A Novel Subspace Approach for Cooperative Localization in Wireless Sensor Networks Using Range Measurements

Frankie K. W. Chan, H. C. So, *Senior Member, IEEE*, and W.-K. Ma, *Member, IEEE*

***Abstract*—Estimating the positions of sensor nodes is a fundamental and crucial problem in wireless sensor networks. In this paper, three novel subspace methods for node localization in a fully connected network are devised with the use of range measurements. Biases and mean square errors of the sensor node position estimates are also derived. Computer simulations are included to contrast the performance of the proposed algorithms with the conventional subspace positioning method, namely, classical multidimensional scaling, as well as Cramér-Rao lower bound.**

***Index Terms*—Position estimation, range-based measurements, subspace method, wireless sensor networks.**

I. Introduction

RECENT technological advances in wireless communications and microsystem integration have enabled the development of small, inexpensive, low-power sensor nodes which are able to collect surrounding data, perform small-scale computations and communicate among their neighbors. These wirelessly connected nodes, when working in a collaborative manner, have great potential in numerous remote monitoring and control applications [1]–[3] such as asset management, habitat monitoring, health caring, building automation, battlefield surveillance as well as environment observation and forecasting. Since sensor nodes are often arbitrarily placed with their positions being unknown, sensor positioning is a fundamental and crucial issue for the wireless sensor network (WSN) operation and management.

Node localization methods can be generally classified as the deterministic [4]–[17] and probabilistic approaches [18], [19]. The simplest deterministic technique is to exploit the connectivity information—who is within the communication range of whom—to derive the node positions with the use of the anchor nodes subject to the proximity constraints imposed by the known connections, but it only provides coarse-grain location estimates. Mathematically, this can be formulated as a linear programming or semi-definite programming problem [4]. Apart from connectivity, range-based schemes utilize node-to-node or hop distances and/or angles, which are obtained from the pair-wise time-of-arrival (TOA), time-difference-of-arrival, received signal strength (RSS) and/or angle-of-arrival measurements, for sensor positioning with higher location accuracy, although it is possible to use the average hop length and hop counts between indirectly connected nodes to deduce distance information [5] as well. Assuming that the range measurements errors are Gaussian distributed, the maximum likelihood (ML) methods for node localization correspond to the nonlinear least squares problem [6]–[9]. In spite of attaining optimum estimation performance, the ML approach requires centralized data processing and there is no guarantee of global convergence. In order to ensure a global solution, semi-definite programming (SDP) relaxation to the ML formulation has been proposed [10]–[12]. Alternatively, the range-based measurements can also be converted into linear equations where the node positions are easily solved even in a distributed manner but at the expense of error accumulation [13], [14]. Another range-based positioning technique which ensures global convergence is to employ classical multidimensional scaling (MDS) [15]–[17] which transforms the pair-wise distance information into the relative coordinates of nodes. On the other hand, particle filtering [18], [19] is a representative example of the probabilistic approach, where each sensor stores a conditional density on its own coordinates based on its measurements and the conditional density of its neighbors for node localization, has a high potential of tracking purposes at the cost of significant computational requirements. Nevertheless, relatively less research has been performed in deriving the theoretical performance of the node localization methods, although location performance limits are developed in [20] and [21].

In this paper, a novel subspace approach for WSN positioning, which belongs to the deterministic category with the use of the node-to-node distance estimates deduced from the RSS or TOA measurements, is developed. We assume a fully connected WSN, which is a frequently used assumption, say in the classical multidimensional scaling (MDS) [15]–[17] for localization where distance measurements between all pairs of sensors are required. For a partially connected WSN, there exist techniques for estimating the missing distance information such as shortest-path distance computation [15] and reconstruction based on singular value decomposition [16].

Manuscript received December 17, 2007; revised August 07, 2008. Current version published January 06, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhengyuan (Daniel) Xu. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 119606).

F. K. W. Chan and H. C. So are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong.

W.-K. Ma is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

Digital Object Identifier 10.1109/TSP.2008.2005870

TABLE I
LIST OF SYMBOLS

Symbol	Meaning
$[\mathbf{A}]_{i,j}$	(i,j) entry of matrix $\mathbf{A}$
$\mathbf{A}^T$	transpose of $\mathbf{A}$
$\mathbf{A}^\dagger$	$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$
rank$(\mathbf{A})$	rank of $\mathbf{A}$
vec$(\mathbf{A})$	vectorization of $\mathbf{A}$
$\mathbb{E}\{\mathbf{A}\}$	expectation of $\mathbf{A}$
$\hat{\mathbf{A}}$	estimate of $\mathbf{A}$
$\tilde{\mathbf{A}}$	noise-free $\mathbf{A}$
$\boldsymbol{\Delta}\mathbf{A}$	perturbation of $\mathbf{A}$
span$(\mathbf{A})$	space spanned by the columns of $\mathbf{A}$
$\mathbf{1}_i$	$i \times 1$ vector with all elements 1
$\mathbf{0}_{i\times j}$	$i \times j$ zero matrix
max(i,j)	larger element between i and j
$\mathbf{I}$	identity matrix
$\mathbf{J}$	$[\begin{matrix}1 & 0\end{matrix}]^T$
$\mathbf{e}_{i,j}$	jth column of $\mathbf{I}_i$
sgn	sign function
$\otimes$	Kronecker matrix product
$\sigma^2_{i,j}$	variance of distance measurement between the ith and jth sensors
diag$(a_1, a_2, \cdots, a_k)$	diagonal matrix with diagonal elements $a_1, a_2, \cdots, a_k$

Alternatively, local maps can first be built from the fully connected subsets in the WSN and then patched together to form the global map [17]. This paper focuses on the algorithm development, as well as its performance analysis, and we are targeting applications where a network can be divided into clusters of fully connected subnetworks and each cluster is then handled by the proposed approach. Our major contributions are (i) A new cooperative localization [21] approach for WSNs is devised and its relationship with the classical MDS method is given. Note that cooperative localization is not a well solved problem because the distance measurements between any pairs of sensors, including those of the unknown-location sensors, are utilized to aid in the location estimation. This is much more challenging than the traditional multilateration where only distance measurements between an unknown-location sensor and anchors are employed for localization. From the simulation results, it is observed that the proposed approach can outperform the classical MDS algorithm; and (ii) The performance measures of the proposed methods, namely, biases and mean square errors of the proposed algorithms, are produced. To the best of our knowledge, performance analysis of the classical MDS technique for node localization is not yet available.

The rest of the paper is organized as follows. The development of three versions of our WSN node positioning approach, namely, full-set subspace algorithm, minimum-set subspace algorithm and distributed subspace algorithm, are presented in Section II. The computational complexity of the three subspace algorithms is investigated in Section III. In Section IV, the theoretical performance of the proposed methods is studied. Numerical examples are presented in Section V to corroborate the analytical development and to evaluate the performance of the proposed approach by comparing the classical MDS method, as well as Cramér-Rao lower bound (CRLB) [21]. Finally, concluding remarks are given in Section VI. A list of symbols that are used in the paper is given in Table I.

II. Subspace Positioning Approach

Consider a fully connected network of M sensors in a two-dimensional space (extension to three-dimensional space is straightforward). Let $\mathbf{s}_i = [x_i, y_i]^T$ where T denotes transpose, $i = 1, 2, \cdots, M$, be the position of the ith sensor. Without loss of generality, we assume that the positions of the first k of them, $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_k$, where $k \geq 3$, are known. The task is to find $\mathbf{s}_i$, $i = k+1, k+2, \cdots, M$. In the following, three variants of the proposed subspace approach, namely, the full-set subspace algorithm, minimum-set subspace algorithm as well as distributed subspace algorithm are derived.

A. Full-Set Subspace Algorithm

Let $\mathbf{X}_{\text{fs}} \in \mathbb{R}^{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)\times 2}$ where ${}_m\mathrm{C}_n = m!/[n!(m-n)!]$, contains all the differences of $x_i - x_j$ and $y_i - y_j$, $i > j$, with at least one of the $\mathbf{s}_i$ and $\mathbf{s}_j$ being unknown. The matrix $\mathbf{X}_{\text{fs}}$ has the form:

$$\mathbf{X}_{\text{fs}} = \begin{bmatrix} x_{k+1}\mathbf{1}_k - \mathbf{x}_k & y_{k+1}\mathbf{1}_k - \mathbf{y}_k \\ x_{k+2}\mathbf{1}_{k+1} - \mathbf{x}_{k+1} & y_{k+2}\mathbf{1}_{k+1} - \mathbf{y}_{k+1} \\ \vdots & \vdots \\ x_M\mathbf{1}_{M-1} - \mathbf{x}_{M-1} & y_M\mathbf{1}_{M-1} - \mathbf{y}_{M-1} \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} \mathbf{x}_i &= [x_1 \quad x_2 \quad \cdots \quad x_i]^T \\ \mathbf{y}_i &= [y_1 \quad y_2 \quad \cdots \quad y_i]^T \end{aligned}$$

and $\mathbf{1}_i$ denotes the $i \times 1$ vector with all elements 1. The matrix $\mathbf{X}_{\text{fs}}$ can be expressed as an affine combination of matrices with coefficients x_i and y_i, $i = k+1, k+2, \cdots, M$:

$$\mathbf{X}_{\text{fs}} = \sum_{i=1}^{2(M-k)} \phi_i \mathbf{X}_i + \mathbf{X}_0 \tag{2}$$

where ϕ_i is the ith element of the vector $\boldsymbol{\phi}_{\text{fs}}$ which is defined as

$$\boldsymbol{\phi}_{\text{fs}} = [x_{k+1} \quad y_{k+1} \quad x_{k+2} \quad y_{k+2} \quad \cdots \quad x_M \quad y_M]^T.$$

Here, $[\mathbf{X}_0]_{i,j} \in \{0, -x_1, -y_1, -x_2, -y_2, \cdots, -x_k, -y_k\}$ and $[\mathbf{X}_k]_{i,j} \in \{-1, 0, 1\}$. As an illustration, consider the case when there are 5 nodes where 2 of their positions, namely, (x_4, y_4) and (x_5, y_5), are unknown. From (1), $\mathbf{X}_{\text{fs}}$ is of the form shown in (3) at the bottom of the page. Based on (2) and (3), the corresponding basis matrices $\mathbf{X}_i$, $i = 0, 1, \cdots, 4$, are easily constructed as follows:

$$\mathbf{X}_0 = \begin{bmatrix} -x_1 & -x_2 & -x_3 & -x_1 & -x_2 & -x_3 & 0 \\ -y_1 & -y_2 & -y_3 & -y_1 & -y_2 & -y_3 & 0 \end{bmatrix}^T \quad (4)$$

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (5)$$

$$\mathbf{X}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}^T \quad (6)$$

$$\mathbf{X}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (7)$$

and

$$\mathbf{X}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T. \quad (8)$$

That is, removing all x_4, y_4, x_5 and y_5 in $\mathbf{X}_{\text{fs}}$ yields $\mathbf{X}_0$. While $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$ and $\mathbf{X}_4$ contain the signs of x_4, y_4, x_5 and y_5 in $\mathbf{X}_{\text{fs}}$, respectively, where the sign value is set to zero when the corresponding element is absent.

