

The results clearly show that the sparse LMS outperforms the standard LMS, which gives a good estimate for the actual spectrum. Consequently, the periodic property of the source signal can be revealed based on the estimated spectral peaks. The results also imply that the proposed sparse algorithms are workable for a practical AR process as commented in Remark 4.

## V. CONCLUSION

In this paper, we have addressed the problem of in-network distributed estimation for sparse vectors. In order to exploit the sparsity of the vector of interest, both the  $\ell_1$ - and  $\ell_0$ -norm are added into the cost function of the standard LMS, and some sparse algorithms are developed correspondingly. The mean stability and mean-square convergence of the proposed algorithms have been analyzed, and the rules for selecting the intensity of the zero-attracting term have been derived. Results show that the proposed sparse LMS algorithms can reduce the mean-square errors, whilst slightly speed up the convergence, if suitable intensities are selected. The  $\ell_1$ -RWLMS and  $\ell_0$ -LMS are much better than the  $\ell_1$ -LMS. Moreover, an application of the proposed algorithms in spectrum estimation has been presented. It is worth to point out that, the proposed methodology for deriving the sparse solutions is systematic. It can be easily extended to the other distributed algorithms, such as incremental LMS [13] and diffusion recursive least squares (RLS) [22].

## REFERENCES

- [1] W. U. Bajwa, J. Haupt, G. Raz, and R. Nowak, "Compressed channel sensing," in *Proc. 42nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, Mar. 2008, pp. 5–10.
- [2] M. Kocic, D. Brady, and M. Stojanovic, "Sparse equalization for real time digital underwater acoustic communications," in *Proc. OCEANS*, San Diego, CA, Oct. 1995, pp. 1417–1422.
- [3] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [4] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Roy. Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.
- [5] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johson, Jr., "Exploiting sparsity in adaptive filters," *IEEE Signal Process. Lett.*, vol. 50, no. 8, pp. 1883–1894, Aug. 2002.
- [6] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [7] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for identification," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2009, pp. 3125–3128.
- [8] K. Shi and P. Shi, "Convergence analysis of sparse LMS algorithms with  $\ell_1$ -norm penalty based on white input signal," *Signal Process.*, vol. 90, pp. 3289–3293, Dec. 2010.
- [9] Y. Gu, J. Jin, and S. Mei, " $\ell_0$  norm constraint LMS algorithm for sparse system identification," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 774–777, Sep. 2009.
- [10] D. P. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 115–118, Feb. 2004.
- [11] B. Jelfs, S. Javidi, P. Vayanos, and D. Mandic, "Characterisation of signal modality: Exploiting signal nonlinearity in machine learning and signal processing," *J. Signal Process. Syst.*, vol. 61, pp. 105–115, Oct. 2010.
- [12] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.
- [13] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.
- [14] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.
- [15] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

- [16] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," *Proc. 13th OCML*, pp. 82–90, 1998.
- [17] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, pp. 1–58, 1992.
- [18] P. Stoica and R. Moses, "On biased estimators and the unbiased Cramer-Rao lower bound," *Signal Process.*, vol. 21, no. 4, pp. 349–350, Oct. 1990.
- [19] M. L'azaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garc'ia, "Adaptively biasing the weights of adaptive filters," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 58, no. 7, pp. 3890–3895, Jul. 2010.
- [20] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.
- [21] P. Stoica and R. Moses, *Spectral Analysis of Signals*. Englewood Cliffs, NJ: Prentice-Hall, 2005.
- [22] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. Electron., Commun., Comput. Sci.*, vol. 90, no. 8, pp. 1504–1510, Aug. 2007.

## Dynamic Multidimensional Scaling for Low-Complexity Mobile Network Tracking

Hadi Jamali-Rad and Geert Leus

**Abstract**—Cooperative localization of mobile sensor networks is a fundamental problem which becomes challenging for anchorless networks where there is no pre-existing infrastructure to rely on. Two cooperative mobile network tracking algorithms based on novel dynamic multidimensional scaling (MDS) ideas are proposed. The algorithms are also extended to operate in partially connected networks. Compared with recently proposed algorithms based on the extended and unscented Kalman filter (EKF and UKF), the proposed algorithms have a considerably lower computational complexity. Furthermore, model-independence, scalability, as well as an acceptable accuracy make our proposed algorithms a good choice for practical mobile network tracking.

**Index Terms**—Anchor-less localization, mobile network tracking, subspace tracking.

## I. INTRODUCTION

Cost and energy prohibitive global positioning systems (GPS) motivate researchers to focus on estimating the location of sensor nodes using their pairwise distances in a cooperative context [1]. Studies on cooperative network localization can be divided into two main categories, i.e., anchored and anchorless localization. Anchored localization algorithms rely on distance measurements between the unknown-

Manuscript received November 25, 2011; revised February 10, 2012 and April 23, 2012; accepted April 23, 2012. Date of publication May 04, 2012; date of current version July 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kainan Thomas Wong. This work was supported by NWO-STW under the VICI program (Project 10382). Part of this work has been presented at the *Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 22–27, 2011 and the *32nd WIC/IEEE SP Symp. on Information Theory and Signal Processing in the Benelux (WICSP)*, Brussels, Belgium, May 10–11, 2011.

The authors are with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2826 CD Delft, The Netherlands (e-mail: h.jamali-rad@tudelft.nl; g.j.t.leus@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2012.2197751

location nodes and the anchor nodes, whereas anchorless ones can work without such information and determine the *relative* location of the sensor nodes from pairwise distance measurements. Such a relative location map could for instance be useful to determine the distribution of the nodes, but other applications might require an additional relative or absolute frame of reference. One popular anchorless localization algorithm for a static network is classical multidimensional scaling (MDS) [2] or its distributed version [3].

Surprisingly, the problem of *cooperative network* localization for *mobile* sensor networks has not been efficiently solved yet. There are a lot of studies in the literature on single and multiple target tracking using the extended and unscented Kalman filter (EKF and UKF) as well as particle filters (PFs) [4]; however, they are mainly noncooperative classical target tracking approaches. For *anchored* localization, studies in [5]–[9] investigate the problem of localizing a mobile target or network using distance measurements in an MDS-based context. In [6], for instance, a Jacobian-like mobile network tracking algorithm is proposed by exploiting the Nyström approximation. However, this approach is noncooperative.

