Correspondence

Efficient Weighted Multidimensional Scaling for Wireless Sensor Network Localization

Frankie K. W. Chan and H. C. So

Abstract—**Localization of sensor nodes is a fundamental and important problem in wireless sensor networks. Although classical multidimensional scaling (MDS) is a computationally attractive positioning method, it is statistically inefficient and cannot be applied in partially-connected sensor networks. In this correspondence, a weighted MDS algorithm is devised to circumvent these limitations. It is proved that the estimation performance of the proposed algorithm can attain Cramér–Rao lower bound (CRLB) for sufficiently small noise conditions. Computer simulations are included to contrast the performance of the proposed algorithm with the classical MDS and distributed weighted MDS algorithms as well as CRLB.**

Index Terms—**Localization, multidimensional scaling, wireless sensor networks.**

I. Introduction

Wireless sensor networks (WSNs), consisting of inexpensive smart nodes, have gained intensive research interest over the past few years. WSNs have wide applications in military, medical and commercial areas such as battlefield surveillance, doctor and patient tracking inside a hospital, machine diagnosis as well as building automation [1]–[4]. Usually, the sensor nodes are deployed randomly. To accomplish the mentioned applications, the positions of the sensor nodes must be estimated first. As a result, the sensor localization is one of the important signal processing tasks in WSNs. Generally, sensor positioning involves two steps: first, observations related to the location information between the sensors are measured, and typical measurement models include time-of-arrival (TOA), time-difference-of-arrival, angle-of-arrival, and received-signal-strength [5]. Second, positions of the nodes are estimated with the use of the distance or bearing information derived from the measurements and locations of the known-position sensors, referred to as anchors. The main focus of this correspondence lies in the second step as we assume that the TOA information is already converted to noisy distance measurements.

Positioning algorithms in WSNs can be categorized as centralized and distributed approaches. In centralized localization, all distance measurements are sent to a central unit for calculating the sensor positions. Centralized processing is advantageous in the sense that the solution obtained is generally more accurate and a global map is available. However, the central processor may be unavailable in some applications and is unable to perform heavy computation particularly for large-scale sensor networks. While in distributed localization, each node performs self-localization using the distances it measures and the local information it collects from its neighbors. Without the need of a powerful central processor, the computational burden is shared among all nodes in this scalable approach at the expense of possibly performance degradation. The algorithm development in this work corresponds to the centralized approach. For comprehensive discussions on centralized and distributed localization, the interested reader is referred to [1], [3], and [6].

Assuming that the distance measurements are corrupted by white Gaussian noise, the maximum-likelihood (ML) algorithm provides optimal estimation accuracy [7], [8]. As the ML cost function is multimodal, sufficiently close initial guess is required to obtain the global solution. On the other hand, semidefinite relaxation (SDR) algorithm [9] approximates the sensor localization problem by a convex optimization and hence global solution is guaranteed. A computationally attractive range-based positioning technique is to employ classical multidimensional scaling (MDS) [4], [10]–[12] which transforms the pairwise distance information into the relative coordinates of nodes. However, the MDS method is suboptimal and it requires all pairwise distance measurements of sensors to produce the global solution. These requirements may not be satisfied in practice and thus limit its application. Metric MDS [10], [11] is an improvement to the classical MDS and a representative technique is the scaling by majorizing a complicated function (SMACOF) algorithm [13]. Based on SMACOF, Costa *et al.* have proposed a distributed weighted multidimensional scaling (dwMDS) algorithm [14], which corresponds to nonlinear weighted least squares (WLS) methodology, to increase node localization accuracy. In this correspondence, we exploit classical MDS to develop a weighted multidimensional scaling (WMDS) position estimator whose accuracy can attain Cramér–Rao lower bound (CRLB) for sufficiently small noise conditions. Unlike the dwMDS algorithm, the proposed WMDS algorithm is based on linear WLS. The proposed algorithm also generalizes the work of [15] which considers single source localization. Our contributions are summarized as i) developing accurate range-based positioning algorithm for multiple sensor localization, ii) extending the proposed algorithm to operate in partially-connected WSNs, and iii) producing the theoretical performance of the proposed algorithm. It is noteworthy that our previous works on source localization include [16]–[18]. Both [16] and [17] address the problem of single-source positioning where the former is a modified version of the classical MDS scheme while the latter utilizes the subspace technique. The work of [18] generalizes [17] to positioning of multiple nodes in WSNs. As the standard least squares (LS) technique is employed in [16]–[18], their estimation performance is suboptimal. While WLS is exploited in this work to attain optimum estimation accuracy in node localization.

The rest of the correspondence is organized as follows. The problem formulation and algorithm development of the WMDS algorithm are presented in Section II. In Section III, performance analysis of the WMDS algorithm is provided. The WMDS methodology is extended to partially connected WSNs in Section IV. In Section V, numerical examples are provided to demonstrate the performance of the WMDS algorithm by comparing with classical MDS and dwMDS methods as well as CRLB. Finally, conclusions are drawn in Section VI. Before proceeding further, the mathematical symbols used in this correspondence are first introduced in Table I.

Manuscript received October 24, 2008; accepted May 06, 2009. First published June 10, 2009; current version published October 14, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhi Tian. This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 119606).

The authors are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: ckwf@hkexperts.com).

