
Sparse multidimensional scaling for blind tracking in sensor networks

R. Rangarajan¹, R. Raich², and A. O. Hero III³

¹ University of Michigan rangaraj@eecs.umich.edu

² University of Michigan ravivr@eecs.umich.edu

³ University of Michigan hero@eecs.umich.edu

1 Introduction

In this chapter, we consider the problem of tracking a moving target using sensor network measurements. We assume no prior knowledge of the sensor locations and so we refer to this tracking as ‘blind’. We use the distributed weighted multidimensional scaling (dwMDS) algorithm to obtain estimates of the sensor positions. Since dwMDS can only find sensor position estimates up to rotation and translation, there is a need for alignment of sensor positions from one time frame to another. We introduce a sparsity constraint to dwMDS to align current time sensor positions estimates with those of the previous time frame. In the presence of a target, location estimates of sensors in the vicinity of the target will vary from their initial values. We use this phenomenon to perform link level tracking relative to the initially estimated sensor locations.

Wireless sensor networks have been deployed for a number of monitoring and control applications such as target tracking [28], environmental monitoring [29], manufacturing logistics [26], geographic routing, and precision agriculture [44]. For many target tracking applications such as anomaly detection [21, 45], species distribution and taxonomy [19], and surveillance [4], the main purpose of the sensor network is to locate and track changes in remote environments. For example, species distribution and classification are currently documented using sightings, captures, and trap locations, which involve considerable manpower, time, and effort. Deploying mobile sensors with cameras can improve remote counts of the species as they move around in the environment. For surveillance applications, the sensors must be able to locate where the intruders or the vehicles are moving in the network. Another example is the problem of locating equipment in a warehouse. The sensors that tag the equipment must register their physical locations and activate an alarm if they are about to exit the building. As another example, in secure protocol and network routing it is critical to track anomalies such as worm activity, flash crowds, outages, and denial of service attacks in the network.

Automatic self-configuration and self-monitoring of sensor networks is the key enabling technology for these tracking applications. To respond to changes in the sensor network, it is critical to know where the changes are occurring. Data measurements from the sensors must be registered to their physical locations in the network in order to make optimal decisions. For dense sensor networks, the large size makes it impractical for humans to manually enter the physical location of the sensors and it is too expensive to attach the GPS to every device in the network. The sensors must have the capabilities to automatically estimate their relative positions and detect changes in the network at low cost, e.g., with minimum battery power.

Self-localization algorithms can be broadly classified into two categories, centralized strategies and decentralized strategies. In a centralized approach, all the data collected by the sensors must be communicated to the fusion center which then makes a decision based on this information. Algorithms that use multidimensional scaling (MDS) [40], maximum likelihood estimation [30], and convex optimization [14] have been proposed for centralized estimation and have shown to perform well. However, this may be impractical when the sensors operate with limited power and bandwidth. For networks with thousands of sensors, transmission of sensor data to a fusion center overwhelm the low-bandwidth capacity of sensor networks. Furthermore, remote sensors are frequently battery operated and battery replacement may be infeasible or expensive.

The need to conserve power and bandwidth has set the stage for more efficient decentralized strategies for localization. Among the popular approaches are adaptive trilateration [32, 39] and successive refinement [9, 23] algorithms. In trilateration, each sensor gathers information about its location with respect to anchor nodes, also referred to as seeds [31], through a shortest path. Using the range estimates from the seeds, a sensor uses trilateration to estimate its location in the network. In successive refinement algorithms, each sensor localizes its position in its own coordinate system based on the information communicated from only its neighbors. Sensors refine their location estimates iteratively using updates from neighboring sensors and finally merge their local coordinates systems, effectively finding the solution to the localization problem. Recently, there has been research emphasis on localization based on a moving target, called a mobile in [6, 34, 42]. The mobile moves randomly in the network while transmitting signals thereby allowing the sensors to calibrate their range to the mobile. This provides a large number of measurements with greater diversity which helps overcome environmental obstacles and enables improved estimation of the sensor node locations.

Most localization algorithms assume the presence of anchor nodes, i.e., certain sensors which have knowledge of their positions in the network. In the absence of anchor nodes, the sensor location estimates are only accurate up to a rotation and translation. The intuition behind this result is the following: consider an ad-hoc network of N sensors. The objective is to find the N sensor locations given the $N(N - 1)/2$ inter-sensor distance measurements.

The distance information depends only on the differences in the sensor locations so that the positions of the N sensors in the network can be rotated and translated without changing these distances. In this chapter, we present a sparsity constrained dwMDS algorithm, which can localize the relative positions of the sensor nodes even in the absence of anchor nodes. The dwMDS algorithm proposed in [9] is a successive refinement method, where a global cost function is divided into multiple local cost functions at each sensor location and the computational load involved in finding the sensor location estimates is divided among the sensors in a distributed fashion. The allocation of non-negative continuous weights to the measured data overcomes the problem of combining local maps to one global map, a problem that is common to other decentralized methods [23]. We call our new algorithm sparsity penalized dwMDS. More importantly, we explain how the anchorless sparse dwMDS algorithm can efficiently track changes in the network.

Sensor localization is frequently viewed as an essential prelude to the monitoring and tracking of active phenomena. Target tracking and detection has been one such motivating application of sensor networks [24, 43, 1]. Most target tracking applications assume known sensor locations or estimate the location of sensor nodes separately before employing the tracking algorithm. The standard model used for describing the state dynamics of a moving target is the linear Gaussian model [37]. When the measurement model is also Gaussian, the optimal tracker is given by the Kalman filter. For nonlinear state space and measurement models, other techniques such as Extended Kalman Filter (EKF) [24], unscented Kalman Filter (UKF) [43], and Gaussian sum approximation [1] have been proposed. Particle filtering algorithms were then formulated for target tracking, where the probability density of the state is approximated by a point mass function on a set of discrete points [13]. The discrete points are chosen through importance sampling. The advantage of particle filtering is its applicability to a large range of densities, noise processes, and measurement models. More recently, researchers have looked at the simpler problem of tracking in a binary sensing modality [2, 25]. The sensor outputs a high, when the target is within a sensing range and outputs a low, when the target falls outside its range. Based on the fusion of the sensor outputs, an approximate link level trajectory can be realized to track the target. Such a binary sensing modality has limited accuracy but requires minimal power consumption and has the advantage of analytical tractability [41]. This procedure can also be interpreted as a target detection problem implemented for multiple time steps.

Distributed target detection methods have been proposed in the literature [33] in the context of designing an optimal decision statistics at the sensor fusion center. The detection problem has also been addressed for under communication constraints, where the sensor transmitting the information needs to send an optimal summary of the gathered information to the fusion center [7]. In the context of anomaly detection in internet data, approximate density of incoming traffic is constructed for each location. Distance between densities

is then used as a similarity measure in the MDS algorithm to form a map of the internet network. By performing MDS over time, it is shown that anomalies such as network scans, worm attacks, and denial of service attacks can be identified and classified [36, 16]. For wormhole detection in ad hoc sensor networks, most research efforts require mobile nodes equipped with special hardware or GPS devices [22, 5] to localize the wormhole.

In contrast to the methods proposed in the literature, we present the sparsity penalized dwMDS algorithm which localizes the sensor nodes in the absence of anchors and tracks multiple targets amongst the sensor links. The principle behind our proposed algorithm is the following: in the ‘acquisition phase’ or initialization, an initial estimate of sensor locations is acquired. Once the sensors have been initially localized, it is only the network topology that is critical to the problem of tracking. Hence, during the tracking phase, we introduce a sparsity constraint to the dwMDS problem formulation, which attempts to fix the alignment of the sensor network with respect to the alignment of the localized network at the previous time instance. By doing so, we keep monitoring the network with respect to a fixed geometry obtained by the localization algorithm at the first time instance ($t = 1$). The sparsity constraint only reassigns a small fraction of the sensor locations, while maintaining the locations of remaining sensors close to their previous estimates. When the sensor network is then used for tracking, only the sensors affected by the presence of a target are perturbed, while the rest of the location estimates remain unchanged. Based on the differences in the sensor location estimates between two time-frames, we propose a novel perturbation based link level tracking algorithm, which accurately localizes a target to within a small set of sensor links. Figure 1 shows the localization process in the absence of targets. The actual sensor locations are marked as circles and the anchor nodes are highlighted using squares. The sensors communicate among themselves and the anchor nodes to obtain location estimates indicated as crossed circles. Figure 2 shows the localization process in the presence of a target. The measurements of the sensor nodes closest to the target are affected and the sensors appear further apart than they are in reality. This change in the sensor location estimates can be used to perform link level tracking.