On the other hand, in [10] an *anchorless* localization scheme for mobile network localization based on the theory of factor graphs is proposed in which each node requires knowledge about its own movement model as a probability distribution, which is not so simple to acquire in a real application. In [11] and [12], cooperative network localization algorithms based on the EKF and the UKF are developed which incorporate the locations of the nodes as well as their velocities in a state-space model. Although velocity measurements of the nodes aid cooperative network localization, it requires the use of costly Doppler sensors, and hence, we avoid using it here. Inspired by the elegance of MDS localization, we propose to use two novel subspace tracking algorithms (Section II) to track the variations in the signal eigenvectors and corresponding eigenvalues of the time-varying double-centered distance matrix. We show that this leads to a *dynamic MDS* paradigm which enables us to track the relative locations of a mobile network using only pairwise distance measurements. The absolute locations of the mobile nodes can then be recovered by the help of an absolute frame of reference provided by a few anchor nodes. In order to circumvent the limitations of the classical MDS, we then also propose an extension for partially connected mobile networks (Section IV). A detailed computational complexity analysis as well as the posterior Cramér–Rao bound (PCRB) derivation (Section III) together with extensive simulation results (Section V) illustrate that the proposed algorithms are scalable, acceptably accurate and have a much lower computational complexity compared to algorithms based on the EKF [11] and the UKF [12].

## II. DYNAMIC MULTIDIMENSIONAL SCALING

In this section, we formulate the problem of cooperative network localization and develop the dynamic MDS idea.

### A. Problem Formulation

We consider a network of  $N$  mobile wireless sensor nodes, living in a  $D$ -dimensional space ( $D < N$ ). Let  $\mathbf{x}_{i,k}$  be the actual coordinate vector of the  $i$ th sensor node at the  $k$ th snapshot of the mobile network, or equivalently, let  $\mathbf{X}_k = [\mathbf{x}_{1,k}, \dots, \mathbf{x}_{N,k}]$  be the matrix of coordinates. Let us consider an environment with line-of-sight (LOS) conditions between the nodes and let us assume that time of flight (ToF) and/or received signal strength (RSS) information is already converted into noisy distance measurements as  $r_{i,j,k} = d_{i,j,k} + v_{i,j,k}$ , where  $d_{i,j,k} = \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|$  is the noise-free Euclidean distance and  $v_{i,j,k} \sim \mathcal{N}(0, \sigma_{v,i,j,k}^2)$  is the additive white noise both at the  $k$ th snapshot. The problem considered herein can now be stated as follows. Given the pairwise noisy distance measurements  $r_{i,j,k}$  at each snapshot of the mobile network, determine the location of the mobile nodes and keep

their track (up to a translation and orthogonal transformation). In case of a network with fixed nodes, the squared noisy distance measurements  $r_{i,j,k}^2$  between the nodes can be collected in a distance matrix  $\mathbf{D}_k$ , i.e.,  $[\mathbf{D}_k]_{i,j} = r_{i,j,k}^2$ , after which the double-centered distance matrix can be calculated as  $\mathbf{B}_k = -\mathbf{\Gamma} \mathbf{D}_k \mathbf{\Gamma} / 2$  using the centering operator  $\mathbf{\Gamma} = \mathbf{I}_N - \mathbf{1}_N \mathbf{1}_N^T$ , where  $\mathbf{I}_N$  denotes an  $N \times N$  identity matrix and  $\mathbf{1}_N$  represents an  $N \times 1$  vector of all ones. For the  $k$ th snapshot of the mobile network, the well-known MDS approach [2], [3], [5] then finds the locations as the solution to  $\min_{\tilde{\mathbf{X}}} \|\mathbf{B}_k - \tilde{\mathbf{X}}^T \tilde{\mathbf{X}}\|_F^2$ , where the minimum is taken over all  $D \times N$  matrices  $\tilde{\mathbf{X}}$  and  $\|\cdot\|_F$  denotes the Frobenius norm. The solution can be found by means of the eigenvalue decomposition (EVD) of  $\mathbf{B}_k$ , which can be expressed as

$$\mathbf{B}_k = [\mathbf{U}_{1,k} \quad \mathbf{U}_{2,k}] \begin{bmatrix} \mathbf{\Sigma}_{1,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{2,k} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{1,k}^T \\ \mathbf{U}_{2,k}^T \end{bmatrix} \quad (1)$$

where  $\mathbf{U}_{1,k}$  and  $\mathbf{U}_{2,k}$ , respectively, represent the  $N \times D$  and  $N \times (N - D)$  matrices containing the orthonormal eigenvectors corresponding to the signal and noise subspace of  $\mathbf{B}_k$ , and  $\mathbf{\Sigma}_{1,k}$  and  $\mathbf{\Sigma}_{2,k}$ , respectively, contain the eigenvalues corresponding to the signal and noise subspace. The MDS estimate of the location matrix up to a translation and orthogonal transformation can then be expressed as

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

In the noiseless case,  $\tilde{\mathbf{X}}_k = \mathbf{\Psi} \mathbf{X}_k \mathbf{\Gamma}$ , where  $\mathbf{\Psi}$  is an arbitrary orthogonal transformation and  $\mathbf{\Gamma}$  translates the nodes such that their center of gravity is at the origin. Although the above procedure can be carried out for every snapshot of the mobile network, the complexity of computing the EVD in (1) can be quite intensive for large  $N$  [13], especially when the nodes have to be monitored continuously. The idea behind the proposed dynamic MDS materialized by two 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  $\mathbf{\Sigma}_{1,k}$  [14], [15]. This can be done by more efficient iterative approaches as follows.

### B. Perturbation Expansion-Based Subspace Tracking

In this subsection, we will present the perturbation expansion-based subspace tracking (PEST) algorithm. The idea is that in a mobile network the new location of a node can be considered as a perturbation of its previous location. Correspondingly, the double-centered distance matrix  $\mathbf{B}_k$  can also be modeled as a perturbed version of  $\mathbf{B}_{k-1}$  ( $\mathbf{B}_k = \mathbf{B}_{k-1} + \Delta \mathbf{B}_k$ ). Now, if the movement of the nodes satisfies the property that the invariant subspace (here, the signal subspace) of the next (perturbed) double-centered distance matrix  $\mathbf{B}_k$  does not contain any vectors that are orthogonal to the invariant subspace of the current  $\mathbf{B}_{k-1}$ , then the two bases respectively spanning the signal and noise subspace of the next double-centered distance matrix follow the expressions [16]

$$\tilde{\mathbf{U}}_{1,k}^u = \tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1} \mathbf{P}_k \quad (3)$$

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

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  $\tilde{\mathbf{U}}_{i,k}$ . Observe that in (3) and (4), different from the expressions in [16], we do not necessarily have the matrices of eigenvectors  $\mathbf{U}_{i,k}$  on the right-hand side. In order to keep the computational complexity as low as possible, we will resort to a first-order approximation to compute  $\mathbf{P}_k$ . However, since we will continuously use first-order approximations, we cannot assume that  $\tilde{\mathbf{U}}_{1,k-1}$  and  $\tilde{\mathbf{U}}_{2,k-1}$  in (3) and (4) are orthonormal bases exactly spanning respectively the signal and noise subspaces of  $\mathbf{B}_{k-1}$ . And thus, the first-order approximation of  $\mathbf{P}_k$  in [16] 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 bases for the perturbed signal and noise subspaces. Thus, we need