Digital Object Identifier 10.1109/TSP.2009.2024869

1053-587X/$26.00 © 2009 IEEE

TABLE I
LIST OF SYMBOLS

Symbol	Meaning
^{T}	transpose
$^{-1}$	inverse
†	pseudo inverse
$\|\cdot\|$	l_2 norm
vec	vectorization
$\otimes$	Kronecker product
$\mathbb{E}$	expectation operator
$\mathbf{1}_i$	$i \times 1$ vector with all elements 1
$\mathbf{I}_i$	$i \times i$ identity matrix
$\mathbf{0}_{i\times j}$	$i \times j$ zero matrix
$\mathbf{e}_i$	ith column of $\mathbf{I}_{\frac{M(M-1)}{2}}$
$\breve{\mathbf{e}}_i$	ith column of $\mathbf{I}_M$
$\grave{\mathbf{e}}_i$	ith column of $\mathbf{I}_{M-k}$
$\min(i,j)$	return i if $i \le j$ and j otherwise
$\max(i,j)$	return i if $i \ge j$ and j otherwise
diag$(\cdot)$	diagonal matrix
cov$(\cdot)$	covariance
$\emptyset$	empty set
$\{\phi\}_{i=1}^k$	$\{\phi_1, \phi_2, \cdots, \phi_k\}$
$\mathcal{N}(\mu, \sigma^2)$	Gaussian density with mean μ and variance σ^2

II. WEIGHTED MULTIDIMENSIONAL SCALING

In this section, the problem of WSN localization is formulated and the WMDS algorithm will be devised.

A. Problem Formulation

We consider two-dimensional WSN localization problem as extension to three-dimensional case is straightforward. Suppose there are M nodes. The position of ith sensor is denoted by $\boldsymbol{\phi}_i = [x_i \; y_i]^T$. The distance measurement between the ith and jth sensor is

$$r_{i,j} = r_{j,i} = d_{i,j} + q_{i,j}, \qquad i,j = 1,2,\cdots,M \tag{1}$$

where $d_{i,j} = \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|$ is the noise-free distance and $q_{i,j} \sim \mathcal{N}(0, \sigma_{i,j}^2)$ is the uncorrelated noise. We assume that $\{\sigma_{i,j}^2\}$ have been accurately estimated and are known *a priori* [7], [9], [19]. Without loss of generality, let $\boldsymbol{\phi}_i$, $i = 1,2,\cdots,k$, be the positions of anchors where $k \ge 3$. The task is to estimate $\{\boldsymbol{\phi}_i\}_{i=k+1}^M$ using $\{r_{i,j}\}$ and $\{\boldsymbol{\phi}_i\}_{i=1}^k$.

It should be pointed out that under practical situations especially in indoor environments, non-line-of-sight (NLOS) propagation is a major source of error, which can introduce large positive biases in the TOA measurements and result in unreliable position estimation. As recent works [20], [21] on source localization under NLOS environments have reported some promising results, we reasonably assume that the NLOS errors in the TOA measurements have been successfully mitigated in this work.

B. Algorithm Development

Assuming that all pairwise noise-free distances are available, the MDS algorithm utilizes the following matrix:

$$\mathbf{B} = \mathbf{J}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{J} = -0.5\mathbf{J}\mathbf{R}\mathbf{J} \tag{2}$$

where

$$\begin{aligned}
\mathbf{J} &= \mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T \\
\mathbf{R} &= \begin{bmatrix} 0 & d_{1,2}^2 & d_{1,3}^2 & \cdots & d_{1,M}^2 \\ d_{2,1}^2 & 0 & d_{2,3}^2 & \cdots & d_{2,M}^2 \\ d_{3,1}^2 & d_{3,2}^2 & 0 & \cdots & d_{3,M}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{M,1}^2 & d_{M,2}^2 & d_{M,3}^2 & \cdots & 0 \end{bmatrix} \\
\boldsymbol{\Phi} &= [\boldsymbol{\Phi}_a \quad \boldsymbol{\Phi}_u] \\
\boldsymbol{\Phi}_a &= [\boldsymbol{\phi}_1 \quad \boldsymbol{\phi}_2 \quad \cdots \quad \boldsymbol{\phi}_k] \\
\boldsymbol{\Phi}_u &= [\boldsymbol{\phi}_{k+1} \quad \boldsymbol{\phi}_{k+2} \quad \cdots \quad \boldsymbol{\phi}_M].
\end{aligned}$$

Without loss of generality, we assume that $\boldsymbol{\Phi}_a\mathbf{1}_k = \mathbf{0}_{2\times 1}$ which is fulfilled by a simple translation using $\boldsymbol{\phi}_i - \sum_{j=1}^k \boldsymbol{\phi}_j/k$, $j = 1,2,\cdots,k$, and thus the actual sensor positions can be recovered in a similar manner. Partitioning $\mathbf{J}$ into upper and lower parts, we have

$$\boldsymbol{\Phi}\mathbf{J} = \boldsymbol{\Phi}_a\mathbf{J}_a + \boldsymbol{\Phi}_u\mathbf{J}_u \tag{3}$$

where

$$\begin{aligned}
\mathbf{J} &= \begin{bmatrix}\mathbf{J}_a^T & \mathbf{J}_u^T\end{bmatrix}^T \\
\mathbf{J}_a &= \begin{bmatrix}\mathbf{I}_k - \frac{1}{M}\mathbf{1}_k\mathbf{1}_k^T & -\frac{1}{M}\mathbf{1}_k\mathbf{1}_{M-k}^T\end{bmatrix} \\
\mathbf{J}_u &= \begin{bmatrix}-\frac{1}{M}\mathbf{1}_{M-k}\mathbf{1}_k^T & \mathbf{I}_{M-k} - \frac{1}{M}\mathbf{1}_{M-k}\mathbf{1}_{M-k}^T\end{bmatrix}.
\end{aligned}$$

By further rearrangement, (3) can be expressed as

$$\boldsymbol{\Phi}\mathbf{J} = \breve{\boldsymbol{\Phi}}_u\breve{\boldsymbol{\Phi}}_a \tag{4}$$

where

$$\begin{aligned}
\breve{\boldsymbol{\Phi}}_u &= [\mathbf{I}_2 \quad \boldsymbol{\Phi}_u] \\
\breve{\boldsymbol{\Phi}}_a &= \begin{bmatrix}\mathbf{J}_a^T\boldsymbol{\Phi}_a^T & \mathbf{J}_u^T\end{bmatrix}^T \\
\mathbf{J}_a^T\boldsymbol{\Phi}_a^T &= \begin{bmatrix}\boldsymbol{\Phi}_a & \mathbf{0}_{2\times(M-k)}\end{bmatrix}^T.
\end{aligned} \tag{5}$$