The multidimensional similarity matrix [22]

$$\mathbf{F}_{\text{fs}} = \mathbf{X}_{\text{fs}}\mathbf{X}_{\text{fs}}^T \quad (9)$$

We assume that $\mathbf{X}_{\text{fs}}$ is of full column rank as there are only two columns. In practice this assumption is usually true since we only need any two rows of $\mathbf{X}_{\text{fs}}$ to be linearly independent. Then, we have $\text{rank}(\mathbf{F}_{\text{fs}}) = \text{rank}(\mathbf{X}_{\text{fs}}) = 2$.

The matrix $\mathbf{F}_{\text{fs}}$ is constructed from $\{d_{i,j}\}$ where $d_{i,j}$ denotes the distance between the ith sensor and jth sensor. The noise-free node-to-node distance is

$$d_{i,j} = d_{j,i} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \; i, j = 1, 2, \cdots, M. \quad (10)$$

With the use of scalar dot product and cosine rule, we have

$$\begin{aligned} (\mathbf{s}_i - \mathbf{s}_j)^T(\mathbf{s}_m - \mathbf{s}_j) &= d_{i,j}d_{j,m}\cos(\theta_{i,j,m}) \\ &= 0.5\left(d_{i,j}^2 + d_{j,m}^2 - d_{i,m}^2\right) \quad (11) \end{aligned}$$

where $\theta_{i,j,m}$ is the angle between the vectors $(\mathbf{s}_i - \mathbf{s}_j)$ and $(\mathbf{s}_m - \mathbf{s}_j)$. Extending (11) yields

$$\begin{aligned} &(\mathbf{s}_i - \mathbf{s}_j)^T(\mathbf{s}_m - \mathbf{s}_n) \\ &\quad = (\mathbf{s}_i - \mathbf{s}_j)^T\left(\mathbf{s}_m - \mathbf{s}_j - (\mathbf{s}_n - \mathbf{s}_j)\right) \\ &\quad = 0.5\left(d_{i,j}^2 + d_{j,m}^2 - d_{i,m}^2 - \left(d_{i,j}^2 + d_{j,n}^2 - d_{i,n}^2\right)\right) \\ &\quad = -0.5\left(d_{i,m}^2 - d_{i,n}^2 - d_{j,m}^2 + d_{j,n}^2\right) \\ &\quad = -0.5\left(\mathbf{e}_{M,i}^T\mathbf{R}\mathbf{e}_{M,m} - \mathbf{e}_{M,i}^T\mathbf{R}\mathbf{e}_{M,n}\right. \\ &\qquad \left. -\mathbf{e}_{M,j}^T\mathbf{R}\mathbf{e}_{M,m} + \mathbf{e}_{M,j}^T\mathbf{R}\mathbf{e}_{M,n}\right) \\ &\quad = -0.5(\mathbf{e}_{M,i} - \mathbf{e}_{M,j})^T\mathbf{R}(\mathbf{e}_{M,m} - \mathbf{e}_{M,n}). \quad (12) \end{aligned}$$

where $[\mathbf{R}]_{i,j} = d_{i,j}^2$ and $\mathbf{e}_{M,i}$ stands for the ith column of the $M \times M$ identity matrix, denoted by $\mathbf{I}_M$. Employing (12), $\mathbf{F}_{\text{fs}}$ can be expressed as:

$$\mathbf{F}_{\text{fs}} = -0.5\mathbf{K}\mathbf{R}\mathbf{K}^T \quad (13)$$

where

$$\begin{aligned} \mathbf{K} &= [\mathbf{H}_{k+1} \quad \mathbf{H}_{k+2} \quad \cdots \quad \mathbf{H}_M]^T \\ \mathbf{H}_i &= [\mathbf{e}_{M,i} - \mathbf{e}_{M,1} \quad \mathbf{e}_{M,i} - \mathbf{e}_{M,2} \quad \cdots \quad \mathbf{e}_{M,i} - \mathbf{e}_{M,i-1}] \\ &= \left[-\mathbf{I}_{i-1} \quad \mathbf{1}_{i-1} \quad \mathbf{0}_{(i-1)\times(M-i)}\right]^T, \\ &\quad i = k+1, k+2, \cdots, M \end{aligned}$$

and $\mathbf{0}_{i\times j}$ represents the $i \times j$ zero matrix. It is noteworthy to point out that in the classical MDS method [15]–[17], double centering is applied to the matrix of squared distances, namely, $\mathbf{R}$, and the resultant matrix of $-0.5\mathbf{J}\mathbf{R}\mathbf{J}$, where $\mathbf{J} = \mathbf{I}_M - \mathbf{1}_M\mathbf{1}_M^T/M$, is utilized for positioning. As a result, substituting $\mathbf{K} = \mathbf{J}$, the formulation of (13) will correspond to the MDS approach. An advantage of the proposed approach over the MDS scheme is that the step of aligning the relative locations to physical locations is not required, which will be shown shortly. Furthermore, we will see that different forms of $\mathbf{K}$ are allowed for node localization in the development of the minimum-set and distributed subspace algorithms.

Since $\mathbf{F}_{\text{fs}}$ is symmetric positive semidefinite and of rank 2, it can be decomposed using eigenvalue factorization as

$$\mathbf{F}_{\text{fs}} = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^T + \mathbf{U}_n\mathbf{\Lambda}_n\mathbf{U}_n^T \quad (14)$$

where $\mathbf{\Lambda}_s = \text{diag}(\lambda_1, \lambda_2) \succ \mathbf{0}$, $\mathbf{\Lambda}_n = \text{diag}(\lambda_3, \cdots, \lambda_{{}_M\mathrm{C}_2 - {}_k\mathrm{C}_2})$ with $\lambda_3 = \lambda_4 = \cdots = \lambda_{{}_M\mathrm{C}_2 - {}_k\mathrm{C}_2} = 0$ and the columns of $\mathbf{U}_s \in \mathbb{R}^{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)\times 2}$ and $\mathbf{U}_n \in \mathbb{R}^{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)\times({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)}$ are called the signal and noise eigenvectors, respectively. As $\mathbf{X}_{\text{fs}}$ lies in the signal subspace spanned by $\mathbf{U}_s$, we have

$$\mathbf{U}_n^T\mathbf{X}_{\text{fs}} = \mathbf{0}_{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)\times 2} \quad (15)$$

$$\mathbf{X}_{\text{fs}} = \begin{bmatrix} x_4 - x_1 & x_4 - x_2 & x_4 - x_3 & x_5 - x_1 & x_5 - x_2 & x_5 - x_3 & x_5 - x_4 \\ y_4 - y_1 & y_4 - y_2 & y_4 - y_3 & y_5 - y_1 & y_5 - y_2 & y_5 - y_3 & y_5 - y_4 \end{bmatrix}^T. \quad (3)$$

or

$$\sum_{i=1}^{2(M-k)} \phi_i \mathbf{U}_n^T \mathbf{X}_i = -\mathbf{U}_n^T \mathbf{X}_0. \tag{16}$$

When disturbance is present, noise-free distances $\{d_{i,j}\}$ are available only for $k \geq i > j \geq 1$. In our study, we generalize all the available distance information, denoted by $r_{i,j}$, as

$$r_{i,j} = r_{j,i} = d_{i,j} + q_{i,j}$$

where $q_{i,j} = q_{j,i}$ is the zero-mean noise with variance $\sigma_{i,j}^2 > 0$ if $\max(i,j) = k{+}1, k{+}2, \cdots, M$, and $\sigma_{i,j}^2 = 0$ otherwise, with $\max(i,j) = i$ if $i \geq j$. This means that for practical purposes, $\{r_{i,j}\}$ will be substituted for $\{d_{i,j}\}$ in (11) and (12) when the latter is unavailable in forming $\mathbf{F}_{\text{fs}}$. As a result, the equal signs in (15) and (16) should be replaced by the approximate equal signs and $\lambda_3 \geq \lambda_4 \geq \cdots \geq \lambda_{{}_M\mathrm{C}_2 - {}_k\mathrm{C}_2} \geq 0$. Taking vectorization on both sides of (16) yields

$$\mathbf{A}_{\text{fs}} \boldsymbol{\phi}_{\text{fs}} \approx \mathbf{b}_{\text{fs}} \tag{17}$$

where

$$\mathbf{A}_{\text{fs}} = \left[\mathrm{vec}\left(\mathbf{U}_n^T \mathbf{X}_1\right) \ \mathrm{vec}\left(\mathbf{U}_n^T \mathbf{X}_2\right) \ \cdots \ \mathrm{vec}\left(\mathbf{U}_n^T \mathbf{X}_{2(M-k)}\right)\right] \tag{18}$$

and

$$\mathbf{b}_{\text{fs}} = -\mathrm{vec}\left(\mathbf{U}_n^T \mathbf{X}_0\right) \tag{19}$$

with $\mathrm{vec}(\cdot)$ being the vectorization operator. A simple and straightforward estimate of $\boldsymbol{\phi}_{\text{fs}}$, denoted by $\hat{\boldsymbol{\phi}}_{\text{fs}}$, is obtained by the least squares (LS):

$$\hat{\boldsymbol{\phi}}_{\text{fs}} = \mathbf{A}_{\text{fs}}^{\dagger} \mathbf{b}_{\text{fs}} \tag{20}$$

where $\mathbf{A}^{\dagger} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T$ is the pseudo-inverse of $\mathbf{A}$. Note that $\mathbf{A}_{\text{fs}}$ is assumed to be of full column rank in (20). We refer this subspace based estimate to as the full-set solution because all admissible pairs in $\{\mathbf{s}_i\}$ have been employed.