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

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

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

Now by using  $\mathbf{B}_k = \mathbf{B}_{k-1} + \Delta \mathbf{B}_k$  we can rewrite (6) as

$$\begin{aligned} & -\underbrace{\mathbf{P}_k \tilde{\mathbf{U}}_{1,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{2,k-1} \mathbf{P}_k}_{\text{2nd order}} + \underbrace{\mathbf{P}_k \tilde{\mathbf{U}}_{1,k-1}^T \Delta \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1}}_{\text{2nd order}} \\ & + \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T \Delta \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1}}_{\text{2nd order}} + \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T \Delta \mathbf{B}_k \tilde{\mathbf{U}}_{2,k-1} \mathbf{P}_k}_{\text{2nd order}} \\ & - \mathbf{P}_k \tilde{\mathbf{U}}_{1,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{1,k-1} - \tilde{\mathbf{U}}_{2,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{1,k-1} \\ & + \tilde{\mathbf{U}}_{2,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{2,k-1} \mathbf{P}_k = \mathbf{0}. \end{aligned} \quad (7)$$

Note that for small perturbations,  $\mathbf{P}_k$  in (7) will be close to a zero matrix. Thus, by neglecting the second-order terms, we obtain

$$\begin{aligned} & -\mathbf{P}_k \tilde{\mathbf{U}}_{1,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}^T \Delta \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1} \\ & + \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{1,k-1}}_{\neq 0} + \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{2,k-1} \mathbf{P}_k}_{\neq 0} = \mathbf{0}. \end{aligned} \quad (8)$$

Different from the derivations in [16], the third and fourth terms in (8) are close but not equal to zero due to the successive first-order approximations as explained earlier. It is notable that (8) 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 (8). By defining

$$\tilde{\Sigma}_{1,k-1} = \tilde{\mathbf{U}}_{1,k-1}^T \mathbf{B}_{k-1} \tilde{\mathbf{U}}_{1,k-1}, \quad (9)$$

this results in

$$\mathbf{P}_k = \tilde{\mathbf{U}}_{2,k-1}^T \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1} \tilde{\Sigma}_{1,k-1}^{-1}. \quad (10)$$

To avoid updating  $\tilde{\mathbf{U}}_{2,k}^u$  in (3), we use the property that  $\tilde{\mathbf{U}}_{1,k-1} \tilde{\mathbf{U}}_{1,k-1}^T + \tilde{\mathbf{U}}_{2,k-1} \tilde{\mathbf{U}}_{2,k-1}^T = \mathbf{I}_N$ . Together with (10), this allows us to rewrite (3) as

$$\tilde{\mathbf{U}}_{1,k}^u = \tilde{\mathbf{U}}_{1,k-1} + (\mathbf{I} - \tilde{\mathbf{U}}_{1,k-1} \tilde{\mathbf{U}}_{1,k-1}^T) \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1} \tilde{\Sigma}_{1,k-1}^{-1}. \quad (11)$$

Now, to be able to use the above formula in an iterative fashion we can normalize it using any orthonormalization process like Gram-Schmidt (GS) factorization. We call the orthonormalized result  $\tilde{\mathbf{U}}_{1,k}$ . 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 the matrix of eigenvectors  $\mathbf{U}_{1,k}$ . To this aim, we look for a transformation matrix  $\mathbf{A}_k$  to map  $\tilde{\mathbf{U}}_{1,k}$  to  $\mathbf{U}_{1,k}$  as follows:

$$\tilde{\mathbf{U}}_{1,k} = \mathbf{U}_{1,k} \mathbf{A}_k. \quad (12)$$

Note that since  $\tilde{\mathbf{U}}_{1,k}$  and  $\mathbf{U}_{1,k}$  are isometries,  $\mathbf{A}_k$  will be a unitary matrix according to the definition in (12). To be able to estimate the locations using (2), we also need to calculate  $\Sigma_{1,k}$ , which depends on the value of  $\mathbf{U}_{1,k}$  and  $\mathbf{B}_k$  as  $\Sigma_{1,k} = \mathbf{U}_{1,k}^T \mathbf{B}_k \mathbf{U}_{1,k}$ . From (9), and using (12), we finally obtain

$$\tilde{\Sigma}_{1,k} = (\mathbf{U}_{1,k} \mathbf{A}_k)^T \mathbf{B}_k (\mathbf{U}_{1,k} \mathbf{A}_k) = \mathbf{A}_k^T \Sigma_{1,k} \mathbf{A}_k. \quad (13)$$

From (13),  $\mathbf{A}_k$  and  $\Sigma_{1,k}$  can be calculated by the EVD of  $\tilde{\Sigma}_{1,k}$ . 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  $\tilde{\Sigma}_{1,k}$  is a  $D \times D$  matrix (the number of dimensions  $D$  is in practice at most 3), 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 overall PEST algorithm is summarized in Algorithm 1. Increasing the measurement interval decreases the computational cost but introduces larger perturbations, which leads to a degraded result. To heal this degradation, we can divide  $\Delta \mathbf{B}_k$  in  $P$  proportional parts and run the PEST algorithm  $P$  times in each snapshot by successively applying these partial perturbations as shown by the following measurement update equation:

$$\mathbf{B}_{k,p} = \mathbf{B}_{k-1} + p \frac{\Delta \mathbf{B}_k}{P}, \quad p = 1, \dots, P.$$

We call this modified algorithm the modified PEST.

### C. Power Iteration-Based Subspace Tracking

Power iterations can also be used to efficiently calculate an invariant subspace of a diagonalizable matrix (like  $\mathbf{B}_k$ ) [13]. Power iterations are normally used in an iterative manner till an acceptable accuracy is reached. Depending on the 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 [13]. 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). Here, instead of using (11) as for the PEST, an unorthonormalized version of  $\tilde{\mathbf{U}}_{1,k}$  can be calculated using

$$\tilde{\mathbf{U}}_{1,k}^u = \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1}. \quad (14)$$

Note that the resulting  $\tilde{\mathbf{U}}_{1,k}$  after orthonormalization will again be an orthonormal basis spanning the desired signal subspace. Thus, the same EVD calculations as in (13) for PEST are required to obtain the matrix of eigenvectors. The overall PIST algorithm is summarized in Algorithm 1. We emphasize that the proposed algorithms are *anchorless* in the sense that the relative position of the mobile nodes (also called network configuration in this context) can continuously be calculated without requiring any anchor nodes. However, determining the absolute 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. Hence, if recovering the absolute locations is also of interest, e.g., for comparison purposes, then a possible additional step can be implemented for every snapshot of the mobile network.