Employing matrix inversion lemma, $\breve{\boldsymbol{\Phi}}_a^\dagger$ can be simplified using (5) as

$$\breve{\boldsymbol{\Phi}}_a^\dagger = \breve{\boldsymbol{\Phi}}_a^T\left(\breve{\boldsymbol{\Phi}}_a\breve{\boldsymbol{\Phi}}_a^T\right)^{-1} = \begin{bmatrix}\boldsymbol{\Phi}_a^\dagger & \frac{-1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T \\ \mathbf{0}_{(M-k)\times 2} & \mathbf{I}_{M-k}\end{bmatrix}. \tag{6}$$

From (4) and (6), we get

$$\boldsymbol{\Phi}\mathbf{J}\breve{\boldsymbol{\Phi}}_a^\dagger = \breve{\boldsymbol{\Phi}}_u \Rightarrow \boldsymbol{\Phi}\mathbf{J}\breve{\boldsymbol{\Phi}}_a^\dagger\grave{\boldsymbol{\Phi}}_u = \mathbf{0}_{2\times(M-k)} \tag{7}$$

where

$$\grave{\boldsymbol{\Phi}}_u = \begin{bmatrix}\boldsymbol{\Phi}_u^T & -\mathbf{I}_{M-k}\end{bmatrix}^T.$$

Combining (7) and (2) yields

$$\mathbf{B}\breve{\boldsymbol{\Phi}}_a^\dagger\grave{\boldsymbol{\Phi}}_u = \mathbf{0}_{M\times(M-k)}. \tag{8}$$

Basically, (8) corresponds to a set of linear equations with unknown $\boldsymbol{\Phi}_u$ and it generalizes the work of [15] which addresses single source localization. Furthermore, unlike [15], the derivation of (8) does not involve eigenvalue decomposition (EVD).

In practice, $\mathbf{B}$ is approximated by $\hat{\mathbf{B}}$ which is constructed by the noisy distance measurements:

$$\hat{\mathbf{B}} = -0.5\mathbf{J}\hat{\mathbf{R}}\mathbf{J} \tag{9}$$

where

$$\hat{\mathbf{R}} = \begin{bmatrix} 0 & r_{1,2}^2 & r_{1,3}^2 & \cdots & r_{1,M}^2 \\ r_{2,1}^2 & 0 & r_{2,3}^2 & \cdots & r_{2,M}^2 \\ r_{3,1}^2 & r_{3,2}^2 & 0 & \cdots & r_{3,M}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{M,1}^2 & r_{M,2}^2 & r_{M,3}^2 & \cdots & 0 \end{bmatrix}$$

and (8) does not hold. Note that in the classical MDS approach, EVD is performed on $\hat{\mathbf{B}}$ to give a LS estimate for $\mathbf{B}$ which corresponds to a standard LS scheme. While our proposed approach constructs linear equations from $\hat{\mathbf{B}}$ without using EVD and exploits WLS to solve the equations as follows.

Let $\mathbf{\Xi}$ be the left-hand side of (8):

$$\mathbf{\Xi} = \hat{\mathbf{B}}\breve{\mathbf{\Phi}}_a^{\dagger}\grave{\mathbf{\Phi}}_u = \mathbf{H}_L\mathbf{\Phi}_u - \mathbf{H}_R \tag{10}$$

where

$$\begin{aligned} \mathbf{H}_L &= \hat{\mathbf{B}} \begin{bmatrix} \mathbf{\Phi}_a^{\dagger} \\ \mathbf{0}_{(M-k)\times 2} \end{bmatrix} \\ \mathbf{H}_R &= \hat{\mathbf{B}} \begin{bmatrix} \frac{-1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T \\ \mathbf{I}_{M-k} \end{bmatrix} \end{aligned}$$

Vectorizing both sides of (10) yields

$$\boldsymbol{\xi} = \mathrm{vec}(\mathbf{\Xi}) = \breve{\mathbf{H}}_L\boldsymbol{\phi}_u - \mathbf{h}_R. \tag{11}$$

where

$$\begin{aligned} \breve{\mathbf{H}}_L &= (\mathbf{I}_{M-k} \otimes \mathbf{H}_L) \\ \boldsymbol{\phi}_u &= \mathrm{vec}(\mathbf{\Phi}_u) \\ \mathbf{h}_R &= \mathrm{vec}(\mathbf{H}_R). \end{aligned}$$

As the noises in $\boldsymbol{\xi}$ are not independent and identically distributed, it is well known [22] that the standard LS estimator can only provide a suboptimal solution. To improve accuracy, we utilize WLS technique and the corresponding cost function to be minimized is given by

$$\min_{\boldsymbol{\phi}_u}(\breve{\mathbf{H}}_L\boldsymbol{\phi}_u - \mathbf{h}_R)^T\mathbf{W}(\breve{\mathbf{H}}_L\boldsymbol{\phi}_u - \mathbf{h}_R) \tag{12}$$

where $\mathbf{W}$ is the weighting matrix and an optimal choice is the Markov estimate [19], [23]:

$$\mathbf{W} = \left\{\mathbb{E}(\boldsymbol{\xi}\boldsymbol{\xi}^T)\right\}^{\dagger}. \tag{13}$$

This WLS estimator is also known as the best linear unbiased estimator [22]. The inverse of (13) is calculated by rewriting (11) as follows:

$$\boldsymbol{\xi} = -0.5\mathrm{vec}\left(\mathbf{J}\hat{\mathbf{R}}\breve{\mathbf{\Phi}}_a^{\dagger}\grave{\mathbf{\Phi}}_u\right) = \mathbf{F}\mathrm{vec}(\hat{\mathbf{R}}) \tag{14}$$

where

$$\mathbf{F} = -0.5\left(\left(\breve{\mathbf{\Phi}}_a^{\dagger}\grave{\mathbf{\Phi}}_u\right)^T \otimes \mathbf{J}\right).$$