Link level tracking has many attractive features, the most important of which is that it does not require a physical model for the target, which is fundamental to most tracking algorithms in the literature [3]. Moreover, the goal of certain sensor networks is to obtain an estimate of the location of the targets, or detect changes in the network. For example, in military applications, the sensors can locate a target relative to the network and the network can activate the appropriate sensors to identify the target. For animal tracking in biological research, it is sufficient to have a low resolution tracking algorithm to monitor animal behavior and interactions with their own clan and with other species.

We introduce the sparsity constrained dwMDS algorithm for simultaneous sensor localization and link level tracking in this chapter. We give a flavor of

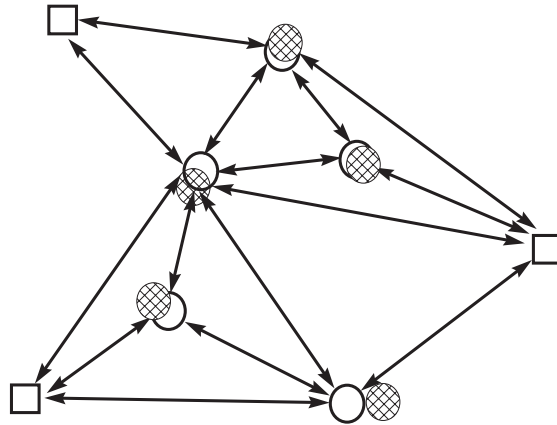


Fig. 1. Localization in the absence of target. Anchor nodes (square), true sensor locations (circle), estimated sensor coordinates (crossed circle).

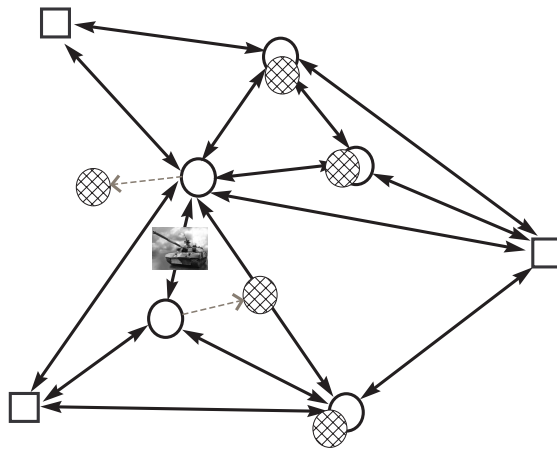


Fig. 2. Link level tracking based on localization in the presence of target.

how the algorithm can be extended to estimate actual target coordinates using standard tracking algorithms. Furthermore, the algorithm we present here can be used to design optimal sensor scheduling strategies for tracking to limit power consumption in sensor networks. We incorporate the sparsity constraint such that the localization algorithm is still distributed in its implementation to minimize communication and computational costs.

This chapter is organized as follows: Section 2 formally introduces the problem of sensor localization. Section 3 introduces the classical MDS al-

gorithm and its variations. We then present our sparsity penalized dwMDS algorithm in Section 4. In Section 5, we explain how this algorithm can be applied for link level tracking. Finally, Section 6 concludes this chapter by discussing the extensions of this formulation for model-based multiple target tracking and sensor management strategies.

2 Problem formulation

We begin by introducing the nomenclature used in this chapter. We denote vectors in \mathbb{R}^M by boldface lowercase letters and matrices in $\mathbb{R}^{M \times N}$ by boldface uppercase letters. The identity matrix is denoted by \mathbf{I} . We use $(\cdot)^T$ to denote the transpose operator. We denote the l_2 -norm of a vector by $\|\cdot\|$, i.e., $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$. A Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix \mathbf{C} is denoted as $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$.

The purpose of the sparsity constrained MDS algorithm is to simultaneously localize and track targets. We first formally state the sensor localization problem. Consider a network of $N = n + m$ nodes in d dimensional space. The localization algorithms can be applied to arbitrary d ($d < N$) dimensional spaces. Since applications for localization typically occur in physical space, we will restrict our attention to $d = 2, 3$ dimensions. Let $\{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d$ be the true location of the n sensors. The m sensor nodes $\{\mathbf{x}_i\}_{i=n+1}^{n+m}$ are anchor nodes, i.e., whose locations are known. We introduce the anchor nodes to keep the formulation as general as possible. Later, we set $m = 0$ for anchor free localization. Denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ as the $d \times N$ matrix of actual sensor locations. Let $\mathbf{D} = (d_{i,j})_{i,j=1}^N$ be the matrix of the true inter-sensor distances, where $d_{i,j}$ denotes the distance between sensor i and sensor j . It is common that some wireless sensor networks may have imperfect a priori knowledge about the locations of certain sensor nodes. This information is encoded through parameters r_i and $\bar{\mathbf{x}}_i$, where $\bar{\mathbf{x}}_i$ is the sensor location and r_i is the corresponding confidence weight. If $\bar{\mathbf{x}}_i$ is unavailable, then we set $r_i = 0$. The problem setting is explained through an illustration of a sensor network in Fig. 3. We assign weights $w_{i,j}$ for measurements between sensors i and j to indicate the accuracy of the distance estimate. In this sensor network, each sensor communicates to its three nearest neighbors and hence, the weights corresponding to links between non neighboring sensors are zero.

Sensor localization is the process of estimating the location of the n sensor nodes $\{\mathbf{x}_i\}_{i=1}^n$ given $\{\mathbf{x}_i\}_{i=n+1}^{n+m}$, $\{r_i\}$, $\{\bar{\mathbf{x}}_i\}$ and pairwise range measurements $\{\delta_{i,j}^t\}$ taken over time $t = 1, 2, \dots, K$. The indices (i, j) run over a subset of $\{1, 2, \dots, N\} \times \{1, 2, \dots, N\}$. The range measurements can be obtained by sensing modalities such as time-of-arrival (TOA), received signal strength (RSS), or proximity.

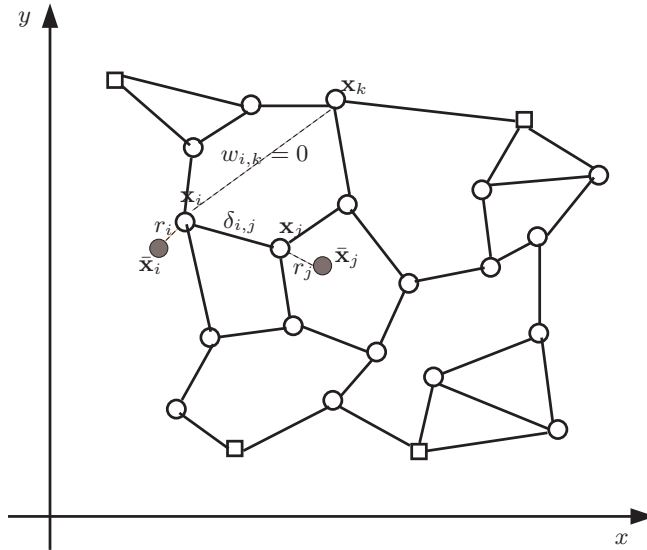


Fig. 3. Sensor localization setup: Anchor nodes (square), sensor nodes (circle), a priori sensor locations (blocked circle). The communicating sensors are connected using solid lines. The non neighboring sensor links have zero weight.

3 Classical MDS and variations

Multidimensional scaling (MDS) is a methodology for recovering underlying low dimensional structure in high dimensional data. The measured data can come from confusion matrices, group data, or any other (dis)similarity measures. MDS has found numerous applications in cognitive science, marketing, ecology, information science, and manifold learning [11, 12]. In the context of sensor localization, the goal in MDS is to discover the sensor locations (lower dimensional embedding) from the inter-sensor distances obtained by a given sensing method (high dimensional data).

