

TRACKING MULTIPLE DYNAMIC TARGETS WITH MULTIDIMENSIONAL SCALING

Davide Macagnano and Giuseppe Thadeu Freitas de Abreu
 Centre for Wireless Communications - University of Oulu
 P.O.Box 4500, 90014-Finland
 [macagnan, giuseppe]@ee.oulu.fi

ABSTRACT

We consider the problem of tracking multiple targets in the presence of imperfect and incomplete ranging information, focusing on the impact of target dynamics. The targets are assumed to describe independent, continuous and differentiable trajectories with non-stationary (dynamic) statistics, *i.e.*, with variable velocities and accelerations. The impact of such dynamics onto the performance, computational complexity and memory requirements of two tracking techniques, namely, the Kalman Filter (KF) and Multidimensional Scaling (MDS), is investigated. The main feature of the MDS-based tracking algorithm, which we proposed in an earlier work, is that tracking is performed over the eigenspace of a Nyström-Gram kernel matrix constructed with no *a-priori* knowledge of the statistics of target trajectories. Consequently, tracking becomes a problem of updating the eigenspace given new input data, which is achieved with an iterative Jacobian eigen-decomposition technique. An advantage of this technique over the KF is that tracking accuracy is independent on target dynamics. Furthermore, the number of iterations required to update the eigenspace, is shown to grow only logarithmically with the target dynamics and with the number of simultaneously tracked targets. As a result, the MDS-based tracking algorithm with Jacobian eigenspace updating becomes more efficient than the KF as soon as a relatively small number of targets are simultaneously tracked, and/or target dynamics exceeds a certain threshold.

I INTRODUCTION

Sensor mobility is a topic of growing interest in wireless sensor networks (WSNs). Although many traditional sensor network application scenarios tend to be static, mobility finds itself at the heart of several new system concepts and techniques for WSNs. The combination of mobility as a feature, and localization as a need or application, gives rise to the problem of target tracking in sensor networks, which is subjected to distinct challenges to those met in more traditional contexts such as array antennas, radar or GPS technology.

Specific difficulties vary with the application, but some of the major challenges for tracking algorithms in the WSN context are stringent constraints on the power consumption of sensors activated for tracking purposes [1], the requirement for low-complexity algorithms [2], and the ability to track a potentially large number of sensors simultaneously [3]. Here, a warehouse-like scenario is considered, in which a small number of inter-wired anchor nodes placed at known fixed locations provide the infra-structure used to track several mobile devices simultaneously. Target nodes may be thought of as tags, which cannot perform ranging but transmit periodic signals that are used by anchors to perform anchor-to-target ranging.

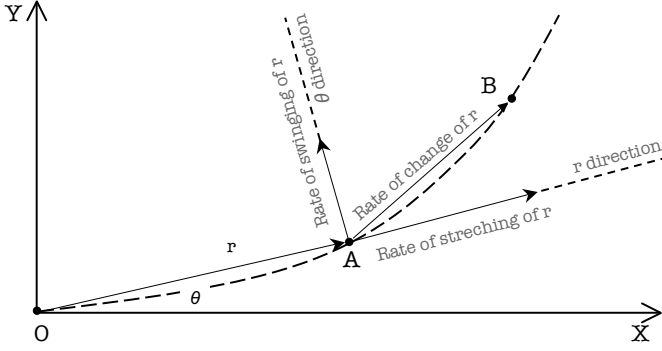
The main challenges in the aforementioned scenario are that targets may have different dynamics, and be in large numbers. Two possible solutions to these problems are considered in this article. The first is the extended Kalman Filter (EKF), a well-understood Bayesian technique known for its low-complexity, performance and stability as a tracking algorithm [4]. The other is a less-known multidimensional-scaling (MDS)-based technique recently introduced [5], in which tracking is performed not on the cartesian system of coordinates, but through a Jacobian adaptation of the eigenspace of a Gram kernel matrix constructed (via Nyström approximation) using only anchor-to-anchor and anchor-to-target ranging information [6]. We focus on the comparison of the performances and complexity requirements of the aforementioned methods in relation to the variation of target dynamics, which was not performed in [5].

