation optimization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4491–4498.

[8] T.-M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Distance-based cooperative relative localization for leader-following control of MAVs," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3641–3648, 2019.

[9] T. H. Nguyen and L. Xie, "Relative transformation estimation based on fusion of odometry and UWB ranging data," *arXiv preprint arXiv:2202.00279*, 2022.

[10] N. Trawny, X. S. Zhou, K. Zhou, and S. I. Roumeliotis, "Interrobot transformations in 3D," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 226–243, 2010.

[11] N. Trawny, "Cooperative localization: On motion-induced initialization and joint state estimation under communication constraints," Ph.D. dissertation, University of Minnesota, 2010.

[12] J. Verschelde, "Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation," *ACM Transactions on Mathematical Software (TOMS)*, vol. 25, no. 2, pp. 251–276, 1999.

[13] B. Jiang, B. D. O. Anderson, and H. Hmam, "3-D relative localization of mobile systems using distance-only measurements via semidefinite optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1903–1916, 2019.

[14] N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "3D relative pose estimation from distance-only measurements," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1071–1078.

[15] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-body attitude control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 30–51, 2011.

[16] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[17] K. M. Anstreicher, "Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming," *Journal of Global Optimization*, vol. 43, no. 2, pp. 471–484, 2009.

[18] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[19] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[20] M. S. Arulampalam, s. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[21] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5303–5310.