Note that from (6), $\mathbf{1}_M^T\breve{\mathbf{\Phi}}_a^{\dagger} = \mathbf{0}_{1\times M}$ and hence $\mathbf{J}\breve{\mathbf{\Phi}}_a^{\dagger} = \breve{\mathbf{\Phi}}_a^{\dagger}$ in (14) is obtained. Expressing $\mathrm{vec}(\hat{\mathbf{R}})$ as $\mathbf{\Psi}\mathbf{r}$, (14) can be written as

$$\mathbf{F}\mathrm{vec}(\hat{\mathbf{R}}) = \mathbf{F}\mathbf{\Psi}\mathbf{r} \tag{15}$$

where

$$\begin{aligned} \mathbf{\Psi} &= \begin{bmatrix} \mathbf{\Psi}_1^T & \mathbf{\Psi}_2^T & \cdots & \mathbf{\Psi}_M^T \end{bmatrix}^T \\ \mathbf{\Psi}_i &= \begin{bmatrix} \mathbf{E}_{\beta,i}^T & \mathbf{0}_{\frac{M(M-1)}{2}\times 1} & \mathbf{E}_{\gamma,i}^T \end{bmatrix}^T \\ \mathbf{E}_{\beta,i} &= \begin{bmatrix} \mathbf{e}_{\alpha(i,1)} & \cdots & \mathbf{e}_{\alpha(i,i-1)} \end{bmatrix}^T \\ \mathbf{E}_{\gamma,i} &= \begin{bmatrix} \mathbf{e}_{\alpha(i+1,i)} & \cdots & \mathbf{e}_{\alpha(M,i)} \end{bmatrix}^T \\ \alpha(i,j) &= 0.5(2jM - 2M - j^2 - j) + i \\ \mathbf{r} &= \begin{bmatrix} r_{1,2}^2 & \cdots & r_{1,M}^2 & r_{2,3}^2 \cdots & r_{2,M}^2 & \cdots & r_{M-1,M}^2 \end{bmatrix}^T. \end{aligned}$$

Squaring (1) and assuming that the noise power is sufficiently small, we have

$$\begin{aligned} r_{i,j}^2 - d_{i,j}^2 &= 2d_{i,j}q_{i,j} + q_{i,j}^2, \quad \max(i,j) > k \\ &\approx 2d_{i,j}q_{i,j}. \end{aligned} \tag{16}$$

The covariance of $r_{i,j}^2$ can then be approximated by

$$\mathrm{cov}\left(r_{i,j}^2\right) \approx 4r_{i,j}^2\sigma_{i,j}^2. \tag{17}$$

Based on (13) and (17), $\mathbf{W}$ is calculated as

$$\mathbf{W} = (\mathbf{F}\breve{\mathbf{\Psi}}\mathbf{Q}\breve{\mathbf{\Psi}}^T\mathbf{F}^T)^{\dagger} \tag{18}$$

where

$$\begin{aligned} \mathbf{Q} &= 4\mathbf{\Lambda}\mathbf{\Sigma}\mathbf{\Lambda} \\ \mathbf{\Sigma} &= \mathrm{diag}\left(\sigma_{1,k+1}^2, \cdots, \sigma_{M-1,M}^2\right) \\ \mathbf{\Lambda} &= \mathrm{diag}(r_{1,k+1}, \cdots, r_{M-1,M}). \end{aligned}$$

Here, $\breve{\mathbf{\Psi}}$ equals $\mathbf{\Psi}$ but with the columns corresponds to those distances between anchors being removed. The columns removed are contained in the set $\mathbf{\Theta}(M,k) = \{(i-1)M - (i(i-1)/2) + 1, \cdots, (i-1)M - (i(i-1)/2) + k - i | i = 1, 2, \cdots, k-1\}$. To speed up the weighting matrix inversion, fast algorithms are available [24], [25]. The WLS position estimate of (12) is

$$\hat{\boldsymbol{\phi}}_u = \left(\breve{\mathbf{H}}_L^T\mathbf{W}\breve{\mathbf{H}}_L\right)^{-1}\breve{\mathbf{H}}_L^T\mathbf{W}\mathbf{h}_R. \tag{19}$$

As the ideal weighting matrix for $\mathbf{W}$ is a function of $\boldsymbol{\phi}_u$ which is to be determined, an iterative procedure to update weighting matrix is thus required. In the beginning, $\mathbf{W}$ is set to $\mathbf{I}_{M(M-k)}$ to get an initial estimate of $\boldsymbol{\phi}_u$. The whole WMDS algorithm is summarized as follows.

1) Set $\mathbf{W} = \mathbf{I}_{M(M-k)}$.
2) Calculate $\hat{\boldsymbol{\phi}}_u$ using (19).
3) Construct $\mathbf{W}$ using (18).
4) Repeat Steps 2) to 3) until a stopping criterion is satisfied.

The typical stopping criteria are i) norm of parameter difference in two successive iterations is less than a small positive number, ii) number of iterations reaches a certain limit, and iii) combination of i) and ii).

It is worthy to point out that this iterative weighting strategy has been a standard technique [26] and is commonly referred to as the iterative quadratic maximum-likelihood (IQML) approach [27], [28] in the field of signal processing. To the best of our knowledge, the global convergence for this form of iterative relaxation algorithm has not yet been proved. While local convergence proof for the IQML algorithm is available for limited applications [29]. Nevertheless, simulation results in Section V show that the algorithm is able to achieve the global solution with performance attaining the CRLB when the noises in the distance measurements are sufficiently small.