The remainder of the article is as follows. In section II, target dynamics is quantified into a tangible metric, and a simple trajectory model utilized throughout the article to simulate the continuous pseudo-random motion of targets with variable dynamics is introduced. In section III the formulation of the EKF and iterated EKF (IEKF) tracking algorithms considered in this work are briefly reviewed, and the strong relationship between their performances and the dynamic metric defined in section II is illustrated. In section IV, the alternative MDS-based tracking technique is revised. That section is concluded with a study of the impact of target dynamics and quantity onto the average number of iterations required by the Jacobian method to update the eigenspace of the last observation given new ranging inputs. Finally, the complexity (as a function of the number of targets) and performance (as a function of target dynamics in the presence of noisy ranging information) of the EKF/IEKF- and MDS-based tracking algorithms are studied. The results indicate that the eigenspace-based MDS tracking algorithm is preferable to the EKF/IEKF method when the number of targets to be simultaneously tracked is larger than just a couple, and when these targets exhibit different dynamics.

II DYNAMICS AND TRAJECTORY

In order to study the effect of variable target dynamics onto the performance of low-complexity (EKF/IEKF- and MDS-based) tracking algorithms, one needs to quantify target dynamics in the form of a tangible metric. One such metrics is the *velocity*, as defined in classical analytical mechanics,

Given the nature of the tracking problem under consideration, in which ranging information is sampled periodically, it is convenient to calculate such a dynamic metric directly from the discrete set of points along the targets' trajectories corresponding to their locations at sampling instants, rather than from the continuous functions that describe those trajectories.


 Figure 1: Quantities used to compute the dynamic metric ν .

Consider a set of points corresponding to three consecutive samples of the location of a target moving along a continuous trajectory, as illustrated in figure 1. The three points describe two consecutive steps of the motion of the target. Making use of a polar coordinate system centered at the first of these points, the velocity associated with the discrete observation of the target's motion is given by [7, pp.18],

$$\vec{v} = \vec{v}_r + \vec{v}_\theta, \quad (1)$$

where

$$\vec{v}_r = \frac{d\vec{r}}{dt}; \quad \vec{v}_\theta = \vec{r} \cdot \frac{d\theta}{dt}. \quad (2)$$

While in conventional analytical mechanics one may be interested in the quantity \vec{v} computed taking into account an infinitesimal time increment t , in our case we are concerned with a dynamic metric that captures the amount of variation of the target location as *per* the observations. This is why even if the motion occurs in 3 dimensions, the physics of the problem can be fully described in 2 dimensions.

Let $\mathbf{x}(t)$ denote the coordinates of a target at instant t , sampled with periodicity T . From equations (1) and (2),

$$\mathbf{v}_1 \triangleq \mathbf{x}(t+T) - \mathbf{x}(t), \quad (3)$$

$$\mathbf{v}_2 \triangleq \mathbf{x}(t+2T) - \mathbf{x}(t+T), \quad (4)$$

$$\nu = \sqrt{|\mathbf{v}_2 - \mathbf{v}_1|^2 - \text{acos}\left(\frac{\langle \mathbf{v}_1; \mathbf{v}_2 \rangle}{|\mathbf{v}_1| \cdot |\mathbf{v}_2|}\right) \cdot |\mathbf{v}_1|^2}, \quad (5)$$

where $\langle \mathbf{x}; \mathbf{y} \rangle$ denotes inner product.

The trajectory model adopted in this paper is briefly described by the equations below [5]. An example of a typical trajectory obtained with this model is given in figure 2.

$$t(s) = \left(\frac{2\sigma_t}{\sqrt{N_R}} \sum_{n=1}^{N_R} \alpha_{n,1} \cos(2\pi f_{n,1}s + u_{n,1}) \right)^2, \quad (6)$$

$$g(s) = \frac{2\sigma_g}{\sqrt{N_R}} \sum_{n=1}^{N_R} \alpha_{n,2} \cos(2\pi f_{n,2}t(s) + u_{n,2}), \quad (7)$$

$$f_{n,i} = \frac{1}{T_C} \cos\left(\frac{2\pi(n-1) + u_{n,2+i}}{N_R}\right), \quad (8)$$

$$\alpha_{n,i} = \begin{cases} \sqrt{2} \sin(u_{1,4+i}) & \text{for } n = 1, \\ 2 \sin(u_{n,4+i}) & \text{for } 1 < n \leq N_R, \end{cases} \quad (9)$$

where T_C is the coherence time of the process and u are random variables uniformly distributed between $[0, 1]$.

