The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

An Efficient Least-Squares Trilateration Algorithm for Mobile Robot Localization

Yu Zhou, *Member, IEEE*

***Abstract* — A novel efficient trilateration algorithm is presented to estimate the position of a target object, such as a mobile robot, in a 2D or 3D space. The proposed algorithm is derived from a nonlinear least-squares formulation, and provides an optimal position estimate from a number (greater than or equal to the dimension of the environment) of reference points and corresponding distance measurements. Using standard linear algebra techniques, the proposed algorithm has low computational complexity and high operational robustness. Error analysis has been conducted through simulations on representative examples. The results show that the proposed algorithm has lower systematic error and uncertainty in position estimation when dealing with erroneous inputs, compared with representative closed-form methods.**

I. Introduction

This paper presents a novel efficient trilateration algorithm which facilitates the self-localization of autonomous mobile robots in 2D and 3D environments.

A. Trilateration Principle

Trilateration refers to positioning an object based on the measured distances between the object and multiple reference points at known positions [1,2]. (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 [3].)

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

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

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 $\mathbf{p}_i$. Equation (1) represents a circle in $\mathbb{R}^2$ or a sphere in $\mathbb{R}^3$, centered at $\mathbf{p}_i$ with a radius of r_i. Solving a system of (1) is equivalent to finding the intersection point/points of a set of circles in $\mathbb{R}^2$ or spheres in $\mathbb{R}^3$.

In reality, trilateration 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 [4]. 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. Thus, it is necessary to determine a "best approximation".

B. Review of Existing Algorithms

Though straightforward in concept, the trilateration problem is far from trivial to solve, due to the nonlinearity of (1) and the errors in $\mathbf{p}_i$ and r_i. A number of algorithms have been proposed to solve the trilateration problem, including both closed-form and numerical solutions.

To determine the 3D position of an object based on the distance measurements from three reference points, Fang provided a closed-form solution by referencing to the base plane defined by the three reference points [5]. A similar formulation was presented by Ziegert and Mize [6]. Independent of the choice of any particular frame of reference, Manolakis derived a more general closed-form solution [4]. 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 [4] were fixed by Rao [7]. Recently Thomas and Ros proposed an alternative closed-form solution using Cayley-Menger determinants which are related to the geometry of the tetrahedra formed by the object and three reference points [2]. In a more general context, Coope presented a closed-form solution for determining the intersection points of n spheres in $\mathbb{R}^n$ based on Gaussian elimination [8].

In general, closed-form solutions have low computational complexity when the solution of (1) exists. They also facilitate the theoretical analysis of the algorithm performance [2,4]. However, in general closed-form solutions do not accommodate the situation that the involved spheres (circles) do not intersect at one point, i.e. no solution exists for (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 a "best approximation" which minimizes the residuals of (1) in some appropriate form. Numerical methods are in general necessary in order to provide such an estimate, as indicated in [2,8].

Manuscript received on March 1, 2009.

Yu Zhou is with the Department of Mechanical Engineering, State University of New York at Stony Brook, Stony Brook, NY11794, USA (phone: 631-632-8322; fax: 631-632-8544; e-mail: yuzhou@notes.cc.sunysb.edu).

Foy 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 [9]. Incorporating the distance measurement errors, Nadivi et al. compared three statistical methods, 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 [10]. Hu and Tang 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 [11]. Coope 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 [8]. In another work, Pent et al. defined a probabilistic model of the distance measurement error and used the extended Kalman filtering to solve the trilateration problem [12].

In general, numerical methods are available to provide an optimal estimation of the position of a target object, in particular when no solution exists for (1). Moreover, numerical methods are in general not limited to dealing with n spheres in $\mathbb{R}^n$. In fact, more accurate position estimate 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 linearize the trilateration problem [9,10,12], which introduces extra errors into position estimation. Many numerical methods involve a searching process, such as Newton's method and the steepest descent method, which iteratively improves an initial guess towards a converged position estimate [8-10]. However, most of these search algorithms are sensitive to the choice of the initial guess, and a global convergence towards the desirable position estimate is not guaranteed.