Classical MDS [18] provides a closed-form solution to the sensor locations when the inter-sensor measurements are the inter-sensor Euclidean distances, i.e., in the absence of noise or nonlinear effects. We assume all pairwise range measurements are available, and so we can compute the complete matrix of distances:

$$d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}. \quad (1)$$

Denote by $\mathbf{D}^{(2)}$ the matrix of squared distances, i.e., $\mathbf{D}^{(2)} = (d_{i,j}^2)_{i,j=1}^N$. Then $\mathbf{D}^{(2)}$ can be rewritten as

$$\mathbf{D}^{(2)} = \boldsymbol{\psi} \mathbf{1}^T - 2\mathbf{X}^T \mathbf{X} + \mathbf{1} \boldsymbol{\psi}^T, \quad (2)$$

where $\mathbf{1}$ is an N -element vector of ones and $\boldsymbol{\psi} = [\mathbf{x}_1^T \mathbf{x}_1, \mathbf{x}_2^T \mathbf{x}_2, \dots, \mathbf{x}_N^T \mathbf{x}_N]^T$. Let $\mathbf{H} = \mathbf{I} - (1/N)\mathbf{1}\mathbf{1}^T$. Multiplying on the left of $\mathbf{D}^{(2)}$ by $-1/2\mathbf{H}$ and the right by \mathbf{H} , we obtain

$$\mathbf{A} = -\frac{1}{2}\mathbf{H}\mathbf{D}^{(2)}\mathbf{H} = \mathbf{H}\mathbf{X}^T\mathbf{X}\mathbf{H}. \quad (3)$$

Given \mathbf{A} , one can discover the matrix \mathbf{X} to a rotation and translation by solving the following variational problem

$$\min_{\mathbf{Y}} \|\mathbf{A} - \mathbf{Y}^T\mathbf{Y}\|_F^2, \quad (4)$$

where $\|\cdot\|_F$ indicates the Frobenius norm and the search space is over all full rank $d \times N$ matrices. The solution to \mathbf{X} is then given by

$$\mathbf{X} = \text{diag}(\lambda_1^{1/2}, \dots, \lambda_d^{1/2})\mathbf{V}_1^T, \quad (5)$$

where the singular value decomposition (SVD) of \mathbf{A} is given by

$$\mathbf{A} = [\mathbf{V}_1 \ \mathbf{V}_2] \text{diag}(\lambda_1, \dots, \lambda_d, \lambda_{d+1}, \dots, \lambda_N) [\mathbf{V}_1 \ \mathbf{V}_2]^T. \quad (6)$$

The matrix \mathbf{V}_1 consists of the eigenvectors of the first d eigenvalues $\lambda_1, \dots, \lambda_d$, while the rest of the $N - d$ eigenvectors are represented as \mathbf{V}_2 . The term $\text{diag}(\lambda_1, \dots, \lambda_N)$ refers to a $N \times N$ diagonal matrix with λ_i as its i^{th} diagonal element. Though the solution to the classical MDS is obtained in closed-form, the algorithm has the following deficiencies:

1. MDS requires knowledge of all inter-sensor distances. Obtaining all pairwise range measurements is prohibitive due to the size of the sensor network and the limited power of the sensors. In our problem formulation, this implies that $w_{i,j} \neq 0, \forall i, j$, which makes MDS fall under the category of a centralized approach, i.e., all the information needs to be transmitted to the fusion center which then performs the MDS algorithm. Due to power and bandwidth limitations in the sensor network, this process is infeasible.
2. The inter-sensor range measurements $\delta_{i,j}$ are corrupted by environment and receiver noise which further degrades the quality of the measurements, i.e., $\delta_{i,j}$ is only an estimate of the inter-sensor distance $d_{i,j}$.
3. MDS uses the squared distance matrix which tends to amplify the measurement noise, resulting in poor performance.

As mentioned in Section 1, there has been significant effort directed towards designing decentralized strategies for sensor localization. However, consistent reconstruction of the sensor locations is attainable only in the presence of anchor nodes. If the current localization algorithms are implemented for anchor free localization, the geometry of the sensor network assumes different alignments as localization is performed over various time instants. This makes it

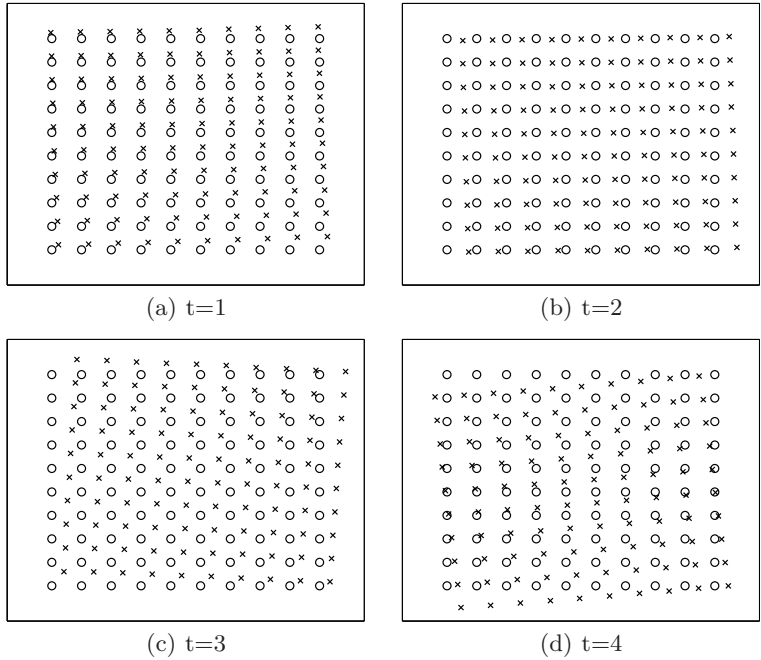


Fig. 4. Anchor free sensor localization by dwMDS. True sensor locations (circle), estimated sensor locations (cross).

impossible to locate where the changes are occurring in the network. To illustrate this phenomenon, we implement the dwMDS algorithm for sensor localization in the absence of anchor nodes and in the absence of target. We provide snapshots of the sensor location estimates (cross) along with their actual locations (circle) in Fig. 4 as a function of time. Observe that the geometry of the network is maintained, while the true locations are subject to rotation and translation. Now consider a target moving through this network. In this scenario, the localization process is affected by two factors: the lack of anchor nodes and some inaccurate inter-sensor measurements in the vicinity of the target. With anchor free localization, the process of tracking a target becomes extremely difficult. To overcome this problem, we propose our sparsity constrained dwMDS algorithm that aligns the current sensor location estimates to those of previous time frames.

4 Sparsity penalized MDS

Consider using the MDS algorithm independently to obtain the sensor location estimates at time t and at time $t - 1$. Alignment between these two sets of

points can be performed in various ways. For example, in Procrustes analysis [17] alignment is performed by finding the optimal affine transformation of one set of nodes that yields the set closest to the second set of points in the least squares sense. However, this procedure cannot guarantee that many sensor locations estimates will remain unchanged from their previously estimated values. The errors in the sensor location estimates between two time steps may accumulate over time resulting in alignment errors. In contrast, we introduce a sparseness penalty on the distances between the sensor location estimates at time t (\mathbf{x}_i) and at time $t - 1$ ($\mathbf{x}_i^{(t-1)}$) directly to the sensor localization algorithm. Construct a vector of Euclidean distances between the location estimates at time t and at time $t - 1$

$$\mathbf{g}^{(t)} = \left[\|\mathbf{x}_1 - \mathbf{x}_1^{(t-1)}\|, \dots, \|\mathbf{x}_n - \mathbf{x}_n^{(t-1)}\| \right]^T. \quad (7)$$

Define the l_0 -measure of a vector $\mathbf{v} = [v_1, v_2, \dots, v_n]$ as the number of nonzero elements given by

$$\|\mathbf{v}\|_0 \triangleq \sum_{i=1}^n I(v_i \neq 0), \quad (8)$$

where $I(\cdot)$ is the indicator function. Using an l_0 -constraint on the distance vector $\mathbf{g}^{(t)}$ of the form $\|\mathbf{g}^{(t)}\|_0 \leq q$, we guarantee that no more than q of the location estimates will vary from their previous time frame values. Minimizing a cost function under the l_0 -constraint requires a combinatorial search which is computationally infeasible. Define the l_p -measure of a vector \mathbf{v} as

$$\|\mathbf{v}\|_p \triangleq \left(\sum_{i=1}^n |v_i|^p \right)^{1/p}. \quad (9)$$

For a quadratic cost function, an l_p -constraint ($0 < p \leq 1$) induces a sparse solution. Among all l_p sparsifying constraints, only $p = 1$ offers a convex relaxation to the l_0 -constraint [15]. To promote sparsity, we next advocate the use of the l_p -constraint as a penalty term via the Lagrange multiplier in the dwMDS algorithm to solve for the sensor location estimates. Hence the term *sparsity penalized MDS*.

