Robotica (2011) volume 29, pp. 375–389. © Cambridge University Press 2010
doi:10.1017/S0263574710000196

A closed-form algorithm for the least-squares trilateration problem

Yu Zhou*

Department of Mechanical Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794, USA

(Received in Final Form: April 16, 2010. First published online: May 20, 2010)

SUMMARY
Trilateration is the most adopted external reference-based localization technique for mobile robots, given the correspondence of external references. The nonlinear least-squares trilateration formulation provides an optimal position estimate from a general number (greater than or equal to the dimension of the environment) of reference points and corresponding distance measurements. This paper presents a novel closed-form solution to the nonlinear least-squares trilateration problem. The performance of the proposed algorithm in dealing with erroneous inputs of reference points and distance measurements has been analyzed through representative examples. The proposed trilateration algorithm has low computational complexity, high operational robustness, and reduced systematic error and uncertainty in position estimation. The effectiveness of the proposed algorithm has been further verified through an experimental test.

KEYWORDS: Localization; Trilateration; Mobile robot.

1. Introduction

This paper presents a novel closed-form trilateration algorithm, which facilitates the self-localization of autonomous mobile robots in two-dimensional (2D) and three-dimensional (3D) environments.

1.1. Concept of trilateration

Trilateration refers to positioning an object based on the measured distances between the object and multiple reference points at known positions.[1] (People tend to call it "multilateration" when more than three reference points are used to position the object. However, "multilateration" has been used to name another process of position estimation based on the measured differences in the distances between the object and three or more reference points.[2]) Given the correspondence of external references, trilateration is the most adopted external reference-based localization technique for mobile robots. A number of trilateration systems can be found in literature, such as those in refs. [3–12].

In principle, trilateration locates an object by solving a system of equations in the form

$$(\mathbf{p}_i - \mathbf{p}_0)^T(\mathbf{p}_i - \mathbf{p}_0) = r_i^2, \tag{1}$$

where $\mathbf{p}_0$ denotes the unknown position of the object, $\mathbf{p}_i$ the known position of the ith reference point, and r_i the known distance between $\mathbf{p}_0$ and $\mathbf{p}_i$. Solving a system of Eq. (1) is equivalent to finding the intersection point/points of a set of circles in $\mathbb{R}^2$ or spheres in $\mathbb{R}^3$:

(1) In $\mathbb{R}^2$, Eq. (1) defines a circle centered at the reference point $\mathbf{p}_i$ with a radius of r_i. With two reference points, two circles defined by Eq. (1) in general intersect at two points symmetric to the base line – the straight line connecting the two reference points (Fig. 1). The object is located at one of the two intersection points. In particular, when the object is collinear with the two reference points, the two circles are tangent to each other, and the object is located at the point of tangency. With three or more reference points, three or more circles defined by Eq. (1) intersect at one unique point where the object is located (Fig. 2).
(2) In $\mathbb{R}^3$, Eq. (1) defines a sphere centered at the reference point $\mathbf{p}_i$ with a radius of r_i. With three reference points, three spheres defined by Eq. (1) in general intersect at two points symmetric to the base plane – the plane determined by the three reference points (Fig. 3). The object is located at one of the two intersection points. In particular, when the object is coplanar with the three reference points, the three spheres intersect at one unique point where the object is located. With four or more reference points, four or more spheres defined by Eq. (1) intersect at one unique point where the object is located (Fig. 4).

Whenever obtaining two intersection points, one must decide which of them the object is actually located at. In many cases, one of the two intersection points naturally distinguishes itself as the only reasonable choice based on some simple criteria, e.g., a mobile robot moving in an indoor environment and localizing itself with respect to landmarks on the ceiling must be located under the ceiling instead of above the ceiling (where the ceiling is the base plane).

In reality, positioning error arises due to the inaccuracy in measuring distances and mapping reference points, and is largely affected by the geometrical arrangement of the reference points and the object.[13] As a result, the involved circles or spheres may not intersect at the actual position of the object, or even may not intersect at all. It is thus necessary to determine a "best approximation" of the object position, in particular when no intersection point is available (i.e., no real solution exists for the system of Eq. (1)).

* Corresponding author. E-mail: yuzhou@notes.cc.sunysb.edu

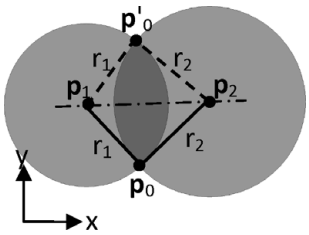

Fig. 1. Two-dimensional trilateration with two reference points.

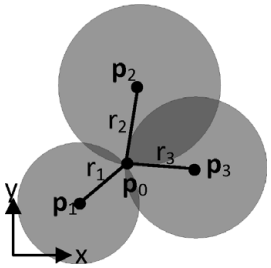

Fig. 2. Two-dimensional trilateration with three reference points.

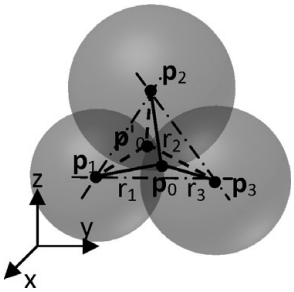

Fig. 3. Three-dimensional trilateration with three reference points.

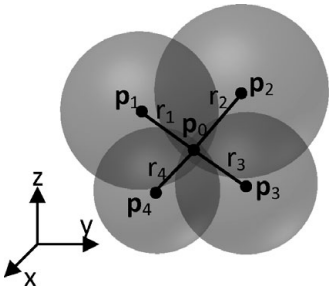

Fig. 4. Three-dimensional trilateration with four reference points.

1.2. Some existing trilateration algorithms

Though straightforward in concept, the trilateration problem is far from trivial to solve due to the nonlinearity of Eq. (1) and the errors in measuring distances and mapping reference points. A number of algorithms have been proposed in order to effectively solve the trilateration problem, including both closed-form and numerical solutions.

Fang[14] provided a closed-form solution for determining the 3D navigation position of a moving object based on the distance measurements from three reference points by referencing the navigation position to the base plane defined by the three reference points. A similar formulation was presented by Ziegert and Mize,[15] applied to measuring the tool position in a machine tool using a linear displacement measuring device – the laser ball bar. In order to use those formulas, however, the positions of reference points must be defined in or transformed into the "base" frame. Moreover, the error in position estimation is correlated to the chosen frame. Independent of the choice of any particular frame of reference, Manolakis[13] derived a more general closed-form solution for estimating the 3D position of an object based on the distance measurements from three reference points. His work shows that the positioning error is affected by the ranging errors, the geometrical arrangement of the object and reference points, and the nonlinearity of the algorithm. A few typos in ref. [13] were fixed by Rao.[16] Recently, Thomas and Ros[17] proposed an alternative closed-form solution for locating an object based on the simultaneous distance measurements from three reference points, using the formulation of Cayley–Menger determinants which are related to the geometry of the tetrahedra formed by the object and three reference points. Closely related to trilateration, Coope[18] presented a closed-form solution for determining the intersection points of n spheres in $\mathbb{R}^n$ based on Gaussian elimination. A more robust, though not closed-form, solution based on the orthogonal decomposition was also presented in ref. [18].

In general, closed-form solutions have low computational complexity when the solution of Eq. (1) exists. They also facilitate the theoretical analysis of the algorithm performance. Closed-form expressions of the relationship between the error in position estimation and that in distance measurement can be found in refs. [13, 17]. However, existing closed-form solutions do not accommodate the situation that the involved circles or spheres do not intersect, i.e., no real solution exists for Eq. (1). Moreover, existing closed-form solutions only solve for the intersection points of n spheres in $\mathbb{R}^n$. They do not apply to determining the intersection point of $N > n$ spheres in $\mathbb{R}^n$, where small errors in distance measuring and reference point mapping can easily cause the involved spheres to fail to intersect at one point. In order to determine the physically existing location of the target object, even if no intersection point exists, it is always necessary to determine an optimal estimate which minimizes the residuals of Eq. (1) in some appropriate form. Numerical methods are in general necessary in order to provide such an estimate, as indicated in refs. [17, 18].

Foy[19] presented a numerical algorithm called Taylor-series estimation which solves the simultaneous set of algebraic position equations by iteratively improving an initial guess with local linear least-sum-squared-error corrections. Incorporating the distance measurement errors, Nadivi *et al.*[20] compared three statistical methods for estimating the 3D position of a point via trilateration, a linear least-squares estimator, an iteratively reweighted least-squares estimator, and a nonlinear least-squares technique, and showed that, in general, the nonlinear least-squares method performs the best. Hu and Tang[21] gave a geometric explanation of the optimal result in least-squares-based trilateration, which is the point of tangency between the hyperellipsoid, determined by the standard deviation of the positioning error, and the intersection of the hypersurfaces, determined by the constraints among the measurements. Coope[18] also suggested a nonlinear least-squares method to obtain the approximate solution, which minimizes the sum of the

difference between the measured and estimated distances. In another work, to locate mobile stations (MS) in a 2D GSM cellular communication network, Pent *et al.*[22] defined a probabilistic model of the MS-BTS (base station) distance measurement error and used the extended Kalman filtering (EKF) to solve the trilateration problem, where the dynamic system was the MS, and the observations were the distance measurements.

In general, numerical methods are available to provide an optimal estimation of the position of a target object, in particular when no real solution exists for Eq. (1). Moreover, numerical methods are in general not limited to dealing with n spheres in $\mathbb{R}^n$. In fact, higher certainty in position estimation is expected as the number of involved reference points increases. However, compared with closed-form solutions, numerical methods in general have higher computational complexity, and closed-form performance analysis is in general not available. Many numerical methods involve a "local optimization" searching process, such as Newton's method and the steepest descent method, which iteratively improves an initial guess toward a converged position estimate.[18–20] However, most of these search algorithms are sensitive to the choice of the initial guess, and in general converge to a local minimum in the vicinity of the initial guess. Though "global optimization" search methods are available, such as the simulated annealing method[23,24] and the genetic algorithm,[23,25] the lack of a general, systematic way to tune the various algorithm-specific parameters and decision criteria cause inconvenience in autonomous applications, and the relatively high time complexity of these methods makes them not suitable for real-time applications.

1.3. Overview of the proposed trilateration algorithm

Built on and extending our recent work[26] (based on "An efficient least-squares trilateration algorithm for mobile robot localization" by Yu Zhou, which appeared in the Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. © 2009 IEEE), we propose a novel closed-form trilateration algorithm which estimates the position of a target object, e.g., a mobile robot, based on the simultaneous distance measurements from multiple reference points, by solving the nonlinear least-squares trilateration problem using standard linear algebra techniques. The proposed trilateration algorithm intends to combine the merits of existing closed-form and numerical solutions. It has the following salient features:

(1) In the closed form and using standard linear algebra techniques, the proposed trilateration algorithm has low computational complexity and facilitates closed-form performance analysis.
(2) Solving the least-squares trilateration problem, the proposed algorithm provides an optimal or at least near optimal estimate of the intersection point of $N \geq n$ spheres constructed from erroneous distance measurements and reference positions in $\mathbb{R}^n$ (where $n =$ 2 or 3), and is not limited to solving for the intersection points of exactly n spheres in $\mathbb{R}^n$.

This paper focuses on the derivation and performance analysis of the proposed trilateration algorithm. Section 2 will derive and explain the proposed trilateration algorithm in detail. Section 3 will perform the error analysis of the proposed trilateration algorithm through representative examples. Section 4 will present an experimental test of the proposed trilateration algorithm. Section 5 will summarize this work.

2. Proposed Trilateration Algorithm

2.1. Nonlinear least-squares formulation

The goal of the proposed trilateration algorithm is to estimate the position of a target object based on the simultaneous distance measurements from multiple reference points at known positions. In order to obtain an optimal position estimate of the target object from potentially inaccurate distance measurements and reference positions, we target our algorithm to solve the nonlinear least-squares trilateration problem. That is, we define an optimal approximation of the object position in $\mathbb{R}^n$ ($n = 2$ or 3 corresponding to a 2D or 3D environment, with the global frame of reference attached to the environment) as

$$\mathbf{p}_0 = \arg\min_{\mathbf{p}_{0,est}} S(\mathbf{p}_{0,est}), \tag{2}$$

where $S(\mathbf{p}_{0,est}) = \sum_{i=1}^{N} [(\mathbf{p}_i - \mathbf{p}_{0,est})^T(\mathbf{p}_i - \mathbf{p}_{0,est}) - r_i^2]^2$, $\mathbf{p}_{0,est}$ denotes an estimate of the object position $\mathbf{p}_0$, $\mathbf{p}_i$ the premapped position of the ith reference point, r_i the measured distance between $\mathbf{p}_0$ and $\mathbf{p}_i$, and N the number of reference points used to determine $\mathbf{p}_0$. Here, we are not constrained to the case of $N = n$. Instead, we are going to give a solution to the general case of $N \geq n$ ($N \in Z^+$).