C. *Overview of the Proposed Algorithm*

Addressing the above-listed issues of existing tirlateration algorithms with an emphasis on mobile robotics, we propose an alternative algorithm which estimates the position of a target object, such as a mobile robot, based on the simultaneous distance measurements from multiple reference points, by solving a nonlinear least-squares formulation of trilateration using standard linear algebra techniques. The proposed algorithm provides an optimal position estimate of the intersection point of N≥n spheres in $\mathbb{R}^n$ (where n=2 or 3), which is not limited to solving for the intersection points of exact n spheres in $\mathbb{R}^n$. The proposed algorithm does not depend on the techniques which tend to be affected by algebraic singularities, such as matrix inversion, and hence has high operational robustness. Though not in closed form, the proposed algorithm has a low computational complexity.

The layout of this paper is as follows. In Section II, we will derive and explain the proposed algorithm in detail. In Section III, we will analyze the performance of the proposed algorithm through simulations with representative examples. Section IV will summarize this work.

II. Proposed Trilateration Algorithm

A. *Nonlinear Least-Squares Formulation*

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

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

where $S(\mathbf{p}_0) = \sum_{i=1}^{N} [(\mathbf{p}_i - \mathbf{p}_0)^T (\mathbf{p}_i - \mathbf{p}_0) - r_i^2]^2$, $\mathbf{p}_0$ denotes an estimate of the robot position, $\mathbf{p}_i$ the pre-mapped 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≥n (and $N \in \mathbb{Z}^+$).

Equation (2) presents a nonlinear optimization problem. Search-based optimization algorithms are commonly used to solve this category of problems, including both local optimization algorithms, e.g. the steepest descent method and the Newton-Raphson method, and global optimization algorithms, e.g. the simulated annealing and the genetic algorithm [13]. The steepest descent method and the Newton-Raphson method in general converge to a local minimum in the vicinity of the initial guess, and the global optimization depends on the choice of the initial guess. Meanwhile, for the method of simulated annealing and the genetic algorithm, in order to reach the global minimum, the various algorithm parameters and decision criteria of these methods need to be tuned to fit with the specific problem. The lack of general, systematic methods for a mobile robot to automatically generate the initial guess and choose algorithm-specific parameters and criteria onboard to guarantee the global convergence in position estimation causes inconvenience in using these search-based algorithms. In addition, the relatively high time complexity of the simulated annealing method and the genetic algorithm makes them not suitable for real-time applications.

We here derive an algorithm to solve the least-squares formulation of trilateration in (2) using standard linear algebra techniques, which guarantees globally optimal position estimates and has low computational complexity.

B. Derivation

We notice that, given $\mathbf{p}_i$ and r_i, solving (2) 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 $\mathbf{a} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{p}_i\mathbf{p}_i^T\mathbf{p}_i - r_i^2\mathbf{p}_i)$,

$\mathbf{B} = \frac{1}{N}\sum_{i=1}^{N}[-2\mathbf{p}_i\mathbf{p}_i^T - (\mathbf{p}_i^T\mathbf{p}_i)\mathbf{I} + r_i^2\mathbf{I}]$ and $\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i$.

To simplify (3), we introduce a linear transform

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

and obtain an equation containing no quadratic term of $\mathbf{q}$

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

Define $\mathbf{f} = \mathbf{a} + \mathbf{B}\mathbf{c} + 2\mathbf{c}\mathbf{c}^T\mathbf{c}$ and $\mathbf{D} = \mathbf{B} + 2\mathbf{c}\mathbf{c}^T + (\mathbf{c}^T\mathbf{c})\mathbf{I}$, we rewrite (5) as

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

Moreover, we notice that in fact

$$\mathbf{D} - (\mathbf{q}^T\mathbf{q})\mathbf{I} = -\frac{2}{N}\sum_{i=1}^{N}\mathbf{p}_i\mathbf{p}_i^T + 2\mathbf{c}\mathbf{c}^T, \tag{7}$$

which is an n×n symmetric matrix and does not contain $\mathbf{q}$. Defining $\mathbf{H} = \mathbf{D} - (\mathbf{q}^T\mathbf{q})\mathbf{I}$, we obtain from (6)

$$\mathbf{f} + \mathbf{H}\mathbf{q} = \mathbf{0}. \tag{8}$$

Equation (8) is a linear system of n equations of the unknown n-dimensional vector $\mathbf{q}$. If $\mathbf{H}$ is full-rank (invertible), $\mathbf{q}$ can be calculated easily as $\mathbf{q} = -\mathbf{H}^{-1}\mathbf{f}$ or using numerical methods such as Gaussian elimination [14]. However, it may happen that $\mathbf{H}$ is not full-rank. In fact, without making more general proof, 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 $\mathbf{p}_i$ are at the same "height", i.e. with the same value of x, y or z, $\mathbf{H}$ has a zero row and a zero column and hence a rank of n-1. In these cases, (8) does not represent a system of n independent linear equations, and hence we cannot uniquely determine $\mathbf{q}$ from only (8). Instead, additional constraints need to be found to construct a new system of n independent equations so that the specific solution can be obtained.