The cost function of the dwMDS algorithm [9] is motivated by the variational formulation of the classical MDS, which attempts to find sensor location estimates that minimize the inter-sensor distance errors. Keeping in mind that it is the geometry of the sensor network which is crucial for tracking, we present a novel extension of the dwMDS algorithm through the addition of the sparseness inducing l_p -constraint. At any time t , we seek to minimize the overall cost function $C^{(t)}$ given by

$$C^{(t)} = \sum_{1 \leq i \leq n} \sum_{i \leq j \leq n+m} \sum_{1 \leq l \leq M} w_{i,j}^{(t),l} \left(\delta_{i,j}^{(t),l} - d_{i,j}(\mathbf{X}) \right)^2 + \sum_{i=1}^n r_i \|\bar{\mathbf{x}}_i - \mathbf{x}_i\|^2 + \lambda \|\mathbf{g}^{(t)}\|_p^p. \quad (10)$$

The Euclidean distance $d_{i,j}(\mathbf{X})$ is defined in (1). For each time t , there are M range measurements $\delta_{i,j}^{(t),l}$ for each sensor link i, j . As in [9], the weights $w_{i,j}^{(t),l}$ can be chosen to quantify the accuracy of the predicted distances. When no measurement is made between sensor i and sensor j , $w_{i,j}^{(t),l} = 0$. Furthermore, the weights are symmetric, i.e., $w_{i,j}^{(t),l} = w_{j,i}^{(t),l}$, and $w_{i,i}^{(t),l} = 0$. If available, the a priori information of sensor locations is encoded through the penalty terms $\{r_i \|\bar{\mathbf{x}}_i - \mathbf{x}_i\|^2\}$. Finally, we introduce an l_p -constraint ($0 \leq p \leq 1$) on the distances between the sensor locations at time t and the estimated sensor locations at time $t - 1$. The Lagrange multiplier of the sparseness penalty is denoted as λ . We can tune the value of λ to yield the desired sparsity level in $\mathbf{g}^{(t)}$. Later, when we apply the algorithm for tracking, the sparseness will be advantageous as only those sensors which are highly affected by the target will vary from their initial positions, thereby allowing for a detection of the target through the process of relative sensor localization. To solve this optimization problem, we propose to use the successive refinement technique, where each sensor node i updates its location estimate by minimizing the global cost function $C^{(t)}$, after observing range measurements at node i and receiving position estimates from its neighboring nodes.

4.1 Minimizing cost function by optimization transfer

Unlike classical MDS for which we could obtain a closed-form expression for the estimates, there is no closed-form solution to minimizing $C^{(t)}$. Therefore, we solve the local nonlinear least squares problem iteratively using a quadratic majorization function similar to SMACOF (Scaling by MAjorizing a COmplicated Function [20]). This procedure can be viewed as a special case of optimization transfer algorithms through surrogate objective functions [27], e.g., the popular EM algorithm.

A majorizing function $T(\mathbf{x}, \mathbf{y})$ of $C(\mathbf{x})$ is a function $T : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, which satisfies the following properties: $T(\mathbf{x}, \mathbf{y}) \geq C(\mathbf{x})$, $\forall \mathbf{y}$ and $T(\mathbf{x}, \mathbf{x}) = C(\mathbf{x})$. In other words, the majorizing function upper bounds the original cost function. Using this property, we can formulate an iterative minimization procedure as follows: denote the initial condition as \mathbf{x}_0 . Starting from $n = 1$, obtain \mathbf{x}_n by solving

$$\mathbf{x}_n = \arg \min_{\mathbf{x}} T(\mathbf{x}, \mathbf{x}_{n-1}),$$

until a convergence criterion for $C(\mathbf{x})$ is met. We can easily observe that this iterative scheme always produces a non increasing sequence of cost functions, i.e.,

$$C(\mathbf{x}_{n+1}) \leq T(\mathbf{x}_{n+1}, \mathbf{x}_n) \leq T(\mathbf{x}_n, \mathbf{x}_n) = C(\mathbf{x}_n).$$

The first and last relations follows from the properties of majorizing functions while the middle inequality follows from the fact that \mathbf{x}_{n+1} minimizes $T(\mathbf{x}, \mathbf{x}_n)$. Now the trick is to choose a majorizing function that can be minimized analytically, e.g., a linear or quadratic function. We propose a

quadratic majorizing function $T^{(t)}(\mathbf{X}, \mathbf{Y})$ for the global cost $C^{(t)}(\mathbf{X})$. Minimizing $C^{(t)}(\mathbf{X})$ through the majorization algorithm is the simple task of minimizing the quadratic function $T^{(t)}(\mathbf{X}, \mathbf{Y})$, i.e.,

$$\frac{\partial T^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i} = 0, \quad i = 1, 2, \dots, n. \quad (11)$$

If we denote the estimates of the sensor nodes at iteration k as \mathbf{X}^k , the recursion for the update of location estimates for node i from (11) is given by

$$\mathbf{x}_i^k = \frac{1}{a_i} (c_i + \mathbf{X}^{k-1} \mathbf{b}_i^{k-1}), \quad (12)$$

where \mathbf{b}_i^{k-1} , a_i , and c_i are defined in (32)-(35) respectively. The details of the derivation of the sparsity penalized MDS algorithm can be found in Section 8. For each sensor i , the j^{th} element of the vector \mathbf{b}_i^{k-1} depends on the weight $w_{i,j}$. Since the weights of the nodes not in the neighborhood of the sensor are zero, the corresponding elements in the vector \mathbf{b}_i^{k-1} are also zero; therefore the update rule for node i in (12) will depend only on the location of its nearest neighbors and not on the entire matrix \mathbf{X}^{k-1} . This facilitates the distributed implementation of the algorithm. The proposed algorithm is summarized in Fig. 5. We illustrate the majorization procedure in Fig. 6. The original cost function (solid) and the corresponding surrogate (dotted) is presented for every iteration, along with the track of the estimates at iteration k (circle). Our

Inputs: $\{\bar{w}_{i,j}^{(t)}\}$, $\{\bar{\delta}_{i,j}^{(t)}\}$, $\{r_i\}$, $\{\bar{\mathbf{x}}_i\}$, $\{\mathbf{x}_i^{(t-1)}\}$, ϵ , \mathbf{X}^0 (initial condition for iterations).

Set $k = 0$, compute cost function $C^{(t),0}$ and a_i from equations (10) and (34) respectively

repeat

- $k=k+1$
- for $i = 1$ to n
 - compute \mathbf{b}_i^{k-1} from equation (32)
 - $\mathbf{x}_i^k = \frac{1}{a_i} (c_i + \mathbf{X}^{k-1} \mathbf{b}_i^{k-1})$
 - compute $C_i^{(t),k}$
 - update $C^{(t),k}$ to $C^{(t),k} - C_i^{(t),k-1} + C_i^{(t),k}$
 - communicate \mathbf{x}_i^k to neighbors of sensor i (nodes for which $w_{i,j} > 0$)
 - communicate $C^{(t),k}$ to next node $((i+1) \bmod n)$
- end for

until $C^{(t),k} - C^{(t),k-1} < \epsilon$

Fig. 5. Description of the sparsity constrained MDS algorithm.

proposed algorithm introduces a sparseness penalty on the distance between estimate at time $t-1$ and the current estimate. If the sparsity regularization parameter λ is not chosen properly, many sensor positions estimates might

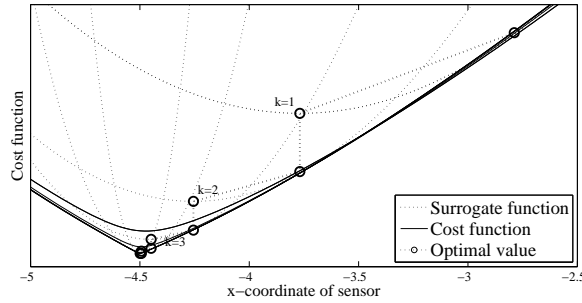


Fig. 6. Majorization procedure: cost function (solid curve), surrogate function (dotted curve), optimal location estimate at each iteration (circle). Only a single coordinate is updated in this picture.

slowly vary with time, thereby creating cumulative error in the sensor localization. An interesting way to counteract this problem would be to penalize the distance between the current estimate and the initial estimate at $t = 1$. Using such a constraint would mean that the sensors are always compared to the fixed initial frame and errors do not accumulate over time. The implementation of this algorithm would be straightforward as it would simply involve changing the index $t - 1$ to 1 in the original algorithm presented in Fig. 5. However, using the estimate from time $t - 1$ has the property that it is easily adapted to the case of mobile sensors.

