Anchorless Cooperative Localization for Mobile Wireless Sensor Networks

Hadi Jamali Rad, Toon van Waterschoot, and Geert Leus
Delft University of Technology - TU Delft
Faculty of EEMCS
2628 CD, Delft, The Netherlands
{h.jamalirad, t.j.m.vanwaterschoot, g.j.t.leus}@tudelft.nl

Abstract

We propose two algorithms for anchorless cooperative network localization in mobile wireless sensor networks (WSNs). In order to continuously localize the mobile network, given the pairwise distance measurements between different wireless sensor nodes, we propose to use subspace tracking to track the variations in signal eigenvectors and corresponding eigenvalues of the double-centered distance matrix. We compare the computational complexity of the proposed algorithms with a recently developed anchorless algorithm exploiting the extended Kalman filter (EKF) as well as an anchored algorithm exploiting ordinary least squares (LS). We show that our proposed algorithms are computationally efficient, and hence, are appropriate for practical implementations. Simulation results further illustrate that the proposed algorithms have an acceptable accuracy in comparison with the aforementioned algorithms and are more robust to an increasing sampling period of the measurements.

1 Introduction

Many of the current research efforts on WSN localization have focused on proposing solutions to derive the location of the nodes using their pairwise distance measurements in a cooperative context. The aforementioned studies can be divided into two main categories, i.e., anchorless and anchored localization, where in the former there are no nodes with known locations (so-called anchors) and determining the relative location of the nodes is the ultimate goal. One popular solution to find the relative locations of the nodes based on distance measurements in a fixed network is to use multidimensional scaling (MDS) or its distributed version for large networks [1]. On the other hand, by exploiting the availability of the anchor nodes and the knowledge of their locations, a set of linear equations can be obtained. This is the basis of the so-called weighted MDS (WMDS) which avoids an eigenvalue decomposition (EVD) and attains the Cramer-Rao bound (CRB). This idea has been developed in [2] for multiple unknown nodes in a static network.

The problem of cooperative localization for *mobile* WSNs is of special interest, and surprisingly has not been efficiently solved yet. In [3], an anchorless localization scheme for mobile (dynamic) networks called SPAWN is proposed based on the theory of factor graphs. In this scheme, each node requires knowledge about its own movement model as a probability distribution in order to do predictions, which is not so simple to be acquired in a real application and additionally increases the computational complexity significantly. In [4], an EKF-based method is developed which incorporates the location of the nodes as well as their velocities in a state-space model. Although velocity measurements of the nodes are beneficial to cooperative network localization, it requires

This work was supported by NWO-STW under the VICI program (project 10382).

the use of Doppler sensors, which increases the implementation cost, and hence, we avoid using them.

Inspired by the simplicity and robustness of MDS localization in fixed WSNs, we propose an MDS-based cooperative localization for a mobile network. It is worth mentioning that classical MDS involves an eigenvalue decomposition (EVD) calculation on a double-centered distance measurement matrix, which has a high complexity for a large network. For a mobile network, computationally intensive EVD calculations should be conducted in each snapshot of the mobile network to localize the nodes in an ad-hoc fashion. To avoid this problem, we propose to use two novel subspace tracking algorithms to track the variations in the signal eigenvectors and corresponding eigenvalues due to variations in the double-centered distance matrix. We show that this can enable us to estimate the next location of the moving nodes in the network given their previous location estimates.

The main advantages of the proposed algorithms can be described as follows. First, the proposed algorithms are computationally efficient, and hence, are suitable choices for practical implementations. Besides, the algorithms have an acceptable positioning accuracy and they are more robust to an increasing sampling period (T_s) of the measurements compared to the other algorithms under consideration. Finally, the algorithms do not rely on the movement model of the nodes (i.e., they are non-parametric) and can be applied to many practical scenarios. The remainder of this paper is organized as follows. In Section 2, we present the system model underlying our analysis and evaluations. Section 3 describes the proposed cooperative localization algorithms based on subspace tracking. Section 4 compares the computational complexity of the algorithms under consideration in this paper. Section 5 provides simulation results for mobile sensor networks with different network parameters. Concluding remarks are presented in Section 6.