It is worth mentioning that the subspace principle used in the above development is in essence the same as that in multiple signal classification (MUSIC) algorithm for direction-of-arrival (DOA) estimation [23]. Thus, the full-set subspace algorithm may loosely be seen as "a MUSIC method for a different signal processing problem". We however need to point out a notable difference. In the conventional MUSIC approach, the DOAs are estimated from the noise subspace through a line search or nonlinear optimization. On the other hand, in the full-set subspace algorithm, a closed-form expression of (20) is available for estimation of the node positions as they are linear in (15) or (16).

For the special case where there is only one sensor being unknown, that is, $M = k + 1$, $\mathbf{U}_n \in \mathbb{R}^{k \times (k-2)}$ and the matrices $\mathbf{A}_{\text{fs}}$, $\boldsymbol{\phi}_{\text{fs}}$ and $\mathbf{b}_{\text{fs}}$ are, respectively,

$$\begin{aligned} \mathbf{A}_{\text{fs}} &= \left[\mathrm{vec}\left(\mathbf{U}_n^T [\mathbf{1}_k \quad \mathbf{0}_{k\times 1}]\right) \quad \mathrm{vec}\left(\mathbf{U}_n^T [\mathbf{0}_{k\times 1} \quad \mathbf{1}_k]\right)\right] \\ \boldsymbol{\phi}_{\text{fs}} &= [x_{k+1} \quad y_{k+1}]^T \end{aligned}$$

and

$$\mathbf{b}_{\text{fs}} = \mathrm{vec}\left(\mathbf{U}_n^T [\mathbf{x}_k \quad \mathbf{y}_k]\right).$$

The LS solution, $\hat{\boldsymbol{\phi}}_{\text{fs}} = [\hat{x}_{k+1} \ \ \hat{y}_{k+1}]^T$, is then calculated as

$$\begin{aligned} \hat{\boldsymbol{\phi}}_{\text{fs}} &= \mathbf{A}_{\text{fs}}^{\dagger} \mathbf{b}_{\text{fs}} \\ &= \begin{bmatrix} \mathbf{U}_n^T \mathbf{1}_k & \mathbf{0}_{(k-2)\times 1} \\ \mathbf{0}_{(k-2)\times 1} & \mathbf{U}_n^T \mathbf{1}_k \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{U}_n^T \mathbf{x}_k \\ \mathbf{U}_n^T \mathbf{y}_k \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{1}_k & 0 \\ 0 & \mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{1}_k \end{bmatrix}^{-1} \\ &\quad \times \begin{bmatrix} \mathbf{1}_k^T \mathbf{U}_n & \mathbf{0}_{1\times(k-2)} \\ \mathbf{0}_{1\times(k-2)} & \mathbf{1}_k^T \mathbf{U}_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_n^T \mathbf{x}_k \\ \mathbf{U}_n^T \mathbf{y}_k \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{1}_k\right)^{-1} \mathbf{1}_k^T \mathbf{U}_n & \mathbf{0}_{1\times(k-2)} \\ \mathbf{0}_{1\times(k-2)} & \left(\mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{1}_k\right)^{-1} \mathbf{1}_k^T \mathbf{U}_n \end{bmatrix} \\ &\quad \times \begin{bmatrix} \mathbf{U}_n^T \mathbf{x}_k \\ \mathbf{U}_n^T \mathbf{y}_k \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{1}_k\right)^{-1} \mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{x}_k \\ \left(\mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{1}_k\right)^{-1} \mathbf{1}_k^T \mathbf{U}_n \mathbf{U}_n^T \mathbf{y}_k \end{bmatrix}. \end{aligned} \tag{21}$$

In particular, when $M = 4$, $\mathbf{U}_n$ is a vector and (21) can be further simplified as

$$\hat{\boldsymbol{\phi}}_{\text{fs}}^T = \frac{\mathbf{U}_n^T}{\mathbf{U}_n^T \mathbf{1}_3} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}$$

which is exactly the solution suggested in [22]. Hence our proposed positioning method is a generalization of [22] in the sense that various numbers of sensors with known and unknown positions are allowed.

B. Minimum-Set Subspace Algorithm

Instead of utilizing all admissible pairs in $\{s_i\}$, it is possible to use a reduced set of the pairs via considering the rank deficiency of $\mathbf{F}_{\text{fs}}$. In doing so, the computational complexity of the full-set subspace algorithm can be significantly reduced. We refer the resultant method to as minimum-set subspace algorithm and its derivation is given as follows. From (13), it can be shown that

$$\mathbf{H}_{i+1}[\mathbf{I}_{i-1} \quad -\mathbf{1}_{i-1}]^T = \mathbf{H}_i, \quad i = k+1, k+2, \cdots, M-1$$

which further implies

$$\mathrm{span}(\mathbf{H}_{k+1}) \subset \mathrm{span}(\mathbf{H}_{k+2}) \subset \cdots \subset \mathrm{span}(\mathbf{H}_M)$$

where $\mathrm{span}(\mathbf{A})$ is the space spanned by the columns of $\mathbf{A}$ and that

$$\mathrm{rank}(\mathbf{K}) = \mathrm{rank}(\mathbf{H}_M) = M - 1.$$

Therefore, dimension reduction can be achieved by replacing $\mathbf{F}_{\text{fs}} \in \mathbb{R}^{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)\times({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)}$ by $\mathbf{F}_{\text{ms}} \in \mathbb{R}^{(M-1)\times(M-1)}$ of the form:

$$\mathbf{F}_{\text{ms}} = \mathbf{X}_{\text{ms}} \mathbf{X}_{\text{ms}}^T = -0.5 \mathbf{H}_M^T \mathbf{R} \mathbf{H}_M \tag{22}$$

TABLE II
COMPUTATIONAL COMPLEXITY OF THE THREE SUBSPACE AND MDS ALGORITHMS

	SVD	**LS**
Full-Set	$4\left({}_M\mathrm{C}_2 -_k \mathrm{C}_2\right)^3 + 22\left({}_M\mathrm{C}_2 -_k \mathrm{C}_2\right)^2$	$2\left({}_M\mathrm{C}_2 -_k \mathrm{C}_2 - 2\right)(M-k)^2 +$ $4\left({}_M\mathrm{C}_2 -_k \mathrm{C}_2 - 2\right)(M-k) +$ $2(M-k)^2$
Minimum-Set	$4(M-1)^3 + 22(M-1)^2$	$2(M-3)(M-k)^2 +$ $4(M-3)(M-k) +$ $2(M-k)^2$
Distributed (per sensor)	$4k^3(M-k) + 22k^2(M-k)^2$	$2(k-2)^2 + 4(k-2) + 2$
MDS	$4M^3 + 22M^2$	$10(M-k) + 12k + 4$

where

$$\mathbf{X}_{\mathrm{ms}} = \begin{bmatrix} x_M - x_1 & x_M - x_2 & \cdots & x_M - x_{M-1} \\ y_M - y_1 & y_M - y_2 & \cdots & y_M - y_{M-1} \end{bmatrix}^T .$$

In general, we can extract any $(M-1)$ rows of $\mathbf{K}$ in (13) to construct $\mathbf{F}_{\mathrm{ms}}$ such that it is of full rank in the presence of noise. As a result, the choice of $\mathbf{F}_{\mathrm{ms}}$ is not unique because different rows of $\mathbf{K}$ can be selected as long as the rank requirement is satisfied. Moreover, different choices will generally give different estimation accuracy. The position estimate development of this simpler scheme follows the same procedure as that in (14)–(20):

$$\mathbf{A}_{\mathrm{ms}}\boldsymbol{\phi}_{\mathrm{ms}} \approx \mathbf{b}_{\mathrm{ms}} \Rightarrow \hat{\boldsymbol{\phi}}_{\mathrm{ms}} = \mathbf{A}_{\mathrm{ms}}^{\dagger}\mathbf{b}_{\mathrm{ms}} \tag{23}$$

where $\mathbf{A}_{\mathrm{ms}} \in \mathbb{R}^{2(M-3)\times 2(M-k)}$, $\boldsymbol{\phi}_{\mathrm{ms}} \in \mathbb{R}^{2(M-k)}$ and $\mathbf{b}_{\mathrm{ms}} \in \mathbb{R}^{2(M-3)}$ have definitions corresponding to $\mathbf{A}_{\mathrm{fs}}$, $\boldsymbol{\phi}_{\mathrm{fs}}$ and $\mathbf{b}_{\mathrm{fs}}$, respectively, and $\hat{\boldsymbol{\phi}}_{\mathrm{ms}}$ is the LS estimate of $\boldsymbol{\phi}_{\mathrm{ms}}$. We refer this method to as the minimum-set solution.

C. Distributed Subspace Algorithm

The full-set and minimum-set subspace algorithms need centralized processing and are not suitable for certain WSN applications where distributed computing is required. In the following, we modify the subspace approach so that $(M-k)$ sensors can share the computations simultaneously. Note that our proposed distributed algorithm utilizes all available distance information simultaneously to produce the solution with no iteration. This is different from the distributed algorithm in [17] which assumes a partially connected network where localized mapping and alignment are performed from sensor nodes along a route from the starting anchor to an ending anchor in an iterative manner.