III. Performance Analysis

In this section, we are going to prove that the covariance of $\hat{\boldsymbol{\phi}}$ can attain the CRLB [1] if $\{\sigma_{i,j}^2\}$ are sufficiently small. The covariance of the WLS position estimate of (19) is well approximated by using the noise-free $\breve{\mathbf{H}}_L$ and the actual weighting matrix [22]

$$\begin{aligned}\operatorname{cov}(\hat{\boldsymbol{\phi}}_u) &\approx \left(\breve{\mathbf{H}}_L^T \mathbf{W} \breve{\mathbf{H}}_L\right)^{-1} \\ &= \left\{\left(\mathbf{I}_{M-k} \otimes \mathbf{B}\begin{bmatrix}\boldsymbol{\Phi}_a^\dagger \\ \mathbf{0}_{(M-k)\times 2}\end{bmatrix}\right)^T (\mathbf{F}\breve{\boldsymbol{\Psi}}\mathbf{Q}\breve{\boldsymbol{\Psi}}^T\mathbf{F}^T)^\dagger \right.\\ &\qquad \left. \times \left(\mathbf{I}_{M-k} \otimes \mathbf{B}\begin{bmatrix}\boldsymbol{\Phi}_a^\dagger \\ \mathbf{0}_{(M-k)\times 2}\end{bmatrix}\right)\right\}^{-1}. \end{aligned} \tag{20}$$

On the other hand, the Fisher information matrix (FIM) of $\boldsymbol{\phi}_u$ is [22]

$$\text{FIM} = \mathbf{D}^T \boldsymbol{\Sigma}^{-1} \mathbf{D} \tag{21}$$

where

$$\mathbf{D} = \begin{bmatrix} \frac{\boldsymbol{\phi}_{k+1}^T - \boldsymbol{\phi}_1^T}{d_{1,k+1}} & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \frac{\boldsymbol{\phi}_{k+2}^T - \boldsymbol{\phi}_1^T}{d_{1,k+2}} & \cdots & \mathbf{0}_{1\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \cdots & \frac{\boldsymbol{\phi}_M^T - \boldsymbol{\phi}_1^T}{d_{1,M}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \cdots & \frac{\boldsymbol{\phi}_M^T - \boldsymbol{\phi}_{M-1}^T}{d_{M-1,M}} \end{bmatrix}.$$

Comparing (20) and (21), it suffices to show that

$$\mathbf{D} = \breve{\boldsymbol{\Lambda}}^{-1} (\mathbf{F}\breve{\boldsymbol{\Psi}})^\dagger \breve{\mathbf{H}}_L \tag{22}$$

where

$$\breve{\boldsymbol{\Lambda}} = \operatorname{diag}(d_{1,k+1}, d_{1,k+2}, \cdots, d_{M-1,M}).$$

Proving (22) is equivalent to show that

$$\left(\left(\breve{\boldsymbol{\Phi}}_a^\dagger \grave{\boldsymbol{\Phi}}_u\right)^T \otimes \mathbf{J}\right) \breve{\boldsymbol{\Psi}}\breve{\boldsymbol{\Lambda}}\mathbf{D} = \mathbf{I}_{M-k} \otimes \left(\mathbf{B}\begin{bmatrix}\boldsymbol{\Phi}_a^\dagger \\ \mathbf{0}_{(M-k)\times 2}\end{bmatrix}\right). \tag{23}$$

The left-hand side of (23) can be expressed as

$$\left(\left(\breve{\boldsymbol{\Phi}}_a^\dagger \grave{\boldsymbol{\Phi}}_u\right)^T \otimes \mathbf{J}\right) \breve{\boldsymbol{\Psi}}\breve{\boldsymbol{\Lambda}}\mathbf{D} = \begin{bmatrix}\mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_{2(M-k)-1} & \mathbf{g}_{2(M-k)}\end{bmatrix} \tag{24}$$

where

$$\begin{aligned} \mathbf{g}_i &= \operatorname{vec}\left(\mathbf{J}\mathbf{G}_i\left(\breve{\boldsymbol{\Phi}}_a^\dagger \grave{\boldsymbol{\Phi}}_u\right)\right) \\ &= \operatorname{vec}\left(\mathbf{J}\mathbf{G}_i\begin{bmatrix}\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T \\ -\mathbf{I}_{M-k}\end{bmatrix}\right) \\ \mathbf{G}_i &= \begin{cases}\mathbf{S}_i + \mathbf{S}_i^T, & i \text{ is odd} \\ \mathbf{T}_i + \mathbf{T}_i^T, & i \text{ is even}\end{cases} \\ \mathbf{S}_i &= \left(x_{k+\frac{i+1}{2}}\mathbf{1}_M - \boldsymbol{\Phi}^T\mathbf{L}_1\right)\breve{\mathbf{e}}_{k+\frac{i+1}{2}}^T \\ \mathbf{T}_i &= \left(y_{k+\frac{i+1}{2}}\mathbf{1}_M - \boldsymbol{\Phi}^T\mathbf{L}_2\right)\breve{\mathbf{e}}_{k+\frac{i}{2}}^T \\ \mathbf{L}_1 &= [1 \quad 0]^T \\ \mathbf{L}_2 &= [0 \quad 1]^T. \end{aligned}$$

When i is odd, we can partition $\mathbf{S}_i$ and $\mathbf{S}_i^T$ into left and right parts to obtain

$$\begin{aligned}\mathbf{G}_i\begin{bmatrix}\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T \\ -\mathbf{I}_{M-k}\end{bmatrix} &= \left(\mathbf{S}_i + \mathbf{S}_i^T\right)\begin{bmatrix}\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T \\ -\mathbf{I}_{M-k}\end{bmatrix} \\ &= \mathbf{Z}_{i,1} + \mathbf{Z}_{i,2} + \mathbf{Z}_{i,3} + \mathbf{Z}_{i,4}\end{aligned} \tag{25}$$

where

$$\begin{aligned} \mathbf{Z}_{i,1} &= \mathbf{S}_i\mathbf{L}_3\left(\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T\right) \\ \mathbf{Z}_{i,2} &= -\mathbf{S}_i\mathbf{L}_4 \\ \mathbf{Z}_{i,3} &= \mathbf{S}_i^T\mathbf{L}_3\left(\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T\right) \\ \mathbf{Z}_{i,4} &= -\mathbf{S}_i^T\mathbf{L}_4 \\ \mathbf{L}_3 &= \begin{bmatrix}\mathbf{I}_k & \mathbf{0}_{k\times(M-k)}\end{bmatrix}^T \\ \mathbf{L}_4 &= \begin{bmatrix}\mathbf{0}_{(M-k)\times k} & \mathbf{I}_{M-k}\end{bmatrix}^T. \end{aligned}$$