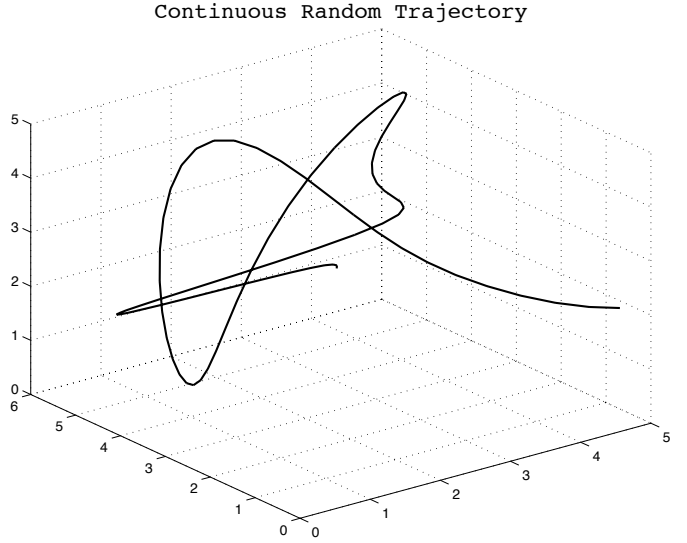


Figure 2: A typical continuous pseudo-random trajectory obtained with the trajectory model described by equations (6) through (8).

III THE KALMAN FILTER

The state prediction equations of the EKF/IEKF are,

$$\vec{\mathbf{Z}}_{k|k-1} = \mathbf{P} \cdot \vec{\mathbf{Z}}_{k-1}, \quad (10)$$

$$\hat{\mathbf{Q}}_{k|k-1} = \mathbf{P} \cdot \hat{\mathbf{Q}}_{k-1} \cdot \mathbf{P}^T + \mathbf{W}, \quad (11)$$

where \mathbf{W} is a constant uncertainty matrix, $\vec{\mathbf{Z}}$ is a vector containing the coordinates of all targets and parameters related to the rate of update, and \mathbf{P} is the constant transition matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_{[n \times n]} & \mu \cdot \mathbf{I}_{[n \times n]} \\ \mathbf{0} & \mathbf{I}_{[n \times n]} \end{bmatrix} \otimes \mathbf{I}_{[M \times M]}, \quad (12)$$

where $\mathbf{I}_{[n \times n]}$ is the identity matrix and μ a small constant.

In the equations above and hereafter, n and M denote the dimension of the Euclidean space and the total number of targets, respectively. In order to simplify the notation, subscripts are used (with different font types and brackets) to label variables, refer to the time index, and specify the dimension of matrices.

The Kalman correction equations are,

$$\vec{\mathbf{Z}}_k = \vec{\mathbf{Z}}_{k|k-1} + \mathbf{B}_k \cdot \epsilon_k, \quad (13)$$

$$\hat{\mathbf{Q}}_k = (\mathbf{I}_{[2nM \times 2nM]} - \mathbf{B}_k \cdot \mathbf{J}_k) \cdot \hat{\mathbf{Q}}_{k|k-1}, \quad (14)$$

where the Kalman gain \mathbf{B}_k , the innovation vector ϵ_k and the Jacobian \mathbf{J}_k , at the k -th iteration, are given by

$$\mathbf{B}_k = \hat{\mathbf{Q}}_{k|k-1} \cdot \mathbf{J}_k^T \cdot (\mathbf{J}_k \cdot \hat{\mathbf{Q}}_{k|k-1} \cdot \mathbf{J}_k^T + \hat{\mathbf{S}}_k)^{-1}, \quad (15)$$

$$\epsilon_k = \vec{\mathbf{D}}_{AM} - \mathcal{D}(\hat{\mathbf{X}}_{k|k-1}; \mathbf{X}_A), \quad (16)$$

$$\mathbf{J}_k = \nabla \mathcal{D}(\mathbf{X}; \mathbf{X}_A) \big|_{\hat{\mathbf{X}}_{k|k-1}}, \quad (17)$$

where ∇ is the differential operator, $\mathcal{D}(\mathbf{X}; \mathbf{X}_A)$ is a function that returns the Euclidean distances between the coordinates of targets \mathbf{X} and anchors \mathbf{X}_A , and $\hat{\mathbf{S}}_k$ is the covariance matrix,

$$\hat{\mathbf{S}}_k = \frac{1}{k} \left[(k-1) \cdot \hat{\mathbf{S}}_{k-1} - \epsilon_k \cdot \epsilon_k^T + \mathbf{E} \right], \quad (18)$$

where \mathbf{E} is a constant uncertainty.