2.2. Algorithm derivation

Targeting to derive an efficient algebraic solution, we notice that, given $\mathbf{p}_i$ and r_i, solving Eq. (2) for $\mathbf{p}_0$ is equivalent to solving

$$\frac{\partial S(\mathbf{p}_0)}{\partial \mathbf{p}_0} = \mathbf{a} + \mathbf{B}\mathbf{p}_0 + [2\mathbf{p}_0\mathbf{p}_0^T + (\mathbf{p}_0^T\mathbf{p}_0)\mathbf{I}]\mathbf{c} - \mathbf{p}_0\mathbf{p}_0^T\mathbf{p}_0 = \mathbf{0}, \tag{3}$$

where $\mathbf{0}$ denotes the n-dimensional zero vector, $\mathbf{I}$ denotes the $n \times n$ identity matrix, $\mathbf{a} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{p}_i\mathbf{p}_i^T\mathbf{p}_i - r_i^2\mathbf{p}_i)$ (n-dimensional vector), $\mathbf{B} = \frac{1}{N}\sum_{i=1}^{N}[-2\mathbf{p}_i\mathbf{p}_i^T - (\mathbf{p}_i^T\mathbf{p}_i)\mathbf{I} + r_i^2\mathbf{I}]$ ($n \times n$ symmetric matrix), and $\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i$ (n-dimensional vector).

By introducing a linear transform

$$\mathbf{p}_0 = \mathbf{q} + \mathbf{c}, \tag{4}$$

where $\mathbf{q}$ is an n-dimensional vector with the kth element denoted as q_k, we obtain from Eq. (3) an equation containing no quadratic term of $\mathbf{q}$

$$(\mathbf{a} + \mathbf{B}\mathbf{c} + 2\mathbf{c}\mathbf{c}^T\mathbf{c}) + \{\mathbf{B}\mathbf{q} + [2\mathbf{c}\mathbf{c}^T + (\mathbf{c}^T\mathbf{c})\mathbf{I}]\mathbf{q}\} - \mathbf{q}\mathbf{q}^T\mathbf{q} = \mathbf{0}. \tag{5}$$

By defining $\mathbf{f} = \mathbf{a} + \mathbf{B}\mathbf{c} + 2\mathbf{c}\mathbf{c}^T\mathbf{c}$ (n-dimensional vector with the kth element denoted as f_k) and $\mathbf{D} = \mathbf{B} + 2\mathbf{c}\mathbf{c}^T + (\mathbf{c}^T\mathbf{c})\mathbf{I}$

($n \times n$ symmetric matrix), we rewrite Eq. (5) as

$$\mathbf{f} + [\mathbf{D} - (\mathbf{q}^T\mathbf{q})\mathbf{I}]\mathbf{q} = \mathbf{0}. \tag{6}$$

In order to solve Eq. (6) for $\mathbf{q}$ efficiently, we introduce a simplification. First, we have

$$\begin{aligned}\mathbf{D} - (\mathbf{q}^T\mathbf{q})\mathbf{I} &= \mathbf{B} + 2\mathbf{c}\mathbf{c}^T + \left(\mathbf{c}^T\mathbf{c}\right)\mathbf{I} - [(\mathbf{p}_0 - \mathbf{c})^T(\mathbf{p}_0 - \mathbf{c})]\mathbf{I} \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[-2\mathbf{p}_i\mathbf{p}_i^T - \left(\mathbf{p}_i^T\mathbf{p}_i\right)\mathbf{I} + r_i^2\mathbf{I}\right] \\ &\quad + 2\mathbf{c}\mathbf{c}^T + \left(\mathbf{c}^T\mathbf{c}\right)\mathbf{I} - [(\mathbf{p}_0 - \mathbf{c})^T(\mathbf{p}_0 - \mathbf{c})]\mathbf{I}. \end{aligned} \tag{7}$$

Next, we notice that, with perfect $\mathbf{p}_i$ and r_i, we have from Eq. (1)

$$\sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} (\mathbf{p}_i - \mathbf{p}_0)^T(\mathbf{p}_i - \mathbf{p}_0), \tag{8}$$

while, with realistic measurements of $\mathbf{p}_i$ and r_i, we have a good approximation:

$$\sum_{i=1}^{N} r_i^2 \approx \sum_{i=1}^{N} (\mathbf{p}_i - \mathbf{p}_0)^T(\mathbf{p}_i - \mathbf{p}_0). \tag{9}$$

Substituting Eqs. (9) into (7), we obtain

$$\begin{aligned}\mathbf{D} - (\mathbf{q}^T\mathbf{q})\mathbf{I} &\approx \frac{1}{N}\sum_{i=1}^{N}\left\{-2\mathbf{p}_i\mathbf{p}_i^T - \left(\mathbf{p}_i^T\mathbf{p}_i\right)\mathbf{I}\right. \\ &\quad \left. + [(\mathbf{p}_i - \mathbf{p}_0)^T(\mathbf{p}_i - \mathbf{p}_0)]\mathbf{I}\right\} + 2\mathbf{c}\mathbf{c}^T \\ &\quad + \left(\mathbf{c}^T\mathbf{c}\right)\mathbf{I} - [(\mathbf{p}_0 - \mathbf{c})^T(\mathbf{p}_0 - \mathbf{c})]\mathbf{I}, \\ &= -\frac{2}{N}\sum_{i=1}^{N}\mathbf{p}_i\mathbf{p}_i^T + 2\mathbf{c}\mathbf{c}^T, \end{aligned} \tag{10}$$

which is an $n \times n$ symmetric matrix containing no $\mathbf{q}$. Defining $\mathbf{H} = -\frac{2}{N}\sum_{i=1}^{N}\mathbf{p}_i\mathbf{p}_i^T + 2\mathbf{c}\mathbf{c}^T$ and denoting the (k, l) element of $\mathbf{H}$ as $H_{k,l}$, we obtain from Eq. (6)

$$\mathbf{f} + \mathbf{H}\mathbf{q} = \mathbf{0}. \tag{11}$$

Equation (11) is a linear system of n equations of the unknown n-dimensional vector $\mathbf{q}$. If $\mathbf{H}$ has full rank (and is hence invertible), $\mathbf{q}$ can be calculated easily as $\mathbf{q} = -\mathbf{H}^{-1}\mathbf{f}$ or using numerical methods such as Gaussian elimination.[27] However, it may happen that $\mathbf{H}$ does not have full rank. In fact, we have verified by symbolic computation that the $\mathbf{H}$ constructed from an arbitrary set of $N = n$ independent reference points $\mathbf{p}_i$ in $\mathbb{R}^2$ (where $n = 2$) or $\mathbb{R}^3$ (where $n = 3$) has a rank of $n - 1$, though the $\mathbf{H}$ constructed from $N > n$ $\mathbf{p}_i$ in general has a rank of n. Moreover, when all $\mathbf{p}_i$ have the same value of x, y, or z coordinate, $\mathbf{H}$ will have a zero row and a zero column and hence a rank of $n - 1$. In these cases, Eq. (11) does not represent a system of n independent linear equations, and hence we cannot uniquely determine $\mathbf{q}$ from only Eq. (11). Instead, additional constraints need to be found to construct a complete system of n independent equations so that the specific solution can be obtained.

Here we propose a unified solution procedure for $\mathbf{H}$ with a rank of either n (full-rank) or $n - 1$.

First, we construct a degenerated system of $n - 1$ linear equations from Eq. (11) by subtracting the last equation from the other $n - 1$ equations

$$\mathbf{H}'\mathbf{q}' = \mathbf{f}' + \mathbf{h}'q_n, \tag{12}$$

where $\mathbf{q}'$ is an $n - 1$ dimensional vector with the kth element as $q'_k = q_k$, $\mathbf{H}'$ is an $(n-1)\times(n-1)$ matrix with the (k,l) element as $H'_{k,l} = H_{k,l} - H_{n,l}$, $\mathbf{f}'$ is an $n - 1$ dimensional vector with the kth element as $f'_k = -(f_k - f_n)$, and $\mathbf{h}'$ is an $n - 1$ dimensional vector with the kth element as $h'_k = -(H_{k,n} - H_{n,n})$.

Next, from Cramer's rule[28] and the definition of matrix determinant, we obtain

$$q_k = q'_k = \frac{\det\left(\mathbf{H}_k^f\right)}{\det(\mathbf{H}')} + \frac{\det\left(\mathbf{H}_k^h\right)}{\det(\mathbf{H}')}q_n, \quad \forall k \in \{1, \ldots, n-1\}, \tag{13}$$

where "det" means the matrix determinant, $\mathbf{H}_k^f$ is the matrix formed by replacing the kth column of $\mathbf{H}'$ with $\mathbf{f}'$, and $\mathbf{H}_k^h$ is the matrix formed by replacing the kth column of $\mathbf{H}'$ with $\mathbf{h}'$. In particular, because $\mathbf{H}'$, constructed from the $\mathbf{H}$ with a rank of n or $n - 1$, has a full rank of $n - 1$, $\det(\mathbf{H}') \neq 0$.

Equation (13) defines q_k in terms of q_n, which means that $\mathbf{q}$ can be completely determined upon the determination of q_n. To solve for q_n, one more independent equation is needed. In fact, one valid constraint is

$$\sum_{k=1}^{n} q_k^2 = \mathbf{q}^T\mathbf{q}, \tag{14}$$

where $\mathbf{q}^T\mathbf{q}$ can be obtained from $\mathbf{D} - (\mathbf{q}^T\mathbf{q})\mathbf{I} = \mathbf{H}$ as

$$\mathbf{q}^T\mathbf{q} = -\frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i^T\mathbf{p}_i + \frac{1}{N}\sum_{i=1}^{N} r_i^2 + \mathbf{c}^T\mathbf{c}. \tag{15}$$

Substituting Eq. (13) into Eq. (14), we obtain

$$\sum_{k=1}^{n-1}[\det\left(\mathbf{H}_k^f\right) + \det\left(\mathbf{H}_k^h\right)q_n]^2 + \det(\mathbf{H}')^2 q_n^2 = \det(\mathbf{H}')^2\mathbf{q}^T\mathbf{q}, \tag{16}$$

which is a quadratic equation of q_n. Written into the canonical form, Eq. (16) becomes

$$aq_n^2 + bq_n + c = 0, \tag{17}$$

where $a = \sum_{k=1}^{n-1}\det(\mathbf{H}_k^h)^2 + \det(\mathbf{H}')^2$, $b = 2\sum_{k=1}^{n-1}[\det(\mathbf{H}_k^f)\det(\mathbf{H}_k^h)]$, $c = \sum_{k=1}^{n-1}\det(\mathbf{H}_k^f)^2 - \det(\mathbf{H}')^2\mathbf{q}^T\mathbf{q}$.

Equation (17) is well-known solvable in the closed form as

$$q_n = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{18}$$

In particular, since $\det(\mathbf{H}') \neq 0$, $a \neq 0$.

Then, by substituting the resulting q_n into Eq. (13), we obtain $q_k, \forall k \in \{1, \ldots, n-1\}$.

Furthermore, by substituting the resulting $\mathbf{q}$ into Eq. (4), we obtain $\mathbf{p}_0$.

The above process generally results in two candidates of $\mathbf{p}_0$, due to the duality of Eq. (18). However, only one of those two candidates is the valid estimate of $\mathbf{p}_0$. Therefore, a judgment needs to be made to pick the correct one. The judging criterion is usually very simple, such as that $\mathbf{p}_0$ is known on one specific side of the base plane (or base line) defined by the reference points, or that current estimate of $\mathbf{p}_0$ should be close enough to the last one.

2.3. Algorithm summary

Following the above derivation, the proposed trilateration algorithm is summarized as Algorithm 1. Since each step of calculation in Algorithm 1 has its closed-form expression, the proposed algorithm, as whole, indeed provides a closed-form solution to the trilateration problem. This facilitates the closed-form performance analysis in Section 3. Solving the least-squares trilateration problem, the proposed algorithm provides an optimal or at least near-optimal estimation of the position of the target object based on its distances from N reference points, where N can be any integer greater than or equal to 2 in a 2D environment or 3 in a 3D environment. Using standard linear algebra techniques, the proposed algorithm is highly tractable and convenient to implement, and has low computational complexity. Without depending on the techniques which tend to be affected by algebraic singularities, such as matrix inversion, the proposed algorithm has high operational robustness.