2 System Model

We consider a network of N mobile wireless sensor nodes, living in a D-dimensional space ($D < N$). Let $\{\mathbf{x}_{i,k}\}_{i=1}^{N}$ be the actual vector coordinates of the sensor nodes, or equivalently, let $\mathbf{X}_k = [\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{N,k}]$ be the matrix of coordinates at snapshot k. By collecting the noisy pairwise distance measurements $d_{i,j,k} = \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\| + v_{i,j,k}$ between the nodes in a distance matrix $\mathbf{D}_k$, i.e. $[\mathbf{D}_k]_{i,j} = d^2_{i,j,k}$, the double-centered distance matrix can be calculated as $\mathbf{B}_k = -1/2\boldsymbol{\Upsilon}\mathbf{D}_k\boldsymbol{\Upsilon}$, where $v_{i,j,k} \sim \mathcal{N}(0, \sigma^2_{i,j,k})$ is the independent and identically distributed (i.i.d.) noise and $\boldsymbol{\Upsilon}$ is the centering operator [1]. In case of a network with fixed nodes, $\mathbf{B}_k$ can be used in the classical MDS to recover the locations of the nodes $\mathbf{X}_k$ (up to a translation and orthogonal transformation) by means of the EVD as described in [1].

One trivial solution for a mobile scenario is to perform these computationally intensive EVD calculations for every snapshot of the mobile network. Instead, we propose two low-complexity localization algorithms. The proposed algorithms are anchorless in the sense that the relative positions of the mobile nodes can continuously be calculated without requiring information about the anchor nodes. Still, determining the exact location of the nodes (removing the unknown translation and orthogonal transformation) requires a coordinate system consisting of at least $D+1$ anchor nodes with known locations. It is assumed that these anchor nodes are equipped with long distance transmission devices to continuously localize themselves with respect to (w.r.t) a central coordinate system. Since the number of anchors is generally small compared to the total number of nodes in a network, this requirement is not intensive from a computational and power consumption point of view.

3 Proposed Subspace Tracking Algorithms

We start by considering the noiseless case ($v_{i,j,k} = 0$) and we expect that the presence of noise will only slightly degrade the performance of the algorithms. In the noiseless case, the double-centered distance matrix $\mathbf{B}_k$ will be a symmetric $N \times N$ matrix of rank D. For the k-th snapshot of the mobile network, the trivial approach is to find the locations as the minimum of min $\|\mathbf{B}_k - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\|^2$ over all $D \times N$ matrices $\tilde{\mathbf{X}}$. The EVD of $\mathbf{B}_k$ can be expressed in the following form

$$\mathbf{B}_k = \begin{bmatrix}\mathbf{U}_{1,k} & \mathbf{U}_{2,k}\end{bmatrix} \begin{bmatrix}\mathbf{\Sigma}_{1,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix} \begin{bmatrix}\mathbf{U}_{1,k}^T \\ \mathbf{U}_{2,k}^T\end{bmatrix} = \mathbf{U}_{1,k}\mathbf{\Sigma}_{1,k}\mathbf{U}_{1,k}^T. \tag{1}$$

Then the location matrix (up to a translation and orthogonal transformation) can be written as

$$\tilde{\mathbf{X}}_k = \mathbf{\Sigma}_{1,k}^{\frac{1}{2}}\mathbf{U}_{1,k}^T. \tag{2}$$

Although the above procedure can be done for every snapshot of a mobile network, the complexity of computing the EVD in (1) can be quite intensive for large N [5]. The idea behind the proposed subspace tracking algorithms is that in order to calculate the location of the nodes using (2), we only need to update the D signal eigenvectors in $\mathbf{U}_{1,k}$ and their corresponding eigenvalues in $\mathbf{\Sigma}_{1,k}$. This can be done by more efficient iterative approaches as proposed in the following.

3.1 Perturbation-Expansion-Based Subspace Tracking

In this section, we will explain the idea behind the perturbation-expansion-based subspace tracking (PEST). If the movement of the nodes satisfies the property that the invariant subspace of the next (perturbed) double-centered distance matrix ($\mathbf{B}_k = \mathbf{B}_{k-1} + \Delta\mathbf{B}_k$) does not contain any vectors that are orthogonal to the invariant subspace of the current double-centered distance matrix ($\mathbf{B}_{k-1}$), the two bases respectively spanning the signal and noise subspace of the next double-centered distance matrix follow the expressions [6]

$$\begin{aligned} \tilde{\mathbf{U}}_{1,k}^u &= \tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k, \quad (3)\\ \tilde{\mathbf{U}}_{2,k}^u &= -\tilde{\mathbf{U}}_{1,k-1}\mathbf{P}_k^T + \tilde{\mathbf{U}}_{2,k-1}, \quad (4)\end{aligned}$$