---

#### Algorithm 1: PEST/PIST

---

- 1: Start with an initial location guess
  - 2: **for**  $k = 1$  to  $K$  (movement steps) **do**
  - 3:   Calculate  $\tilde{\mathbf{U}}_{1,k}^u$  using (11) (for PEST) or (14) (for PIST)
  - 4:   GS orthonormalization  $\tilde{\mathbf{U}}_{1,k} = GS(\tilde{\mathbf{U}}_{1,k}^u)$
  - 5:   Calculate  $\tilde{\Sigma}_{1,k}$ ,  $\mathbf{A}_k$  and  $\Sigma_{1,k}$  using (9) and (13)
  - 6:   Calculate  $\mathbf{U}_{1,k}$  using (12)
  - 7:   Location estimation using (2)
  - 8: **end for**
-

TABLE I  
COMPUTATIONAL COMPLEXITY

Algo.	Multiplication	GS	SQRT	Matrix inverse	EVD	Total FLOPS
PEST	$4N^2D + 3ND^2 + ND$	$1(N \times D)$	2 (Scalar)	$1(D \times D)$	$1(D \times D)$	$4DN^2 + (5D^2 + D)N + 2D^3 + 6D^2 + 24$
PIST	$2N^2D + 2ND^2 + ND$	$1(N \times D)$	2 (Scalar)	-	$1(D \times D)$	$4DN^2 + (5D^2 + D)N + D^3 + 24$
EKF [11]	$(D/2)N^5 + (5D^2/2 - D)N^4 + (12D^3 - 5D^2/2 + 2D)N^3 + (4D^2 - D/2 + 2)N^2 - 2N$	-	$DN(N - 1)/2$ (Scalar)	$2(2DN \times 2DN)$	-	$(D/2)N^5 + (5D^2/2 - D)N^4 + (28D^3 - 5D^2/2 + 2D)N^3 + (52D^2 + 11D/2 + 2)N^2 + (-6D - 2)N$
UKF [12]	$(7D/2)N^5 + (6D^2 - 3D + 1/2)N^4 + (16D^3/3 - 8D^2 + 25D/2 - 1)N^3 + (D^2 - 9D + 2)N^2 + (D - 3/2)N$	-	$2DN^3 - (2D - 1/2)N^2 - 1/2N$ (Scalar) and $1(2DN \times 2DN)$	$1(N(N - 1)/2 \times N(N - 1)/2)$	-	$(1/8)N^6 + (7D/2 - 3/8)N^5 + (6D^2 - 3D + 19/8)N^4 + (16D^3/3 - 8D^2 + 73D/2 - 33/8)N^3 + (D^2 - 33D + 19/2)N^2 + (D - 3/2)N$

### III. ANALYSIS OF THE PROPOSED ALGORITHMS

In the following, we analyze the computational complexity and attainable accuracy of the algorithms under consideration.

#### A. Computational Complexity

We define the computational complexity as the number of operations required to estimate the locations for a single snapshot. For the sake of simplicity, we do not count the number of additions and subtractions, due to their negligible complexity in comparison with the other operations. Also, we consider the same complexity for multiplications and divisions, and hence, we present the sum of them as the number of floating point operations (FLOPS). The results of the complexity calculations for the PEST, the PIST, the EKF, and the UKF algorithms are summarized in Table I. The last column in the table presents the maximum number of FLOPS. To calculate this value, we assume that Gauss-Jordan elimination requiring  $N^3 + 6N^2$  multiplications is used to calculate the inverse of an  $N \times N$  matrix. Further, we assume that Newton's method is used to calculate a scalar square root (SQRT) which requires 12 multiplications and a Cholesky decomposition is used to calculate a matrix square root which requires  $2N^3/3$  multiplications for an  $N \times N$  matrix [8]. Moreover, a GS orthonormalization process is considered which requires  $2ND^2$  multiplications for an  $N \times D$  matrix [17]. For a  $D \times D$  matrix EVD computation, we consider a maximum number of  $D^3$  multiplications [17]. As can be seen in the table, both PEST and PIST have a quadratic complexity in  $N$  while it is of order 5 and 6 in  $N$  for the EKF and UKF, respectively. As can be seen, the considerably lower computational complexity is a significant gain for the proposed algorithms, especially for large networks (large  $N$ ). Finally, the Jacobian-like algorithm proposed in [6] although being noncooperative approximately leads to a complexity order of 3 in  $N$  which is still one order of magnitude larger than our complexity. An advantage of this low complexity is that the central unit of our algorithm can simply be one of the nodes of the network.

#### B. Tracking Accuracy

To derive the tracking accuracy, let us assume that the nodes move according to the following state-space model:

$$\mathbf{s}_k = \mathbf{\Phi}_k \mathbf{s}_{k-1} + \mathbf{w}_k + \mathbf{m}_{k-1} \quad (15)$$

$$\mathbf{r}_k = \mathbf{h}(\mathbf{s}_k) + \mathbf{v}_k \quad (16)$$

where  $\mathbf{s}_k = [\mathbf{x}_{1,k}^T, \dots, \mathbf{x}_{N,k}^T, \dot{\mathbf{x}}_{1,k}^T, \dots, \dot{\mathbf{x}}_{N,k}^T]^T$  is a column vector of length  $2DN$  containing the locations and velocities at the  $k$ th snapshot, and  $\mathbf{r}_k = [r_{1,2,k}, r_{1,3,k}, \dots, r_{N-1,N,k}]^T$  is the column vector of pairwise distance measurements of length  $N(N - 1)/2$  at the  $k$ th snapshot. Next,  $\mathbf{h}(\cdot)$  is a deterministic observation function which relates the locations of the nodes (inside  $\mathbf{s}_k$ ) to their corresponding pairwise distances and  $\mathbf{m}_{k-1}$  is an optional control input at the  $(k - 1)$ th snapshot [18]. Further,  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are vectors with zero mean Gaussian

entries with standard deviation (std)  $\sigma_{w,k}$  and  $\sigma_{v,i,j,k}$ , respectively. For the sake of clarity, we denote the elements of the state vector as  $\mathbf{s}_k = [\mathbf{s}_{l,k}^T, \mathbf{s}_{v,k}^T]^T$ , where  $\mathbf{s}_{l,k}$  of length  $DN$  represents the vectorized version of the locations and  $\mathbf{s}_{v,k}$  of length  $DN$  represents the vectorized version of the corresponding velocities. The lower bound on the mean squared error (MSE) of estimation for any discrete-time non-linear filtering problem can be computed via the posterior Cramér-Rao bound (PCRB) [19]. For our problem, i.e., estimating the locations in the state vector  $\mathbf{s}_k$  using all the previous and current measurements  $\mathbf{r}_0, \dots, \mathbf{r}_k$ , the lower bound on the MSE covariance matrix (matrix of the state error) of any unbiased estimator is given by

$$\mathbb{E}\{[\hat{\mathbf{s}}_k - \mathbf{s}_k][\hat{\mathbf{s}}_k - \mathbf{s}_k]^T\} \geq \mathbf{J}_k^{-1} \quad (17)$$