Algorithm 1: *Nonlinear Least-Squares Trilateration in* $\mathbb{R}^n$ *(*$n \in \{2,3\}$*)*

Input: A set of N reference points $\{\mathbf{p}_i \mid i \in Z^+, n \leq i \leq N\}$, and the corresponding set of distances between $\mathbf{p}_i$ and the unknown position $\mathbf{p}_0$ — $\{r_i \mid i \in Z^+, n \leq i \leq N\}$.

Output: $\mathbf{p}_0$.

(1) Calculate $\mathbf{a}$, $\mathbf{B}$, $\mathbf{c}$, $\mathbf{H}$, $\mathbf{f}$, $\mathbf{H}'$, $\mathbf{h}'$, $\mathbf{f}'$, $\mathbf{H}_k^h$ and $\mathbf{H}_k^f$.
(2) Calculate $\mathbf{q}^T\mathbf{q}$ from Eq. (15).
(3) Calculate q_n from Eq. (18).
(4) Calculate $q_k, \forall k \in \{1, \ldots, n-1\}$, from Eq. (13).
(5) Calculate $\mathbf{p}_0$ from Eq. (4).
(6) Choose one of the two candidates of $\mathbf{p}_0$.
(7) Return $\mathbf{p}_0$.

3. Performance Analysis

The proposed trilateration algorithm estimates the position of a target object based on the simultaneous distance measurements from multiple reference points at known positions. The input to the algorithm includes the mapped positions of the reference points, $\mathbf{p}_i$, which are defined in a global frame of reference attached to the environment, and the measured distances between the object and reference points, r_i. In practice, errors arise in $\mathbf{p}_i$ due to inaccurate mapping of the reference points, and errors arise in r_i due to imperfect distance measurement of the range sensors. These input errors will cause output errors in the estimation of the object position $\mathbf{p}_0$. This section discusses the influence of these input errors on the accuracy of the estimation output.

3.1. Performance indices

We define an $(n+1)N$-dimensional input vector $\mathbf{x} = [\cdots \mathbf{p}_i^T \cdots r_i \cdots]^T$ which consists of all $\mathbf{p}_i$ and r_i. Thus $\mathbf{p}_0$ is a function of $\mathbf{x}$, i.e., $\mathbf{p}_0 = \mathbf{p}_0(\mathbf{x})$. Denoting the actual value and random error of $\mathbf{x}$ as $\bar{\mathbf{x}}$ and $\delta\mathbf{x}$ respectively, we have $\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}$ for the measurement of $\mathbf{x}$. Correspondingly, the actual value and output estimate of $\mathbf{p}_0$ are defined by $\mathbf{p}_0(\bar{\mathbf{x}})$ and $\mathbf{p}_0(\mathbf{x}) = \mathbf{p}_0(\bar{\mathbf{x}} + \delta\mathbf{x})$ respectively, and the estimation error of $\mathbf{p}_0$ is $\delta\mathbf{p}_0(\mathbf{x}) = \mathbf{p}_0(\bar{\mathbf{x}} + \delta\mathbf{x}) - \mathbf{p}_0(\bar{\mathbf{x}})$.

From the Taylor expansion, we obtain

$$\delta\mathbf{p}_0(\mathbf{x}) = \mathbf{p}_0(\bar{\mathbf{x}} + \delta\mathbf{x}) - \mathbf{p}_0(\bar{\mathbf{x}}) = \sum_{k=1}^{\infty} \frac{1}{k!} \frac{\partial^k \mathbf{p}_0(\bar{\mathbf{x}})}{\partial \mathbf{x}^{T^k}} \delta\mathbf{x}^{[k]}. \tag{19}$$

Here we follow the notation of matrix calculus.[29] The matrix derivative $\frac{\partial \mathbf{F}}{\partial \mathbf{M}}$ of a function $\mathbf{F}$: $\mathbf{M}_{n\times m} \rightarrow \mathbf{F}_{p\times q}$ is represented as an $n\times m$ block matrix whose (k_1,k_2) entry is a $p\times q$ matrix $\frac{\partial \mathbf{F}}{\partial M_{k_1k_2}}$, where $M_{k_1k_2}$ is the (k_1,k_2) entry of $\mathbf{M}$. $\delta\mathbf{x}^{[k]}$ denotes the kth Kronecker power of $\delta\mathbf{x}$, i.e., $\delta\mathbf{x}^{[k]} = \delta\mathbf{x} \otimes \delta\mathbf{x} \otimes \cdots \otimes \delta\mathbf{x}$.

We assume that the random errors of the input quantities have zero mean values, i.e., $\mathrm{E}(\delta\mathbf{x}) = \mathbf{0}$. Neglecting the high-order terms in Eq. (19), we can obtain, based on the definition of mean and variance of random variables, the mean vector $\mathrm{E}(\delta\mathbf{p}_0)$ and variance matrix $\mathrm{var}(\delta\mathbf{p}_0)$ of the output error $\delta\mathbf{p}_0$ as

$$E[\delta\mathbf{p}_0(\mathbf{x})] \approx \frac{1}{2} \frac{\partial^2 \mathbf{p}_0(\bar{\mathbf{x}})}{\partial \mathbf{x}^{T^2}} vec[\mathrm{var}(\delta\mathbf{x})], \tag{20}$$

$$\mathrm{var}[\delta\mathbf{p}_0(\mathbf{x})] \approx \frac{\partial \mathbf{p}_0(\bar{\mathbf{x}})}{\partial \mathbf{x}^T} \mathrm{var}(\delta\mathbf{x}) \frac{\partial \mathbf{p}_0(\bar{\mathbf{x}})^T}{\partial \mathbf{x}}, \tag{21}$$

where $vec(\mathbf{M})$ denotes the vector created from a matrix $\mathbf{M}$ by stacking its columns, $\frac{\partial \mathbf{p}_0}{\partial \mathbf{x}^T} = [\cdots \frac{\partial \mathbf{p}_0}{\partial \mathbf{p}_i^T} \cdots \frac{\partial \mathbf{p}_0}{\partial r_i} \cdots]$, and $\frac{\partial^2 \mathbf{p}_0}{\partial \mathbf{x}^{T^2}} = [\cdots \frac{\partial^2 \mathbf{p}_0}{\partial \mathbf{p}_i^T \partial \mathbf{p}_j^T} \cdots \frac{\partial^2 \mathbf{p}_0}{\partial \mathbf{p}_i^T \partial r_j} \cdots \frac{\partial^2 \mathbf{p}_0}{\partial r_i \partial \mathbf{p}_j^T} \cdots \frac{\partial^2 \mathbf{p}_0}{\partial r_i \partial r_j} \cdots]$. Equations (20) and (21) agree with the formulation in ref. [13]. They show that the mean and variance of the output error are directly related to the variance of the input error.

Adopting the scheme of error analysis in refs. [13, 17], we evaluate the impact of the position error of $\mathbf{p}_i$, $\delta\mathbf{p}_i$, and the impact of the distance error of r_i, δr_i, on $\delta\mathbf{p}_0$ separately.

To evaluate the impact of $\delta\mathbf{p}_i$ on $\delta\mathbf{p}_0$, we assume that the measurement errors in mapping the reference points are zero-mean random variables and uncorrelated from one another with the same standard deviation σ_p for

each coordinate. We define two performance indices, the normalized total bias B_p, which represents the systematic estimation error, and the normalized total standard deviation error S_p, which represents the uncertainty of position estimation

$$B_p(\mathbf{p}_0) = \frac{|E(\delta\mathbf{p}_0)|}{\sigma_p^2} = \frac{1}{2}\left|\frac{\partial^2\mathbf{p}_0}{\partial\mathbf{p}^{T^2}} vec(\mathbf{I}_{nN})\right| = \frac{1}{2}\left|\sum_{i=1}^{N}\frac{\partial^2\mathbf{p}_0}{\partial\mathbf{p}_i^{T^2}} vec(\mathbf{I}_n)\right|, \tag{22}$$

$$S_p(\mathbf{p}_0) = \frac{\sqrt{Tr[\mathrm{var}(\delta\mathbf{p}_0)]}}{\sigma_p} = \sqrt{Tr\left(\frac{\partial\mathbf{p}_0}{\partial\mathbf{p}^T}\frac{\partial\mathbf{p}_0^T}{\partial\mathbf{p}}\right)}, \tag{23}$$

where $\mathbf{p} = [\cdots \mathbf{p}_i^T \cdots]^T$ is an nN-dimensional vector consisting of all $\mathbf{p}_i$, $|\mathbf{v}|$ denotes the norm of a vector $\mathbf{v}$, and $Tr(\mathbf{M})$ denotes the trace of a matrix $\mathbf{M}$. We notice that B_p and S_p are independent of σ_p.

Similarly, to evaluate the impact of δr_i on $\delta\mathbf{p}_0$, we assume that the errors in measuring distances are zero-mean random variables and uncorrelated from one another with the same standard deviation σ_r. We define two performance indices, the normalized total bias B_r and the normalized total standard deviation error S_r as

$$B_r(\mathbf{p}_0) = \frac{|E(\delta\mathbf{p}_0)|}{\sigma_r^2} = \frac{1}{2}\left|\frac{\partial^2\mathbf{p}_0}{\partial\mathbf{r}^{T^2}} vec(\mathbf{I}_N)\right| = \frac{1}{2}\left|\sum_{i=1}^{N}\frac{\partial^2\mathbf{p}_0}{\partial r_i^2}\right|, \tag{24}$$

$$S_r(\mathbf{p}_0) = \frac{\sqrt{Tr[\mathrm{var}(\delta\mathbf{p}_0)]}}{\sigma_r} = \sqrt{Tr\left(\frac{\partial\mathbf{p}_0}{\partial\mathbf{r}^T}\frac{\partial\mathbf{p}_0^T}{\partial\mathbf{r}}\right)}, \tag{25}$$

where $\mathbf{r} = [\cdots r_i \cdots]^T$ is an N-dimensional vector consisting of all r_i. We notice that B_r and S_r are independent of σ_r.

The differential terms of $\mathbf{p}_0$ relevant to Eqs. (22)–(25) include $\frac{\partial\mathbf{p}_0}{\partial\mathbf{p}_i^T}$, $\frac{\partial\mathbf{p}_0}{\partial r_i}$, $\frac{\partial^2\mathbf{p}_0}{\partial\mathbf{p}_i^{T^2}}$ and $\frac{\partial^2\mathbf{p}_0}{\partial r_i^2}$. They are derived and listed in the Appendix.

As discussed before, due to the duality of Eq. (18), we generally obtain two candidates of $\mathbf{p}_0$ from computation. Correspondingly, we can generally obtain two $\frac{\partial\mathbf{p}_0}{\partial\mathbf{x}^T}$ and two $\frac{\partial^2\mathbf{p}_0}{\partial\mathbf{x}^{T^2}}$, and hence two values for each of B_p, S_p, B_r, and S_r. In order to distinguish between them, we denote the values for B_p, S_p, B_r, and S_r corresponding to $+\sqrt{b^2-4ac}$ in Eq. (18) as B_{p+}, S_{p+}, B_{r+}, and S_{r+} respectively, and those corresponding to $-\sqrt{b^2-4ac}$ as B_{p-}, S_{p-}, B_{r-}, and S_{r-} respectively. As a result, in practice the prediction of the estimation error will, in general, depend on whether $\mathbf{p}_0$ is corresponding to $+\sqrt{b^2-4ac}$ or $-\sqrt{b^2-4ac}$.

An error analysis on the proposed trilateration algorithm has been conducted, and the results of the above-defined performance indices are reported in the following sections. Without loss of generality, our error analysis focuses on $\mathbb{R}^3$, with an understanding that the trilateration in $\mathbb{R}^2$ is indeed a downgrade of the trilateration in $\mathbb{R}^3$. Through representative examples, we test the proposed trilateration algorithm with three reference points at first and then with four reference points.

3.2. Trilateration in $\mathbb{R}^3$ with three reference points

Following the representative examples in refs. [13, 17], we examine a three-reference case in $\mathbb{R}^3$ in which the *XY* coordinates of the three reference points form an equilateral triangle inscribed in a circle centered at the origin of the frame of reference with a radius of 1000. The reference points are located at $\mathbf{p}_1 = [-500\sqrt{3} \quad -500 \quad 0]^T$, $\mathbf{p}_2 = [0 \quad 1000 \quad 0]^T$ and $\mathbf{p}_3 = [500\sqrt{3} \quad -500 \quad 0]^T$. We also define two square data acquisition regions as $R_+ = \{\mathbf{p}_0 = [x,\ y,\ z]^T | z = 8000,\ -4000 \le x,\ y \le 4000\}$ and $R_- = \{\mathbf{p}_0 = [x, y, z]^T | z = -8000, -4000 \le x, y \le 4000\}$, which are symmetric to the base plane defined by the reference points and corresponding to $+\sqrt{b^2-4ac}$ and $-\sqrt{b^2-4ac}$ in Eq. (18) respectively. As a result, we obtain B_{p+}, S_{p+}, B_{r+}, and S_{r+} across R_+, and B_{p-}, S_{p-}, B_{r-}, and S_{r-} across R_-.