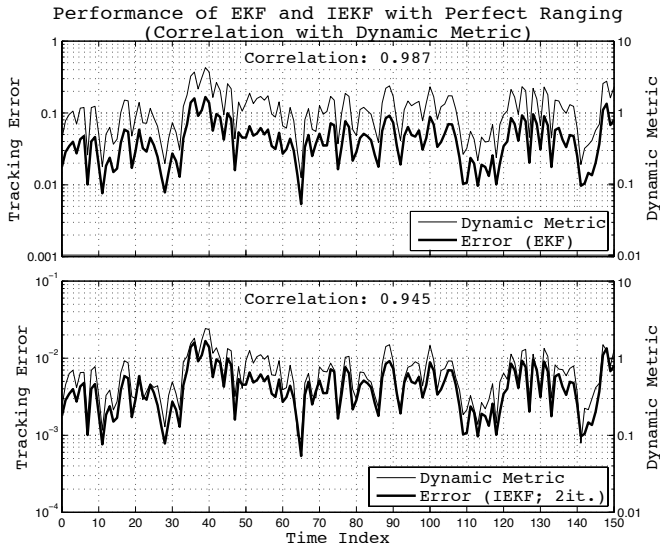


Figure 3: Variation of target dynamics metric ν and performances of EKF and IEKF tracking algorithms.

Figure 3 shows a snapshot of the variation of a target's dynamic metric ν , and the corresponding errors obtained with the EKF and IEKF algorithms. All plots were obtained under perfect (noiseless, unbiased) ranging conditions. The figure illustrates that ν is highly correlated with the performance of the EKF-based tracking algorithms, although iterating the EKF (IEKF) reduces the error due to variable target dynamics and thus the correlation with ν .

IV MDS-BASED TRACKING ALGORITHM

Unlike the KF, the MDS-based tracking algorithm first introduced in [5], does not rely on any model nor on more than the last estimate and current observations in order to update the estimates of the location of tracked target. Consequently, the performance the MDS-based tracking technique is immune to the non-stationarity of the target's dynamics, although the complexity of the algorithm is not.

This is because the complexity of this tracking algorithm depends on the number of iterations taken by the Jacobian technique to update the eigenspace corresponding to the previous target locations given new observations.

For the sake of clarity the structure of the MDS-based tracking algorithm proposed in [5] is here briefly reviewed. The algorithm is formulated over an incomplete Euclidean distance matrix (EDM) $\tilde{\mathbf{D}}$ which collects all the anchor-anchor \mathbf{D}_{A^2} and anchor-target distances $\tilde{\mathbf{D}}_{AM}(t)$ at each sampling instance.

$$\tilde{\mathbf{D}} = [d_{i,j}(t)] = \begin{bmatrix} \mathbf{D}_{A^2} & \tilde{\mathbf{D}}_{AM}(t) \\ \tilde{\mathbf{D}}_{AM}^T(t) & \mathbf{0} \end{bmatrix} \quad (19)$$

At any time t , the MDS techniques allows one to recover the location $\mathbf{Y}(t)$ of all sensors in the network (up to rotation, scaling, translation and reflection) from the complete Gram matrix $\mathbf{G}(t)$. Mathematically we have

$$\mathbf{Y}(t) = [\mathbf{V}(t)]_{1:n} \cdot [\mathbf{\Lambda}(t)]_{1:n}^{\frac{1}{2}}, \quad (20)$$

$$\mathbf{G}(t) = \mathbf{V}(t) \cdot \mathbf{\Lambda}(t) \cdot \mathbf{V}(t)^T, \quad (21)$$

where a complete Gram-matrix $\mathbf{G}(t)$ can be obtained from the incomplete $\tilde{\mathbf{D}}$ using the Nyström approximation [6],

$$\mathbf{G}(t) \approx \begin{bmatrix} \mathbf{G}_A & \mathbf{G}_M(t) \\ \mathbf{G}_M(t)^T & \mathbf{G}_M(t)^T \cdot \mathbf{G}_A^{-1} \cdot \mathbf{G}_M(t) \end{bmatrix}, \quad (22)$$

with [8],

$$\mathbf{G}_M(t) = -\frac{1}{2} \cdot (\tilde{\mathbf{D}}_{AM}(t) + \mathbf{C}_1 - \mathbf{C}_2 - \mathbf{C}_3(t)), \quad (23)$$