where $\mathbf{P}_k$ is a coefficient matrix, $\tilde{\mathbf{U}}_{i,k}$ represents an orthonormal basis spanning the same subspace as the matrix of eigenvectors $\mathbf{U}_{i,k}$, and $\tilde{\mathbf{U}}_{i,k}^u$ is an unorthonormalized version of $\tilde{\mathbf{U}}_{i,k}$. To compute $\mathbf{P}_k$ in (3) and (4), we will resort to a first-order approximation. However, since we will continuously use first-order approximations, we can not assume that $\tilde{\mathbf{U}}_{1,k-1}$ and $\tilde{\mathbf{U}}_{2,k-1}$ in (3) and (4) are exact orthonormal bases spanning respectively the signal and noise subspaces of $\mathbf{B}_{k-1}$. And thus, the first-order approximation of $\mathbf{P}_k$ in [6] does not hold anymore, and we need to derive a new $\mathbf{P}_k$. The value of $\mathbf{P}_k$ should satisfy the necessary and sufficient condition for $\tilde{\mathbf{U}}_{1,k}^u$ and $\tilde{\mathbf{U}}_{2,k}^u$ to be new bases for the new perturbed signal and noise subspaces, and thus we need

$$(\tilde{\mathbf{U}}_{2,k}^u)^T\mathbf{B}_k\tilde{\mathbf{U}}_{1,k}^u = \mathbf{0}. \tag{5}$$

We can expand (5) by substituting (3) and (4) as follows

$$(-\tilde{\mathbf{U}}_{1,k-1}\mathbf{P}_k^T + \tilde{\mathbf{U}}_{2,k-1})^T\mathbf{B}_k(\tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k) = \mathbf{0}. \tag{6}$$

After neglecting the second-order terms, we obtain

$$\begin{aligned}&-\mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1}+\tilde{\mathbf{U}}_{2,k-1}^T\Delta\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}\\&+\underbrace{\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1}}_{\neq\mathbf{0}}+\underbrace{\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{2,k-1}}_{\neq\mathbf{0}}\mathbf{P}_k=\mathbf{0}.\end{aligned}\tag{7}$$

Different from the derivations in [6], the third and fourth terms in (7) are not exactly equal to zero and also the value of their elements increases with each iteration due to the fact that we are using first-order approximations in each snapshot. It is notable that (7) is linear in the elements of $\mathbf{P}_k$ and can easily be solved w.r.t $\mathbf{P}_k$. However, this requires a $DN \times DN$ matrix inverse calculation which is undesirable due to its high complexity. Therefore, we confine our approximation of $\mathbf{P}_k$ to the first three terms in (7). By defining

$$\tilde{\mathbf{\Sigma}}_{1,k-1}=\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1},\tag{8}$$

this results in

$$\mathbf{P}_k=\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}(\tilde{\mathbf{\Sigma}}_{1,k-1})^{-1}.\tag{9}$$

To avoid updating $\tilde{\mathbf{U}}_{2,k}^u$ in (3), we use $\tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{U}}_{1,k-1}^T+\tilde{\mathbf{U}}_{2,k-1}\tilde{\mathbf{U}}_{2,k-1}^T=\mathbf{I}$ ($\mathbf{I}$ represents the identity matrix). Together with (9), this allows us to rewrite (3) as

$$\tilde{\mathbf{U}}_{1,k}^u=\tilde{\mathbf{U}}_{1,k-1}+(\mathbf{I}-\tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{U}}_{1,k-1}^T)\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{\Sigma}}_{1,k-1}^{-1}.\tag{10}$$

Now, to be able to use the above formula in an iterative manner we should normalize it using any possible orthonormalization process like Gram-Schmidt (GS) factorization. We call the orthonormalized result $\tilde{\mathbf{U}}_{1,k}$. As described in [6] and as can be seen from the above derivations, $\tilde{\mathbf{U}}_{1,k}$ is an approximation of the orthonormal basis which spans the same subspace as its corresponding signal eigenvectors in $\mathbf{U}_{1,k}$. However, to be able to calculate the relative locations using (2), we have to find $\mathbf{U}_{1,k}$. To this aim, we look for a matrix $\mathbf{A}_k$ so that