Analyzing the effect of $\delta\mathbf{p}_i$ on $\delta\mathbf{p}_0$, we notice that B_{p+} and B_{p-} are symmetric to the base plane, so are S_{p+} and S_{p-}, i.e., $B_p(\mathbf{p}_0 = [x, y, \pm z]^T) = B_{p+}(\mathbf{p}_0 = [x, y, z]^T) = B_{p-}(\mathbf{p}_0 = [x, y, -z]^T)$ and $S_p(\mathbf{p}_0 = [x, y, \pm z]^T) = S_{p+}(\mathbf{p}_0 = [x, y, z]^T) = S_{p-}(\mathbf{p}_0 = [x, y, -z]^T)$. Moreover, B_p and S_p are symmetric to the orthographic projections of the medians of the base triangle onto the data acquisition planes. For example, one of the medians is the y axis, and correspondingly $B_p(\mathbf{p}_0 = [x, y, \pm z]^T) = B_p(\mathbf{p}_0 = [-x, y, \pm z]^T)$ and $S_p(\mathbf{p}_0 = [x, y, \pm z]^T) = S_p(\mathbf{p}_0 = [-x, y, \pm z]^T)$.

The resulting trends of B_p and S_p are shown in Fig. 5. Both B_p (systematic error) and S_p (estimation uncertainty) increase as $\mathbf{p}_0$ moves away from the centroid of the base triangle. The minimum values of B_p and S_p are obtained at the centers of the data acquisition regions $\mathbf{p}_0 = [0, 0, \pm 8000]^T$ which correspond to the centroid of the base triangle, where $B_{p,min} = 0.0055$ and $S_{p,min} = 9.3277$.

Compared with the results reported in ref. [13, 17] which were generated from the exactly same example, the proposed algorithm has lower S_p values (Figure 4a in ref. [17] reports that $S_p^2 \approx 170$ ($S_p = 13.04$) when $\mathbf{p}_0 = [0, 0, 8000]^T$, and $S_p^2 \approx 192$ ($S_p = 13.86$) when $\mathbf{p}_0 = [-4000, 4000, 8000]^T$, which are consistent with the results in ref. [13]. Since only S_p was reported in ref. [17], we only make a comparison in S_p here.), which means that the proposed trilateration algorithm has a reduced uncertainty in position estimation when using imperfectly mapped reference points.

Analyzing the effect of δr_i on $\delta\mathbf{p}_0$, we notice that similar to B_p and S_p, B_{r+} and B_{r-} are symmetric to the base plane, so are S_{r+} and S_{r-}; B_r and S_r are symmetric to the orthographic projections of the medians of the base triangle onto the data acquisition planes; B_r and S_r increase as $\mathbf{p}_0$ moves away from the centroid of the base triangle. The minimum values of B_r and S_r are obtained at the centers of the data acquisition regions, where $B_{r,min} = 0.0054$ and $S_{r,min} = 9.3277$. Moreover, the values of B_r and S_r are very close, if not equal, to those of B_p and S_p respectively (Table I). For this reason, we do not present the figures for B_r and S_r specifically.

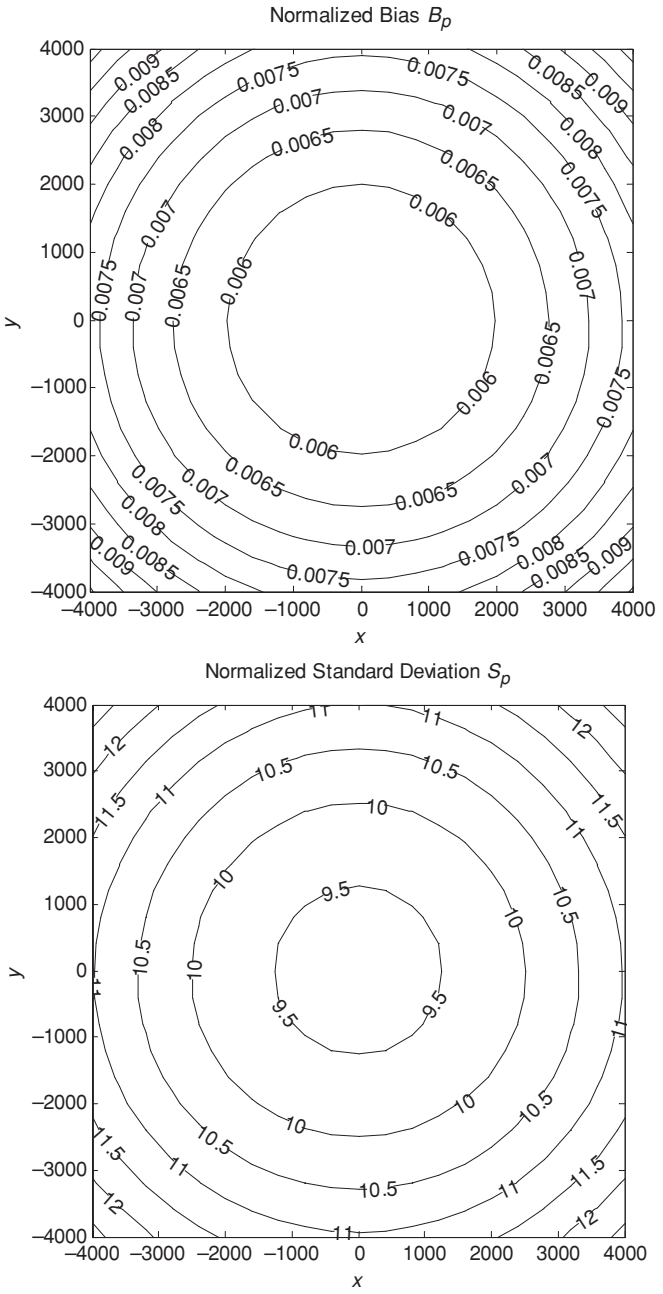

Fig. 5. Normalized total bias B_p and normalized total standard deviation error S_p obtained from the three-reference example.

Compared with the results reported in refs. [13, 17], the proposed algorithm has significantly lower B_r values (Ref. [17] reports that the maximum B_r on the edge of the same data acquisition region is about 0.03 which agrees with the result in ref. [13], while our result is about 0.01.), which means that the proposed trilateration algorithm has a reduced systematic error in position estimation when using erroneous distance measurements.

3.3. Trilateration in $\mathbb{R}^3$ with four reference points

To test the performance of the proposed trilateration algorithm with $N > n$ reference points, we examine a four-reference case in $\mathbb{R}^3$ in which the *XY* coordinates of the four reference points form a square inscribed in the same circle as in the three-reference example (centered at the origin of the frame of reference with a radius of 1000). The reference points are located at $\mathbf{p}_1 = [-500\sqrt{2} \quad -500\sqrt{2} \quad 0]^T$, $\mathbf{p}_2 = [-500\sqrt{2} \quad 500\sqrt{2} \quad 0]^T$, $\mathbf{p}_3 = [500\sqrt{2} \quad 500\sqrt{2} \quad 0]^T$, and $\mathbf{p}_4 = [500\sqrt{2} \quad -500\sqrt{2} \quad 0]^T$. Using the same two square data acquisition regions as the three-reference case, we calculate B_{p+}, S_{p+}, B_{r+}, and S_{r+} across R_+, and B_{p-}, S_{p-}, B_{r-}, and S_{r-} across R_-.

Table I. Comparison between B_p/S_p and B_r/S_r for the three-reference case.

$\mathbf{p}_0$	(0,0,8000)	(−4000, 4000, 8000)	(−2000, 1600, 8000)	(2000, 3200, 8000)
B_p	0.0055	0.0102	0.0064	0.0074
B_r	0.0054	0.0101	0.0062	0.0073
S_p	9.3277	12.7608	10.0348	10.8435
S_r	9.3277	12.7608	10.0348	10.8435

B_p/S_p and B_r/S_r are compared at 4 representative $\mathbf{p}_0$.

Table II. Comparison between B_p/S_p of the three-reference case and those of the four-reference case.

	$B_{p,min}$	$B_{p,max}$	$S_{p,min}$	$S_{p,max}$
Three-reference	0.0055	0.0102	9.3277	12.7608
Four-reference	0.0070	0.0101	8.0780	11.0059

In the three-reference case, $B_{p,min}$ and $\mathbf{S}_{p,min}$ are obtained at $\mathbf{p}_0 = [0, 0, \pm 8000]$, and $B_{p,max}$ and $S_{p,max}$ are obtained at $\mathbf{p}_0 = [\pm 4000, 4000, \pm 8000]$; In the four-reference case, $B_{p,min}$ is obtained at $\mathbf{p}_0 = [0, 0, \pm 8000]$, $B_{p,max}$ is obtained at $\mathbf{p}_0 = [4000, 4000, 8000]$ and $\mathbf{p}_0 = [-4000, -4000, -8000]$, $\mathbf{S}_{p,min}$ is obtained at $\mathbf{p}_0 = [0, 0, \pm 8000]$, and $S_{p,max}$ is obtained at $\mathbf{p}_0 = [\pm 4000, \pm 4000, \pm 8000]$.

The resulting trends of B_{p+} and B_{p-} are shown in Fig. 6. Different from the three-reference case, B_{p+} and B_{p-} are not symmetric to the base plane, i.e., in general $B_{p+}(\mathbf{p}_0 = [x, y, z]^T) \neq B_{p-}(\mathbf{p}_0 = [x, y, -z]^T)$. Instead, B_{p+} and B_{p-} are mutually symmetric to the diagonals of the base square, i.e., $B_{p+}(\mathbf{p}_0 = [x, y, z]^T) = B_{p-}(\mathbf{p}_0 = [-x, -y, -z]^T)$.

S_{p+} and S_{p-} are still symmetric to the base plane, i.e., $S_p(\mathbf{p}_0 = [x, y, \pm z]^T) = S_{p+}(\mathbf{p}_0 = [x, y, z]^T) = S_{p-}(\mathbf{p}_0 = [x, y, -z]^T)$. Moreover, S_p is radially symmetric to the vertical axis which passes through the centers of the base square and data acquisition regions, i.e., $S_p(\mathbf{p}_0 = [x, y, \pm z]^T) = S_p(\mathbf{p}_0 = [-x, y, \pm z]^T) = S_p(\mathbf{p}_0 = [x, -y, \pm z]^T) = S_p(\mathbf{p}_0 = [-x, -y, \pm z]^T)$. The trend of S_p is shown in Fig. 7.

Both B_p and S_p increase as $\mathbf{p}_0$ moves away from the center of the base square. The minimum values of B_p and S_p are obtained at the centers of the data acquisition regions, where $B_{p,min} = B_{p+,min} = B_{p-,min} = 0.0070$ and $S_{p,min} = S_{p+,min} = S_{p-,min} = 8.0780$.