$$\mathbf{C}_1 = \frac{1}{A^2} \cdot [\mathbf{e}_{[1 \times A]} \cdot \mathbf{D}_{A^2} \cdot \mathbf{e}_{[A \times 1]}] \cdot \mathbf{e}_{[A \times M]}, \quad (24)$$

$$\mathbf{C}_2 = \frac{1}{A} \cdot [\mathbf{D}_{A^2} \cdot \mathbf{e}_{[A \times 1]}] \otimes \mathbf{e}_{[1 \times M]}, \quad (25)$$

$$\mathbf{C}_3(t) = \frac{1}{A} \cdot [\mathbf{e}_{[1 \times A]} \cdot \tilde{\mathbf{D}}_{AM}(t)] \otimes \mathbf{e}_{[A \times 1]}. \quad (26)$$

Provided that the coordinates $\mathbf{X}_A(t)$ of at least $A > n$ anchor nodes are known with respect to an absolute system of coordinates, the solution $\mathbf{Y}(t)$ can be re-oriented via the Procrustes transformation [9], returning $\hat{\mathbf{X}}(t)$.

As indicated by equation (20), tracking with the MDS algorithm requires the repetitive computation of the eigen-decomposition of $\mathbf{G}(t)$. This problem can be circumvented by utilizing an iterative eigenspace adaptation technique based on the Jacobian algorithm [10], summarized by

$$\mathbf{A}_{k+1} \leftarrow \mathbf{R}(i_k, j_k, \theta_k) \cdot \mathbf{A}_k \cdot \mathbf{R}(i_k, j_k, \theta_k)^T, \quad (27)$$

where $\mathbf{R}(i_k, j_k, \theta_k)$ are orthogonal similarity transformations corresponding to the Givens rotation matrices, whose optimum rotation angle for the plane associated with the quartet $(a_{i,i}, a_{i,j}, a_{j,i}, a_{j,j})$ is given by,

$$\theta_{\text{Opt}}(i, j) = \frac{1}{2} \cdot \text{atan} \left(\frac{a_{i,j} + a_{j,i}}{a_{i,i} - a_{j,j}} \right). \quad (28)$$

In the Jacobian algorithm, the approximate eigenvector matrix of \mathbf{A} obtained after K_E iterations is given by,

$$\mathbf{V} = \prod_{k=1}^{K_E} \prod_{(i_k, j_k)} \mathbf{R}(i_k, j_k, \theta_k). \quad (29)$$

A Jacobian-like eigen-decomposition algorithm capable of finding a single matrix \mathbf{R} that jointly (approximately) diagonalizes a set of matrices $\mathcal{A} \triangleq \{\mathbf{A}_1, \dots, \mathbf{A}_M\}$ also exists [11].

In this Jacobian-like joint-diagonalization technique, one iterates the expression

$$\mathbf{A}_{k+1} = \mathbf{R}(i_k, j_k, \theta_k) \cdot \mathbf{A}_k \cdot \mathbf{R}(i_k, j_k, \theta_k)^T \quad (30)$$

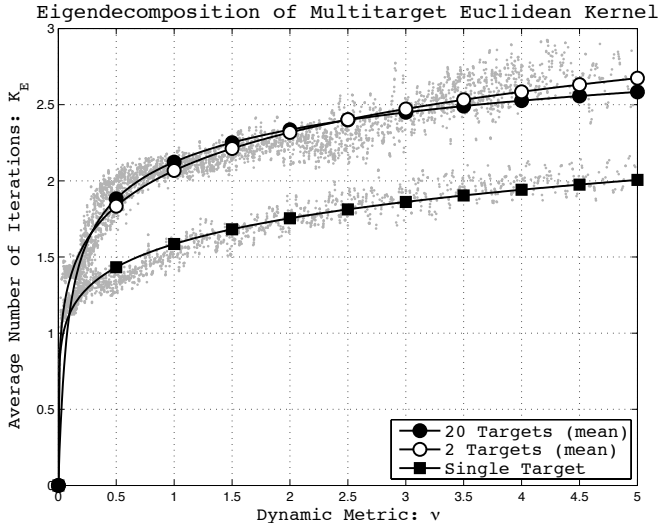
where $\mathbf{R}(i_k, j_k, \theta_k)$ is the solution of

$$\min_{\theta(i,j)} \sum_{m=1}^M \text{off}(\mathbf{R}(i, j, \theta) \cdot \mathbf{A}_m \cdot \mathbf{R}(i, j, \theta)^T), \quad (31)$$