Construct $\mathbf{F}_{\mathrm{dist},i} \in \mathbb{R}^{k\times(M-k)k}$ as:

$$\begin{aligned} \mathbf{F}_{\mathrm{dist},i} =& \mathbf{X}_{\mathrm{dist},i}\left[\mathbf{X}_{\mathrm{dist},k+1}^T \quad \mathbf{X}_{\mathrm{dist},k+2}^T \quad \cdots \quad \mathbf{X}_{\mathrm{dist},M}^T\right] \\ =& -0.5\mathbf{P}_i\mathbf{H}_i^T\mathbf{R} \\ & \times \left[\mathbf{H}_{k+1}\mathbf{P}_{k+1}^T \quad \mathbf{H}_{k+2}\mathbf{P}_{k+2}^T \quad \cdots \quad \mathbf{H}_M\mathbf{P}_M^T\right] \\ & i = k+1, k+2, \cdots, M \end{aligned} \tag{24}$$

where

$$\begin{aligned} \mathbf{X}_{\mathrm{dist},i} &= \begin{bmatrix} x_i - x_1 & x_i - x_2 & \cdots & x_i - x_k \\ y_i - y_1 & y_i - y_2 & \cdots & y_i - y_k \end{bmatrix}^T \\ &= [\mathbf{1}_k \quad \mathbf{0}_{k\times 1}]x_i + [\mathbf{0}_{k\times 1} \quad \mathbf{1}_k]y_i - [\mathbf{x}_k \quad \mathbf{y}_k] \end{aligned}$$

and

$$\mathbf{P}_i = \left[\mathbf{I}_k \quad \mathbf{0}_{k\times(i-1-k)}\right].$$

Since $\mathrm{rank}(\mathbf{F}_{\mathrm{dist},i}) = 2$ in the absence of noise, $\mathbf{F}_{\mathrm{dist},i}$ can be decomposed by singular value decomposition (SVD) as

$$\mathbf{F}_{\mathrm{dist},i} = \mathbf{U}_{s,i}\boldsymbol{\Lambda}_{s,i}\mathbf{V}_{s,i}^T + \mathbf{U}_{n,i}\boldsymbol{\Lambda}_{n,i}\mathbf{V}_{n,i}^T \tag{25}$$

where $\boldsymbol{\Lambda}_{s,i} = \mathrm{diag}(\lambda_{i,1}, \lambda_{i,2})$ and $\boldsymbol{\Lambda}_{n,i} = \mathbf{0}_{(k-2)\times(k-2)}$. Here, the columns of $\mathbf{U}_{s,i} \in \mathbb{R}^{k\times 2}$, $\mathbf{U}_{n,i} \in \mathbb{R}^{k\times(k-2)}$, $\mathbf{V}_{s,i} \in \mathbb{R}^{k(M-k)\times 2}$ and $\mathbf{V}_{n,i} \in \mathbb{R}^{k(M-k)\times(k-2)}$ contain the singular vectors of $\mathbf{F}_{\mathrm{dist},i}$. Similar to the previous development and $\{r_{i,j}\}$ will be substituted for $\{d_{i,j}\}$ when the latter is unavailable, we then have:

$$\mathbf{U}_{n,i}^T\mathbf{X}_{\mathrm{dist},i} \approx \mathbf{0}_{(k-2)\times 2}. \tag{26}$$

Taking vectorization on both sides of (26) with the use of (24) yields

$$\mathbf{A}_{\mathrm{dist},i}\mathbf{s}_i \approx \mathbf{b}_{\mathrm{dist},i} \tag{27}$$

where

$$\mathbf{A}_{\mathrm{dist},i} = \left[\mathrm{vec}\left(\mathbf{U}_{n,i}^T[\mathbf{1}_k \quad \mathbf{0}_{k\times 1}]\right) \quad \mathrm{vec}\left(\mathbf{U}_{n,i}^T[\mathbf{0}_{k\times 1} \quad \mathbf{1}_k]\right)\right]$$

and

$$\mathbf{b}_{\mathrm{dist},i} = \mathrm{vec}\left(\mathbf{U}_{n,i}^T[\mathbf{x}_k \quad \mathbf{y}_k]\right).$$

The LS estimate of $\mathbf{s}_i$, denoted by $\hat{\mathbf{s}}_i$, is given by

$$\hat{\mathbf{s}}_i = \mathbf{A}_{\mathrm{dist},i}^{\dagger}\mathbf{b}_{\mathrm{dist},i} \tag{28}$$

and we refer this approach to as the distributed subspace method.

III. COMPUTATIONAL COMPLEXITY OF SUBSPACE ALGORITHMS

In this Section, the computational requirement of our proposed positioning methods, namely, the full-set, minimum-set as well as distributed subspace algorithms will be examined. The following analysis is based on the SVD and LS operations on which most of the computations spend. First, we will investigate the computational complexity of the SVD in the proposed subspace algorithms. As performing SVD of $\boldsymbol{\Upsilon} \in \mathbb{R}^{m\times n}$ requires $4m^2n + 22n^2$ floating point operations (FLOPS) [24], the numbers of FLOPS involved in the SVD of $\mathbf{F}_{\mathrm{fs}}$, $\mathbf{F}_{\mathrm{ms}}$ and $\mathbf{F}_{\mathrm{dist},i}$ are straightforwardly determined and the results are shown in the first column of Table II. Next, we consider the computational complexity of solving a set of linear equations $\boldsymbol{\Xi}\boldsymbol{\beta} \approx \boldsymbol{\gamma}$ where $\boldsymbol{\Xi} \in \mathbb{R}^{m\times n}$, $\boldsymbol{\beta} \in \mathbb{R}^n$ and $\boldsymbol{\gamma} \in \mathbb{R}^m$ by QR factorization [24] though other methods can be used. The whole LS process requires $2mn^2$ FLOPS for the QR factorization of $\boldsymbol{\Xi} = \mathbf{QR}$,

$2mn$ FLOPS for vector construction of $\boldsymbol{\gamma} = \mathbf{Q}^T\boldsymbol{\gamma}$ as well as n^2 FLOPS for the backward substitution of $\mathbf{R}\boldsymbol{\beta} = \boldsymbol{\gamma}$.

Now we modify the proposed algorithms to enhance computational efficiency. In the following, a fast algorithm for the full-set subspace algorithm is developed and it can be easily extended to the minimum-set and distributed algorithms. Expressing $\mathbf{X}_{\rm fs}$ as

$$\begin{aligned}\mathbf{X}_{\rm fs} &= \sum_{i=1}^{2(M-k)} \phi_i\mathbf{X}_i + \mathbf{X}_0\\ &= \sum_{i=1}^{M-k} \left(x_{i+k}\mathbf{X}_{2(i-1)+1} + y_{i+k}\mathbf{X}_{2i}\right) + \mathbf{X}_0\\ &= \left[\mathbf{L}[x_{k+1} \quad \cdots \quad x_M]^T \quad \mathbf{0}_{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)\times 1}\right]\\ &\quad + \left[\mathbf{0}_{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2)\times 1} \quad \mathbf{L}[y_{k+1} \quad \cdots \quad y_M]^T\right] + \mathbf{X}_0\\ &= \mathbf{L}\boldsymbol{\Phi}_{\rm fs} + \mathbf{X}_0 \end{aligned} \tag{29}$$

where

$$\begin{aligned}\mathbf{L} &= \left[\mathbf{X}_1\mathbf{G} \quad \mathbf{X}_3\mathbf{G} \quad \mathbf{X}_5\mathbf{G} \quad \cdots \quad \mathbf{X}_{2(M-k)-1}\mathbf{G}\right]\\ \mathbf{G} &= [1 \quad 0]^T\end{aligned}$$

and

$$\boldsymbol{\Phi} = \begin{bmatrix} x_{k+1} & y_{k+1}\\ \vdots & \vdots\\ x_M & y_M \end{bmatrix}.$$

Based on (15) and (29), we get

$$\begin{aligned}\mathbf{U}_n^T(\mathbf{L}\boldsymbol{\Phi} + \mathbf{X}_0) &\approx \mathbf{0}_{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)\times 2}\\ \Rightarrow \hat{\boldsymbol{\Phi}}_{\rm fs} &= -\left(\mathbf{U}_n\mathbf{U}_n^T\mathbf{L}\right)^\dagger\left(\mathbf{U}_n\mathbf{U}_n^T\mathbf{X}_0\right)\\ &= -\left(\mathbf{U}_n^T\mathbf{L}\right)^\dagger\mathbf{U}_n^T\mathbf{X}_0\end{aligned} \tag{30}$$

where $\hat{\boldsymbol{\Phi}}_{\rm fs}$ is the LS estimate of $\boldsymbol{\Phi}_{\rm fs}$. Since $\mathbf{A}_{\rm fs} \in \mathbb{R}^{2({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)\times 2(M-k)}$, $\mathbf{b}_{\rm fs} \in \mathbb{R}^{2({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)}$, $\mathbf{U}_n^T\mathbf{L} \in \mathbb{R}^{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)\times(M-k)}$ and $\mathbf{U}_n^T\mathbf{X}_0 \in \mathbb{R}^{({}_M\mathrm{C}_2 - {}_k\mathrm{C}_2 - 2)\times 2}$, the computational complexity of calculating $\hat{\boldsymbol{\phi}}_{\rm fs}$ and $\hat{\boldsymbol{\Phi}}_{\rm fs}$, is, respectively, $16({}_M\mathrm{C}_2 -_k \mathrm{C}_2 - 2)(M-k)^2 + 8({}_M\mathrm{C}_2 -_k \mathrm{C}_2 - 2)(M-k) + 4(M-k)^2$ and $2({}_M\mathrm{C}_2 -_k \mathrm{C}_2 - 2)(M-k)^2 + 4({}_M\mathrm{C}_2 -_k \mathrm{C}_2 - 2)(M-k) + 2(M-k)^2$, which indicates the superiority of using (30). The numbers of FLOPS required to compute the LS solutions of the three subspace algorithms are shown in the second column of Table II. As a comparison, the computational complexity of the MDS method [17] is also included.

Combining the results in Table II, we see that the computational requirements of the minimum-set subspace and MDS methods are comparable, which are of order M^3 or $O(M^3)$. The full-set version is the most computationally demanding as it involves roughly $O(M^6)$ operations. While the distributed variant will be the simplest one when $M \gg k$. It is worthy to note that [10] the typical number of operations for the SDP algorithm is $O(M^3)$ which is identical to those of the MDS as well as proposed minimum-set and distributed methods. Nevertheless, it is believed that the actual computation time of the former is larger than those of the latter according to the simulation comparison between the SDP and MDS approaches in [12].