Compared with the three-reference case (Table II), we notice that there is an increase in the minimum B_p due to the addition of another imperfectly mapped reference point ($B_{p,min} = 0.0070$ for the four-reference case versus $B_{p,min} = 0.0055$ for the three-reference case). However, the maximum B_p decreases slightly ($B_{p,max} = 0.0101$ for the

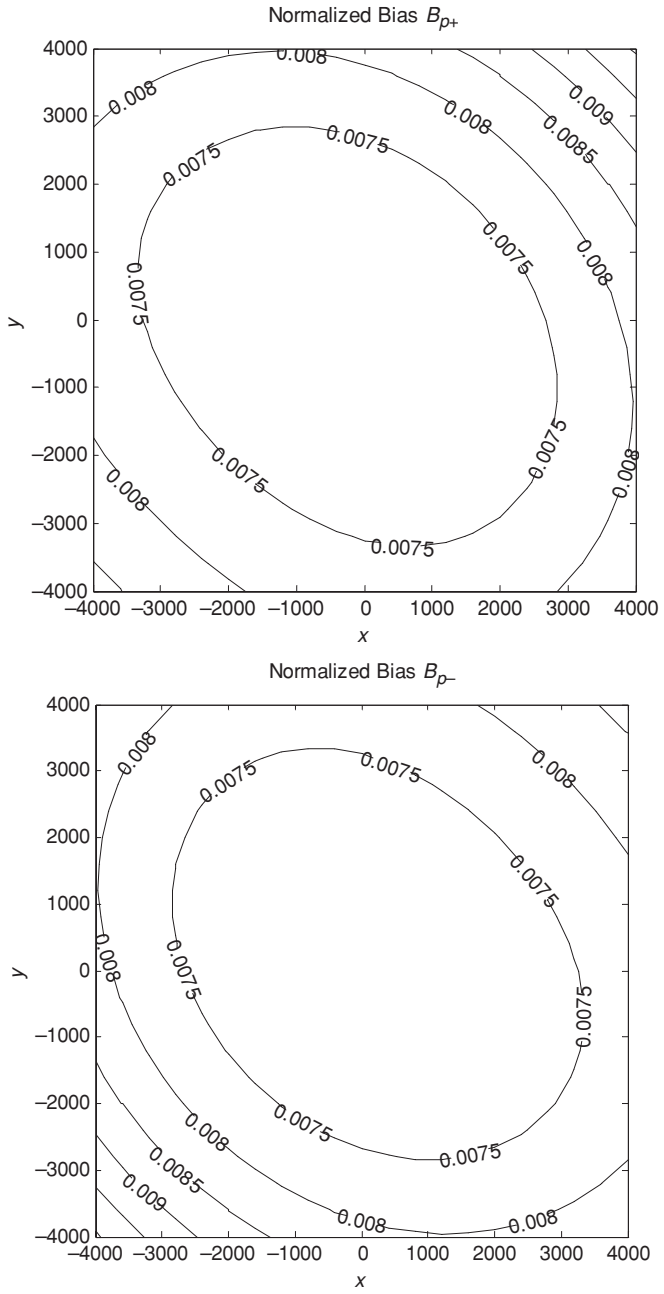

Fig. 6. Normalized total bias B_{p+} and B_{p-} obtained from the four reference example.

four-reference case versus $B_{p,max} = 0.0102$ for the three-reference case). Moreover, a comparison between Figs. 5 and 6 exposes that the values of B_p at the corners of the data acquisition regions in the four-reference case are in general lower than those in the three-reference case. This means that, as $\mathbf{p}_0$ moves away from the center of the data acquisition region, the increase rate of B_p for the four-reference case is less than that of the three-reference case, and hence the systematic estimation error with four reference points tends to be lower than that with three reference points when the object is located farther away. We also notice (from Table II and Figs. 5 and 7) that S_p of the four-reference case is significantly lower than that of the three-reference case across the data acquisition regions, which means that the uncertainty in position estimation, when using imperfectly

Table III. Comparison between S_p and S_r for the four-reference case.

$\mathbf{p}_0$	(0,0,8000)	(−4000, 4000, 8000)	(−2000, 1600, 8000)	(2000, 3200, 8000)
S_p	8.0780	11.0059	8.6836	9.3884
S_r	8.0780	11.0059	8.6836	9.3884

S_p and S_r are compared at 4 representative $\mathbf{p}_0$.

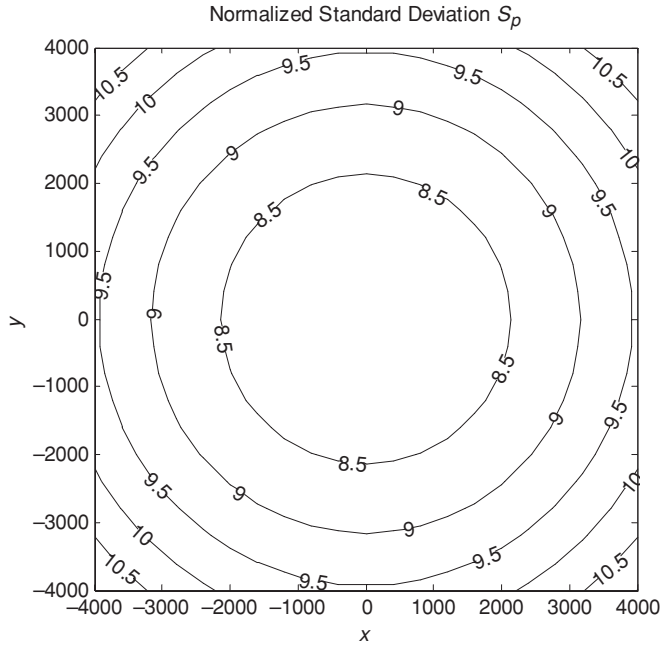

Fig. 7. Normalized total standard deviation error S_p obtained from the four-reference example.

mapped reference points, will decrease by referring to more reference points.

Similar to the three-reference case, B_{r+} and B_{r-} are symmetric to the base plane, so are S_{r+} and S_{r-}. Moreover, B_r and S_r are radially symmetric to the vertical axis which passes through the centers of the base square and data acquisition regions. Both B_r and S_r increase as $\mathbf{p}_0$ moves away from the centroid of the base square. The minimum values of B_r and S_r are obtained at the centers of the data acquisition regions, where $B_{r,min} = 0.0040$ and $S_{r,min} = 8.0780$. The trend of B_r is shown in Fig. 8. We also notice that the values of S_r are very close, if not equal, to those of S_p (Table III). For this reason, we do not present the figure for S_r specifically.

Compared with the values in the three-reference case (from Table IV and Figs. 5, 7, and 8), both B_r and S_r in the four-reference case are significantly lower across the data acquisition regions, which means that both the systematic error and the uncertainty in position estimation, when using erroneous distance measurements, will decrease by combining more distance measurements.

3.4. Approximation performance of the error indices

In the above error analysis, the B_p, S_p, B_r and S_r are calculated directly from Eqs. (22)–(25). In order to validate

Table IV. Comparison between B_r/S_r of the three-reference case and those of the four-reference case.

	$B_{r,min}$	$B_{r,max}$	$S_{r,min}$	$S_{r,max}$
Three-reference	0.0054	0.0101	9.3277	12.7608
Four-reference	0.0040	0.0075	8.0780	11.0059

In the three-reference case, $B_{r,min}$ and $\mathbf{S}_{r,min}$ are obtained at $\mathbf{p}_0 = [0, 0, \pm 8000]$, and $B_{r,max}$ and $S_{r,max}$ are obtained at $\mathbf{p}_0 = [\pm 4000, 4000, \pm 8000]$; In the four-reference case, $B_{r,min}$ and $S_{r,min}$ are obtained at $\mathbf{p}_0 = [0, 0, \pm 8000]$, and $B_{r,max}$ and $S_{r,max}$ are obtained at $\mathbf{p}_0 = [\pm 4000, \pm 4000, \pm 8000]$.

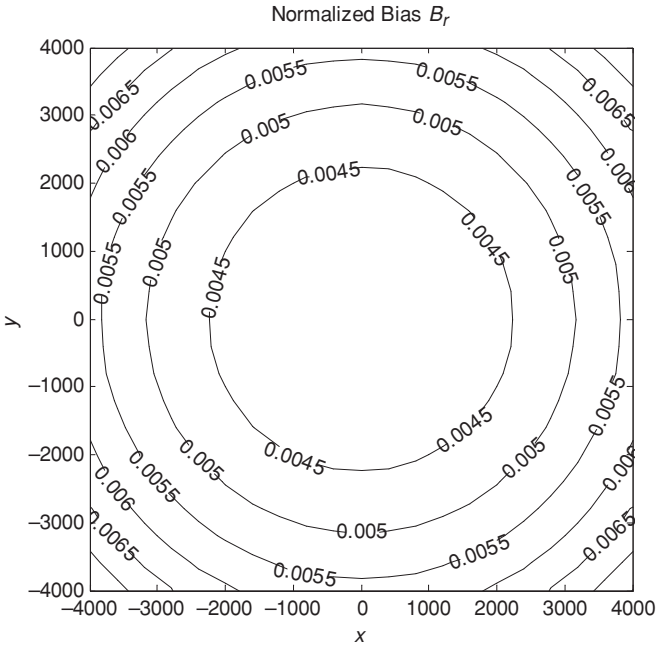

Fig. 8. Normalized total bias B_r obtained from the four-reference example.

those closed-form formulas in approximating the relationship between the input and output errors, the results calculated from Eqs. (22)–(25) are compared with those obtained from numerical simulations.

In the numerical simulations, to evaluate the impact of $\delta\mathbf{p}_i$ on $\delta\mathbf{p}_0$, we assume that the components of $\delta\mathbf{p}_i$ are uncorrelated from one another and follow the Gaussian distribution with the zero mean and the same standard deviation σ_p. Given the correct position of the target object $\mathbf{p}_0$, the correct positions of the reference points $\mathbf{p}_i$ and the correct distances between the object and reference points r_i, we generate a number of erroneous $\mathbf{p}_i$ according to the random distribution of $\delta\mathbf{p}_i$, substitute them into Algorithm 1 to calculate corresponding erroneous $\mathbf{p}_0$, obtain the statistics of $\delta\mathbf{p}_0$, and then calculate B_p and S_p from $\mathrm{E}(\delta\mathbf{p}_0)$ and $\mathrm{var}(\delta\mathbf{p}_0)$ respectively. To evaluate the impact of δr_i on $\delta\mathbf{p}_0$, we assume that δr_i are uncorrelated from one another and follow the Gaussian distribution with the zero mean and the same standard deviation σ_r. Given the correct $\mathbf{p}_0$, $\mathbf{p}_i$ and r_i, we generate a number of erroneous r_i according to the random distribution of δr_i, substitute them into Algorithm 1 to calculate corresponding erroneous $\mathbf{p}_0$, obtain the statistics of $\delta\mathbf{p}_0$, and then calculate B_r and S_r from $\mathrm{E}(\delta\mathbf{p}_0)$ and $\mathrm{var}(\delta\mathbf{p}_0)$ respectively.

Different from the closed-form calculation using Eqs. (22)–(25) which is independent of σ_p and σ_r, the numerical simulation must be carried out based on specified σ_p and σ_r. We set σ_p and σ_r with different values ($\sigma_p, \sigma_r \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$), run the simulation with 10,000 samples for each value, and calculate B_p, S_p, B_r, and S_r across the data acquisition regions. It turns out that the resulting B_p, S_p, B_r, and S_r are consistent across the range of σ_p and σ_r, which verifies that B_p, S_p, B_r, and S_r are largely independent of σ_p and σ_r, as indicated in Eqs. (22)–(25).

The comparison between the B_p, S_p, B_r, and S_r obtained from Eqs. (22)–(25) and those obtained from the numerical simulations verifies that Eqs. (22)–(25) provide a good closed-form approximation of the effect of $\delta\mathbf{p}_i$ and δr_i on $\delta\mathbf{p}_0$ (Table V).

4. Experimental Test

The effectiveness of the proposed trilateration algorithm has also been verified through an experiment of trilaterating an onboard stereovision unit. Focusing on testing the estimation error under the combined effect of realistic input errors and the execution speed of the proposed trilateration algorithm, we used the stereovision system only for obtaining the distance measurements between the unit and reference points. The measured distances were input to the proposed trilateration algorithm to position the unit in 3D space.

The experiment system consisted of a reference pattern and an onboard stereovision unit:

1. Two reference patterns were used for three-reference and four-reference trilateration respectively, where bright LEDs were adopted to represent reference points. The three-reference pattern contains three symmetric reference points which define an equilateral base triangle, while the four-reference pattern contains four symmetric reference points which define a base square, both inscribed in a circle with a radius of 550 mm. Since they were built in the same way, we only show the four-reference pattern here (Fig. 9a). For either three-reference or four-reference trilateration, the corresponding pattern was hung under

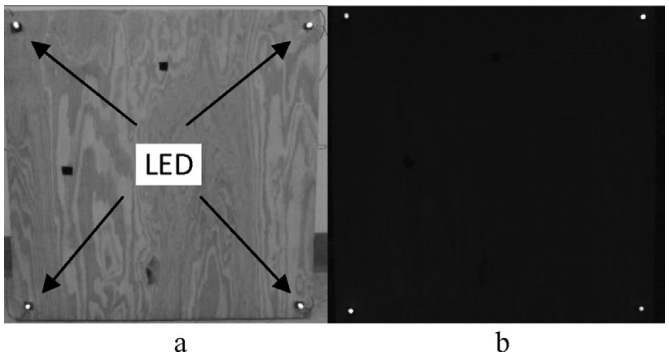

Fig. 9. Reference pattern used in the experiment.

Table V. Comparison between B_p, S_p, B_r, and S_r calculated from Eqs. (22)–(25) and those obtained from numerical simulations.