4.2 Implementation

Weights: When RSS measurements are used to compute distance estimates, the weights are set using the locally weighted regression methods (LOESS) scheme [8] similar to one used in the dwMDS algorithm [9]. The weight assignment is given by

$$w_{i,j} = \begin{cases} \exp(-\delta_{i,j}^2/h_{i,j}^2), & \text{if } i \text{ and } j \text{ are neighbors} \\ 0, & \text{otherwise,} \end{cases}$$

where $h_{i,j}$ is the maximum distance measured by either sensor i or j . A naive equal weight assignment to all measurements is also shown to work well with our algorithm.

Initialization: For the successive refinement procedure, the sensor locations estimates must be initialized for every time frame. Though several initialization algorithms have been proposed in the literature, we use a naive random initialization. We would like to point out that the initialization is not a critical component to our algorithm, as we are solely interested in the alignment of sensors in the network and not on the exact locations. Irrespective of the

initial estimates, the sparseness penalty will ensure that the estimated sensor locations are relatively close to those of previous time frames. Our algorithm is found to be robust with respect to the initial estimates.

Neighborhood selection: Traditionally, the neighbors are chosen based on the distance measure obtained from the RSS measurements, i.e., select all sensors within a distance R as your neighbors. When the RSS measurements are noisy, there is a significant bias in the neighborhood selection rule. This method has a tendency to select sensors which are, on average, less than the actual distances $\|\mathbf{x}_i - \mathbf{x}_j\|$. A simple two-stage adaptive neighborhood selection rule is proposed in [9] to overcome the effect of this bias. We use this selection rule in our algorithm.

Range measurement models: The inter-sensor measurements can be obtained by RSS, TOA, or proximity. Any one of these approaches can be used in our algorithm. Our sparsity constrained MDS algorithm is fairly robust to either of these measurement models. For the simulations in this chapter, we use the RSS to obtain a range measurement between two sensors. It can be shown through the central limit theorem (CLT) that the RSS is log-normal in its distribution [10], i.e., if $P_{i,j}$ is the measured power by sensor i transmitted by sensor j in milliWatts, then $10 \log_{10}(P_{i,j})$ is Gaussian. Thus $P_{i,j}$ in dBm is typically modeled as

$$\begin{aligned} P_{i,j} &= \mathcal{N}(\bar{P}_{i,j}, \sigma_0^2) \\ \bar{P}_{i,j} &= P_0 - 10n_p \log\left(\frac{d_{i,j}}{d_0}\right), \end{aligned} \quad (13)$$

where $\bar{P}_{i,j}$ is the mean received power at distance $d_{i,j}$, σ_0 is the standard deviation of the received power in dBm, and P_0 is received power in dBm at a reference distance d_0 . n_p is referred to as the path-loss exponent that depends on the multipath in the environment. Given the received power, we use maximum likelihood estimation to compute the range, i.e., distance between the sensor nodes i and j . The maximum likelihood estimator of $d_{i,j}$ is given by

$$\hat{\delta}_{i,j} = 10^{((P_0 - P_{i,j})/10n_p)}. \quad (14)$$

Simulation of tracker without a target

The simulation parameters are chosen as follows: we deploy a 10×10 uniform grid of sensors in a network. We consider anchor free localization, i.e., $m = 0$ and we assume we make a single inter-sensor measurement ($M = 1$). We set the sparseness parameter λ to produce a change in the location estimates for only a small portion of the sensors. The value of λ will depend on the size of the network and the noise in the measurements. If the RSS measurements are very noisy, then range estimates become inaccurate which tend to vary the sensor

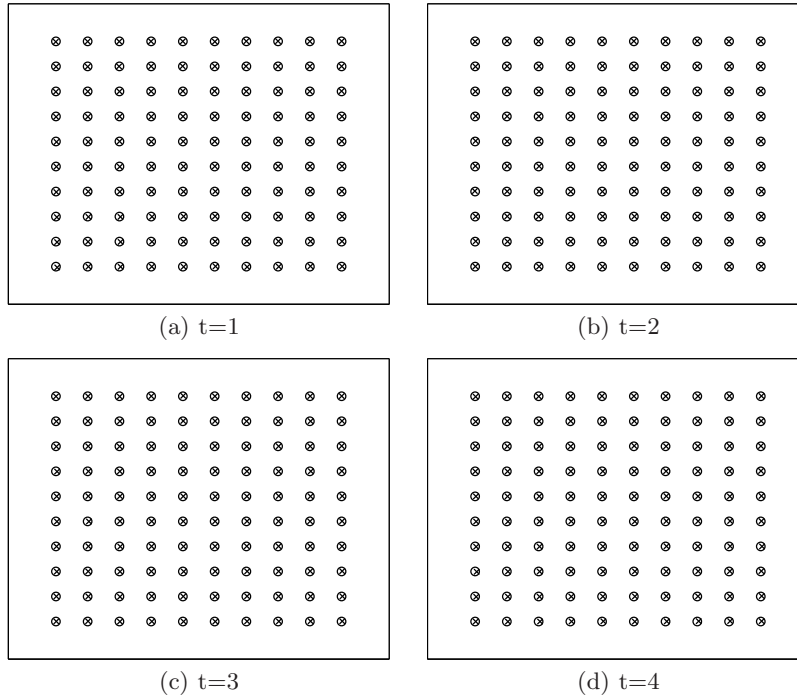


Fig. 7. Anchor free sensor localization by sparsity penalized MDS. True sensor locations (circle), sensor position estimates (cross).

location estimates. Hence λ is selected to ensure that sensor location estimates remain aligned with the previous time frame estimates. In this simulation, we set $\lambda = 0.1$ and the noise variance $\sigma_0 = 0.15$. Each sensor communicates with its 15 nearest neighbors. The weights of the RSS measurements were chosen based on the LOESS scheme described earlier. The weights of links for non communicating sensors were set to zero. We demonstrate the performance of the sparsity constrained MDS algorithm on this sensor network as a function of time in Fig. 7. The true locations are denoted as circles and the estimated locations as crosses.

5 Tracking using sparse MDS

Here we present an algorithm for performing link level tracking using the sparsity constrained MDS algorithm. By link level tracking, we refer to localization of targets to within a set of inter-sensor links. Link level tracking is attractive in the sense that there is no need to assume a physical model for a target. However, it is important to know the effect of the target on the

inter-sensor measurements. Researchers have proposed various models for the signal strength measurements ranging from the traditional linear Gaussian model to the binary sensing model. These are approximate statistical models and the distribution of the measurements in the presence of a target remains an open question.

To model the statistics under the setting of vehicle tracking, we conducted experiments using RF sensors hardware in the presence of a target. We constructed a fine grid of locations, where the target was placed and RSS measurements were recorded between two static sensors for positions on the grid. Upon gathering the data, we fit the following statistical model in the presence of target. The RSS measurements at sensor link i, j are distributed as

$$\begin{aligned} P_{i,j}^k | \hat{P}_{i,j} &\sim \mathcal{N}(\hat{P}_{i,j}, \sigma_0^2), \text{ i.i.d.}, \quad k = 1, 2, \dots, M \\ \hat{P}_{i,j} &\sim \mathcal{N}(\bar{P}_{i,j}, \sigma_1^2), \end{aligned} \quad (15)$$

where $P_{i,j}^k$ is the k^{th} inter-sensor measurement when the target is in the neighborhood of the sensors. The M sensor link measurements are correlated through the random variable $\hat{P}_{i,j}$. The values obtained from our actual experiments were $\sigma_0 \approx 0.1463\text{dBm}$ and $\sigma_1 \approx 1.5\text{dBm}$. The noise variance in the measurements σ_1 was roughly an order of 10 times larger than σ_0 . In other words, RSS measurements tend to have a larger variance due to scattering and attenuation of the signals in the presence of a target. A confidence measure for such a log-normal distribution of the RSS data is obtained using the Kolmogorov-Smirnov (KS) test in [35] and the model is shown to work well for sensor localization. We assume this statistical model for the RSS measurements, when the target is within a specified distance R of the sensor link i, j . The distance R depends on the reflectivity of the object. If the object is highly reflective, then the variation in the RSS measurements is detected by more links.