IV. Statistical Analysis

In this Section, the biases and mean square errors of the sensor position estimates of the proposed approach are derived. Although we focus on analyzing the full-set subspace algorithm, the produced results are applicable to the minimum-set and distributed subspace algorithms. Let $\mathbf{X}_{\rm fs} = \tilde{\mathbf{X}}_{\rm fs} + \boldsymbol{\Delta}\mathbf{X}_{\rm fs}$, $\mathbf{F}_{\rm fs} = \tilde{\mathbf{F}}_{\rm fs} + \boldsymbol{\Delta}\mathbf{F}_{\rm fs}$ and $\mathbf{U}_n = \tilde{\mathbf{U}}_n + \boldsymbol{\Delta}\mathbf{U}_n$ where $\tilde{\mathbf{X}}_{\rm fs}$, $\tilde{\mathbf{F}}_{\rm fs}$ and $\tilde{\mathbf{U}}_n$ are, respectively, the noise-free matrix that contains the co-ordinate differences, noise-free multidimensional similarity matrix and vectors that span the noise-free noise subspace while $\boldsymbol{\Delta}\mathbf{X}_{\rm fs}$, $\boldsymbol{\Delta}\mathbf{F}_{\rm fs}$ and $\boldsymbol{\Delta}\mathbf{U}_n$ are the corresponding residual components due to errors in range measurements. Similarly, we express $\mathbf{A}_{\rm fs}$, $\mathbf{b}_{\rm fs}$ and $\hat{\boldsymbol{\phi}}_{\rm fs}$ as

$$\begin{aligned}\mathbf{A}_{\rm fs} &= \tilde{\mathbf{A}}_{\rm fs} + \boldsymbol{\Delta}\mathbf{A}_{\rm fs}\\ \mathbf{b}_{\rm fs} &= \tilde{\mathbf{b}}_{\rm fs} + \boldsymbol{\Delta}\mathbf{b}_{\rm fs}\end{aligned}$$

and

$$\hat{\boldsymbol{\phi}}_{\rm fs} = \boldsymbol{\phi} + \boldsymbol{\Delta}\boldsymbol{\phi}_{\rm fs}$$

where $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$ and $\boldsymbol{\phi}$ are the noise-free matrices while $\boldsymbol{\Delta}\mathbf{A}_{\rm fs}$, $\boldsymbol{\Delta}\mathbf{b}_{\rm fs}$ and $\boldsymbol{\Delta}\boldsymbol{\phi}_{\rm fs}$ are the corresponding noise components. By applying the first-order approximation of $\boldsymbol{\Delta}\mathbf{U}_n$ [25]

$$\boldsymbol{\Delta}\mathbf{U}_n \approx -\left(\tilde{\mathbf{F}}_{\rm fs}^\dagger\right)^T \boldsymbol{\Delta}\mathbf{F}_{\rm fs}^T\tilde{\mathbf{U}}_n \tag{31}$$

the perturbations $\boldsymbol{\Delta}\mathbf{A}_{\rm fs}$ and $\boldsymbol{\Delta}\mathbf{b}_{\rm fs}$ from (18) and (19) are, respectively

$$\begin{aligned}\boldsymbol{\Delta}\mathbf{A}_{\rm fs} = \Big[&-\mathrm{vec}\left(\tilde{\mathbf{U}}_n^T\boldsymbol{\Delta}\mathbf{F}_{\rm fs}\tilde{\mathbf{F}}_{\rm fs}^\dagger\mathbf{X}_1\right) \quad \cdots\\ &-\mathrm{vec}\left(\tilde{\mathbf{U}}_n^T\boldsymbol{\Delta}\mathbf{F}_{\rm fs}\tilde{\mathbf{F}}_{\rm fs}^\dagger\mathbf{X}_{2(M-k)}\right)\Big]\end{aligned} \tag{32}$$

and

$$\boldsymbol{\Delta}\mathbf{b}_{\rm fs} = \mathrm{vec}(\tilde{\mathbf{U}}_n^T\boldsymbol{\Delta}\mathbf{F}_{\rm fs}\tilde{\mathbf{F}}_{\rm fs}^\dagger\mathbf{X}_0). \tag{33}$$

To determine $\boldsymbol{\Delta}\boldsymbol{\phi}_{\rm fs}$, we premultiply $\mathbf{A}_{\rm fs}^T\mathbf{A}_{\rm fs}$ on both sides of (20) to obtain

$$\mathbf{A}_{\rm fs}^T\mathbf{A}_{\rm fs}\hat{\boldsymbol{\phi}}_{\rm fs} = \mathbf{A}_{\rm fs}^T\mathbf{b}_{\rm fs}. \tag{34}$$

By analyzing the perturbations of (34) and neglecting the second-order terms, we get

$$\begin{aligned}\mathbf{A}_{\rm fs}^T\mathbf{A}_{\rm fs}\hat{\boldsymbol{\phi}}_{\rm fs} &= \Big(\tilde{\mathbf{A}}_{\rm fs}^T\tilde{\mathbf{A}}_{\rm fs} + \tilde{\mathbf{A}}_{\rm fs}^T\boldsymbol{\Delta}\mathbf{A}_{\rm fs} + \boldsymbol{\Delta}\mathbf{A}_{\rm fs}^T\tilde{\mathbf{A}}_{\rm fs}\\ &\qquad + \boldsymbol{\Delta}\mathbf{A}_{\rm fs}^T\boldsymbol{\Delta}\mathbf{A}_{\rm fs}\Big)\left(\boldsymbol{\phi} + \boldsymbol{\Delta}\boldsymbol{\phi}_{\rm fs}\right)\\ &\approx \tilde{\mathbf{A}}_{\rm fs}^T\tilde{\mathbf{A}}_{\rm fs}\boldsymbol{\phi} + \tilde{\mathbf{A}}_{\rm fs}^T\tilde{\mathbf{A}}_{\rm fs}\boldsymbol{\Delta}\boldsymbol{\phi}_{\rm fs} + \tilde{\mathbf{A}}_{\rm fs}^T\boldsymbol{\Delta}\mathbf{A}_{\rm fs}\boldsymbol{\phi}\\ &\quad + \boldsymbol{\Delta}\mathbf{A}_{\rm fs}^T\tilde{\mathbf{A}}_{\rm fs}\boldsymbol{\phi}\\ &= \tilde{\mathbf{A}}_{\rm fs}^T\tilde{\mathbf{b}}_{\rm fs} + \tilde{\mathbf{A}}_{\rm fs}^T\tilde{\mathbf{A}}_{\rm fs}\boldsymbol{\Delta}\boldsymbol{\phi}_{\rm fs} + \tilde{\mathbf{A}}_{\rm fs}^T\boldsymbol{\Delta}\mathbf{A}_{\rm fs}\boldsymbol{\phi}\\ &\quad + \boldsymbol{\Delta}\mathbf{A}_{\rm fs}^T\tilde{\mathbf{b}}_{\rm fs}\end{aligned} \tag{35}$$

and

$$\mathbf{A}_{\text{fs}}^T\mathbf{b}_{\text{fs}} \approx \tilde{\mathbf{A}}_{\text{fs}}^T\tilde{\mathbf{b}}_{\text{fs}} + \tilde{\mathbf{A}}_{\text{fs}}^T\boldsymbol{\Delta}\mathbf{b}_{\text{fs}} + \boldsymbol{\Delta}\mathbf{A}_{\text{fs}}^T\tilde{\mathbf{b}}_{\text{fs}}. \tag{36}$$

Substituting (35) and (36) into (34) yields

$$\begin{aligned}
\tilde{\mathbf{A}}_{\text{fs}}^T\tilde{\mathbf{A}}_{\text{fs}}\boldsymbol{\Delta\phi}_{\text{fs}} &\approx \tilde{\mathbf{A}}_{\text{fs}}^T(\boldsymbol{\Delta}\mathbf{b}_{\text{fs}} - \boldsymbol{\Delta}\mathbf{A}_{\text{fs}}\boldsymbol{\phi}) \\
\Rightarrow \boldsymbol{\Delta\phi}_{\text{fs}} &\approx \tilde{\mathbf{A}}_{\text{fs}}^\dagger(\boldsymbol{\Delta}\mathbf{b}_{\text{fs}} - \boldsymbol{\Delta}\mathbf{A}_{\text{fs}}\boldsymbol{\phi}) \\
&= \tilde{\mathbf{A}}_{\text{fs}}^\dagger\Big(\Big[\text{vec}\Big(\tilde{\mathbf{U}}_n^T\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}\tilde{\mathbf{F}}_{\text{fs}}^\dagger\mathbf{X}_0\Big) \quad \cdots \\
&\qquad \text{vec}\Big(\tilde{\mathbf{U}}_n^T\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}\tilde{\mathbf{F}}_{\text{fs}}^\dagger\mathbf{X}_{2(M-k)}\Big)\Big][1 \quad \boldsymbol{\phi}^T]^T\Big) \\
&= \tilde{\mathbf{A}}_{\text{fs}}^\dagger\Big(\Big[\Big(\Big(\mathbf{X}_0^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\Big)\otimes\tilde{\mathbf{U}}_n^T\Big)\text{vec}(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}) \quad \cdots \\
&\qquad \Big(\Big(\mathbf{X}_{2(M-k)}^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\Big)\otimes\tilde{\mathbf{U}}_n^T\Big)\text{vec}(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}})\Big) \\
&\qquad \times[1 \quad \boldsymbol{\phi}^T]^T\Big) \\
&= \tilde{\mathbf{A}}_{\text{fs}}^\dagger\Bigg(\Big(\mathbf{X}_0^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\Big)\otimes\tilde{\mathbf{U}}_n^T \\
&\qquad + \sum_{i=1}^{2(M-k)}\Big(\tilde{\phi}_i\mathbf{X}_i^T\Big)\tilde{\mathbf{F}}_{\text{fs}}^\dagger\otimes\tilde{\mathbf{U}}_n^T\Bigg) \\
&\quad \times\text{vec}(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}) \\
&= \tilde{\mathbf{A}}_{\text{fs}}^\dagger\Big(\tilde{\mathbf{X}}^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\otimes\tilde{\mathbf{U}}_n^T\Big)\text{vec}(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}})
\end{aligned} \tag{37}$$

where $\otimes$ denotes Kronecker matrix product. We see that only $\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}$ is the random variable in (37) and taking the expected value of $\boldsymbol{\Delta\phi}_{\text{fs}}$ yields

$$\mathbb{E}\{\boldsymbol{\Delta\phi}_{\text{fs}}\} = \tilde{\mathbf{A}}_{\text{fs}}^\dagger\Big(\tilde{\mathbf{X}}^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\otimes\tilde{\mathbf{U}}_n^T\Big)\text{vec}\left(\mathbb{E}\{\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}\}\right) \tag{38}$$

where $\mathbb{E}$ denotes the expectation operator. The (i,j) element of $\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}$ can be expressed as

$$\begin{aligned}
[\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}]_{i,j} = 0.5\,\big(&2d_{k,l}q_{k,l} + q_{k,l}^2 + 2d_{m,n}q_{m,n} + q_{m,n}^2 \\
&-2d_{k,n}q_{k,n} - q_{k,n}^2 - 2d_{m,l}q_{m,l} - q_{m,l}^2\big).
\end{aligned} \tag{39}$$