Here we propose a unified solution procedure for rank($\mathbf{H}$)=n and n-1. First, denoting the kth component of $\mathbf{f}$ as f_k and the kth row of $\mathbf{H}$ as $\mathbf{h}_k^T$, we construct a n-1 dimensional vector $\mathbf{f}'=[f_1- f_n, \ldots, f_{n-1}- f_n]^T$ and a (n-1)×n matrix $\mathbf{H}'=[\mathbf{h}_1-\mathbf{h}_n, \ldots, \mathbf{h}_{n-1}-\mathbf{h}_n]^T$, and obtain from (8)

$$\mathbf{f}' + \mathbf{H}'\mathbf{q} = \mathbf{0}. \tag{9}$$

Next, using orthogonal decomposition [14], we obtain

$$\mathbf{H}' = \mathbf{Q}\mathbf{U}, \tag{10}$$

where $\mathbf{Q}$ is a (n-1)×(n-1) orthogonal matrix and $\mathbf{U}$ a (n-1)×n upper diagonal matrix. Then, pre-multiplying the both sides of (9) by $\mathbf{Q}^T$, we obtain

$$\mathbf{Q}^T\mathbf{f}' + \mathbf{U}\mathbf{q} = \mathbf{0}. \tag{11}$$

Rewriting (11) in its scalar form, we have for the 3D case

$$\begin{cases} v_1 + u_{11}q_1 + u_{12}q_2 + u_{13}q_3 = 0 \\ v_2 + u_{22}q_2 + u_{23}q_3 = 0 \end{cases}, \tag{12}$$

where v_k denotes the kth component of $\mathbf{Q}^T\mathbf{f}'$, u_{kj} the (k,j) entry of $\mathbf{U}$, and q_k the kth component of $\mathbf{q}$. From (12), we obtain

$$\begin{cases} q_1 = (\frac{u_{12}v_2}{u_{11}u_{22}} - \frac{v_1}{u_{11}}) + (\frac{u_{12}u_{23}}{u_{11}u_{22}} - \frac{u_{13}}{u_{11}})q_3 \\ q_2 = -\frac{v_2}{u_{22}} - \frac{u_{23}}{u_{22}}q_3 \end{cases}. \tag{13}$$

Now we have 3 unknowns but 2 equations, and therefore need one more independent equation to solve for $\mathbf{q}$. In fact, one valid constraint is

$$q_1^2 + q_2^2 + q_3^2 = \mathbf{q}^T\mathbf{q}, \tag{14}$$

where $\mathbf{q}^T\mathbf{q}$ can be obtained from $\mathbf{D} - \mathbf{q}^T\mathbf{q} = \mathbf{H}$ 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 (13) into (14), we obtain

$$[(\frac{u_{12}v_2}{u_{11}u_{22}} - \frac{v_1}{u_{11}}) + (\frac{u_{12}u_{23}}{u_{11}u_{22}} - \frac{u_{13}}{u_{11}})q_3]^2 + [\frac{v_2}{u_{22}} + \frac{u_{23}}{u_{22}}q_3]^2 + q_3^2 = \mathbf{q}^T\mathbf{q}, \tag{16}$$

which is a quadratic equation of q_3 and can be solved in closed form. Substituting the resulting q_3 into (13), we can obtain q_1 and q_2. Similarly, for the 2D case, we obtain from (11)

$$q_1 = -\frac{v_1}{u_{11}} - \frac{u_{12}}{u_{11}}q_2. \tag{17}$$

In the 2D case, the constraint (14) becomes

$$q_1^2 + q_2^2 = \mathbf{q}^T\mathbf{q}. \tag{18}$$

Substituting (17) into (18), we obtain

$$[\frac{v_1}{u_{11}} + \frac{u_{12}}{u_{11}}q_2]^2 + q_2^2 = \mathbf{q}^T\mathbf{q}, \tag{19}$$

which is a quadratic equation of q_2 and can be solved in closed form. Substituting the resulting q_2 into (17), we can obtain q_1.