where  $\mathbb{E}(\cdot)$  stands for statistical expectation and  $\hat{\mathbf{s}}_k$  is the state estimate. The recursive PCRB derived in [19] for updating the posterior Fisher information matrix ( $\mathbf{J}_k$ ) for our model expressed by (15) and (16) boils down to

$$\mathbf{J}_k = \left( \mathbf{Q}_k + \mathbf{\Phi}_k \mathbf{J}_{k-1}^{-1} \mathbf{\Phi}_k^T \right)^{-1} + [\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)]^T \mathbf{R}_k^{-1} [\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)] \quad (18)$$

where  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  respectively represent the exact covariance matrices of the process (movement) and measurement noise  $\mathbf{w}_k$  and  $\mathbf{v}_k$ , and the gradient  $\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)$  should be calculated using the true locations. Since we basically estimate the locations of the nodes and not their velocities, the PCRB of our location estimates is given by

$$\text{PCRB}_k = \sum_{i=1}^{DN} [\mathbf{J}_k^{-1}]_{i,i} \quad (19)$$

which we average over different Monte Carlo (MC) realizations of the movement process. It is worth mentioning that the MSE of our location estimates will correspond to the errors on the absolute locations and not on those up to a translation and orthogonal transformation. As mentioned earlier, the absolute locations can be recovered by considering  $l$  anchor nodes with known locations. Now, if we compute (18) for the location estimates of our anchorless network,  $\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)$  and correspondingly  $[\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)]^T \mathbf{R}_k^{-1} [\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)]$  will be rank deficient with a rank of at most  $DM - D - 1$  (due to the unknown translation and orthogonal transformation in every snapshot). To resolve this problem, we try to obtain a bound by reformulating (18) for a network with  $l$  anchor nodes, and modify the process and measurement models as

$$\bar{\mathbf{s}}_k = \bar{\mathbf{\Phi}}_k \bar{\mathbf{s}}_{k-1} + \bar{\mathbf{w}}_k + \bar{\mathbf{m}}_{k-1} \quad (20)$$

$$\bar{\mathbf{r}}_k = \mathbf{h}(\bar{\mathbf{s}}_k) + \bar{\mathbf{v}}_k \quad (21)$$

where  $\bar{\mathbf{s}}_k$ ,  $\bar{\mathbf{w}}_k$ , and  $\bar{\mathbf{m}}_{k-1}$  are  $2D(N - l) \times 1$  vectors calculated by removing the elements corresponding to the locations and velocities of the anchors from  $\mathbf{s}_k$ ,  $\mathbf{w}_k$  and  $\mathbf{m}_{k-1}$ , respectively. Therefore,  $\bar{\mathbf{\Phi}}_k$  will be a  $2D(N - l) \times 2D(N - l)$  matrix relating the previous modified state

vector  $\bar{\mathbf{s}}_{k-1}$  to the next one  $\bar{\mathbf{s}}_k$ . For the modified measurement model,  $\bar{\mathbf{r}}_k$  is an  $(N(N-1)/2 - |\Omega|) \times 1$  vector similar to  $\mathbf{r}_k$  but the noisy distance measurements ( $r_{i,j,k}$ ) between the  $l$  anchors are removed ( $|\cdot|$  denotes the cardinality). The indices of the removed distance measurements are contained in

$$\Omega = \left\{ (i-1)N - \frac{i(i-1)}{2} + 1, \dots, (i-1)N - \frac{i(i-1)}{2} + l - i \mid i = 1, 2, \dots, l - 1 \right\}.$$

The modified sequence of the posterior FIM can then be obtained as

$$\bar{\mathbf{J}}_k = \left( \bar{\mathbf{Q}}_k + \bar{\Phi}_k \bar{\mathbf{J}}_{k-1}^{-1} \bar{\Phi}_k^T \right)^{-1} + [\nabla_{\bar{\mathbf{s}}_k} \mathbf{h}(\bar{\mathbf{s}}_k)]^T \bar{\mathbf{R}}_k^{-1} [\nabla_{\bar{\mathbf{s}}_k} \mathbf{h}(\bar{\mathbf{s}}_k)]$$

where  $\bar{\mathbf{Q}}_k$  is the  $2D(N-l) \times 2D(N-l)$  process noise covariance matrix corresponding to  $\bar{\mathbf{w}}_k$ . The modified measurement noise covariance matrix  $\bar{\mathbf{R}}_k$  will be a  $(N(N-1)/2 - |\Omega|) \times (N(N-1)/2 - |\Omega|)$  diagonal matrix similar to  $\mathbf{R}_k$  but corresponding to  $\bar{\mathbf{v}}_k$ . Further,  $\nabla_{\bar{\mathbf{s}}_k} \mathbf{h}(\bar{\mathbf{s}}_k)$  is a  $(N(N-1)/2 - |\Omega|) \times 2D(N-l)$  matrix similar to  $\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)$  but it is calculated by taking partial derivatives from the remaining  $N(N-1)/2 - |\Omega|$  distance measurements with respect to the  $2D(N-l)$  elements in the modified state vector  $\bar{\mathbf{s}}_k$ .

#### IV. EXTENSION TO PARTIALLY CONNECTED NETWORKS

The derivations of the proposed algorithms in Section II are based on the assumption that all the pairwise distance measurements are available. However, this assumption is not valid for many practical mobile scenarios where the nodes only have a limited communication range. Therefore, we also consider a simple finite-range model where the distances can be measured only if they are below a certain communication range  $r_0$ , otherwise they cannot be measured and they are considered *missing links*. To tackle this problem, there has been a lot of research in the literature to reconstruct the squared distance matrix  $\mathbf{D}_k$  or correspondingly its double-centered version  $\mathbf{B}_k$  by exploiting their specific properties like rank and inertia [20], [21]. However, we are interested in a low-complexity algorithm which also fits to our proposed dynamic MDS model. To this aim, we propose to include an additional inner iterative procedure (iterating  $P$  times in each snapshot) to account for the missing links. In each snapshot, we first construct  $\hat{\mathbf{D}}_k$  from the measured noisy  $\mathbf{D}_k$  as

$$[\hat{\mathbf{D}}_k]_{i,j} = \begin{cases} [\mathbf{D}_k]_{i,j}, & (i,j) \text{ measured} \\ [\hat{\mathbf{D}}_{k-1}]_{i,j}, & (i,j) \text{ missing \& } [\hat{\mathbf{D}}_{k-1}]_{i,j} > r_0^2 \\ r_0^2, & (i,j) \text{ missing \& } [\hat{\mathbf{D}}_{k-1}]_{i,j} \leq r_0^2 \end{cases} \quad (22)$$