Taking the expected value of (39), we have

$$\mathbb{E}\left\{[\boldsymbol{\Delta}\mathbf{F}_{\text{fs}}]_{i,j}\right\} = 0.5\left(\sigma_{k,l}^2 + \sigma_{m,n}^2 - \sigma_{k,n}^2 - \sigma_{m,l}^2\right). \tag{40}$$

As a result, the biases of the position estimates are computed from (38) and (40), and we roughly see that they decrease with the noise powers of the distance measurements. On the other hand, the correlation matrix of $\boldsymbol{\Delta\phi}$ is evaluated using (37) as shown in the Appendix

$$\begin{aligned}
&\mathbb{E}\left\{\boldsymbol{\Delta\phi}_{\text{fs}}\boldsymbol{\Delta\phi}_{\text{fs}}^T\right\} = \tilde{\mathbf{A}}_{\text{fs}}^\dagger\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\otimes\tilde{\mathbf{U}}_n^T\right) \\
&\times\mathbb{E}\left\{\text{vec}(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}})\text{vec}^T(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}})\right\}\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^\dagger\otimes\tilde{\mathbf{U}}_n^T\right)^T(\tilde{\mathbf{A}}_{\text{fs}}^\dagger)^T
\end{aligned} \tag{41}$$

where

$$\begin{aligned}
\mathbb{E}\left\{\text{vec}(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}})\text{vec}^T(\boldsymbol{\Delta}\mathbf{F}_{\text{fs}})\right\} &= 0.25(\mathbf{K}\otimes\mathbf{K})(\mathbf{V}_1+\mathbf{V}_2) \\
&\quad \times(\mathbf{K}^T\otimes\mathbf{K}^T) \\
\mathbf{V}_1 &= \text{vec}(\boldsymbol{\Sigma})\text{vec}^T(\boldsymbol{\Sigma}) \\
&\quad + \text{diag}\Big(4\text{vec}(\tilde{\mathbf{R}})\odot\text{vec}(\boldsymbol{\Sigma}) \\
&\qquad +2\text{vec}(\boldsymbol{\Sigma})\odot\text{vec}(\boldsymbol{\Sigma})\Big) \\
\boldsymbol{\Sigma} &= \begin{bmatrix} 0 & \sigma_{1,2}^2 & \cdots & \sigma_{1,M}^2 \\ \sigma_{2,1}^2 & 0 & \cdots & \sigma_{2,M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{M,1}^2 & \sigma_{M,2}^2 & \cdots & 0 \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
&[\mathbf{V}_2]_{m+(n-1)M,n+(m-1)M} \\
&\quad = \begin{cases} 4d_{m,n}^2\sigma_{m,n}^2 + 2\sigma_{m,n}^4 & : \quad m,n = 1,2,\cdots,M \\ 0 & : \quad \text{otherwise} \end{cases}.
\end{aligned}$$

From the diagonal elements of $\mathbb{E}\{\boldsymbol{\Delta\phi}_{\text{fs}}\boldsymbol{\Delta\phi}_{\text{fs}}^T\}$, we have the mean square errors of the position estimates.

V. Numerical Examples

Computer simulation has been conducted to evaluate the performance of the proposed range-based WSN positioning approach. We compare the mean square position errors (MSPEs) of the subspace algorithms to the classical MDS method [15]–[17], as well as CRLB for cooperative localization [21]. Note that for a fully connected network, the MDS scheme performs eigenvalue decomposition on (13) with $\mathbf{K} = \mathbf{J}$ and we employ the alignment procedure in [17] to obtain the final position estimates. The additive noises $\{q_{i,j}\}$ are zero-mean white Gaussian processes and they are scaled to obtain different signal-to-noise ratio (SNR) conditions. The SNR of all range measurements is set to be identical and it is defined as $\text{SNR} = d_{i,j}^2/\sigma_{i,j}^2$. A WSN of 14 fully connected sensors is considered with 10 of them are being unknown. The 14 sensors are located in a 100 m × 200 m area with corners (0,0)m, (100,0)m, (100,200)m, and (0,200)m. Though different realizations of $\mathbf{F}_{\text{ms}}$ are possible for the minimum-set subspace algorithm, the one in (22) is adopted in the following for simplicity. All simulation results are averages of 1000 independent runs.

In the first scenario, the 10 unknown-position sensors have fixed locations while the anchor sensors are placed at (0,0)m, (100,0)m, (100,100)m, and (0, 100)m and the WSN geometry is shown in Fig. 1. The average of the MSPEs for the 10 unknown-position sensors is plotted in Fig. 2 to show the overall performance. It can be observed that the three proposed schemes have smaller MSPEs than those of the MDS method with the full-set subspace algorithm performs the best, though it is inferior to the CRLB by about 8 dBm^2, in the whole SNR range. Furthermore, the measured MSPEs of the subspace methods agree very well with the theoretical calculation based on (41). In this test, the numbers of FLOPS required in the full-set, minimum-set, distributed subspace (per sensor) and MDS methods are 2635570, 15346, 37778, and 15440, respectively. It is seen that the minimum-set subspace algorithm is the most computationally efficient but gives the worst accuracy among the three

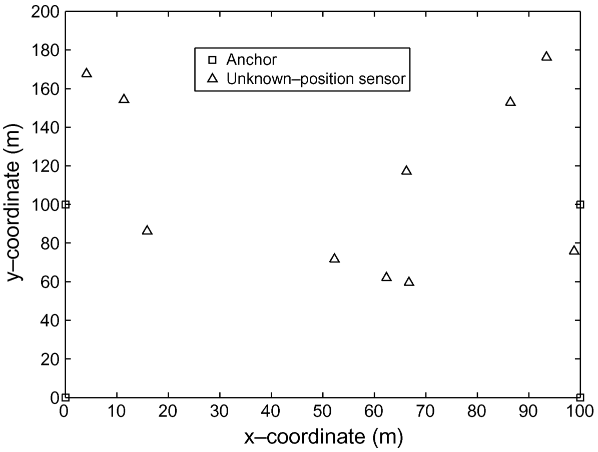


Fig. 1. Positions of the sensors in the first test.

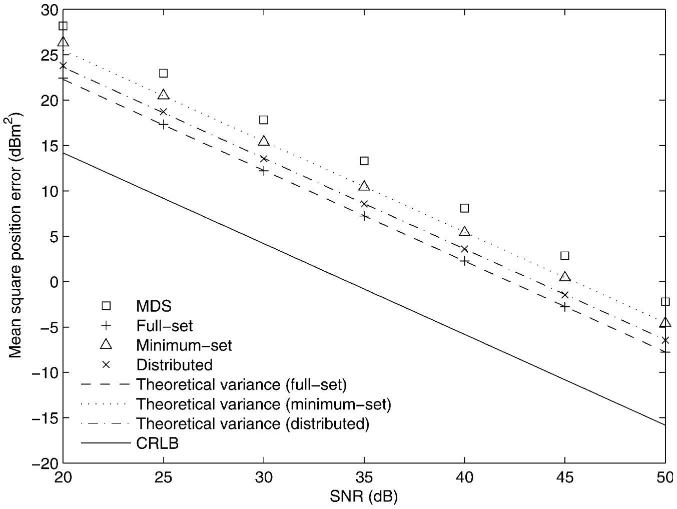


Fig. 2. Mean square position error versus SNR with fixed-position sensors for the first anchor geometry.

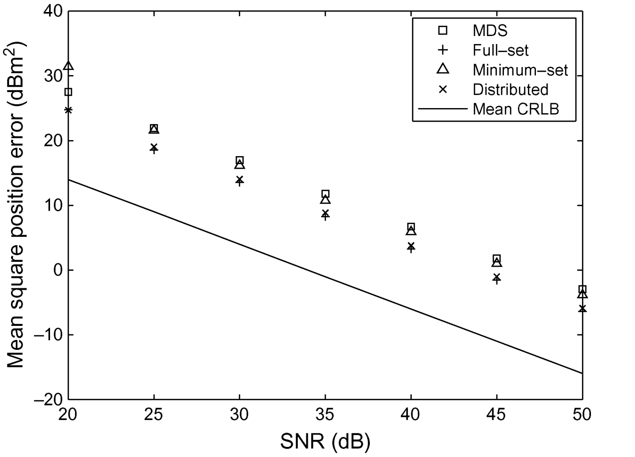


Fig. 3. Mean square position error versus SNR with random-position sensors for the first anchor geometry.