Given the measurement model, we formulate the optimal decision statistic to detect a presence of a target in a particular sensor link using the likelihood ratio test (LRT). For a fixed false alarm level α , the LRT for each link i, j is given by

$$\left| \frac{1}{M} \sum_{l=1}^M P_{i,j}^{(t),l} - P'_{i,j} \right| \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \quad (16)$$

where $\gamma = (\sigma_0/\sqrt{M})Q^{-1}(\alpha/2)$ and $P'_{i,j}$ is the mean received power in the sensor link estimated using an initial set of range measurements. $\{P_{i,j}^{(t),l}\}_{l=1}^M$ is the set of inter-sensor measurements made by link i, j at time t . We assume that the sensor network is in its steady state operation mode. We do not consider the transient effects in the measured data when it is obtained in the absence of any target. This most powerful test of level α yields the probability of correct detection

$$\beta = 2Q \left(Q^{-1}(\alpha/2) \sqrt{\frac{\sigma_0^2}{\sigma_0^2 + M\sigma_1^2}} \right). \quad (17)$$

A derivation of the decision rule and its performance is given in Section 9. The performance of the optimal detector is clearly dependent on the number of samples available for the inter-sensor measurements. As the number of measurements M becomes very large, β in (17) tends to 1. However, if only few samples are available, β may not approach 1 and misdetect type errors may become non negligible. In such a case, instead of using LRT, we can use a test on the variation of the sensor location estimates at time t from their estimates at a previous time. In other words, we can perform a simple hypothesis test for each link of the form,

$$\|d_{i,j}^t - d_{i,j}^T\| \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{i,j}, \quad (18)$$

where $T = 1$ or $T = t - 1$ depending on whether the sensors are static or mobile.

Simulation of tracker in the presence of target

We present our results by simulating moving targets in a uniform 10×10 grid of sensors. We set $m = 0$, i.e., no anchor nodes. We assume no a priori knowledge of the sensor coordinates, i.e., $r_i = 0$. Each sensor communicates only to its 15 nearest neighbors and the weights for those links were chosen by the LOESS strategy. The rest of the weights were set to zero. We obtain $M = 50$ data measurements for each communicating sensor link in the network. We set the sparseness parameter λ to produce a change in the location estimates for only a small portion of the sensors. We allow any number of targets to appear in a sensor network with probability 0.4. Though our algorithm is robust to randomly moving targets in the network, we consider a state-space model for the purposes of this simulation to produce a visually pleasing target trajectory. We apply the sparsity constrained MDS algorithm as multiple targets move through the sensor network.

The results are shown in Fig. 8. The true sensor locations are shown as circles and the estimated sensor locations are indicated using crosses. The sensors corresponding to those sensor links that declared a target present using the distance based target localization algorithm (DBT) in (18) are shown in filled circles. The target trajectories are shown as inverted triangles. We observe that as the targets move, the sparsity constrained MDS algorithm reconstructs sensor location estimates with the majority of them unchanged from the previous time step. Thus, in conjunction with sparse dwMDS, the DBT is able to localize the targets to within a small set of sensor links.

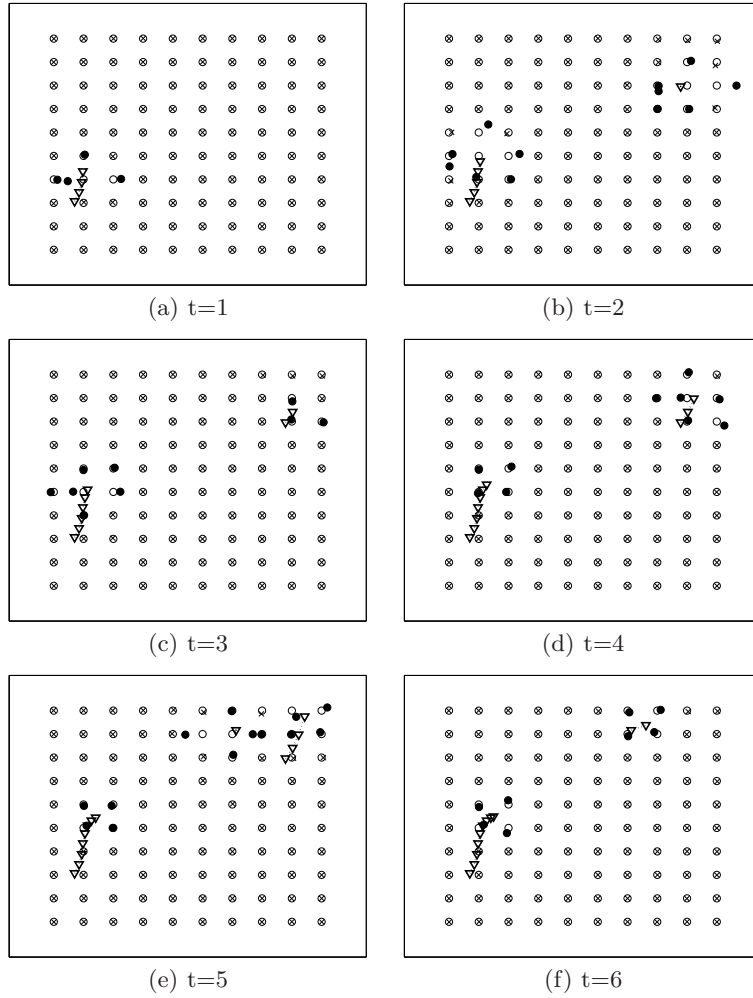


Fig. 8. Anchor free sensor localization by sparsity constrained MDS in the presence of targets. True sensor locations (circle), estimated sensor locations (cross), sensors localizing the target (blocked circle), target trajectory (inverted triangle).

5.1 Numerical Study

We analyze the performance of the localization algorithms using ROC curves. We consider the following setup: we deploy a 10×10 uniform grid of sensors in a network (see Fig. 11). We consider anchor free localization, i.e., $m = 0$ and make a single inter-sensor measurement ($M = 1$) at each time frame. We assume no a priori knowledge of the sensor coordinates, i.e., $r_i = 0$. Each sensor communicates only to its 8 nearest neighbors and the weights for those

links were chosen by the LOESS strategy [9]. The rest of the weights were set to zero. Furthermore, we set noise variances σ_0 and σ_1 defined in (13) and (15), respectively as $\sigma_0 = 1$ and $\sigma_1 = 5\sigma_0 = 5$. Sensor links within a radius $R = 1.5$ indicate the presence of a target, i.e., follow the H_1 hypothesis. We set the reference distance d_0 defined below (13) to be $d_0 = 1$ and the path loss exponent $\eta = 2$. We set the sparseness parameters $\lambda = 2.5$ and $p = 1$ to produce a change in the location estimates for only a small portion ($< 10\%$) of the sensors.

We begin by considering the case of random appearance of targets in the sensor network, i.e., targets appear at different locations every time instant. For the DBT, we set $\tau = 0$ in (18), i.e., we compare our distance estimates to a fixed initial frame. For every time instant, the DBT and the LRT are performed on each active sensor link and the process is repeated for 5000 target locations. The resulting ROC curve is presented in Fig. 9. The ROC for the LRT using simulations is indicated using circles and the corresponding theoretical curve obtained from (17) is shown as a solid line. We observe that the simulation and the theoretical curves match for the LRT. The ROC for the DBT is shown using a dashed line. The DBT algorithm yields higher probability of correct detection than the LRT for most false alarm levels. For example, at false alarm level $\alpha = 0.3$, β for the DBT is approximately 0.89 which is 5% more than that of the LRT, which yields $\beta \approx 0.84$.

The intuition for this result is as follows: in the presence of a target, the RSS measurements of the sensor links are spatially-correlated. The presence of a target in a given link implies that with high probability the target is present in neighboring sensor links. However, the RSS model in (15) specifies only the distribution of the measurements independently on each link. The LRT makes complete use of the RSS measurements but is limited in its performance as the optimal decision statistic for each sensor link i, j is independent of other sensor link measurements. On the other hand, the DBT finds the active sensor links only based on the estimated distances through sparsity penalized MDS. However, since the inter-sensor distances are computed at each sensor using information from its nearest neighbors, this method makes an implicit use of the spatial correlation of the measurements in its decision statistic, which results in an improvement in performance.