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

The above process generally results in two candidates of $\mathbf{p}_0$, due to the duality of (16) and (19). However, only one of the two candidates is the true $\mathbf{p}_0$. 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 last $\mathbf{p}_0$.

C. Summary

Following the above derivation, the proposed trilateration algorithm is summarized as Algorithm 1.

As indicated in the derivation, the proposed trilateration algorithm provides an optimal estimation of the location of a mobile robot 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 has low computational complexity. Without depending on the techniques which tend to be affected by

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

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

Output: $\mathbf{p}_0$.

1) Calculate $\mathbf{a}$, $\mathbf{B}$, $\mathbf{c}$, $\mathbf{f}$, $\mathbf{f}'$, $\mathbf{H}$, $\mathbf{H}'$, $\mathbf{Q}$ and $\mathbf{U}$.
2) Calculate $\mathbf{q}^T\mathbf{q}$ from (15).
3) For 3D trilateration, calculate q_3 from (16); for 2D trilateration, calculate q_2 from (19).
4) For 3D trilateration, calculate q_1 and q_2 from (13); for 2D trilateration, calculate q_1 from (17).
5) Calculate $\mathbf{p}_0$ from (4).
6) Choose one of the two candidates of $\mathbf{p}_0$.
7) Return $\mathbf{p}_0$.

algebraic singularities, such as matrix inversion, the proposed algorithm also has high operational robustness.

III. Error Analysis

The input to the algorithm includes the mapped positions of the reference points, $\mathbf{p}_i$, and the measured distances between the robot and reference points, r_i. In practice, errors arise in $\mathbf{p}_i$ due to inaccurate mapping of the reference points, which happens in both manual and robotic mapping processes; 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 robot position $\mathbf{p}_0$.

A. Performance Indices

We define $\mathbf{p}=\begin{bmatrix}\cdots & \mathbf{p}_i^T & \cdots\end{bmatrix}^T$, $\mathbf{r}=\begin{bmatrix}\cdots & r_i & \cdots\end{bmatrix}^T$ and $\mathbf{x}=\begin{bmatrix}\mathbf{p}^T & \mathbf{r}^T\end{bmatrix}^T$. Denoting the actual value and random error of the measurement $\mathbf{x}$ as $\bar{\mathbf{x}}$ and $\delta\mathbf{x}$ respectively, assuming that the input errors are zero-mean random errors, i.e. $E(\delta\mathbf{x})=\mathbf{0}$, and following a similar derivation as [4], we can obtain the mean vector $E(\delta\mathbf{p}_0)$ 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. Equations (20) and (22) show that the mean and variance of the output error are directly related to the variance of the input error. In particular, to evaluate the impact of the error of $\mathbf{p}$, $\delta\mathbf{p}$, on $\delta\mathbf{p}_0$, we assume that the components of $\delta\mathbf{p}$ are zero-mean random variables and uncorrelated from one another with the same standard deviation σ_p for each coordinate, and, similar to [2,4], 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

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

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

where $|\mathbf{v}|$ 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 the error of $\mathbf{r}$, $\delta\mathbf{r}$, on $\delta\mathbf{p}_0$, we assume that the components of $\delta\mathbf{r}$ are zero-mean random variables and uncorrelated from one another with the same standard deviation σ_r, and define two performance indices correspondingly as

$$B_r=\frac{|E[\delta\mathbf{p}_0(\mathbf{x})]|}{\sigma_r^2}=\frac{1}{2}\left|\frac{\partial^2\mathbf{p}_0}{\partial\mathbf{r}^{T^2}}vec(\mathbf{I})\right|, \tag{24}$$

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

We notice that B_r and S_r are independent of σ_r. We have tested B_p, S_p, B_r and S_r through simulations.

The proposed trilateration algorithm provides an optimal approximation of the intersection point of $N\ge2$ circles in $\mathbb{R}^2$ and $N\ge3$ spheres in $\mathbb{R}^3$. Without loss of generality, our error analysis focuses on $\mathbb{R}^3$. A similar trend can be found for $\mathbb{R}^2$. Through representative examples, we test the proposed algorithm with 3 reference points at first and then with 4 reference points. The proposed trilateration algorithm has been programmed in Matlab. Tested on a Dell Latitude D620 laptop computer with a 1.66 GHz Intel Core 2 CPU, the average running time for the algorithm is <0.0006 second. This means that the proposed algorithm is highly suitable for real-time trilateration tasks.