where the link between nodes  $i$  and  $j$  is denoted by  $(i,j)$ . As is clear from (22), we fill the missing links with their corresponding previously recovered distance estimates, if their value is larger than  $r_0$ ; otherwise we just fill the missing links with  $r_0$  since we know that they should be larger than  $r_0$ . We then use the modified squared distance matrix  $\hat{\mathbf{D}}_k$  to calculate  $\hat{\mathbf{B}}_k$ , which we feed to the PEST or the PIST to calculate the signal eigenvectors and corresponding eigenvalues. Then the relative locations of the nodes are used to recalculate a new set of pairwise distances and to construct a temporary squared distance matrix  $\mathbf{E}_k$  similar to  $\hat{\mathbf{D}}_k$ . Then, we modify  $\hat{\mathbf{D}}_k$  by updating the distances corresponding to the missing links from the recently calculated  $\mathbf{E}_k$  as

$$[\hat{\mathbf{D}}_k]_{i,j} = \begin{cases} [\hat{\mathbf{D}}_k]_{i,j}, & (i,j) \text{ measured} \\ [\hat{\mathbf{D}}_{k-1}]_{i,j} + \rho \left( [\mathbf{E}_k]_{i,j} - [\hat{\mathbf{D}}_{k-1}]_{i,j} \right), & (i,j) \text{ missing} \end{cases} \quad (23)$$

where  $\rho \in (0, 1]$  is a smoothing gain. This gain avoids divergence of the algorithm for cases where the signal subspace is affected due to a large number of missing links. Now, a new  $\hat{\mathbf{B}}_k$  can be calculated from

the recently updated  $\hat{\mathbf{D}}_k$  which can be used for the next (inner) iteration in the same snapshot. The final  $\hat{\mathbf{D}}_k$  from the inner loop will be transferred to the next snapshot. The modified iterative algorithm for partially connected networks is shown in Algorithm 2. Note that these  $P$  inner iterations scale the computational complexity of the algorithms by at most a factor  $P$ . Since in practice  $P \leq 10$ , this does not increase the order of complexity of the modified algorithms for networks with  $N > 10$ . It is noteworthy that different from ranging, communication between each node and the central unit can be accomplished by multi-hop communications.

---

#### Algorithm 2: Extension to Partially Connected Networks

---

- 1: Start with an initial location guess
  - 2: **for**  $k = 1$  to  $K$  (movement steps) **do**
  - 3: Construct  $\hat{\mathbf{D}}_k$  from  $\mathbf{D}_k$  using (22)
  - 4: Calculate  $\hat{\mathbf{B}}_k$  from  $\hat{\mathbf{D}}_k$
  - 5: **for**  $p = 1$  to  $P$  **do**
  - 6: Use PEST/PIST to estimate locations from  $\hat{\mathbf{B}}_k$
  - 7: Calculate new pairwise distances and construct  $\mathbf{E}_k$
  - 8: Update  $\hat{\mathbf{D}}_k$  using (23)
  - 9: Calculate a new  $\hat{\mathbf{B}}_k$  for the next (inner) iteration
  - 10: **end for**
  - 11: **end for**
- 

#### V. SIMULATION RESULTS

We consider a network of  $N$  mobile sensors, living in a two-dimensional space ( $D = 2$ ). The mobile nodes are considered to be moving inside a bounded area of  $100 \times 100$  squared meters determined by its vertices located at (0,0)m, (0,100)m, (100,0)m, and (100,100)m. Note that our proposed algorithms are blind to the movement model, but for the sake of comparison we consider a modified random walk process where  $\Phi = \mathbf{I}_{4N} + \mathbf{F}T_s$ , with  $T_s$  the measurement interval and  $\mathbf{F} = [\mathbf{0}_{2N \times 2N}, \mathbf{I}_{2N \times 2N}; \mathbf{0}_{2N \times 2N}, \mathbf{0}_{2N \times 2N}]$ . We set  $\mathbf{w}_k = [\mathbf{0}^T, \hat{\mathbf{w}}_k^T]^T$ , where we assume that  $\hat{\mathbf{w}}_k$  is a vector with i.i.d. zero-mean Gaussian entries with std  $\sigma_w$ . This movement model does not guarantee that the mobile nodes stay inside the bounded area. To make this happen without greatly violating the predefined movement model in favor of the model-based algorithms (the EKF and the UKF), we propose to slightly change the movement pattern so that each time a node gets closer than a threshold ( $d_0 = 5$  m) to the borders of the covered area, we gradually decrease the velocity of that particular node with a centripetal force. The center of the area is  $\mathbf{c} = (50, 50)$  m. Let us define the  $2N \times 2N$  diagonal matrix  $\mathbf{G}_{k-1} = \text{diag}(\mathbf{g}_{k-1})$ , where  $\mathbf{g}_{k-1}$  is given by

$$[\mathbf{g}_{k-1}]_i = \begin{cases} 0, & [c]_1 - |[s_{l,k-1}]_{2i-1} - [c]_1| < d_0 \\ \left[ \frac{[c]_1 - |[s_{l,k-1}]_{2i-1} - [c]_1|}{[c]_1} \right]^{\frac{1}{\alpha}}, & [c]_1 - |[s_{l,k-1}]_{2i-1} - [c]_1| \geq d_0 \\ 0, & [c]_2 - |[s_{l,k-1}]_{2i} - [c]_2| < d_0 \\ \left[ \frac{[c]_2 - |[s_{l,k-1}]_{2i} - [c]_2|}{[c]_2} \right]^{\frac{1}{\alpha}}, & [c]_2 - |[s_{l,k-1}]_{2i} - [c]_2| \geq d_0 \end{cases}$$

with  $i = 1, 2, \dots, N$ . This equation investigates whether or not the nodes are closer to the borders than the threshold. Now, the velocity of the nodes in the next step will be computed as

$$\mathbf{s}_{v,k} = \mathbf{G}_{k-1} \mathbf{s}_{v,k-1} + \mathbf{G}_{k-1} \hat{\mathbf{w}}_k + \underbrace{\sigma_w [\mathbf{I}_{2N} - \mathbf{G}_{k-1}] \left( \frac{[c]_1 \mathbf{1}_{2N} - \mathbf{s}_{l,k-1}}{\|[c]_1 \mathbf{1}_{2N} - \mathbf{s}_{l,k-1}\|} \right)}_{\hat{\mathbf{m}}_{k-1}} \quad (24)$$

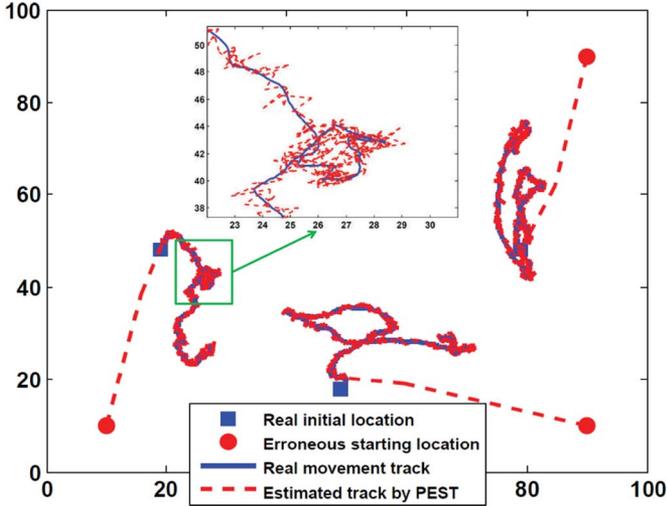


Fig. 1. Tracking of a single realization from erroneous initial locations.