$\mathbf{p}_0$	(0, 0, 8000)	(−4000, 4000, 8000)	(−2000, 1600, 8000)	(2000, 3200, 8000)
B_p	0.0055	0.0102	0.0064	0.0074
$B_{p,sim}$	0.0057	0.0107	0.0066	0.0078
S_p	9.3277	12.7608	10.0348	10.8435
$S_{p,sim}$	9.3609	12.8730	10.0896	10.8947
B_r	0.0054	0.0101	0.0062	0.0073
$B_{r,sim}$	0.0054	0.0111	0.0066	0.0076
S_r	9.3277	12.7608	10.0348	10.8435
$S_{r,sim}$	9.3059	12.7783	10.0170	10.8358
B_p	0.0070	0.0085	0.0073	0.0083
$B_{p,sim}$	0.0077	0.0090	0.0078	0.0087
S_p	8.0780	11.0059	8.6836	9.3884
$S_{p,sim}$	8.0725	11.0442	8.6908	9.4088
B_r	0.0040	0.0075	0.0047	0.0054
$B_{r,sim}$	0.0040	0.0077	0.0047	0.0057
S_r	8.0780	11.0059	8.6836	9.3884
$S_{r,sim}$	8.1271	11.0947	8.7389	9.4609

The closed-form and simulation results of B_p, S_p, B_r, and S_r are compared at 4 representative $\mathbf{p}_0$. The subscript "sim" denotes the results from the numerical simulations. The presented $B_{p,sim}$ and $S_{p,sim}$ are obtained with $\sigma_p = 70$, while the presented $B_{r,sim}$ and $S_{r,sim}$ are obtained with $\sigma_r = 70$.

the ceiling of an indoor experiment environment with the base plane parallel to the flat floor. We used the base plane as the *XY* plane of the global frame of reference, and the center of the base triangle or square as the origin of the frame.

2. A stereovision unit, consisting of two Matrix Vision BlueFox USB 640 × 480 CCD cameras with Kowa 12 mm lenses, was installed on an Activmedia Pioneer3-DX mobile robot (Fig. 10). This stereovision set is supported by a Directed Perception PTU-D46-17 controllable pan/tilt unit which enables the stereovision unit to orient toward the reference pattern. With the mobile robot moving in the flat planar indoor environment, the cameras were kept on a plane parallel to the base plane (*XY* plane of the global frame) at a distance of 1600 mm. The apertures of both the cameras were intentionally minimized such that, with regular ambient lighting, the LEDs were shown as bright spots outstanding from the dark background in the images (Fig. 9b), which provide convenience for segmenting the reference points from the images.

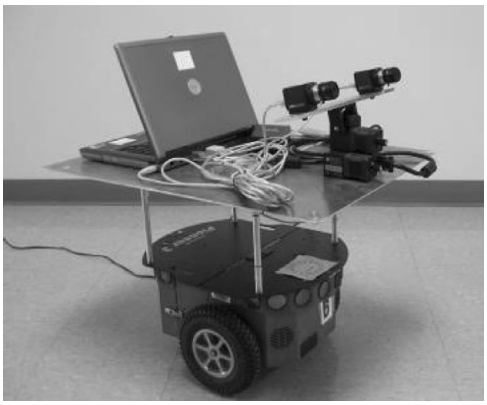

Fig. 10. Onboard experimental system.

The cameras were calibrated using a camera calibration toolbox for Matlab available at http://www.vision.caltech.edu/bouguetj/calib_doc/ and a printed black/white checkerboard pattern to find out the intrinsic parameters of the cameras, such as focal length, image center and distortion coefficients, and the transformation between the two cameras. The calibration results are listed as follows:

1. For the left camera, we have the focal length (pixels) $\mathbf{f} = [1623.4699; 1626.0442] \pm [1.1299; 1.1164]$, image center (pixels) $\mathbf{c} = [329.6443;\ 248.5307] \pm [1.9019;\ 1.6757]$, and distortion coefficients $\mathbf{k} = [-0.3546; 0.3416; 0.0013; 0.0020] \pm [0.0115; 0.1950; 0.00019; 0.00018]$.
2. For the right camera, we have the focal length (pixels) $\mathbf{f} = [1627.7598; 1629.5029] \pm [1.1475; 1.1091]$, image center (pixels) $\mathbf{c} = [334.0203;\ 246.5809] \pm [1.9384;\ 1.5751]$, and distortion coefficients $\mathbf{k} = [-0.3364; 0.2858; 0.0006; -0.0011] \pm [0.0094; 0.1669; 0.00031; 0.00014]$.
3. For the transformation between the two cameras, we have the rotation vector (radians) $\mathbf{v}_R = [-0.0129; 0.2389; -0.\ 0016] \pm [0.0045; 0.0064; 0.0004]$ from which the corresponding rotation matrix can be calculated using the Rodrigues' rotation formula, and translation vector (mm) $\mathbf{v}_T = [-289.0918; 3.2027; 32.6389] \pm [0.1915; 0.1109; 1.3917]$.

In each of the above calibration results, the numbers in the first "[]" are the calibrated values, while the numbers in the second "[]" are the numerical errors which, as

Table VI. Camera position estimation errors.

	Three-reference case		Four-reference case	
R_h	$\lvert E(\delta\mathbf{p}_0)\rvert$	$\lvert \mathrm{STD}(\delta\mathbf{p}_0)\rvert$	$\lvert E(\delta\mathbf{p}_0)\rvert$	$\lvert \mathrm{STD}(\delta\mathbf{p}_0)\rvert$
2000	2.6391	20.7778	1.7574	14.8373
2500	11.8517	24.9709	11.3892	20.1474
3000	14.7132	28.2164	13.4792	25.2913
3500	20.4381	32.3926	18.2349	27.7020
4000	28.3571	49.2009	26.1342	46.9997

In this table, $E(.)$ denotes mean, STD(.) denotes standard deviation, $|.|$ denotes the norm of the vector, and the unit of lengths is millimeter.

indicated by the toolbox, are approximately three times the standard deviations. Since the transformation between the two cameras is referenced to the left camera, the position of the stereovision unit is represented by the position of the left camera.

For data acquisition, we moved the mobile robot to a few different distances from the reference pattern (represented by the horizontal distance R_h from the Z-axis of the global frame which passes through the center of the base triangle/square), and at each distance we acquired data from a number of locations on the circle about the Z-axis with the radius of R_h. At each location, the stereovision unit was oriented toward the reference pattern, a pair of images of the reference pattern was taken by the stereovision unit, and the actual position of the left camera was measured manually. The data were stored in an onboard laptop computer.

The collected images were processed after the data acquisition process. Focusing on evaluating the proposed trilateration algorithm, we segmented the reference points from each pair of images manually (after correcting the image distortions), ran a stereovision program to estimate the distances from the left camera to the reference points based on the camera calibration results, and then implemented the proposed trilateration algorithm to estimate the 3D position of the left camera. The estimation error was calculated by comparing the estimated position with the measured one, which was affected by all the above-mentioned input errors, including the reference positioning errors, distance measurement errors (caused by camera calibration errors and image segmentation errors), as well as the manual camera position measurement errors.

As the result, we obtained a profile of the position estimation error under the experimental settings (Table VI). We notice that, even with the above less accurate measurements, the experimental system achieved an average 3D positioning accuracy of <30 mm in both three-reference and four-reference cases at an observation distance (between the cameras and pattern) of 4.3 m (corresponding to $R_h = 4$ m), which, as we believe, is sufficient for many mobile robot applications. The results also show some improvements in both the estimation accuracy and uncertainty by using more reference points (four reference points versus three reference points in our experiment).

Moreover, due to its closed-form nature, the proposed trilateration algorithm can be implemented at a high speed in practice. We tested both a Matlab program and a compiled standalone executable (exe file) of the algorithm respectively on a laptop PC with a 1.66 G Hz Intel Core 2 CPU. Given the reference positions and distances, for the three-reference trilateration, the Matlab version of the proposed trilateration algorithm takes 3.8310×10^{-4} s in average, while the standalone version takes 3.7006×10^{-4} s in average; for the four-reference trilateration, the Matlab program takes 3.9732×10^{-4} s in average, while the standalone executable takes 3.9333×10^{-4} s. It means that the proposed trilateration algorithm is highly applicable to real-time applications. Moreover, requiring only the reference positions and corresponding distance measurements as the input, the proposed algorithm is ready to be integrated into various ranging-based trilateration systems with different ways of distance measurement, for instance, those systems based on ultrasound[6,10,30,31] and radio frequency signals.[7,8,11]

5. Conclusion

This paper presents a novel closed-form trilateration algorithm which estimates the position of a target object, such as a mobile robot, based on the simultaneous distance measurements from multiple reference points. Solving the nonlinear least-squares trilateration problem, the proposed algorithm provides a near optimal position estimate of the intersection point of $N \geq n$ spheres in $\mathbb{R}^n$ ($n = 2$ for 2D environments and $n = 3$ for 3D environments), not limited to solving for the intersection points of exactly n spheres in $\mathbb{R}^n$. Using standard linear algebra techniques, the proposed algorithm has low computational complexity. Without depending on the techniques which tend to be affected by algebraic singularities, such as matrix inversion, the proposed algorithm has high operational robustness. The performance analysis shows that the algorithm is highly effective, with lower systematic bias and estimation uncertainty than representative closed-form methods, when dealing with erroneous inputs of distance measurements and reference points. The results have also verified that introducing more reference points and corresponding distance measurements into the trilateration process will in general reduce the estimation uncertainty. The experimental test shows further that the proposed trilateration algorithm is highly applicable to real-time applications. Though targeting the applications in mobile robotics, the proposed trilateration algorithm is readily applicable to any ranging-based object localization tasks in various environments and scenarios.

Appendix

In this appendix, we provide the differential terms of $\mathbf{p}_0$ relevant to Eqs. (22)–(25), including $\frac{\partial \mathbf{p}_0}{\partial \mathbf{p}_i^T}$, $\frac{\partial \mathbf{p}_0}{\partial r_i}$, $\frac{\partial^2 \mathbf{p}_0}{\partial \mathbf{p}_i^{T2}}$ and $\frac{\partial^2 \mathbf{p}_0}{\partial r_i^2}$.

$$
\left.
\begin{aligned}
\frac{\partial \mathbf{p}_0}{\partial \mathbf{p}_i^T} &= \frac{\partial \mathbf{q}}{\partial \mathbf{p}_i^T} + \frac{\partial \mathbf{c}}{\partial \mathbf{p}_i^T},\\
\frac{\partial q_k}{\partial \mathbf{p}_i^T} &= \frac{1}{\det(\mathbf{H}')}\left[\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T} + \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T} q_n + \det\left(\mathbf{H}_k^h\right)\frac{\partial q_n}{\partial \mathbf{p}_i^T}\right] - \frac{\det\left(\mathbf{H}_k^f\right) + \det\left(\mathbf{H}_k^h\right) q_n}{\det(\mathbf{H}')^2}\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T},\\
&\qquad\qquad \forall k \in \{1,\ldots,n-1\},\\
\frac{\partial q_n}{\partial \mathbf{p}_i^T} &= \frac{1}{2a}\left[-\frac{\partial b}{\partial \mathbf{p}_i^T} \pm \frac{1}{\sqrt{b^2-4ac}}\left(b\frac{\partial b}{\partial \mathbf{p}_i^T} - 2c\frac{\partial a}{\partial \mathbf{p}_i^T} - 2a\frac{\partial c}{\partial \mathbf{p}_i^T}\right)\right] - \frac{-b \pm \sqrt{b^2-4ac}}{2a^2}\frac{\partial a}{\partial \mathbf{p}_i^T},\\
\frac{\partial a}{\partial \mathbf{p}_i^T} &= 2\sum_{k=1}^{n-1}\left[\det\left(\mathbf{H}_k^h\right)\frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\right] + 2\det(\mathbf{H}')\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T},\\
\frac{\partial b}{\partial \mathbf{p}_i^T} &= 2\sum_{k=1}^{n-1}\left[\det\left(\mathbf{H}_k^h\right)\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T} + \det\left(\mathbf{H}_k^f\right)\frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\right],\\
\frac{\partial c}{\partial \mathbf{p}_i^T} &= 2\sum_{k=1}^{n-1}\left[\det\left(\mathbf{H}_k^f\right)\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T}\right] - 2\mathbf{q}^T\mathbf{q}\det(\mathbf{H}')\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T} - \det(\mathbf{H}')^2\frac{\partial(\mathbf{q}^T\mathbf{q})}{\partial \mathbf{p}_i^T},\\
\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left(\prod_{l\in SS, l\neq l'} H'_{l,\sigma(l)}\frac{\partial H'_{l',\sigma(l')}}{\partial \mathbf{p}_i^T}\right),\\
\frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left(\prod_{l\in SS, l\neq l'} H^h_{k\,l,\sigma(l)}\frac{\partial H^h_{k\,l',\sigma(l')}}{\partial \mathbf{p}_i^T}\right),\\
\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left(\prod_{l\in SS, l\neq l'} H^f_{k\,l,\sigma(l)}\frac{\partial H^f_{k\,l',\sigma(l')}}{\partial \mathbf{p}_i^T}\right),\\
\frac{\partial \mathbf{H}}{\partial \mathbf{p}_i^T} &= \frac{2}{N}\left[\mathbf{I}_n \otimes \left(\mathbf{c}^T - \mathbf{p}_i^T\right) + (\mathbf{c}-\mathbf{p}_i)vec(\mathbf{I}_n)^T\right],\\
\frac{\partial \mathbf{f}}{\partial \mathbf{p}_i^T} &= \frac{\partial \mathbf{a}}{\partial \mathbf{p}_i^T} + \frac{\partial \mathbf{B}}{\partial \mathbf{p}_i^T}(\mathbf{I}_n \otimes \mathbf{c}) + \frac{1}{N}\mathbf{B} + \frac{2}{N}\left[\mathbf{I}_n \otimes \left(\mathbf{c}^T\mathbf{c}\right)\right] + \frac{4}{N}\mathbf{c}\mathbf{c}^T,\\
\frac{\partial \mathbf{a}}{\partial \mathbf{p}_i^T} &= \frac{1}{N}\left[\mathbf{I}_n \otimes \left(\mathbf{p}_i^T\mathbf{p}_i\right) + 2\mathbf{p}_i\mathbf{p}_i^T - r_i^2\right],\ \frac{\partial \mathbf{B}}{\partial \mathbf{p}_i^T} = -\frac{2}{N}\left[\mathbf{I}_n \otimes \mathbf{p}_i^T + \mathbf{p}_i vec(\mathbf{I}_n)^T + \mathbf{p}_i^T \otimes \mathbf{I}_n\right],\\
\frac{\partial \mathbf{c}}{\partial \mathbf{p}_i^T} &= \frac{1}{N}\mathbf{I}_n,\\
\frac{\partial(\mathbf{q}^T\mathbf{q})}{\partial \mathbf{p}_i^T} &= \frac{2}{N}\left(\mathbf{c}^T - \mathbf{p}_i^T\right).
\end{aligned}
\right\} \quad \text{(A1)}
$$