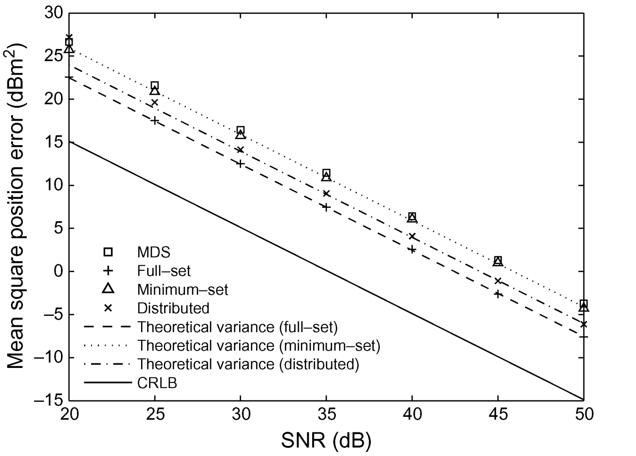


Fig. 4. Mean square position error versus SNR with fixed-position sensors for the second anchor geometry.

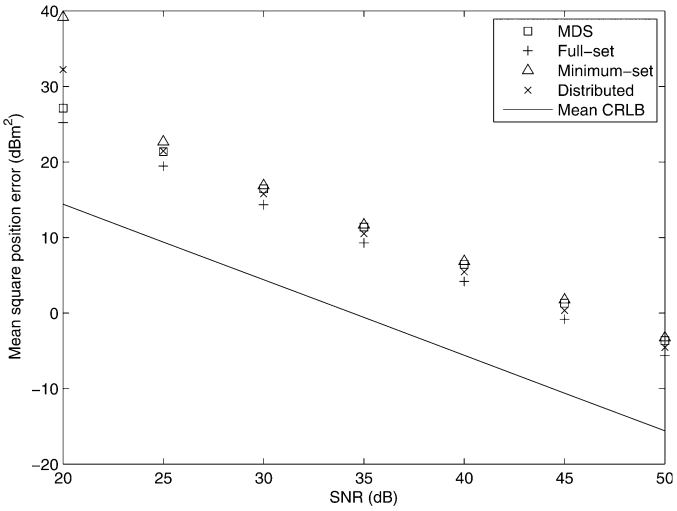


Fig. 5. Mean square position error versus SNR with random-position sensors for the second anchor geometry.

proposed schemes and has comparable complexity with that of the MDS method.

In the second scenario, the positions of the four anchor sensors remain unchanged but those of the 10 unknown-position sensors are located randomly according to the uniform distribution within the 20000 m^2 area in each independent run, and the results are shown in Fig. 3. For each set of WSN geometry, the performance bound is computed and we use the average as the CRLB. Comparing with the first test, both the three proposed and MDS methods have larger MSPEs but the former is still superior to the latter by around 1 to 3.5 dBm^2 for all SNRs. We observe that the full-set and distributed algorithms have comparable estimation performance and are superior to the minimum-set scheme. In summary, the results indicate that the subspace methods outperform the MDS solution for various sensor geometries in an average sense.

The above tests for fixed-position and random-position nodes have been repeated using another anchor geometry where the anchors are placed on the perimeter of the network, that is, they are located at (0,0)m, (100,0)m, (0,200)m, and (100,200)m, and their MSPEs are plotted in Figs. 4 and 5, respectively. It is seen that the findings are similar to those of Figs. 2 and 3 except that the performance of the minimum-set solution is slightly inferior to that of the MDS method.

VI. Conclusion

Three novel subspace based approaches have been devised for node localization in fully connected WSNs using distance measurements. Alternative realizations with smaller computational complexity are also proposed. In addition, theoretical performance measures, namely, biases and mean square errors of the sensor position estimates, are derived and verified by computer simulations. It is shown that the proposed positioning methods are generally superior to the classical multidimensional scaling scheme.

Apart from distance measurements, it is expected [15] that the proposed approaches will work with connectivity-only information because we can employ the shortest path distances between all pairs of nodes for constructing the distance matrix, although its theoretical performance seems to be unavailable. Fully connected network assumption is one obvious drawback of the proposed approaches. In case of partially connected isotropic networks, it is possible to use the average hop length and hop counts between indirectly connected nodes [5] or other means [16] to deduce the remaining requisite distances for the multidimensional similarity matrix construction. Or we can borrow the idea of [17] by building local maps from the fully connected subsets in the WSNs and then patching them together for global positioning. Nevertheless, further research is required for tackling anisotropic sensor networks [26]. In addition, it is believed that improvements of the proposed methodology can be made by finding the best minimum-set solution for a particular network geometry and devising the optimum weighting matrix [27] for solving (17) in a weighted least squares sense. Optimal anchor placement in WSNs is also an interesting research topic and a good starting point is to analytically study the corresponding Cramér-Rao lower bound [28], [29].

Appendix

The correlation matrix of the position estimate obtained by the full-set subspace algorithm is now derived as follows.

$$\begin{aligned}
&\mathbb{E}\left\{\Delta\boldsymbol{\phi}_{\text{fs}}\Delta\boldsymbol{\phi}_{\text{fs}}^T\right\}\\
&= \tilde{\mathbf{A}}_{\text{fs}}^{\dagger}\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^{\dagger}\otimes\tilde{\mathbf{U}}_n^T\right)\mathbb{E}\left\{\text{vec}(\Delta\mathbf{F}_{\text{fs}})\text{vec}^T(\Delta\mathbf{F}_{\text{fs}})\right\}\\
&\quad\times\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^{\dagger}\otimes\tilde{\mathbf{U}}_n^T\right)^T\left(\tilde{\mathbf{A}}_{\text{fs}}^{\dagger}\right)^T\\
&= \frac{1}{4}\tilde{\mathbf{A}}_{\text{fs}}^{\dagger}\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^{\dagger}\otimes\tilde{\mathbf{U}}_n^T\right)\\
&\quad\times\mathbb{E}\left\{\text{vec}(\mathbf{K}\Delta\mathbf{R}\mathbf{K}^T)\text{vec}^T(\mathbf{K}\Delta\mathbf{R}\mathbf{K}^T)\right\}\\
&\quad\times\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^{\dagger}\otimes\tilde{\mathbf{U}}_n^T\right)^T\left(\tilde{\mathbf{A}}_{\text{fs}}^{\dagger}\right)^T\\
&= \frac{1}{4}\tilde{\mathbf{A}}_{\text{fs}}^{\dagger}\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^{\dagger}\otimes\tilde{\mathbf{U}}_n^T\right)(\mathbf{K}\otimes\mathbf{K})\\
&\quad\times\mathbb{E}\left\{\text{vec}(\Delta\text{R})\text{vec}^T(\Delta\text{R})\right\}(\mathbf{K}^T\otimes\mathbf{K}^T)\\
&\quad\times\left(\tilde{\mathbf{X}}_{\text{fs}}^T\tilde{\mathbf{F}}_{\text{fs}}^{\dagger}\otimes\tilde{\mathbf{U}}_n^T\right)^T\left(\tilde{\mathbf{A}}_{\text{fs}}^{\dagger}\right)^T \qquad \text{(A1)}
\end{aligned}$$

where $[\Delta\mathbf{R}]_{i,j} = 2d_{i,j}q_{i,j} + q_{i,j}^2$. Essentially, only $\mathbb{E}\{\text{vec}(\Delta\mathbf{R})\text{vec}^T(\Delta\mathbf{R})\}$ needs to be computed as others are deterministic. Exploiting the symmetric structure of $\Delta\mathbf{R}$ yields

$$\begin{aligned}
\Delta\mathbf{R} &= \sum_{j=2}^{M}\sum_{i=1}^{j-1}[\mathbf{B}]_{i,j}[\Delta\mathbf{R}]_{i,j}\\
&\Rightarrow \text{vec}(\Delta\text{R}) = \sum_{j=2}^{M}\sum_{i=1}^{j-1}\text{vec}\left([\mathbf{B}]_{i,j}\right)[\Delta\mathbf{R}]_{i,j}\\
&\Rightarrow \mathbb{E}\left\{\text{vec}(\Delta\text{R})\text{vec}^T(\Delta\text{R})\right\}\\
&= \mathbb{E}\left\{\sum_{j_1=2}^{M}\sum_{i_1=1}^{j_1-1}\text{vec}\left([\mathbf{B}]_{i_1,j_1}\right)[\Delta\mathbf{R}]_{i_1,j_1}\right.\\
&\qquad\left.\times\sum_{j_2=2}^{M}\sum_{i_2=1}^{j_2-1}\text{vec}^T\left([\mathbf{B}]_{i_2,j_2}\right)[\Delta\mathbf{R}]_{i_2,j_2}\right\}\\
&= \boldsymbol{\Upsilon} + \boldsymbol{\Psi}
\end{aligned}$$

where

$$\begin{aligned}
[\mathbf{B}]_{i,j} &= \mathbf{e}_{M,i}\mathbf{e}_{M,j}^T + \mathbf{e}_{M,j}\mathbf{e}_{M,i}^T\\
\boldsymbol{\Upsilon} &= \mathbb{E}\left\{\sum_{j_1=2}^{M}\sum_{i_1=1}^{j_1-1}\text{vec}([\mathbf{B}]_{i_1,j_1})\,\text{vec}^T([\mathbf{B}]_{i_1,j_1})\,[\Delta\mathbf{R}]_{i_1,j_1}^2\right\}
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\Psi} = \mathbb{E}&\left\{\sum_{j_1=2}^{M}\sum_{i_1=1}^{j_1-1}\sum_{\substack{j_2=2\\ i_2\neq i_1}}^{M}\sum_{\substack{i_2=1\\ j_2\neq j_1}}^{j_2-1}\text{vec}\left([\mathbf{B}]_{i_1,j_1}\right)[\Delta\mathbf{R}]_{i_1,j_1}\right.\\
&\qquad\left.\times\text{vec}^T\left([\mathbf{B}]_{i_2,j_2}\right)[\Delta\mathbf{R}]_{i_2,j_2}\right\}.
\end{aligned}$$