$$\tilde{\mathbf{U}}_{1,k}=\mathbf{U}_{1,k}\mathbf{A}_k.\tag{11}$$

Note that since $\tilde{\mathbf{U}}_{1,k}$ and $\mathbf{U}_{1,k}$ are isometries, $\mathbf{A}_k$ will be a unitary matrix. To be able to estimate the locations according to (2), we also need to calculate $\mathbf{\Sigma}_{1,k}$, which depends on the value of $\mathbf{U}_{1,k}$ and $\mathbf{A}_k$ as follows

$$\mathbf{\Sigma}_{1,k}=\mathbf{U}_{1,k}^T\mathbf{B}_k\mathbf{U}_{1,k}.$$

From (8), and using (11), we finally obtain

$$\begin{aligned}\tilde{\mathbf{\Sigma}}_{1,k}&=(\mathbf{U}_{1,k}\mathbf{A}_k)^T\mathbf{B}_k(\mathbf{U}_{1,k}\mathbf{A}_k),\\&=\mathbf{A}_k^T\mathbf{U}_{1,k}^T\mathbf{B}_k\mathbf{U}_{1,k}\mathbf{A}_k,\\&=\mathbf{A}_k^T\mathbf{\Sigma}_{1,k}\mathbf{A}_k.\end{aligned}\tag{12}$$

From (12), $\mathbf{A}_k$ and $\mathbf{\Sigma}_{1,k}$ can be calculated by an EVD of $\tilde{\mathbf{\Sigma}}_{1,k}$. Note that, our main goal for using perturbation expansion was to avoid computationally intensive EVD calculations, while here we require it again. However, the point is that $\tilde{\mathbf{\Sigma}}_{1,k}$ is a $D\times D$ matrix, which is very small in size compared to the $N\times N$ double-centered distance matrix $\mathbf{B}_k$ for large scale sensor networks. The PEST algorithm is summarized in Algorithm 1.

Algorithm 1 PEST

Start with an initial location guess
for $k = 1$ to K **do**
  Calculate $\tilde{\mathbf{U}}^{u}_{1,k}$ using (10)
  GS orthonormalization $\tilde{\mathbf{U}}_{1,k} = \mathrm{GS}(\tilde{\mathbf{U}}^{u}_{1,k})$
  Calculate $\tilde{\mathbf{\Sigma}}_{1,k}$, $\mathbf{A_k}$ and $\mathbf{\Sigma}_{1,k}$ using (8) and (12)
  Calculate $\mathbf{U_{1,k}}$ using (11)
  Location estimation using (2)
end for

Algorithm 2 PIST

Start with an initial location guess
for $k = 1$ to K **do**
  Calculate $\tilde{\mathbf{U}}^{u}_{1,k} = \mathbf{B_k}\tilde{\mathbf{U}}_{\mathbf{1,k-1}}$
  GS orthonormalization $\tilde{\mathbf{U}}_{1,k} = \mathrm{GS}(\tilde{\mathbf{U}}^{u}_{1,k})$
  Calculate $\tilde{\mathbf{\Sigma}}_{1,k}$, $\mathbf{A_k}$ and $\mathbf{\Sigma}_{1,k}$ using (8) and (12)
  Calculate $\mathbf{U_{1,k}}$ using (11)
  Location estimation using (2)
end for

3.2 Power-Iteration-Based Subspace Tracking

Power iterations can also be used to efficiently calculate an invariant subspace of a diagonalizable matrix (like $\mathbf{B}_k$) [5]. Power iterations are normally used in an iterative manner to reach an acceptable accuracy. Depending on a random initial guess, the number of iterations can be large, which in turn leads to a high computational complexity. Additionally, an inappropriate choice of the initial guess can sometimes lead to instability and divergence problems [5]. To avoid both problems (complexity and divergence) in mobile network localization, we propose to do just one iteration in each snapshot of the mobile network and use the previous estimate of the orthonormal basis as the initial guess for the next estimate. This leads to a scheme that tracks the desired invariant subspace in a similar fashion as PEST, and we call it power-iteration-based subspace tracking (PIST). Note that this power-iteration-based approach leads to a unique orthonormal basis spanning the desired signal subspace. Thus, the same EVD calculations as in (12) are required to obtain the matrix of eigenvectors. The PIST algorithm is shown in Algorithm 2.