As $\breve{\mathbf{e}}_{k+((i+1)/2)}^T\mathbf{L}_3 = \mathbf{0}_{1\times k}$, $\mathbf{Z}_{i,1} = \mathbf{0}_{M\times(M-k)}$. Furthermore, $\mathbf{Z}_{i,3}$ is simplified by noting that

$$\begin{aligned} & x_{k+\frac{i+1}{2}}\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\mathbf{1}_M^T\mathbf{L}_3\left(\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T\right) \\ &= x_{k+\frac{i+1}{2}}\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\mathbf{1}_k^T\left(\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T\right) \\ &= x_{k+\frac{i+1}{2}}\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\mathbf{1}_{M-k}^T \end{aligned} \tag{26}$$

and

$$\begin{aligned} & \breve{\mathbf{e}}_{k+\frac{i+1}{2}}\mathbf{L}_1^T\boldsymbol{\Phi}\mathbf{L}_3\left(\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T\right) \\ &= \breve{\mathbf{e}}_{k+\frac{i+1}{2}}\mathbf{L}_1^T\boldsymbol{\Phi}_a\left(\boldsymbol{\Phi}_a^\dagger\boldsymbol{\Phi}_u + \frac{1}{k}\mathbf{1}_k\mathbf{1}_{M-k}^T\right) \\ &= \breve{\mathbf{e}}_{k+\frac{i+1}{2}}\mathbf{L}_1^T\boldsymbol{\Phi}_u. \end{aligned} \tag{27}$$

Combining (26) and (27), we have

$$\mathbf{Z}_{i,3} = \breve{\mathbf{e}}_{k+\frac{i+1}{2}}\left(x_{k+\frac{i+1}{2}}\mathbf{1}_{M-k}^T - \mathbf{L}_1^T\boldsymbol{\Phi}_u\right). \tag{28}$$

Moreover, we get

$$\mathbf{Z}_{i,2} = -\left(x_{k+\frac{i+1}{2}}\mathbf{1}_M - \boldsymbol{\Phi}^T\mathbf{L}_1\right)\breve{\mathbf{e}}_{k+\frac{i+1}{2}}^T\mathbf{L}_4 \tag{29}$$

$$\begin{aligned} \mathbf{Z}_{i,4} &= -\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\left(x_{k+\frac{i+1}{2}}\mathbf{1}_M^T - \mathbf{L}_1^T\boldsymbol{\Phi}\right)\mathbf{L}_4 \\ &= -\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\left(x_{k+\frac{i+1}{2}}\mathbf{1}_{M-k}^T - \mathbf{L}_1^T\boldsymbol{\Phi}_u\right) = -\mathbf{Z}_{i,3}. \end{aligned} \tag{30}$$

With the use of (28), (30), and (25) is written as

$$\mathbf{Z}_{i,2} = \left(\boldsymbol{\Phi}^T\mathbf{L}_1 - x_{k+\frac{i+1}{2}}\mathbf{1}_M\right)\breve{\mathbf{e}}_{k+\frac{i+1}{2}}^T\mathbf{L}_4. \tag{31}$$

Similarly, when i is even, (25) is

$$\left(\boldsymbol{\Phi}^T\mathbf{L}_2 - y_{k+\frac{i}{2}}\mathbf{1}_M\right)\breve{\mathbf{e}}_{k+\frac{i}{2}}^T\mathbf{L}_4. \tag{32}$$

From (31) and (32), we have, for $i = 1, 3, \cdots, 2(M-k) - 1$

$$\begin{aligned} [\mathbf{g}_i \quad \mathbf{g}_{i+1}] &= \left[\operatorname{vec}\left(\mathbf{J}\boldsymbol{\Phi}^T\mathbf{L}_1\breve{\mathbf{e}}_{k+\frac{i+1}{2}}^T\mathbf{L}_4\right) \right. \\ &\qquad \left. \operatorname{vec}\left(\mathbf{J}\boldsymbol{\Phi}^T\mathbf{L}_2\breve{\mathbf{e}}_{k+\frac{i+1}{2}}^T\mathbf{L}_4\right)\right] \\ &= \left[\left(\mathbf{L}_4^T\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\right) \otimes (\mathbf{J}\boldsymbol{\Phi}^T\mathbf{L}_1) \right. \\ &\qquad \left. \left(\mathbf{L}_4^T\breve{\mathbf{e}}_{k+\frac{i+1}{2}}\right) \otimes (\mathbf{J}\boldsymbol{\Phi}^T\mathbf{L}_2)\right] = \grave{\mathbf{e}}_i \otimes \mathbf{J}\boldsymbol{\Phi}^T. \end{aligned} \tag{33}$$

Using (33), (24) is

$$\begin{aligned}\left(\left(\breve{\mathbf{\Phi}}_a^{\dagger}\grave{\mathbf{\Phi}}_u\right)^T\otimes\mathbf{J}\right)\breve{\mathbf{\Psi}}\breve{\mathbf{\Lambda}}\mathbf{D} &= \mathbf{I}_{M-k}\otimes\mathbf{J}\mathbf{\Phi}^T\\ &= \mathbf{I}_{M-k}\otimes\mathbf{J}\mathbf{\Phi}^T\left(\mathbf{\Phi}_a\mathbf{\Phi}_a^T\right)\left(\mathbf{\Phi}_a\mathbf{\Phi}_a^T\right)^{-1}\\ &= \mathbf{I}_{M-k}\otimes\mathbf{J}\mathbf{\Phi}^T\mathbf{\Phi}\begin{bmatrix}\mathbf{\Phi}_a^{\dagger}\\ \mathbf{0}_{(M-k)\times 2}\end{bmatrix}. \end{aligned}\qquad (34)$$

Since $\mathbf{\Phi}_a\mathbf{1}_k = \mathbf{0}_{2\times 1}$, (34) becomes

$$\mathbf{I}_{M-k}\otimes\mathbf{J}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{J}\begin{bmatrix}\mathbf{\Phi}_a^{\dagger}\\ \mathbf{0}_{(M-k)\times 2}\end{bmatrix} = \mathbf{I}_{M-k}\otimes\left(\mathbf{B}\begin{bmatrix}\mathbf{\Phi}_a^{\dagger}\\ \mathbf{0}_{(M-k)\times 2}\end{bmatrix}\right)\qquad (35)$$

which is the right-hand side of (23).