Next, we consider the case of a moving target, where we assume a standard state-space target motion model (for the purpose of a visually pleasing trajectory). We repeated the same algorithms for 5000 such trajectories. The LRT based algorithm yields the same performance curve as the test is independent of whether the target is moving or not. The resulting ROC curve for the DBT is presented as a dotted line in Fig. 9. Since we continue to base our decision rule on the fixed initial frame ($\tau = 0$), we observe that the performance of the DBT is also similar to the case of random target appearances.

In the case of a moving target, the RSS measurements are also temporally-correlated. Given a set of sensors indicating a presence of a target at a particular time, there is a high probability that the target is in the vicinity of

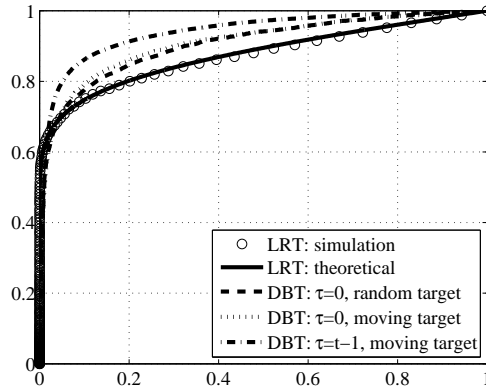


Fig. 9. ROC curve for the LRT and the DBT link level tracking algorithm. LRT (solid line), DBT for a random target with $\tau = 0$ (dashed), DBT for a moving target with $\tau = 0$ (dotted), and DBT for a moving target with $\tau = t - 1$ (dashed dotted).

these sensors at the next time frame. To examine the effect of the temporal correlation, we can compare the current estimated distances to the estimated distances from the previous time-frame rather than the initial frame, i.e., set $\tau = t - 1$ instead of $\tau = 0$. The temporal correlation of the RSS measurements is captured in the DBT through the sparsity constraint used for aligning the sensors locations estimates. In other words, with high probability the sensor location estimates that are perturbed in the previous time-frame will also be perturbed in the current time-frame, thereby increasing the probability of detection.

The results for $\tau = t - 1$ are presented in Fig. 9 using a dashed dotted line. We observe that the performance gains for DBT with $\tau = t - 1$ are higher as compared to DBT with $\tau = 0$ as such a decision rule incorporates both spatial and temporal correlations of the target dynamics. For example, for $\alpha = 0.1$, β for the LRT is 0.75. The result of spatial smoothing alone yields $\beta \approx 0.79$. By performing both spatial and temporal smoothing, we can obtain $\beta \approx 0.86$ through our algorithm, which corresponds to a 15% increase in performance.

We make the following observations for the two proposed tests:

- The DBT for link level tracking outperforms the LRT as it can account for the spatial and the temporal correlations in the target motion.
- The LRT outperforms DBT for low false alarm levels ($\alpha < 0.01$) for the following reasons: first, the DBT we considered is suboptimal as we did not optimize the performance over the choice of sparsity (p, λ) . Furthermore, the LRT uses an optimal decision statistic and the exact measurements to perform the test. Currently, we are in pursuit of finding the optimal sparsity that can yield further improvement in performance.

- The issue of space-time sampling is key to the performance of the DBT. Any scenario that exhibits high spatial correlations (e.g., highly reflective targets or more sensors/unit area) can yield further improvement in performance of the DBT. For example, Fig. 10 illustrates the performance of the DBT for $\tau = 0$ when the number of sensors is increased to 300. By comparing the perturbation to the fixed initial frame, we only perform spatial smoothing of the sensor location estimates. We observe that the denser sampling of sensors have resulted in better spatial smoothing, which eliminates more false alarms resulting in an improved performance. For example, at a false alarm level $\alpha = 0.01$, the DBT with 100 sensors yields $\beta \approx 0.48$, while the DBT with 300 sensors yields $\beta \approx 0.66$. If the sampling time for the sensors and the computation time of the DBT algorithm is much faster than the target motion, the DBT can yield better performance by taking advantage of more temporal correlations.

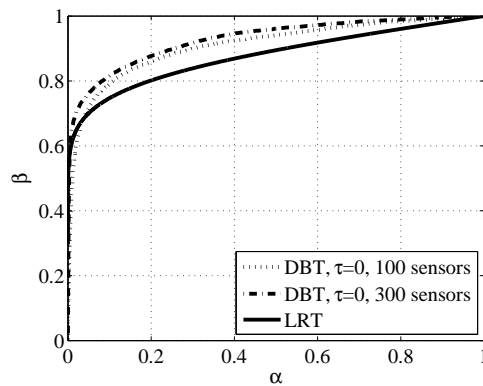


Fig. 10. ROC curve for the LRT and the DBT link level tracking algorithm with $\tau = 0$ for different spatial sampling. LRT (solid line), DBT for a moving target using 10×10 grid of sensors (dotted), and DBT using 300 randomly located sensors (dashed dotted).

- The disadvantage of LRT in this setting is that the test is performed independently on each sensor link. Further improvements in the probability of detection can be achieved when the LRT is derived for the full spatio-temporal model.
- In the performance analysis, we assumed steady state operation, i.e., perfect knowledge of the inter-sensor distances are obtained a priori in the absence of target. If such knowledge is unavailable and distances need to be estimated, the LRT tracker must be modified to a generalized likelihood ratio test (GLRT). The DBT can estimate the initial set of distances more accurately from the RSS measurements by taking advantage of spa-

tial correlations and hence can yield a higher probability of detection than the GLRT.

Spatial localization from link level localization

Our objective is to approximately locate the target relative to the location of the sensors. There are a number of ways in which this link level estimate can be translated into estimated target coordinates in space. For example, one could use as an estimate the midpoint of the convex hull generated by the positions of those sensors that detect the target according to the LRT or the DBT. An example of the midpoint tracking algorithm is shown in Fig. 11. Another estimate can be found by the intersection of convex regions

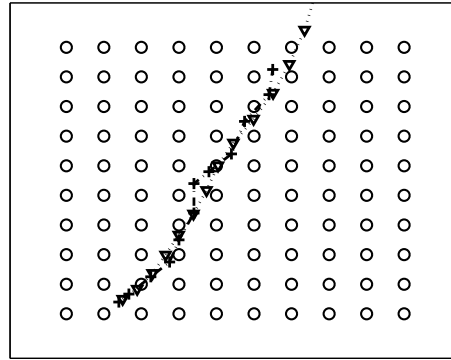


Fig. 11. A simple tracking algorithm based on link level tracking. True sensor locations (circle), true trajectory of the target (diamond), estimated trajectory (plus).

corresponding to the sensor links that show the presence of the target through the optimal decision rule. These estimates do not require a physical model of the target trajectory. However, given a target motion model, standard filtering techniques such as the Kalman filter or particle filter (PF) can be used to obtain refined target position estimates from the link level data.

Future work

Given the set of tagged sensors, i.e., sensor links with high output in the LRT, we have reduced the problem to that of binary sensing, where the knowledge of the presence of the target is stored as the decisions made on each of the links. For accurate estimation of targets positions, we can now use the popular particle filtering techniques proposed on binary sensing models to perform

multi-target tracking given a small set of anchor nodes. Moreover, most sensor networks are remotely operated and limited in power. We can pose a power constraint by limiting the number of inter-sensor measurements to a small s of the $n(n-1)$ ($s \ll n(n-1)$) total sensor links at each time step. The problem of choosing s from $n(n-1)$ links is a combinatorially hard problem. So we propose a convex relaxation to the problem, which chooses the set of active links by minimizing the predicated mean square error of the state of the target. This approach has been shown to achieve near optimal performance in our earlier work [38].

6 Conclusions

In this chapter, we presented the sparsity penalized MDS algorithm for simultaneous localization and tracking. We are interested in tracking a target relative to the sensor coordinates. The subset selection capability of our proposed sparsity constraint allows the algorithm to find only those which have changed their location estimate due to the presence of a target. We use these sensors to perform link level tracking. We formulate a model for the inter-sensor RSS measurements in the presence and absence of a target by conducting actual experiments in free space. Using this model, we illustrated the performance of our algorithm for link level target tracking. Currently, we are in pursuit of optimal sensor scheduling strategies for physical level tracking.