Exploiting $\text{vec}(\mathbf{e}_{M,i}\mathbf{e}_{M,j}^T) = \mathbf{e}_{M,i}\otimes\mathbf{e}_{M,j} = \mathbf{e}_{M^2,(i-1)M+j}$ and the property of $\mathbb{E}\{q_{i,j}q_{k,l}q_{m,n}q_{o,p}\} = \mathbb{E}\{q_{i,j}q_{k,l}\}\mathbb{E}\{q_{m,n}q_{o,p}\}+\mathbb{E}\{q_{i,j}q_{m,n}\}\mathbb{E}\{q_{k,l}q_{o,p}\} + \mathbb{E}\{q_{i,j}q_{o,p}\}\mathbb{E}\{q_{k,l}q_{m,n}\}$, $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Psi}$ can be simplified as

$$\begin{aligned}
\boldsymbol{\Upsilon} = \sum_{j=2}^{M}\sum_{i=1}^{j-1}&\left(\mathbf{e}_{M^2,(i-1)M+j} + \mathbf{e}_{M^2,(j-1)M+i}\right)\\
&\times\left(\mathbf{e}_{M^2,(i-1)M+j} + \mathbf{e}_{M^2,(j-1)M+i}\right)^T\\
&\times\left(4d_{i,j}^2\sigma_{i,j}^2 + 3\sigma_{i,j}^4\right) \qquad \text{(A2)}
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\Psi} = &\sum_{j_1=2}^{M}\sum_{i_1=1}^{j_1-1}\sum_{\substack{j_2=2\\ i_2\neq i_1}}^{M}\sum_{\substack{i_2=1\\ j_2\neq j_1}}^{j_2-1}\text{vec}\left([\mathbf{B}]_{i_1,j_1}\right)\text{vec}^T\left([\mathbf{B}]_{i_2,j_2}\right)\\
&\times\sigma_{i_1,j_1}^2\sigma_{i_2,j_2}^2\\
= &\,\text{vec}(\boldsymbol{\Sigma})\text{vec}^T(\boldsymbol{\Sigma})\odot\left(\mathbf{1}_{M^2}\mathbf{1}_{M^2}^T - \text{sgn}(\boldsymbol{\Upsilon})\right)\big). \qquad \text{(A3)}
\end{aligned}$$

where $\text{sgn}(\cdot)$ is the sign function. Note that in(A2), $[\boldsymbol{\Upsilon}]_{i,j} = 0$ except for the following elements:

$$\begin{aligned}
[\boldsymbol{\Upsilon}]_{(i-1)M+j,(i-1)M+j} &= [\boldsymbol{\Upsilon}]_{(i-1)M+j,(j-1)M+i}\\
&= [\boldsymbol{\Upsilon}]_{(j-1)M+i,(i-1)M+j}\\
&= [\boldsymbol{\Upsilon}]_{(j-1)M+i,(j-1)M+i}\\
&= 4d_{i,j}^2\sigma_{i,j}^2 + 3\sigma_{i,j}^4,\\
&\quad j = 2,3,\cdots,M, i = 1,2,\cdots,j-1
\end{aligned}$$

Hence,

$$\boldsymbol{\Upsilon} + \boldsymbol{\Psi} = \check{\boldsymbol{\Upsilon}} + \text{vec}(\boldsymbol{\Sigma})\text{vec}^T(\boldsymbol{\Sigma}) \qquad \text{(A4)}$$

where

$$\begin{aligned}[\check{\boldsymbol{\Upsilon}}]_{(i-1)M+j,(i-1)M+j} &= [\check{\boldsymbol{\Upsilon}}]_{(i-1)M+j,(j-1)M+i} \\ &= \begin{cases} 4d_{i,j}^2\sigma_{i,j}^2 + 2\sigma_{i,j}^4 & : \quad i,j = 1,2,\cdots,M, i \neq j \\ 0 & : \quad \text{otherwise} \end{cases}\end{aligned}$$

is the covariance matrix of $\mathbb{E}\{\text{vec}(\boldsymbol{\Delta}\mathbf{R})\text{vec}^T(\boldsymbol{\Delta}\mathbf{R})\}$. Substituting (A4) into (A1) yields $\mathbb{E}\{\boldsymbol{\Delta\phi}_{\text{fs}}\boldsymbol{\Delta\phi}_{\text{fs}}^T\}$.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their careful reading and constructive comments, which improved the clarity of this paper.

REFERENCES

[1] A. Hać, *Wireless Sensor Network Designs*. Hoboken, NJ: Wiley, 2003.

[2] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. London, U.K.: CRC, 2005.

[3] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Hoboken, NJ: Wiley, 2005.

[4] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2001, pp. 1655–1663.

[5] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE GlobeCom*, San Antonio, TX, Nov. 2001, vol. 5, pp. 2926–2931.

[6] R. L. Moses, D. Krishnamurthy, and R. M. Patterson, "A self-localization method for wireless sensor networks," *Proc. EURASIP J. Appl. Signal Process.*, vol. 4, pp. 348–358, 2003.

[7] N. Patwari, A. O. Hero, III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, August 2003.

[8] M. Meurer, P. W. Baier, T. Weber, C. A. Jotten, and S. Heilmann, "SPIDER: Enhanced distance based localization of mobile radio terminals," in *Proc. VTC 2004-Fall*, Los Angeles, CA, Sept. 2004, vol. 5, pp. 3521–3525.

[9] A. A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based wireless sensor network localization with flip ambiguity mitigation," in *Proc. VTC 2006-Spring*, Melbourne, Australia, May 2006, vol. 2, pp. 1022–1026.

[10] P. Biswas, T.-C. Liang, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sens. Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.

[11] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Auto. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.

[12] Z. Wei, Large Scale Sensor Network Localization Dep. Statist., Stanford Univ., 2006, Tech. Rep.

[13] A. Savvides, C.-C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. 7th Ann. Int. Conf. Mobile Comput. Netw.*, Rome, Italy, Jul. 2001, pp. 166–179, ACM Press.

[14] C. Savarese, J. M. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor network," in *Proc. ICASSP 2001*, Salt Lake City, UT, May 2001, vol. 4, pp. 2037–2040.

[15] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, pp. 961–974, Nov. 2004.

[16] P. Drineas, A. Jayed, M. Magdon-Ismail, G. Pandurangan, R. Virrankoski, and A. Savvides, "Distance matrix reconstruction from incomplete distance information for sensor network localization," in *Proc. IEEE SECON 2006*, Reston, VA, Sep. 2006, pp. 536–544.

[17] X. Ji and H. Zha, "Robust sensor localization algorithm in wireless ad-hoc sensor networks," in *Proc. Comput. Commun. Netw.*, Oct. 2003, pp. 527–532.

[18] M. Coates, "Distributed particle filters for sensor networks," in *Proc. IEEE Inf. Process. Sens. Netw.*, Apr. 2004, pp. 225–233.

[19] T. Ihler, J. W. Fisher, III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 809–819, Apr. 2005.

[20] Q. Shi, S. Kyperountas, N. S. Correal, and F. Niu, "Performance analysis of relative location estimation for multihop wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 830–838, Apr. 2005.

[21] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, pp. 54–69, Jul. 2005.

[22] Q. Wan, Y.-J. Luo, W.-L. Yang, J. Xu, J. Tang, and Y.-N. Peng, "Mobile localization method based on multidimensional similarity analysis," in *Proc. ICASSP 2005*, Phoenix, AZ, vol. IV, pp. 1081–1084.

[23] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proc. RADC Spectral Estimation Workshop*, Rome, NY, 1979, pp. 243–258.

[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The John Hopkins Univ. Press, 1996, pp. 18, 89, 231–232.

[25] A. J. Weiss and B. Friedlander, "Synchronous DS-CDMA downlink with frequency selective fading," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 158–167, Jan. 1999.

[26] H. Lim and J. C. Hou, "Localization for anisotropic sensor networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 138–149.

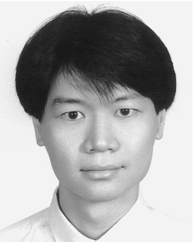
[27] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood and Cramér-Rao bound: Further results and comparisons," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 38, no. 12, pp. 2140–2150, 1990.

[28] Y. Shang, H. Shi, and A. A. Ahmed, "Performance study of localization methods for ad-hoc sensor networks," in *Proc. IEEE Int. Conf. Mobile Ad-hoc and Sens. Syst.*, Fort Lauderdale, FL, Oct. 2004, pp. 184–193.

[29] B. Yang and J. Scheuing, "Cramér-Rao bound and optimum sensor array for source localization from time differences of arrival," in *Proc. ICASSP*, Philadelphia, PA, Mar. 2005, vol. 4, pp. 961–964.

Frankie K. W. Chan received the B.Eng. degree in computer engineering and the M.Phil. degree from the City University of Hong Kong in 2002 and 2005, respectively.

His research interests include statistical signal processing and their applications, with particular attention to frequency estimation, sensor network localization, and related mathematics.



H. C. So (SM'00) was born in Hong Kong. He received the B.Eng. degree from the City University of Hong Kong in 1990 and the Ph.D. degree from The Chinese University of Hong Kong in 1995, both in electronic engineering.

From 1990 to 1991, he was an Electronic Engineer at the Research & Development Division of Everex Systems Engineering, Ltd., Hong Kong. From 1995 to 1996, he worked as a Postdoctoral Fellow at The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor at the Department of Electronic Engineering, City University of Hong Kong. Currently, he is an Associate Professor in the Department of Electronic Engineering at City University of Hong Kong. His research interests include fast and adaptive filter, signal detection, parameter estimation, and source localization.

W.-K. Ma (M'01) received the B.Eng. (First-Class Hons.) in electrical and electronic engineering from the University of Portsmouth, Portsmouth, U.K., in 1995, and the M.Phil. and Ph.D. degrees, both in electronic engineering, from the Chinese University of Hong Kong (CUHK), Hong Kong, in 1997 and 2001, respectively.

He is currently an Assistant Professor in the Department of Electronic Engineering, CUHK. He was with the Institute of Communications Engineering, National Tsing Hua University, Taiwan, also as an Assistant Professor, from 2005 to 2007. He still holds an adjunct position there. Prior to becoming a faculty member, he held various research positions at McMaster University, Canada; CUHK, Hong Kong; and the University of Melbourne, Australia. His research interests are in signal processing and communications, with a recent emphasis on MIMO techniques and convex optimization.

Dr. Ma's Ph.D. dissertation was commended to be of very high quality and received well-deserved honorary mention by the Faculty of Engineering, CUHK, in 2001. Since October 2007, he has been an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.