IV. Extension to Partially Connected Wireless Sensor Networks

The WMDS algorithm derivation in Section III assumes that distance measurements between all sensor pairs are available, which is not always valid for practical situations. In the following, the WMDS algorithm is extended to a partially-connected WSN. Our basic idea is estimate the unavailable or missing distance (MD) measurements prior to employing the proposed method. The estimated positions are then used to update the MDs, and we repeat the estimation process in an iterative manner until a stopping criterion is met.

Let the connectivity function, denoted by $c_{i,j}$ be 1 if the distance measurement between the ith and jth sensor is available and 0 otherwise.

If $c_{i,j} = 0$, then obviously their distance exceeds the maximum communication range, namely, $\mathcal{R}_{\max}$:

$$d_{i,j} > \mathcal{R}_{\max}. \qquad (36)$$

On the other hand, if both the ith and jth sensors connect to the lth sensor, then we have

$$d_{i,j} \leq d_{i,l} + d_{l,j}. \qquad (37)$$

By combining (36) and (37), we have

$$\mathcal{R}_{\min}^{(i,j)} \geq d_{i,j} > \mathcal{R}_{\max} \qquad (38)$$

where $\mathcal{R}_{\min}^{(i,j)} = \min\{d_{i,l} + d_{l,j}\}$, $l = 1, 2, \cdots, M$, $c_{i,l} = c_{l,j} = 1$. In practice, when the maximum communication range is unknown, $\mathcal{R}_{\max}$ is replaced by the largest distance measurement and $\mathcal{R}_{\min}^{(i,j)}$ is approximated by $\min\{r_{i,l} + r_{l,j}\}$. Initially, all MDs are estimated as

$$r_{i,j} = 0.5\left(\mathcal{R}_{\min}^{(i,j)} + \mathcal{R}_{\max}\right), \quad c_{i,j} = 0. \qquad (39)$$

Note that the idea of (39) is modified from the shortest path distance algorithm [12], [30]. Instead of using $r_{i,j} = \mathcal{R}_{\min}^{(i,j)}$ in [12] and [30], we employ the average between $\mathcal{R}_{\min}^{(i,j)}$ and $\mathcal{R}_{\max}$ to get the initial MD estimates. The MD estimation algorithm proceeds as follows: first, $\{r_{i,j} : c_{i,j} = 0, \{l : c_{i,l} = c_{l,j} = 1, l = 1, 2, \cdots, M\} \neq \emptyset\}$ are estimated using (39) and the corresponding $\{c_{i,j}\}$ are set to 1. The process repeats until all MDs have been obtained. After initialization, the WMDS algorithm in Section II is employed and the position estimates obtained are utilized to calculate a new set of MDs. This recursive process is terminated when certain stopping criterion is satisfied. The operation of the WMDS for partially-connected WSNs is depicted in Table II.

TABLE II
WMDS Algorithm for Partially-Connected WSNs

```
While $c_{i,j} = 0,\ i, j = 1, 2, \cdots, M$
    For $i = 1, 2, \cdots, M$
        For $j = 1, 2, \cdots, M$
            Compute $r_{i,j}$ using (39)
        end
    end
end

While stopping criterion is not satisfied
    Estimate $\hat{\phi}_u$ with WMDS algorithm
    Calculate $r_{i,j}$ where $c_{i,j} = 0$
end
```

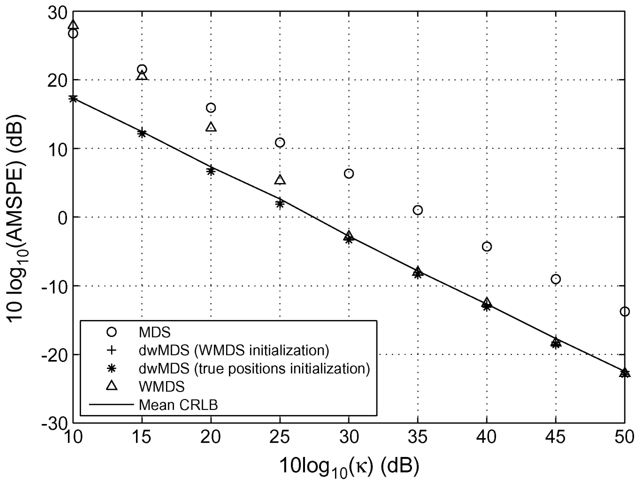

Fig. 1. Average mean square position error versus κ in fully connected network.

V. Numerical Examples

Computer simulations are conducted to evaluate the performance of WMDS algorithm. We consider that there are $M = 24$ sensors in a 100 m × 100 m area with $k = 4$ anchors at positions (0,0) m, (0,100) m, (100,0) m, and (100,100) m. All simulation results are averages of $N = 100$ independent runs and the average mean square position errors (AMSPEs) of the 20 unknown-position sensors are plotted in fully connected and partially connected WSN scenarios. The AMSPE is defined as

$$\text{AMSPE} = \frac{\sum_{i=1}^{N}\sum_{j=k+1}^{M}\left(\hat{x}_j^{(i)} - x_j\right)^2 + \left(\hat{y}_j^{(i)} - y_j\right)^2}{N}$$

where $\hat{\alpha}_j^{(i)}$ is the estimate of α_j in the ith independent run. In the two scenarios, the WMDS algorithm is compared with the classical MDS [12] and dwMDS [14] methods as well as CRLB. The dwMDS algorithm is initialized using the position estimates of WMDS with $\mathbf{W} = \mathbf{I}_{M(M-k)}$ and the true sensor positions. The WMDS algorithm has iterated ten times as further iterations cannot produce noticeable improvement. We assign $\kappa\sigma_{i,j}^2 = d_{i,j}^2$, where κ is a constant is set to make a longer distance to have a larger measurement error.