B. Trilateration in $\mathbb{R}^3$ with 3 Reference Points

Following the representative examples in [2,4], we examine a 3-reference case in $\mathbb{R}^3$ in which the XY coordinates of the 3 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 assume that a mobile robot moves across a square data acquisition region defined as $\{[x,y,z]^T \mid z=8000, -4000\le x,y\le4000\}$.

To evaluate the impact of $\delta\mathbf{p}$ on $\delta\mathbf{p}_0$, we set σ_p with different values ($\sigma_p\in\{10,20,30,40,50,60,70,80,90,100\}$), run the simulation with 10000 samples for each σ_p, and calculate B_p and S_p across the above data acquisition region. The resulting variations of B_p and S_p are consistent across the range of σ_p, as indicated by (22) and (23). Figures 1 and 2 show the variation of B_p and S_p for a representative σ_p=70. We observe that both B_p and S_p increases as the robot moves away from the center of the base triangle defined by the reference points. In particular, at the center of the data acquisition region where $\mathbf{p}_0=[0,0,8000]^T$, we obtain

B_p=0.0057 and S_p=9.36; while at the edge of the data acquisition region where $\mathbf{p}_0$=[-4000,4000,8000]^T, we obtain B_p=0.0107 and S_p=12.87. Compared with the results reported in [2,4] which were generated from the exactly same simulation setting, our algorithm has lower S_p values ([2] 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 means that the proposed trilateration algorithm has a reduced uncertainty in position estimation when using imperfectly mapped reference points.

To evaluate the impact of $\delta\mathbf{r}$ on $\delta\mathbf{p}_0$, we set σ_r with different values ($\sigma_r \in \{10,20,30,40,50,60,70,80,90,100\}$), run

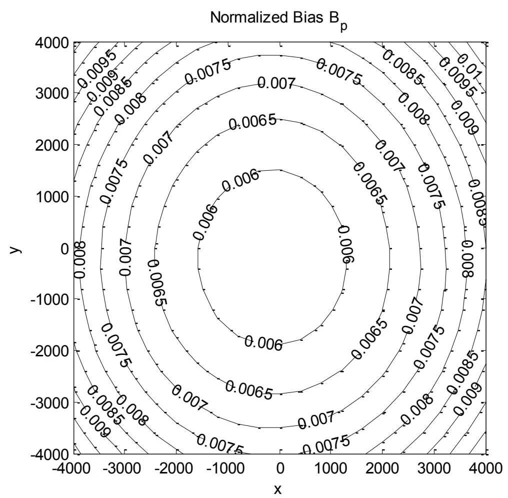

Fig.1. Normalized total bias B_p obtained from the 3-reference example

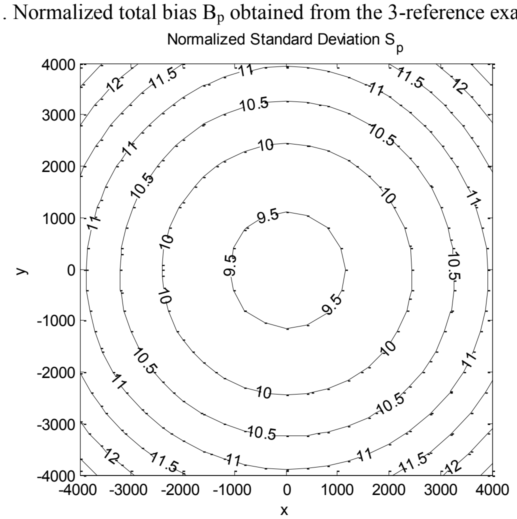

Fig.2. Normalized total standard deviation S_p obtained from the 3-reference example