3.3 EKF Tracking and Ordinary LS

For the sake of comparison, we consider two other algorithms. First, we consider cooperative mobile network localization using the EKF proposed in [4]. However, as we do not have velocity measurements in our setup, we simplify the EKF model of [4]. The discrete-time state and measurement equations can be written as

$$\mathbf{x}_k = \mathbf{\Phi}\mathbf{x}_{k-1} + \mathbf{w}_k, \quad (13)$$
$$\mathbf{d}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad (14)$$

where $\mathbf{x}_k = [\mathbf{x}^T_{1,k}, \ldots, \mathbf{x}^T_{N,k}, \dot{\mathbf{x}}^T_{1,k}, \ldots, \dot{\mathbf{x}}^T_{N,k}]$ is the column vector of length $2DN$ containing the nodes' locations and velocities at the k-th snapshot, $\mathbf{d}_k = [d_{1,2,k}, \ldots, d_{(N-1),N,k}]^T$ is the column vector of pairwise distance measurements of length $N(N-1)/2$ at the k-th snapshot, and $\mathbf{\Phi} = \mathbf{I} + \mathbf{F}T_s$, with T_s the sampling period and $\mathbf{F}$ given by

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{DN\times DN} & \mathbf{I}_{DN\times DN} \\ \mathbf{0}_{DN\times DN} & \mathbf{0}_{DN\times DN} \end{bmatrix}.$$

Algorithm 3 EKFT

Start with an initial location guess
for $k = 1$ to K **do**
  Next state:
  $\hat{\mathbf{x}}_k^- = \mathbf{\Phi}\hat{\mathbf{x}}_{k-1}$
  Next error covariance:
  $\mathbf{P_k^-} = \mathbf{\Phi P_{k-1} \Phi^T + Q}$
  Compute the Kalman gain:
  $\mathbf{K_k} = \mathbf{P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}}$
  Update the state:
  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K_k}\big(\mathbf{d_k} - \mathbf{h}(\hat{\mathbf{x}}_\mathbf{k}^-)\big)$
  Update the error covariance:
  $\mathbf{P_k} = \mathbf{(I - K_k H_k) P_k^-}$
end for

Table 1: Computational Complexity

Algo.	FLOPS for Lin. Op.	Orthonorm.	SQRT	Matrix Inverse	EVD	Total FLOPS
PEST	$4N^2D + 3ND^2 + ND$	$1(N \times D)$	2	$1\ (D \times D)$	$1\ (D \times D)$	$4DN^2 + (5D^2 + D)N + 2D^3 + 6D^2 + 24$
PIST	$2N^2D + 2ND^2 + ND$	$1\ (N \times D)$	2	-	$1\ (D \times D)$	$4DN^2 + (5D^2 + D)N + D^3 + 24$
EKFT	$(D/2)N^5 + (5D^2/2 - D)N^4 + (12D^3 - 5D^2/2 + 2D)N^3 + (4D^2 - D/2 + 2)N^2 - 2N$	-	$DN(N-1)/2$	$2\ (2DN \times 2DN)$	-	$(D/2)N^5 + (5D^2/2 - D)N^4 + (28D^3 - 5D^2/2 + 2D)N^3 + (52D^2 + 11D/2 + 2)N^2 + (-6D - 2)N$
Ordinary LS (l anchors)	$(D^2)N^3 + (l - 2lD^2)N^2 + (D^2(1 + l^2) + D(1+l) - l^2)N - (D + D^2)l$	-	-	$N - l\ \ (D \times D)$	-	$(D^2)N^3 + (l - 2lD^2)N^2 + (D^3 + D^2(7+l^2) + D(1+l) - l^2)N - (D + 7D^2 - D^3)l$

Further, we set $\mathbf{w}_k = [\mathbf{0}^T, \bar{\mathbf{w}}_k^T]^T$, where we assume that the entries of $\bar{\mathbf{w}}_k$ and $\mathbf{v}_k$ are uncorrelated zero-mean white Gaussian noise processes with standard deviation σ_w and $\sigma_{i,j,k}$, respectively. To linearize the measurement equations, we take the Jacobian matrix of $\mathbf{h}(\mathbf{x}_k)$ defined by an $N(N-1)/2 \times 2DN$ matrix $\mathbf{H}_k = \nabla \mathbf{h}(\mathbf{x}_k)$. The EKF tracking (EKFT) algorithm is shown in Algorithm 3. $\mathbf{P}_k$, $\mathbf{R}_k$ and $\mathbf{Q}$ are the covariance matrix of the error in the state estimate, the measurement noise, and the process noise, respectively. We also consider one anchored localization algorithm similar to the WMDS in [2]. We employ the known locations of the anchors to end up with a set of linear equations in the unknown locations and then we use ordinary LS to estimate the location of the unknown nodes. Hence, we consider the WMDS algorithm with the weighting matrix equal to identity matrix and we do not adopt any iterations per snapshot of the movement.