In the first scenario, sensors are fully connected and the sensors are randomly deployed in the 100 m × 100 m area in each run. The mean value of CRLBs for different sensor geometries is employed as the minimum achievable performance bound. In Fig. 1, it is observed that the

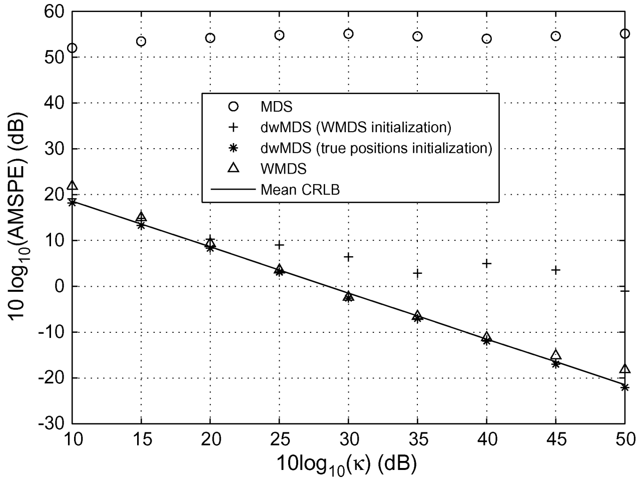

Fig. 2. Average mean square position error versus κ in partially connected network.

AMSPEs of the dwMDS algorithm with both initializations can attain the CRLB in the whole κ range. On the other hand, the AMSPEs of the WMDS algorithm can attain the CRLB when $\kappa \geq 30$ dB. Furthermore, the AMSPE of the MDS algorithm is larger than the CRLB by about 8 dB.

In the second scenario, all the simulation settings are the same as the previous one except that the communication range is set to 70 m. Therefore, the WSN is now partially connected and the average node degree is 15.4 [31]. The same MDs are initialized for both the MDS and WMDS algorithms. It is shown in Fig. 2 that the AMSPEs of the WMDS algorithm can attain the CRLB when $\kappa \geq 25$ dB. On the other hand, the MDS algorithm performs unsatisfactorily and its AMSPEs deviate the CRLB by at least 30 dB. Moreover, the AMSPEs of dwMDS method initialized by true sensor positions can attain the CRLB for the whole κ range. However, the one initialized by WMDS position estimates deviate from the CRLB when $\kappa \geq 25$ dB.

VI. Conclusion

A weighted multidimensional scaling algorithm for node localization in a fully connected sensor network is devised. Modifications to partially connected scenarios are also suggested. Statistical performance is analyzed and simulation results show that its estimation accuracy can attain Cramér–Rao lower bound for sufficiently small noise conditions. As a future work, we will study the local convergence [29] of the proposed algorithm.

References

[1] 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.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[2] N. S. Correal and N. Patwari, "Wireless sensor networks: Challenges and opportunities," in *Proc. MPRG/Virgina Technology Wireless Symp.*, Jun. 2001, pp. 1–9.

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

[4] I. Stojmenovic, *Handbook of Sensor Networks: Algorithms and Architectures*. New York: Wiley, 2005.

[5] K. W. Cheung, H. C. So, Y. T. Chan, and W.-K. Ma, "A constrained least squares approach to mobile positioning: Algorithms and optimality," *EURASIP J. Appl. Signal Process.*, pp. 1–23, 2006.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

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

[8] 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, Aug. 2003.

[9] P. Biswas, T. Lian, T. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 188–220, 2006.

[10] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. New York: Springer, 1997.

[11] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. Boca Raton, FL: Chapman Hall/CRC, 2001.

[12] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Netw.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

[13] P. Groenen, *The Majorization Approach to Multidimensional Scaling: Some Problems and Extensions*. Leiden, The Netherlands: DSWO Press, 1993.

[14] J. A. Costa, N. Patwari, and A. O. Hero, III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[15] H. Wei, Q. Wan, Z. Chen, and S. Ye, "A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1511–1524, Apr. 2008.

[16] K. W. Cheung and H. C. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 460–470, Feb. 2005.

[17] H. C. So and K. W. Chan, "A generalized subspace approach for mobile positioning with time-of-arrival measurements," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5103–5107, Oct. 2007.

[18] F. K. W. Chan, H. C. So, and W.-K. Ma, "A novel subspace approach for node localization in wireless sensor networks using range measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 260–269, Jan. 2009.

[19] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 365–377, Jan. 2008.

[20] H. Miao, K. Yu, and M. Juntti, "Positioning for NLOS propagation: Algorithm derivations and Cramer-Rao bounds," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2568–2580, Sep. 2007.

[21] K. Yu and Y. J. Gau, "Improved positioning algorithms for nonline-of-sight environments," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2342–2353, Jul. 2008.

[22] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[23] T. Soderstrom and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[24] A. B. Israel and T. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. New York: Springer, 2003.

[25] P. Courrieu, "Fast computation of Moore-Penrose inverse matrices," *Neural Inf. Process.—Lett. Rev.*, vol. 8, no. 2, pp. 25–29, Aug. 2005.

[26] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*. New York: Academic, 1977.

[27] R. Kumaresan, L. L. Scharf, and A. K. Shaw, "An algorithm for pole-zero modeling and spectral analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 3, pp. 637–640, Jun. 1986.

[28] Y. Bresler and A. Macovski, "Exact maximum likelihood parameter estimation of superimposed exponential signals in noise," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1081–1088, Oct. 1986.

[29] J. Li, P. Stoica, and Z. Liu, "Comparative study of IQML and MODE direction-of-arrival estimators," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 149–160, Jan. 1998.

[30] G. Yu and S. Wang, "A hierarchical MDS-based localization algorithm for wireless sensor networks," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 25–28, 2008, pp. 748–754.

[31] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw.*, vol. 43, pp. 499–518, 2003.