The summation $\sum_{\sigma\in P_{ss}}$ in $\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}$, $\frac{\partial \det(\mathbf{H}_k^h)}{\partial \mathbf{p}_i^T}$ and $\frac{\partial \det(\mathbf{H}_k^f)}{\partial \mathbf{p}_i^T}$ is computed over all the permutations of the set of numbers $SS = \{1,2,\ldots,n\}$. P_{ss} denotes the set of all n! permutations of SS. $\operatorname{sgn}(\sigma)$ denotes the signature of the permutation σ, where $\operatorname{sgn}(\sigma) = +1$ if σ is an even permutation, and $\operatorname{sgn}(\sigma) = -1$ if it is odd.

$$
\left.
\begin{aligned}
\frac{\partial \mathbf{p}_0}{\partial r_i} &= \frac{\partial \mathbf{q}}{\partial r_i}, \quad \frac{\partial q_k}{\partial r_i} = \frac{1}{\det(\mathbf{H}')}\left[\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial r_i} + \det\left(\mathbf{H}_k^h\right)\frac{\partial q_n}{\partial r_i}\right], \forall k \in \{1,\ldots,n-1\},\\
\frac{\partial q_n}{\partial r_i} &= \frac{1}{2a}\left[-\frac{\partial b}{\partial r_i} \pm \frac{1}{\sqrt{b^2-4ac}}\left(b\frac{\partial b}{\partial r_i} - 2a\frac{\partial c}{\partial r_i}\right)\right], \quad \frac{\partial b}{\partial r_i} = 2\sum_{k=1}^{n-1}\left[\det\left(\mathbf{H}_k^h\right)\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial r_i}\right],\\
\frac{\partial c}{\partial r_i} &= 2\sum_{k=1}^{n-1}\left[\det\left(\mathbf{H}_k^f\right)\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial r_i}\right] - \det(\mathbf{H}')^2\frac{\partial(\mathbf{q}^T\mathbf{q})}{\partial r_i}\\
\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial r_i} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left(\prod_{l\in SS, l\neq l'} H^f_{k\,l,\sigma(l)}\frac{\partial H^f_{k\,l',\sigma(l')}}{\partial r_i}\right), \quad \frac{\partial \mathbf{f}}{\partial r_i} = \frac{\partial \mathbf{a}}{\partial r_i} + \frac{\partial \mathbf{B}}{\partial r_i}\mathbf{c},\ \frac{\partial \mathbf{a}}{\partial r_i} = -\frac{2}{N}r_i\mathbf{p}_i,\ \frac{\partial \mathbf{B}}{\partial r_i} = \frac{2}{N}r_i\mathbf{I}_n,\ \frac{\partial(\mathbf{q}^T\mathbf{q})}{\partial r_i} = \frac{2}{N}r_i.
\end{aligned}
\right\} \quad \text{(A2)}
$$

$$
\left.
\begin{aligned}
\frac{\partial^2 \mathbf{p}_0}{\partial \mathbf{p}_i^{T\,2}} &= \frac{\partial^2 \mathbf{q}}{\partial \mathbf{p}_i^{T\,2}},\\
\frac{\partial^2 q_k}{\partial \mathbf{p}_i^{T\,2}} &= \frac{1}{\det(\mathbf{H}')}\left\{\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^{T\,2}} + \frac{\partial q_n}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\right] + q_n \frac{\partial^2 \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^{T\,2}} + \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\left(\mathbf{I}_n \otimes \frac{\partial q_n}{\partial \mathbf{p}_i^T}\right) + \det\left(\mathbf{H}_k^h\right)\frac{\partial^2 q_n}{\partial \mathbf{p}_i^{T\,2}}\right\}\\
&\quad - \frac{1}{\det(\mathbf{H}')^2}\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\left\{\mathbf{I}_n \otimes \left[\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T} + \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T} q_n + \det\left(\mathbf{H}_k^h\right)\frac{\partial q_n}{\partial \mathbf{p}_i^T}\right]\right\} - \frac{\det\left(\mathbf{H}_k^f\right) + \det\left(\mathbf{H}_k^h\right) q_n}{\det(\mathbf{H}')^2}\frac{\partial^2 \det(\mathbf{H}')}{\partial \mathbf{p}_i^{T\,2}}\\
&\quad + \left\{-\frac{1}{\det(\mathbf{H}')^2}\left[\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T} + \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T} q_n + \det\left(\mathbf{H}_k^h\right)\frac{\partial q_n}{\partial \mathbf{p}_i^T}\right]\right.\\
&\quad \left. + 2\frac{\det\left(\mathbf{H}_k^f\right) + \det\left(\mathbf{H}_k^h\right) q_n}{\det(\mathbf{H}')^3}\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\right\}\left[\mathbf{I}_n \otimes \frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\right], \qquad \forall k \in \{1, \ldots, n-1\},\\
\frac{\partial^2 q_n}{\partial \mathbf{p}_i^{T\,2}} &= \frac{1}{2a}\left\{-\frac{\partial^2 b}{\partial \mathbf{p}_i^{T\,2}} \pm \frac{1}{\sqrt{b^2-4ac}}\left[\frac{\partial b}{\partial \mathbf{p}_i^T}\left(\mathbf{I}_n \otimes \frac{\partial b}{\partial \mathbf{p}_i^T}\right) + b\frac{\partial^2 b}{\partial \mathbf{p}_i^{T\,2}} - 2\frac{\partial c}{\partial \mathbf{p}_i^T}\left(\mathbf{I}_n \otimes \frac{\partial a}{\partial \mathbf{p}_i^T}\right) - 2c\frac{\partial^2 a}{\partial \mathbf{p}_i^{T\,2}} - 2\frac{\partial a}{\partial \mathbf{p}_i^T}\left(\mathbf{I}_n \otimes \frac{\partial c}{\partial \mathbf{p}_i^T}\right) - 2a\frac{\partial^2 c}{\partial \mathbf{p}_i^{T\,2}}\right]\right.\\
&\quad \left. \mp \frac{1}{\sqrt{b^2-4ac}^3}\left(b\frac{\partial b}{\partial \mathbf{p}_i^T} - 2c\frac{\partial a}{\partial \mathbf{p}_i^T} - 2a\frac{\partial c}{\partial \mathbf{p}_i^T}\right)\left[\mathbf{I}_n \otimes \left(b\frac{\partial b}{\partial \mathbf{p}_i^T} - 2c\frac{\partial a}{\partial \mathbf{p}_i^T} - 2a\frac{\partial c}{\partial \mathbf{p}_i^T}\right)\right]\right\}\\
&\quad - \frac{1}{2a^2}\frac{\partial a}{\partial \mathbf{p}_i^T}\left\{\mathbf{I}_n \otimes \left[-\frac{\partial b}{\partial \mathbf{p}_i^T} \pm \frac{1}{\sqrt{b^2-4ac}}\left(b\frac{\partial b}{\partial \mathbf{p}_i^T} - 2c\frac{\partial a}{\partial \mathbf{p}_i^T} - 2a\frac{\partial c}{\partial \mathbf{p}_i^T}\right)\right]\right\} - \frac{-b \pm \sqrt{b^2-4ac}}{2a^2}\frac{\partial^2 a}{\partial \mathbf{p}_i^{T\,2}}\\
&\quad + \left\{-\frac{1}{2a^2}\left[-\frac{\partial b}{\partial \mathbf{p}_i^T} \pm \frac{1}{\sqrt{b^2-4ac}}\left(b\frac{\partial b}{\partial \mathbf{p}_i^T} - 2c\frac{\partial a}{\partial \mathbf{p}_i^T} - 2a\frac{\partial c}{\partial \mathbf{p}_i^T}\right)\right] + \frac{-b \pm \sqrt{b^2-4ac}}{a^3}\frac{\partial a}{\partial \mathbf{p}_i^T}\right\}\left[\mathbf{I}_n \otimes \frac{\partial a}{\partial \mathbf{p}_i^T}\right],\\
\frac{\partial^2 a}{\partial \mathbf{p}_i^{T\,2}} &= 2\sum_{k=1}^{n-1}\left\{\frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\right] + \det\left(\mathbf{H}_k^h\right)\frac{\partial^2 \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^{T\,2}}\right\} + 2\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\right] + 2\det(\mathbf{H}')\frac{\partial^2 \det(\mathbf{H}')}{\partial \mathbf{p}_i^{T\,2}},\\
\frac{\partial^2 b}{\partial \mathbf{p}_i^{T\,2}} &= 2\sum_{k=1}^{n-1}\left[\frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T}\right] + \det\left(\mathbf{H}_k^h\right)\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^{T\,2}} + \frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^T}\right] + \det\left(\mathbf{H}_k^f\right)\frac{\partial^2 \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^{T\,2}}\right],\\
\frac{\partial^2 c}{\partial \mathbf{p}_i^{T\,2}} &= 2\sum_{k=1}^{n-1}\left[\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^T}\right] + \det\left(\mathbf{H}_k^f\right)\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^{T\,2}}\right] - 2\left[\det(\mathbf{H}')\frac{\partial(\mathbf{q}^T\mathbf{q})}{\partial \mathbf{p}_i^T} + \mathbf{q}^T\mathbf{q}\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\right]\left[\mathbf{I}_n \otimes \frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\right]\\
&\quad - 2\mathbf{q}^T\mathbf{q}\det(\mathbf{H}')\frac{\partial^2 \det(\mathbf{H}')}{\partial \mathbf{p}_i^{T\,2}} - 2\det(\mathbf{H}')\frac{\partial \det(\mathbf{H}')}{\partial \mathbf{p}_i^T}\left[\mathbf{I}_n \otimes \frac{\partial(\mathbf{q}^T\mathbf{q})}{\partial \mathbf{p}_i^T}\right] - \det(\mathbf{H}')^2\frac{\partial^2(\mathbf{q}^T\mathbf{q})}{\partial \mathbf{p}_i^{T\,2}},\\
\frac{\partial^2 \det(\mathbf{H}')}{\partial \mathbf{p}_i^{T\,2}} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left\{\sum_{l''\in SS, l''\neq l'}\left[\prod_{l\in SS, l\neq l', l\neq l''} H'_{l,\sigma(l)}\frac{\partial H'_{l'',\sigma(l'')}}{\partial \mathbf{p}_i^T}\right]\left(\mathbf{I}_n \otimes \frac{\partial H'_{l',\sigma(l')}}{\partial \mathbf{p}_i^T}\right) + \prod_{l\in SS, l\neq l'} H'_{l,\sigma(l)}\frac{\partial^2 H'_{l',\sigma(l')}}{\partial \mathbf{p}_i^{T\,2}}\right\},\\
\frac{\partial^2 \det\left(\mathbf{H}_k^h\right)}{\partial \mathbf{p}_i^{T\,2}} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left\{\sum_{l''\in SS, l''\neq l'}\left[\prod_{l\in SS, l\neq l', l\neq l''} H^h_{k\,l,\sigma(l)}\frac{\partial H^h_{k\,l'',\sigma(l'')}}{\partial \mathbf{p}_i^T}\right]\left(\mathbf{I}_n \otimes \frac{\partial H^h_{k\,l',\sigma(l')}}{\partial \mathbf{p}_i^T}\right) + \prod_{l\in SS, l\neq l'} H^h_{k\,l,\sigma(l)}\frac{\partial^2 H^h_{k\,l',\sigma(l')}}{\partial \mathbf{p}_i^{T\,2}}\right\},\\
\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial \mathbf{p}_i^{T\,2}} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left\{\sum_{l''\in SS, l''\neq l'}\left[\prod_{l\in SS, l\neq l', l\neq l''} H^f_{k\,l,\sigma(l)}\frac{\partial H^f_{k\,l'',\sigma(l'')}}{\partial \mathbf{p}_i^T}\right]\left(\mathbf{I}_n \otimes \frac{\partial H^f_{k\,l',\sigma(l')}}{\partial \mathbf{p}_i^T}\right) + \prod_{l\in SS, l\neq l'} H^f_{k\,l,\sigma(l)}\frac{\partial^2 H^f_{k\,l',\sigma(l')}}{\partial \mathbf{p}_i^{T\,2}}\right\},\\
\frac{\partial^2 \mathbf{H}}{\partial \mathbf{p}_i^{T\,2}} &= \frac{2}{N}\left(\frac{1}{N} - 1\right)\{\mathbf{U}_{n,1}[vec(\mathbf{I}_n)^T \otimes \mathbf{I}_n](\mathbf{I}_n \otimes \mathbf{U}_{n,n}) + \mathbf{I}_n \otimes vec(\mathbf{I}_n)^T\}, \frac{\partial^2(\mathbf{q}^T\mathbf{q})}{\partial \mathbf{p}_i^{T\,2}} = \frac{2}{N}\left(\frac{1}{N} - 1\right)vec(\mathbf{I}_n)^T,\\
\frac{\partial^2 \mathbf{f}}{\partial \mathbf{p}_i^{T\,2}} &= \frac{\partial^2 \mathbf{a}}{\partial \mathbf{p}_i^{T\,2}} + \frac{\partial^2 \mathbf{B}}{\partial \mathbf{p}_i^{T\,2}}[\mathbf{I}_n \otimes (\mathbf{I}_n \otimes \mathbf{c})] + \frac{1}{N}\frac{\partial \mathbf{B}}{\partial \mathbf{p}_i^T}\mathbf{U}_{n,n}(\mathbf{I}_n \otimes \mathbf{I}_n)(\mathbf{I}_n \otimes \mathbf{U}_{1,n}) + \frac{1}{N}\frac{\partial \mathbf{B}}{\partial \mathbf{p}_i^T} + \frac{4}{N^2}[\mathbf{c}^T \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{c}^T + \mathbf{c}vec(\mathbf{I}_n)^T],\\
\frac{\partial^2 \mathbf{a}}{\partial \mathbf{p}_i^{T\,2}} &= \frac{2}{N}\left[\mathbf{p}_i^T \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{p}_i^T + \mathbf{p}_i vec(\mathbf{I}_n)^T\right],\\
\frac{\partial^2 \mathbf{B}}{\partial \mathbf{p}_i^{T\,2}} &= -\frac{2}{N}\{\mathbf{U}_{n,1}[vec(\mathbf{I}_n)^T \otimes \mathbf{I}_n](\mathbf{I}_n \otimes \mathbf{U}_{n,n}) + \mathbf{I}_n \otimes vec(\mathbf{I}_n)^T + vec(\mathbf{I}_n)^T \otimes \mathbf{I}_n\}.
\end{aligned}
\right\} \quad \text{(A3)}
$$