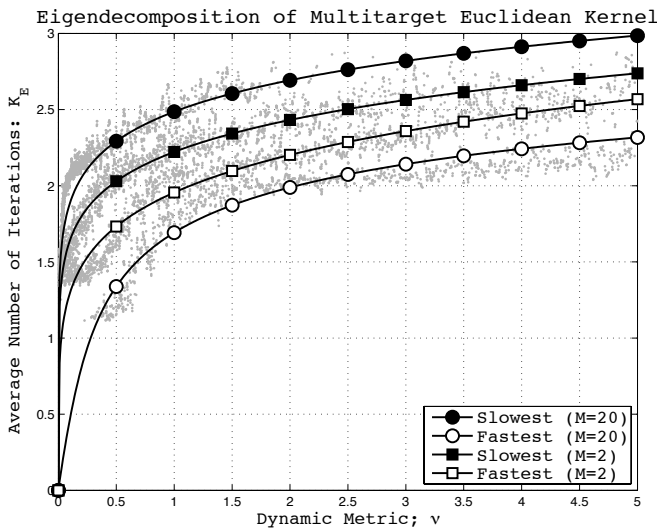
with

$$\text{off}(\mathbf{A}) \triangleq \sum_{i \neq j} \|a_{i,j}\|^2. \quad (32)$$

Convergence and stability of this algorithm are rigorously proved in [11]. In the form presented thereby, however, the algorithm would be too computationally demanding for our needs, due to the optimization step described by equation (31). Fortunately, a closed-form expression for the optimum angles that solve this minimization problem was later discovered [12]. The solution thereby is extremely simple and are applicable for any set of equi-dimensional matrices \mathcal{A} , regardless of symmetry and commutativity properties.



(a) Dependence on average dynamics.



(b) Dependence on min and max dynamics.

Figure 4: Speed of eigenspace tracking \times target dynamics.

As noted in [5], the joint-diagonalization algorithm with closed-form optimum rotation angles can be used as an eigenspectrum tracking algorithm as follows.

Consider the two Gram matrices $\mathbf{G}(t)$ and $\mathbf{G}(t+T)$ corresponding to consecutive observations $\hat{\mathbf{D}}_{AM}(t)$ and $\hat{\mathbf{D}}_{AM}(t+T)$, and assume that the eigenvector matrix $\mathbf{V}(t)$ of $\mathbf{G}(t)$ is known.

Then, $\mathbf{V}(t+T)$ can be easily computed by initializing the Jacobian algorithm with $\mathbf{V}(t)$. This leads to the following solution for the eigen-decomposition of $\mathbf{G}(t+T)$,

$$\mathbf{V}(t+T) = \mathbf{V}(t) \cdot \prod_{k=1}^{K_E} \prod_{(i_k, j_k)} \mathbf{R}(i_k, j_k, \theta_k). \quad (33)$$

One may expect that in this approach, the number of iterations K_E required to update $\mathbf{V}(t)$ into $\mathbf{V}(t+T)$ will be related to the dynamics of the targets. For example, if the location of all targets at time $t+T$ differs only slightly from that at time t , it is fair to expect that K_E will be rather small.

Figure 4 shows the average number of iterations \bar{K}_E as a function of ν , for systems with a single, 2 and 20 targets.

Those and all subsequent figures built from simulations, display scattered data in gray, smoothed via weighted least-squares regression with a polynomial kernel (lowess) [13], superposed by mathematical models fitted to the data, shown in solid lines.

Since in the multi-target scenarios the dynamic metric of each target is generally different at each observation instant, figure 4(a) shows plots of \bar{K}_E against the average of the dynamic metrics of all the targets. It is noticeable that \bar{K}_E grows only in a log-like relationship with $\bar{\nu}$. Surprisingly, it is also found that the behavior of \bar{K}_E as a function of $\bar{\nu}$ does not depend on the number of targets if $M \geq 2$.

This can be better understood from figure 4(b), which shows the variation on \bar{K}_E against the minimum and maximum dynamics amongst the multiple targets. When \bar{K}_E is plotted against the minimum of an M -tuple of dynamic metrics $\nu = \{\nu_1, \dots, \nu_M\}$, it is natural that the curve appears shifted upwards, since the higher dynamics of all the other targets contribute to a larger number of Jacobian iterations.