4 Computational Complexity

We define the computational complexity as the number of operations required to create one estimate of the location of the unknown nodes. For the sake of simplicity, we do not count the number of additions and subtractions as well as the number of multiplications by 1, -1, or powers of 2, due to the negligible complexity in comparison with more general multiplications. Also, we consider the same complexity for multiplications and divisions, and hence, present the sum of them as the number of floating point operations (FLOPS). The results are summarized in Table 1.

The last column in the table presents the total number of FLOPS. To calculate this, we assume that Gauss-Jordan elimination is used to calculate the matrix inverse and $N^3 + 6N^2$ FLOPS are required to calculate the inverse of an $N \times N$ matrix. As well, we assume that the Newton method is used to calculate a scalar square root (SQRT) and 12 FLOPS are required. Moreover, the GS orthonormalization process (Orthonorm.) is considered which requires $2ND^2$ FLOPS for an $N \times D$ matrix. And, for a $D \times D$

matrix EVD computation, we consider a maximum number of D^3 FLOPS. As can be seen in the table, both PEST and PIST have a quadratic complexity in N while it is of order 5 in N (using the matrix inversion lemma) for the EKFT and of order 3 for the ordinary LS. This results in a much higher complexity (especially for large N) for the EKFT and ordinary LS in comparison with the proposed algorithms.

5 Simulation Results

In this section, we compare the performance of the explained algorithms (PEST, PIST, EKFT and ordinary LS) in different mobile network localization scenarios. We consider a network of $N = 14$ mobile sensors, living in a two-dimensional space ($D = 2$). The mobile nodes are considered to be initially deployed in an area of 100 m $\times$100 m. To obtain a fair comparison, we consider the random walk process and measurement model as described for the EKFT in Subsection 3.3. Further, we consider $l = 4$ anchors to linearize the measurement model for the ordinary LS and also $l = 4$ anchors to resolve the unknown translation and orthogonal transformation of the obtained location estimates from the anchorless algorithms using Procrustes analysis as explained in [7]. As explained earlier, the distance measurements are impaired by additive noise. The derivations for the CRB of range estimation in an additive white Gaussian noise (AWGN) channel with attenuation in [8] show that the CRB is inversely proportional to the signal-to-noise ratio (SNR) of the transmissions and directly proportional to the distance powered by the path loss exponent (κ). Therefore, for a free space model ($\kappa = 2$), we consider a constant $\gamma = d^2_{i,j,k}/\sigma^2_{i,j,k}$, which acts like the SNR and punishes the longer distances with larger measurement errors. To be able to quantitatively compare the performances of the algorithms under consideration, we consider the positioning mean squared error (PMSE) of the algorithms at the k-th snapshot, which is defined by

$$\text{PMSE} = \frac{\sum_{m=1}^{M}\sum_{n=l+1}^{N} e^2_{n,m,k}}{M}, \tag{15}$$

where $e_{n,m,k}$ represents the distance between the real location of the n-th node and its estimated location at the m-th Monte Carlo (MC) trial of the k-th snapshot. All simulations are averaged over $M = 100$ independent MC runs where in each run the nodes move toward random directions starting from random initial locations.

Fig. 1 depicts the PMSE performance of the algorithms versus γ at snapshot $k = 30$ for $T_s = 1$ s and $\sigma_w = 0.1$. As can be seen from the figure, the EKFT performs better than the other algorithms for γ values less than about 85 dB. The ordinary LS is performing close to the PIST and their performance does not saturate with γ. PEST has approximately the same performance as PIST till $\gamma = 70$ dB and after that the performance of the PEST saturates. This is because the error due to first order approximations in the PEST becomes dominant after $\gamma = 70$ dB. The EKFT performs better than the proposed algorithms because it has perfect knowledge of the statistical properties of the process and the measurement models while this knowledge is not required for our algorithms (being non-parametric) and this is a considerable advantage. To highlight this effect, we feed the EKFT with $\mathbf{Q}$ and $\mathbf{R}_k$ matrices by keeping their structure but applying a perturbation on the non-zero values. The result (EKFT with imperfect model knowledge) illustrates that the performance of the EKFT degrades and becomes worse than the other algorithms for a large span of γ. Fig. 2 shows the previous scenario but for $T_s = 10$ s and $\sigma_w = 2$. To emphasize the effect of the T_s, we have also increased the value of σ_w, which increases the movement dynamics of the process model. As can be seen from the figure, the performances of both the ordinary LS and the EKFT are affected which can be justified by their dependency on T_s. It is notable that the performance of the EKFT is significantly degraded making it the worst among the algorithms under consideration.