Here $\mathbf{U}_{k,l} = \sum_{i=1}^{k}\sum_{j=1}^{l}\mathbf{E}_{ij}^{(k\times l)} \otimes [\mathbf{E}_{ij}^{(k\times l)}]^T$, where $\mathbf{E}_{ij}^{(k\times l)}$ is the $k\times l$ matrix with the (i,j) element as 1 and all the other elements as zero.

$$
\left.
\begin{aligned}
\frac{\partial^2 \mathbf{p}_0}{\partial r_i^2} &= \frac{\partial^2 \mathbf{q}}{\partial r_i^2},\\
\frac{\partial^2 q_k}{\partial r_i^2} &= \frac{1}{\det(\mathbf{H}')}\left[\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial r_i^2} + \det\left(\mathbf{H}_k^h\right)\frac{\partial^2 q_n}{\partial r_i^2}\right], \quad \forall k \in \{1,\ldots,n-1\},\\
\frac{\partial^2 q_n}{\partial r_i^2} &= \frac{1}{2a}\left[-\frac{\partial^2 b}{\partial r_i^2} \pm \frac{1}{\sqrt{b^2-4ac}}\left(\frac{\partial b}{\partial r_i}^2 + b\frac{\partial^2 b}{\partial r_i^2} - 2a\frac{\partial^2 c}{\partial r_i^2}\right) \mp \frac{1}{\sqrt{b^2-4ac}^3}\left(b\frac{\partial b}{\partial r_i} - 2a\frac{\partial c}{\partial r_i}\right)^2\right],\\
\frac{\partial^2 b}{\partial r_i^2} &= 2\sum_{k=1}^{n-1}\left[\det\left(\mathbf{H}_k^h\right)\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial r_i^2}\right],\\
\frac{\partial^2 c}{\partial r_i^2} &= 2\sum_{k=1}^{n-1}\left[\frac{\partial \det\left(\mathbf{H}_k^f\right)}{\partial r_i}^2 + \det\left(\mathbf{H}_k^f\right)\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial r_i^2}\right] - \det\left(\mathbf{H}'\right)^2\frac{\partial^2(\mathbf{q}^T\mathbf{q})}{\partial r_i^2},\\
\frac{\partial^2 \det\left(\mathbf{H}_k^f\right)}{\partial r_i^2} &= \sum_{\sigma\in P_{ss}}\operatorname{sgn}(\sigma)\sum_{l'=1}^{n-1}\left\{\sum_{l''\in SS, l''\neq l'}\left[\prod_{l\in SS, l\neq l', l\neq l''} H_{k\,l,\sigma(l)}^f\frac{\partial H_{k\,l'',\sigma(l'')}^f}{\partial r_i}\right]\frac{\partial H_{k\,l',\sigma(l')}^f}{\partial r_i} + \prod_{l\in SS, l\neq l'} H_{k\,l,\sigma(l)}^f\frac{\partial^2 H^f_{k\,l',\sigma(l')}}{\partial r_i^2}\right\},\\
\frac{\partial^2 \mathbf{f}}{\partial r_i^2} &= \frac{\partial^2 \mathbf{a}}{\partial r_i^2} + \frac{\partial^2 \mathbf{B}}{\partial r_i^2}\mathbf{c}, \quad \frac{\partial^2 \mathbf{a}}{\partial r_i^2} = -\frac{2}{N}\mathbf{p}_i, \quad \frac{\partial^2 \mathbf{B}}{\partial r_i^2} = \frac{2}{N}\mathbf{I}_n, \quad \frac{\partial^2(\mathbf{q}^T\mathbf{q})}{\partial r_i^2} = \frac{2}{N}.
\end{aligned}
\right\} \tag{A4}
$$

Acknowledgments

Special thanks are given to Mr. Xionghui Lu and Mr. Xu Zhong for their assistance in the process of experimental data acquisition.

References

1. J. Borenstein, H. R. Everett, L. Feng and D. Wehe, "Mobile robot positioning: Sensors and techniques," *J. Robot. Syst.* **14**(4), 231–249 (1997).
2. "Multilateration," Wikimedia Foundation, Inc. [Online]. Available http://en.wikipedia.org/wiki/Multilateration.
3. A. El-Rabbany, *Introduction to GPS: The Global Positioning System* (Artech House, Norwood, MA, 2002).
4. V. Ashkenazi, D. Park and M. Dumville, "Robot positioning and the global navigation satellite system," *Ind. Robot* **27**(6), 419–426 (2000).
5. F. Folster and H. Rohling, "Data association and tracking for automotive radar networks," *IEEE Trans. Intell. Transp. Syst.* **6**(4), 370–377 (2005).
6. A. Ward, A. Jones and A. Hopper, "A new location technique for the active office," *IEEE Pers. Commun.* **4**(5), 42–47 (1997).
7. P. Bahl and V. N. Padmanabhan, "User Location and Tracking in an In-building Radio Network," *Microsoft Research Technical Report MSR-TR-99-12*, Redmond, WA, Microsoft Corporation (1999).
8. J. Hightower, G. Borriello and R. Want, "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength," *UW CSE Technical Report 2000-02-02* (University of Washington, Seattle, WA, 2000).
9. N. B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System," *Sixth ACM/IEEE International Conference on Mobile Computing and Networking*, Boston, MA (2000), pp. 32–43.
10. A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster, "The anatomy of a context-aware application," *Wirel. Netw.* **1**, 1–16 (2001).
11. F. Ahmad and M. G. Amin, "Noncoherent approach to through-the-wall radar localization," *IEEE Trans. Aerosp. Electron. Syst.* **42**(4), 1405–1419 (2006).
12. Y. Zhou, W. Liu and P. Huang, "Laser-Activated RFID-Based Indoor Localization System for Mobile Robots," *Proceedings of 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy (2007) pp. 4600–4605.
13. D. E. Manolakis, "Efficient solution and performance analysis of 3-D position estimation by trilateration," *IEEE Trans. Aerosp. Electron. Syst.* **32**(4), 1239–1248 (1996).
14. B. Fang, "Trilateration and extension to global positioning system navigation," *J. Guid.* **9**(6), 715–717 (1986).
15. J. Ziegert and C. D. Mize, "The laser ball bar: a new instrument for machine tool metrology," *Precis. Eng.* **16**(4), 259–267 (1994).
16. S. K. Rao, "Comments on "efficient solution and performance analysis of 3-D position estimation by trilateration," *IEEE Trans. Aerosp. Electron. Syst.* **34**(2), 681 (1998).
17. F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Trans. Robot.* **21**(1), 93–101 (2005).
18. I. D. Coope, "Reliable computation of the points of intersection of n spheres in R^n," *Aust. N.Z. Ind. Appl. Math. J.* **42**(E), C461–C477 (2000).
19. W. H. Foy, "Position-location solutions by Taylor-series estimation," *IEEE Trans. Aerosp. Electron. Syst.* **AES-12**(2), 187–194 (1976).
20. W. Navidi, W. S. Murphy Jr. and W. Hereman, "Statistical methods in surveying by trilateration," *Comput. Stat. Data Anal.* **27**, 209–217 (1998).
21. W. C. Hu and W. H. Tang, "Automated least-squares adjustment of triangulation-trilateration figures," *J. Surv. Eng.* 133–142 (2001).
22. M. Pent, M. A. Spirito and E. Turco, "Method for positioning GSM mobile stations using absolute time delay measurements," *Electron. Lett.* **33**(24), 2019–2020 (1997).
23. J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Hoboken, NJ (John Wiley & Sons, 2005).
24. S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *Science* **220**(4598), 671–680 (1983).

25. D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers* (World Scientific, Singapore, 1999).
26. Y. Zhou, "An efficient Least-Squares Trilateration Algorithm for Mobile Robot Localization," *Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (St. Louis, MO, 2009).
27. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (The Johns Hopkins University Press, Baltimore, MD, 1996).
28. E. Kreyszig, *Advanced Engineering Mathematics*, 8th ed. (John Wiley & Sons, New York, 1999).
29. A. Weinmann, *Uncertain Models and Robust Control* (Springer-Verlag, Wien, New York, 1991).
30. J. Urena, A. Hernandez, A. Jimenez, J. M. Villadangos, M. Mazo, J. C. Garcia, J. J. Garcia, F. J. Alvarez, C. De Marziani, M. C. Perez, J. A. Jimenez, A. R. Jimenez and F. Seco, "Advanced sensorial system for an acoustic LPS," *Microprocess. Microsyst.* **31**, 393–401 (2007).
31. X. Cheng, H. Shu, Q. Liang and D. H. Du, "Silent positioning in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.* **57**(3), 1756–1766 (2008).