Likewise, the curve describing the relationship between \bar{K}_E and the maximum ν in the M -tuple ν appears shifted downwards. What is interesting about these results, however, is that the amount of these shifts are symmetric, with respect to the curve obtained with $\bar{\nu}$, regardless of the number of targets. The conclusion that can be drawn is that \bar{K}_E is not strongly dependent on ν which, in turn, implies that the complexity of the MDS-based algorithm is mainly determined by the number of targets, and only slightly affected by the target dynamics. We leave further comparisons of the EKF and MDS-based tracking algorithms to the next subsection.

V COMPLEXITY AND PERFORMANCE

A detailed account of the complexity and memory requirements of the MDS and EKF tracking algorithms was performed in [5]. For convenience, here only the complexity and memory orders of these algorithms are shown in Table 1.

In figure 5(a), the variation of the complexity of the MDS tracking algorithm against the target dynamics are shown, for a single- and two-target scenarios, with perfect (noiseless and unbiased) ranging. The complexity figure is the average number of floating point operations (flops) incurred in the execution of the algorithm, computed using the relations shown in Table 1, and where \bar{K}_E was calculated with models corresponding to the solid lines shown in figure 4(a).

All curves are obtained using perfect ranging information so as to isolate the effect of the target dynamics on the tracking algorithms from the influence of noise. It is visible that, indeed, the complexity of the MDS method is mostly independent on the target dynamics (except for small ν), which corroborates the conclusions drawn in the last subsection.

Next, we compare the complexity of the MDS-based tracking scheme against that of the EKF technique, as a function of the number of targets. The results are shown in figure 5(b). It can be seen that the MDS-based method is preferable to the EKF algorithm if more than a couple of targets are to be tracked simultaneously.

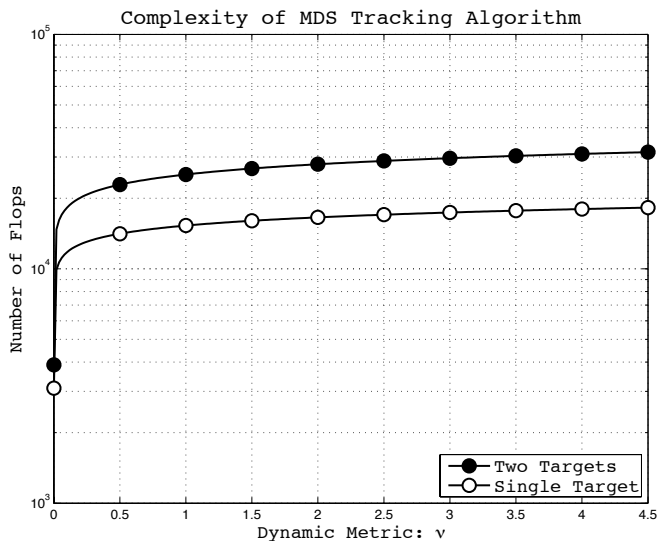
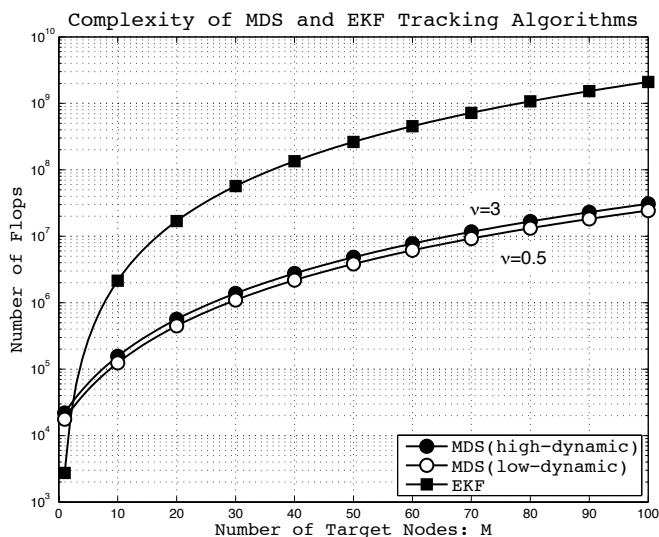

 (a) Effect of ν on the complexity of MDS-tracking.

 (b) Effect of M on the complexity of EKF- and MDS-tracking.

Figure 5: Complexity of tracking algorithms as a function of the average target dynamics with perfect ranging information.