where the third term  $\check{\mathbf{m}}_{k-1}$  is the  $2N \times 1$  nonzero vector in the optional control input ( $\mathbf{m}_{k-1} = [\mathbf{0}^T \check{\mathbf{m}}_{k-1}^T]^T$ ), which imposes a centripetal force directed towards the center of the area  $c$ . Note that the elements of the  $\mathbf{G}_{k-1}$  matrix are 0 for the nodes that have passed the threshold, and therefore only the third term pulls them back into the area. For those nodes that have not passed the threshold, the elements of  $\mathbf{G}_{k-1}$  are close to 1 since  $\alpha$  is chosen to be a large integer  $10 < \alpha < 20$ , and therefore (24) acts very close to the classical random walk ( $\mathbf{s}_{v,k} = \mathbf{s}_{v,k-1} + \check{\mathbf{w}}_k$ ) for those nodes. Inspired by the CRB for range estimation in additive white Gaussian noise, following [7] and [9], for a realistic free-space model we introduce a constant  $\gamma = d_{i,j,k}^2 / \sigma_{v,i,j,k}^2$ , which punishes the longer distances with larger measurement errors. For a quantitative comparison, we consider the positioning root mean-squared error (PRMSE) of the algorithms at the  $k$ th snapshot, which is defined by

$$\text{PRMSE}_k = \sqrt{\frac{\sum_{m=1}^M \sum_{n=1}^N e_{n,m,k}^2}{M}} \quad (25)$$

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 MC trial of the  $k$ th snapshot. All simulations are averaged over  $M = 100$  independent MC runs where in each run the nodes move in random directions starting from random initial locations. For the sake of comparison, we also simulate the cooperative network localization method of [11] based on the EKF and also the algorithm in [12] based on the UKF modified to our setup. Fig. 1 illustrates a realization of the mobile network ( $N = 3$ ) were for the sake of clarity only the PEST is plotted (we show in the following simulations that both algorithms have very close performances). For all simulations, to be able to plot and/or evaluate the results based on the absolute locations, we resolve the unknown translation and orthogonal transformation of our location estimates by considering  $l = 3$  anchor nodes. In general, for all the simulations, we initialize the algorithms with random erroneous locations. Here, for the sake of visibility, we initialize the algorithm close to the borders of the covered area, which is far from the real initial locations. As is clear, convergence is a matter of a few steps. During our simulations we observed that random initializations lead to divergence of the EKF in many of its runs, while the UKF and (even better) our proposed algorithms are robust against erroneous initializations.

Fig. 2 shows the PRMSE performance of the algorithms versus  $\gamma$  for  $N = 10$ ,  $T_s = 0.1$  s,  $\sigma_w = 0.1$  and at the snapshot  $k = 250$ , where all the algorithms have converged. We also plot the performance

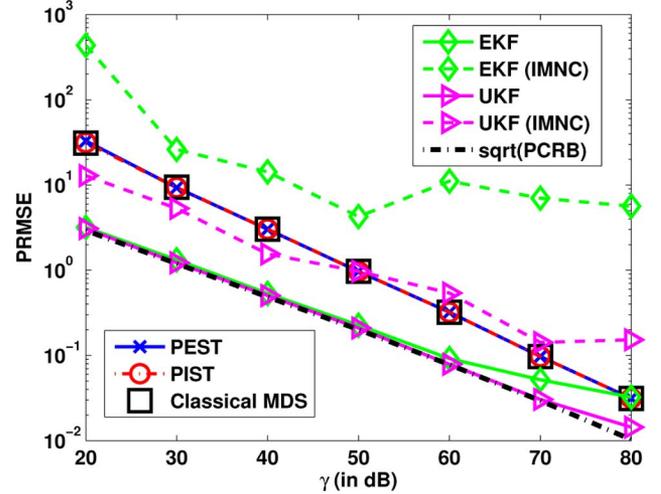


Fig. 2. PRMSE performance for  $T_s = 0.1$  s.

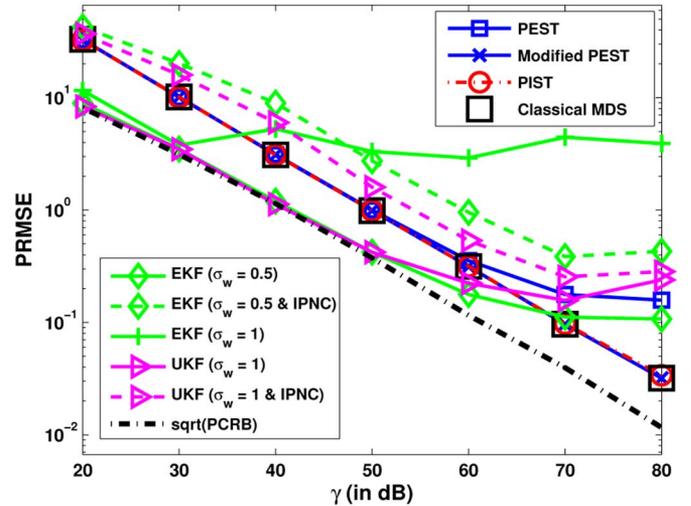


Fig. 3. PRMSE performance for  $T_s = 1$  s.

of classical MDS and the derived PCRB as the performance bounds of the algorithms. From the figure, the PEST and the PIST perform very close to each other and attain the classical MDS performance while they are much more computationally efficient. The EKF performs better than the proposed algorithms in terms of accuracy, and the UKF is even better than the EKF (closer to the PCRB) but they both come at the price of a much higher complexity and depend on the information about the process and measurement models. That is why if we feed both the EKF and UKF with imperfect measurement noise covariance (IMNC) information (here,  $\mathbf{R}_k$  with 40% error), the EKF diverges drastically while the UKF degrades and performs worse than the proposed algorithms for  $\gamma > 50$  dB. Beyond the computational efficiency, this is another advantage of our proposed *model-independent* algorithms over model-based ones (the EKF and the UKF). Fig. 3 depicts the same results as Fig. 2 ( $N = 10$ ) but for  $T_s = 1$  s and  $\sigma_w = 0.5$  and 1. Increasing  $\sigma_w$  boosts the effect of increasing  $T_s$ . From the figure, by increasing  $T_s$  and  $\sigma_w$  the EKF diverges drastically even with perfect model information while in a similar situation the UKF is just degraded for  $\gamma > 50$  dB. The PIST performs superb and the PEST is a little bit degraded for  $\gamma > 60$  dB, which can be healed by using the modified PEST as explained in Section II-B. Again we investigate the model-dependency of the EKF and the UKF by feeding them with