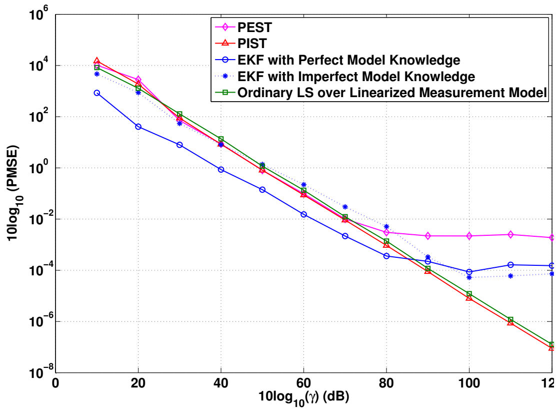

Figure 1: PMSE for $T_s = 1$ s

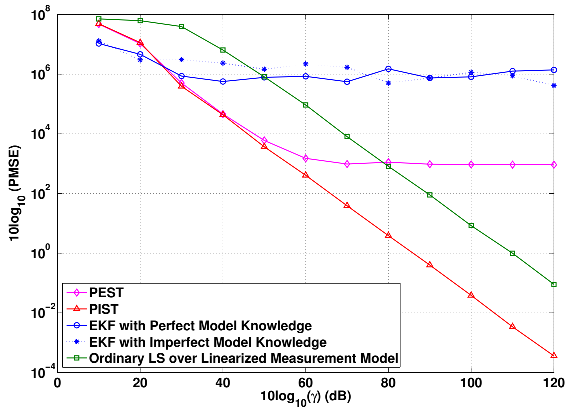

Figure 2: PMSE for $T_s = 10$ s

6 Conclusions

Classical MDS is a popular cooperative localization scheme in static WSNs. However, computing an EVD for each snapshot of a mobile network is computationally intensive. To overcome this problem, we have proposed two novel algorithms based on subspace tracking to track the variations in the signal eigenvectors and corresponding eigenvalues of the double-centered distance matrix. It has been shown that the proposed algorithms have a low computational complexity and an acceptable localization accuracy compared to the algorithms using the EKF and the ordinary LS. Future work will be conducted on the distributed realization of the proposed algorithms thereby focusing on networks with partial connectivity.

References

[1] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed Weighted MDS for Node Localization in Sensor Networks," *ACM Trans. on Sensor Networks*, vol. 2, pp. 39–64, Feb. 2006.

[2] F. K. W. Chan, and H. C. So, "Efficient Weighted Multidimensional Scaling for Wireless Sensor Networks," *IEEE Trans. on Signal Proc.*, vol. 57, pp. 4548–4553, Nov. 2009.

[3] U. Ferner, H. Wymeersch, and M. Z. Win, "Cooperative Anchorless Localization for Large Dynamic Networks" in *Proc. Intl. Conf. on Ultra-Wideband. ICUWB*, Sept. 2008, pp. 181-185.

[4] L. Dong, "Cooperative Network Localization via Node Velocity Estimation," in *Proc. IEEE WCNC*, April 2009, pp. 1–6.

[5] G. H. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 3rd ed., 1996.

[6] R. J. Vaccaro, "Weighted Subspaces Fitting Using Subspace Perturbation Expansion," in *Proc. IEEE ICASSP*, May 1998, pp. 1973-1976.

[7] K.W. Cheung and H. C. So, "A Multidimensional Scaling Framework for Mobile Location Using Time-of-arrival Measurements," *IEEE Trans. on Signal Proc.*, vol. 53, pp. 460-470, Feb. 2005.

[8] Y. Wang, G. Leus, A.-J. van der Veen, "Cramer-Rao Bound for Range Estimation," in *Proc. IEEE ICASSP*, May 2009, pp. 3301–3304.