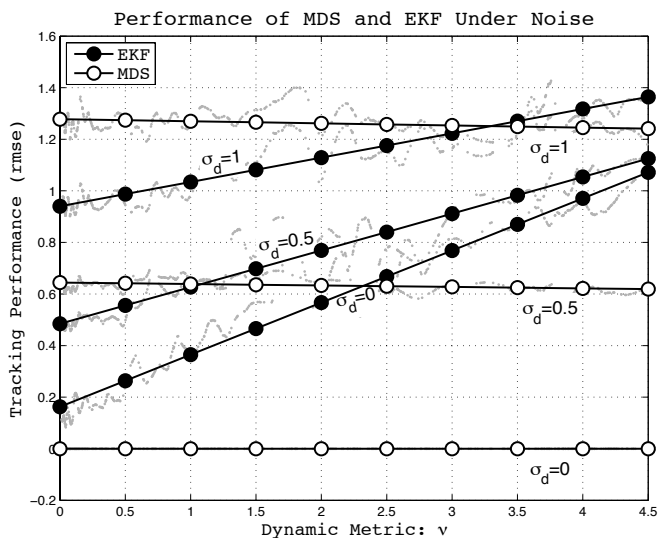

 Figure 6: Performance of the MDS and EKF tracking algorithms as a function of $\bar{\nu}$ under perfect and noisy ranging.

Table 1: Requirements of Tracking Algorithms

	EKF	MDS
Complexity:	$\mathcal{O}(K_I A^3 M^3)$	$\mathcal{O}(K_E (A + M)^3)$
Memory:	$\mathcal{O}(A^2 M^2)$	$\mathcal{O}((A + M)^2)$

Finally, we compare the error performances of the two tracking techniques under the effect of zero-mean, independent and identically distributed (i.i.d) white Gaussian noise added to the true target coordinates before computing their Euclidean distances to each of the anchors. Curves for both the noiseless and noisy cases with two different noise variances are shown.

The figure illustrates the fact that the KF performs poorly if tracked targets are sufficiently dynamic, *i.e.*, move with trajectories characterized by pseudo-random processes of non-stationary statistics. This is because in such conditions, the KF's ability to filter the noise affecting the observations is undermined by its inability to quickly adapt to new dynamic conditions. Of course, one could improve the performance of KF-based techniques by including a block that estimates target dynamics. This would, however, increase the overall complexity of the method, extending further the gains (in complexity reduction) provided by the alternative MDS-based technique.

Together, figures 5(b) and 6 summarize the main conclusion of our study, namely, that a subspace-based tracking algorithm is advantageous, from both complexity and performance points of view, compared to KF-based methods. Furthermore, since the MDS tracking technique here described incorporates no noise-filtering capabilities, even better results are expected if the approach is coupled with an *a posteriori* smoothing block.

REFERENCES

- [1] T. Vercauteren, D. Guo, and X. Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 4, pp. 712 – 723, Apr. 2005.
- [2] M. Rydström, A. Urruela, E. Ström, and A. Svensson, "Low complexity tracking for ad-hoc automotive sensor networks," in *Proc. IEEE SECON'04*, Oct. 4 - 7, 2004, pp. 585 – 591.
- [3] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. IEEE 44-th CDC&ECC'05*, Dec. 12-15 2005, pp. 6698 – 6703.
- [4] F. Daum, "Nonlinear filters: beyond the kalman filter," *IEEE Aerospace and Electronic Systems Mag.*, vol. 20, no. 8, pp. 57 – 69, Aug. 2005.
- [5] D. Macagnano and G. Abreu, "Tracking multiple targets with multidimensional scaling," in *WPMC'06*, Sep. 17 - 20, 2006.
- [6] J. C. Platt, "FastMap, MetricMap, and Landmark MDS are all Nyström algorithms," Microsoft Research Technical Report, Tech. Rep. MSR-TR-2004-26, 2004.
- [7] R. L. Halfman, *Dynamics: Particles, Rigid Bodies and Systems*. Reading, MA: Addison-Wesley, 1962, vol. 1.
- [8] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*.
- [9] P. D. Fiore, "Efficient linear solution of exterior orientation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 2, pp. 140 – 148, February 2001.
- [10] G. Golub and C. van Loan, *Matrix Computations*, 3rd ed. New York, NY: Johns Hopkins Univ. Press, Nov. 1996.
- [11] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, "Numerical methods for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 4, pp. 927 – 949, 1993.
- [12] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161 – 164, 1996.
- [13] J. Fox, *Nonparametric Simple Regression: Smoothing Scatterplots*, ser. Quantitative Applications in the Social Sciences. Sage University Paper, 2000.