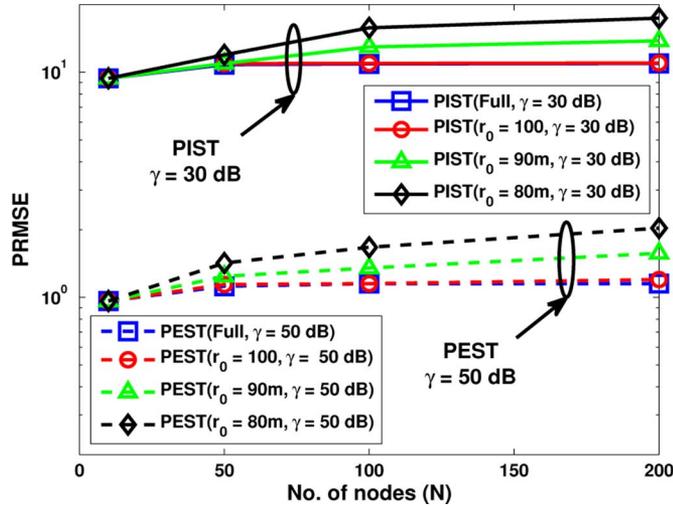


Fig. 4. Partial connectivity and scalability.

imperfect process noise covariance (IPNC) information, e.g., a scaled  $\sigma_w$  is adopted here. The results are interesting since both algorithms degrade significantly and perform worse than the proposed algorithms for all  $\gamma$ . Notably, the UKF is much more robust against an increase of  $T_s$ , and the proposed algorithms are even more *robust* than the UKF, and this makes them cost-efficient algorithms for practical scenarios.

Finally, Fig. 4 investigates two important issues, i.e., scalability and tackling partial connectivity. For the sake of clarity, we plot the performance of the PIST for  $\gamma = 30$  dB and the one of the PEST for  $\gamma = 50$  dB both for  $T_s = 0.1$  s. From the figure, the performance of the algorithms in fully connected networks remains almost the same with increasing the size of the network up to  $N = 200$  (i.e., *scalability*). For partially connected networks, we decrease  $r_0$  from the maximum distance in the network  $r_{\max} = 100\sqrt{2} \approx 141$  m to  $r_0 = 100, 90$ , and 80 m. As can be seen, the performance of the algorithms in partially connected networks (for  $r_0 < 100$  m) gradually deviates from that of the fully connected network. In our simulations, we observe that  $r_0 = 100$  m and  $r_0 = 80$  m approximately correspond to, respectively, 30% and 50% misconnectivity in the network which is considerable. Note that decreasing  $r_0$  further leads to many possible configurations which are not rigid anymore and thus in principle there will be no solution for the reconstruction problem. This might lead to large estimation errors by our algorithm in cases where the signal subspace is badly damaged due to the large number of missing links.

## VI. CONCLUSION

We have proposed two cooperative mobile network tracking algorithms based on a novel dynamic MDS. We have also extended the proposed algorithms to operate in more realistic partially connected networks. The proposed algorithms are model independent. It has been shown that the proposed algorithms are characterized by a low computational complexity, an acceptable accuracy, and robustness against the measurement interval of the network, which makes them a superb choice for practical implementations. As a future work, we will explore a distributed implementation of the proposed algorithms.

## REFERENCES

- [1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, 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] Y. Shang, W. Rumi, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Apr. 2003, pp. 201–212.
- [3] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted MDS for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.
- [4] I. M. F. Caballero, L. Merrino, and A. Ollero, "A particle filtering method for wireless sensor network localization with an aerial robot beacon," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2008, pp. 596–601.
- [5] 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.
- [6] D. Macagnano and G. T. F. de Abreu, "Tracking multiple targets with multidimensional scaling," in *Proc. Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Sep. 2006, pp. 1–5.
- [7] F. K. W. Chan and H. C. So, "Efficient weighted multidimensional scaling for wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4548–4553, Nov. 2009.
- [8] Z. Wang and S. A. Zekavat, "A new TOA-DOA node localization for mobile ad-hoc networks: Achieving high performance and low complexity," in *Proc. Int. Conf. Telecommun. (ICT)*, 2010, pp. 836–842.
- [9] H. Jamali-Rad, T. van Waterschoot, and G. Leus, "Cooperative localization using efficient Kalman filtering for mobile wireless sensor networks," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Aug.–Sep. 2011, pp. 1984–1988.
- [10] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, pp. 427–450, Apr. 2009.
- [11] L. Dong, "Cooperative network localization via node velocity estimation," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2009, pp. 1–6.
- [12] M. A. Caceres, F. Sottile, R. Gallero, and M. A. Spirito, "Hybrid GNSS-ToA localization and tracking via cooperative unscented Kalman filter," in *Proc. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2010, pp. 272–276.
- [13] G. H. Golub and C. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
- [14] H. Jamali-Rad, A. Amar, and G. Leus, "Cooperative mobile network localization via subspace tracking," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2011, pp. 2612–2615.
- [15] H. Jamali-Rad, T. van Waterschoot, and G. Leus, "Anchorless cooperative localization for mobile wireless sensor networks," in *Proc. 32nd WIC/IEEE SP Symp. on Information Theory and Signal Processing in the Benelux (WICSP)*, May 2011, pp. 1–8.
- [16] R. J. Vaccaro, "Weighted subspaces fitting using subspace perturbation expansions," Dept. of Elec. Eng., Katholieke Universiteit Leuven, Tech. Rep. No. ESAT-SISTA/TR 1997-45, Jun. 1997 [Online]. Available: <ftp://ftp.esat.kuleuven.ac.be/SISTA/guests/reports/vaccaro.ps.Z>
- [17] Y. Hua, "Asymptotical orthonormalization of subspace matrices without square root," *IEEE Signal Process. Mag.*, vol. 21, pp. 56–61, Jul. 2004.
- [18] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [19] P. Tichavski, H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," in *IEEE Trans. Signal Process.*, May 1998, vol. 46, pp. 1386–1396.
- [20] P. Drineas, A. Javed, M. Magdon-Ismael, G. Pandurangan, R. Virrankoski, and A. Savvides, "Distance matrix reconstruction from incomplete distance information for sensor network localization," in *Proc. Annu. IEEE Commun. Soc. Conf. Sensor, Mesh, Ad Hoc Commun., Netw. (SECON)*, Sep. 2006, pp. 536–544.
- [21] P. A. Regalia and J. Wang, "On distance reconstruction for sensor network localization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2010, pp. 2866–2869.