the simulation with 10000 samples for each σ_r, and calculate B_r and S_r across the above data acquisition region. The resulting variations of B_r and S_r are consistent across the range of σ_r, as indicated by (24) and (25). Moreover, the simulation results show that B_r and S_r have similar trends of variation to those of B_p and S_p respectively, and their values are very close to B_p and S_p respectively. In particular, for σ_r=70, at $\mathbf{p}_0$=[0,0,8000]^T, we obtain B_r=0.0054 and S_r=9.31; while at $\mathbf{p}_0$=[-4000,4000,8000]^T, we obtain B_r=0.0111 and S_r=12.78. They are very close to the values of B_p and S_p at the same points. For this reason, we do not present the figures for B_r and S_r specifically. Compared with the results reported in [2,4] which were generated from the exactly same simulation setting, our algorithm has significant lower B_r values ([2] reports that the maximum B_r on the edge of the same data acquisition region is about 0.03 while our result is 0.0111.), which means that the proposed trilateration algorithm has a reduced systematic error in position estimation when using erroneous distance measurements.

C. Trilateration in $\mathbb{R}^3$ with 4 Reference Points

To test the performance of the proposed trilateration algorithm with N>n reference points, we examine a 4-reference case in $\mathbb{R}^3$ in which the XY coordinates of the 4 reference points form a square inscribed in the same circle as in the 3-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 $\mathbf{p}_4 = [500\sqrt{2} \quad -500\sqrt{2} \quad 0]^T$. Same as the 3-reference example, we also assume that the mobile robot moves across a square data acquisition region defined as $\{[x,y,z]^T | z=8000, -4000 \le x,y \le 4000\}$.

Figures 3 and 4 show the variations of B_p and S_p (taken at σ_p=70). Similar to the 3-referecne case, both B_p and S_p increases as the robot moves away from the center of base square defined by the reference points. In particular, at the center of the data acquisition region where $\mathbf{p}_0$=[0,0,8000]^T, we obtain B_p=0.0077 and S_p=8.07; while at the edge of the data acquisition region where $\mathbf{p}_0$=[-4000,4000,8000]^T, we obtain B_p=0.0090 and S_p=11.04. Compared with the 3-reference case, we notice that there is an increase in the minimum B_p due to the addition of another imperfectly mapped reference point. However, the maximum B_p decreases. Moreover, S_p becomes lower, which means that the uncertainty in position estimation, when using imperfectly mapped reference points, will decrease by referring to more reference points.

The variation of S_r is very close to that of S_p. In particular, for σ_r=70, at $\mathbf{p}_0$=[0,0,8000]^T, we obtain S_r=8.13; while at $\mathbf{p}_0$=[-4000,4000,8000]^T, we obtain S_r=11.10. For this reason, we do not present the figures for S_r specifically. However, the variation of B_r (Fig.5) is significantly different from that of B_p. In particular, for σ_r=70, at $\mathbf{p}_0$=[0,0,8000]^T, we obtain B_r=0.0040; at $\mathbf{p}_0$=[-4000,4000,8000]^T, we obtain B_r=0.0077.

Compared with those of the 3-reference case, both B_r and S_r are significantly lower, 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.

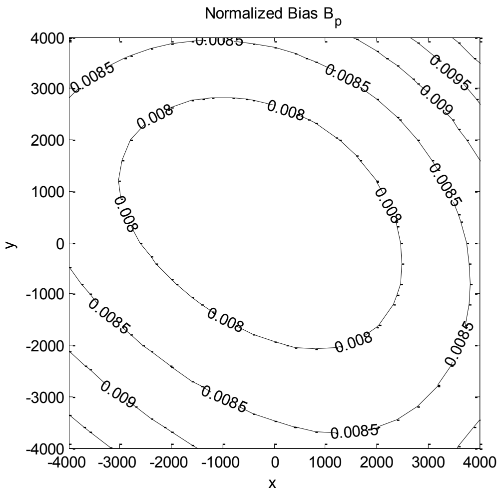

Fig.3. Normalized total bias B_p obtained from the 4-reference example

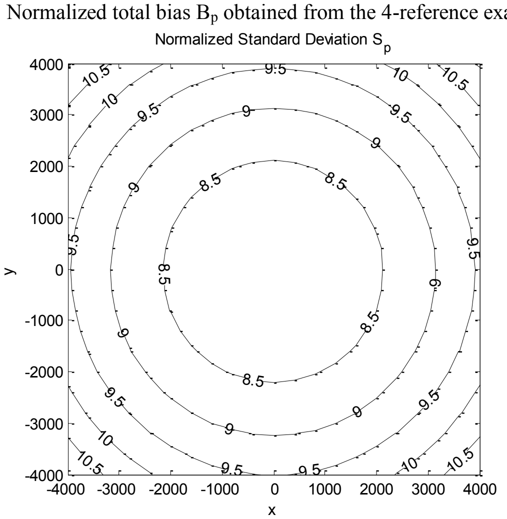

Fig.4. Normalized total standard deviation S_p obtained from the 4-reference example

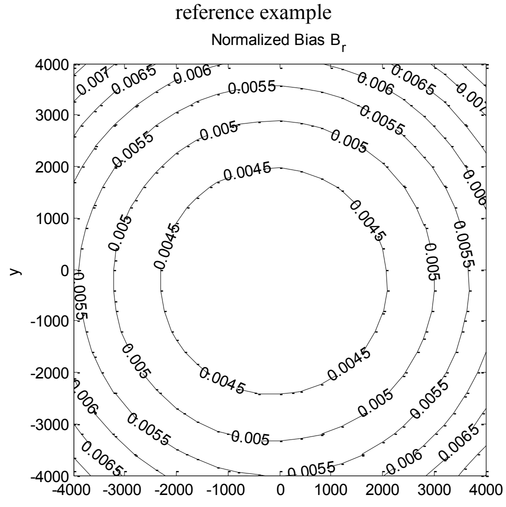

Fig.5. Normalized total bias B_r obtained from the 4-reference example

IV. CONCLUSION

This paper presents an efficient 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. Solving the nonlinear least-squares formulation of trilateration, the proposed algorithm provides an 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 exact n spheres in $\mathbb{R}^n$. Using standard linear algebra techniques, the proposed algorithm, though not in the closed form, has low computational complexity and is highly applicable to real-time applications. 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 simulation results show 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 simulations also show that introducing more reference points and corresponding distance measurements into the trilateration process will in general reduce the estimation uncertainty. Though targeting the applications in mobile robotics, it is our belief that the proposed trilateration algorithm is applicable to any ranging-based object localization tasks in various environments and scenarios.

REFERENCES

[1] J. Borenstein, H. R. Everett, L. Feng, D. Wehe, "Mobile robot positioning: sensors and techniques", *Journal of Robotic Systems*, 14(4): 231-249, 1997.

[2] F. Thomas, L. Ros, "Revisiting trilateration for robot localization", *IEEE Transactions on Robotics*, 21(1): 93-101, 2005.

[3] "Multilateration", http://en.wikipedia.org/wiki/Multilateration.

[4] D. E. Manolakis, "Efficient solution and performance analysis of 3-D position estimation by trilateration", *IEEE Transactions on Aerospace and Electronic Systems*, 32(4): 1239-1248, 1996.

[5] B. Fang, "Trilateration and extension to global positioning system navigation", *Journal of Guidance*, 9(6): 715-717, 1986.

[6] J. Ziegert, C. D. Mize, "The laser ball bar: a new instrument for machine tool metrology", *Precision Engineering*, 16(4): 259-267.

[7] S. K. Rao, "Comments on "efficient solution and performance analysis of 3-D position estimation by trilateration"", *IEEE Transactions on Aerospace and Electronic Systems*, 34(2): 681, 1998.

[8] I. D. Coope, "Reliable computation of the points of intersection of n spheres in R^n*", The Australian & New Zealand Industrial and Applied Mathematics Journal*, 42(E): C461–C477, 2000.

[9] W. H. Foy, "Position-location solutions by Taylor-series estimation", *IEEE Transactions on Aerospace and Electronics Systems*, AES-12(2): 187-194, 1976.

[10] W. Navidi, W. S. Murphy Jr., W. Hereman, "Statistical methods in surveying by trilateration", *Computational Statistics & Data Analysis*, 27: 209-217, 1998.

[11] W. C. Hu, W. H. Tang, "Automated least-squares adjustment of triangulation-trilateration figures", *Journal of Surveying Engineering*, 133-142, 2001.

[12] M. Pent, M. A. Spirito, E. Turco, "Method for positioning GSM mobile stations using absolute time delay measurements", *Electronics Letters*, 33(24): 2019-2020, 1997.

[13] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, John Wiley & Sons, 2005.

[14] G. H. Golub, C. F. Van Loan, *Matrix Computations, 3rd Edition*, The Johns Hopkins University Press, 1996.