7 Acknowledgements

We would like to thank Mr. Xing Zhou for his enthusiastic assistance in conducting the experiments discussed in this chapter. This work was partially supported by a grant from the National Science Foundation CCR-0325571.

8 Appendix: Derivation of sparsity penalized dwMDS

To simplify our derivation, we divide the global cost function into multiple local cost functions as follows:

$$C^{(t)} = \sum_{i=1}^n C_i^{(t)} + c^{(t)}, \quad (19)$$

where $c^{(t)}$ is a constant independent of the sensor locations \mathbf{X} and the local cost function at each sensor node i is

$$C_i^{(t)} = \sum_{j=1, j \neq i}^n \bar{w}_{i,j}^{(t)} (\bar{\delta}_{i,j}^{(t)} - d_{i,j}(\mathbf{X}))^2 + 2 \sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} (\bar{\delta}_{i,j}^{(t)} - d_{i,j}(\mathbf{X}))^2 + r_i \|\bar{\mathbf{x}}_i - \mathbf{x}_i\|^2 + \lambda \|\mathbf{x}_i - \mathbf{x}_i^{(t-1)}\|^p, \quad (20)$$

where $\bar{w}_{i,j}^{(t)} = \sum_{l=1}^M w_{i,j}^{(t),l}$ and $\bar{\delta}_{i,j}^{(t)} = \sum_{l=1}^M w_{i,j}^{(t),l} \delta_{i,j}^{(t),l} / \bar{w}_{i,j}^{(t)}$. The cost function $C_i^{(t)}$ depends only the measurements made by sensor node i and the positions of the neighboring nodes, i.e., nodes for which $w_{i,j}^{(t),l} > 0$; $C_i^{(t)}$ is local to node i [9]. The local cost function in (20) can be rewritten as

$$C_i^{(t)}(\mathbf{X}) = c_1^{(t)} + c_2^{(t)}(\mathbf{X}) - c_3^{(t)}(\mathbf{X}) + c_4^{(t)}(\mathbf{X}), \quad (21)$$

where

$$\begin{aligned} c_1^{(t)} &= \sum_{j=1, j \neq i}^n \bar{w}_{i,j}^{(t)} (\bar{\delta}_{i,j}^{(t)})^2 + 2 \sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} (\bar{\delta}_{i,j}^{(t)})^2 \\ c_2^{(t)}(\mathbf{X}) &= \sum_{j=1, j \neq i}^n \bar{w}_{i,j}^{(t)} d_{i,j}^2(\mathbf{X}) + 2 \sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} d_{i,j}^2(\mathbf{X}) + r_i \|\bar{\mathbf{x}}_i - \mathbf{x}_i\|^2 \\ c_3^{(t)}(\mathbf{X}) &= 2 \sum_{j=1, j \neq i}^n \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} d_{i,j}(\mathbf{X}) + 4 \sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} d_{i,j}(\mathbf{X}) \\ c_4^{(t)}(\mathbf{X}) &= \lambda \|\mathbf{x}_i - \mathbf{x}_i^{(t-1)}\|^p. \end{aligned} \quad (22)$$

The term $c_1^{(t)}$ is independent of \mathbf{x}_i . The term $c_2^{(t)}$ is quadratic in \mathbf{x}_i . Terms $c_3^{(t)}$ and $c_4^{(t)}$ are neither affine nor quadratic functions of \mathbf{x}_i . A majorizing function for the term $c_3^{(t)}$ is motivated by the following Cauchy-Schwarz inequality,

$$d_{i,j}(\mathbf{X}) = \|\mathbf{x}_i - \mathbf{x}_j\| \geq \frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{y}_i - \mathbf{y}_j)}{d_{i,j}(\mathbf{Y})}, \quad \forall \mathbf{Y}, \quad (23)$$

where $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]$. For $c_4^{(t)}$, we present a quadratic majorizing function, which can be obtained from the following relation

$$\alpha^{p/2} \leq \alpha_0^{p/2} + \frac{p}{2} (\alpha - \alpha_0) (\alpha_0)^{(\frac{p}{2}-1)}, \quad \forall \alpha, \alpha_0 > 0. \quad (24)$$

The above inequality follows from a linear approximation to the concave function $f(\alpha) = \alpha^{p/2}$ via Taylor series expansion. Choosing $\alpha = \|\mathbf{x}_i - \mathbf{x}_i^{t-1}\|^2$ and $\alpha_0 = \|\mathbf{y}_i - \mathbf{x}_i^{t-1}\|^2$ yields

$$\|\mathbf{x}_i - \mathbf{x}_i^{t-1}\|^p \leq \|\mathbf{y}_i - \mathbf{x}_i^{t-1}\|^p + \frac{p}{2} \frac{\|\mathbf{x}_i - \mathbf{x}_i^{t-1}\|^2 - \|\mathbf{y}_i - \mathbf{x}_i^{t-1}\|^2}{\|\mathbf{y}_i - \mathbf{x}_i^{t-1}\|^{2-p}}, \quad (25)$$

the majorizing function for the $c_4^{(t)}$ term. Substituting the inequalities from (23) and (25) in (21), we obtain the majorizing function for the local cost function as

$$\begin{aligned}
 T_i^{(t)}(\mathbf{X}, \mathbf{Y}) &= c_1^{(t)} + \sum_{j=1, j \neq i}^n \bar{w}_{i,j}^{(t)} d_{i,j}^2(\mathbf{X}) + 2 \sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} d_{i,j}^2(\mathbf{X}) + r_i \|\bar{\mathbf{x}}_i - \mathbf{x}_i\|^2 \\
 &+ 2 \sum_{j=1, j \neq i}^n \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} \frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{y}_i - \mathbf{y}_j)}{d_{i,j}(\mathbf{Y})} \\
 &+ 4 \sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} \frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{y}_i - \mathbf{y}_j)}{d_{i,j}(\mathbf{Y})} \\
 &+ \lambda \|\mathbf{y}_i - \mathbf{x}_i^{(t-1)}\|^p + \frac{\lambda p \|\mathbf{x}_i - \mathbf{x}_i^{(t-1)}\|^2 - \|\mathbf{y}_i - \mathbf{x}_i^{(t-1)}\|^2}{2 \|\mathbf{y}_i - \mathbf{x}_i^{(t-1)}\|^{2-p}}.
 \end{aligned} \tag{26}$$

Since $T_i^{(t)}(\mathbf{X}, \mathbf{Y})$ is a majorizing function to $C_i^{(t)}(\mathbf{X})$, it is easy to verify that the function $T^{(t)}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^n T_i^{(t)}(\mathbf{X}, \mathbf{Y})$ is a majorizing function to the global cost function $C^{(t)}(\mathbf{X})$. The partial derivative of $T^{(t)}(\mathbf{X}, \mathbf{Y})$ with respect to \mathbf{x}_i is straightforward as all the expressions in (26) are linear or quadratic in \mathbf{x}_i . The partial derivative of $T^{(t)}(\mathbf{X}, \mathbf{Y})$ with respect to \mathbf{x}_i is given by

$$\frac{\partial T^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i} = \frac{\partial T_i^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i} + \sum_{k \neq i} \frac{\partial T_k^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i}, \tag{27}$$

where

$$\begin{aligned}
 \frac{\partial T_i^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i} &= 2 \sum_{j=1, j \neq i}^n \left(\bar{w}_{i,j}^{(t)} (\mathbf{x}_i - \mathbf{x}_j) - \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} \frac{(\mathbf{y}_i - \mathbf{y}_j)}{\|\mathbf{y}_i - \mathbf{y}_j\|} \right) \\
 &+ 4 \left(\sum_{j=n+1}^{n+m} \bar{w}_{i,j}^{(t)} (\mathbf{x}_i - \mathbf{x}_j) - \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} \frac{(\mathbf{y}_i - \mathbf{y}_j)}{\|\mathbf{y}_i - \mathbf{y}_j\|} \right) \\
 &+ 2r_i (\mathbf{x}_i - \bar{\mathbf{x}}_i) + \lambda p \frac{(\mathbf{x}_i - \mathbf{x}_i^{(t-1)})}{\|\mathbf{y}_i - \mathbf{x}_i^{(t-1)}\|^{2-p}}
 \end{aligned} \tag{28}$$

and

$$\frac{\partial T_k^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i} = 2 \left(\bar{w}_{i,k}^{(t)} (\mathbf{x}_i - \mathbf{x}_k) - \bar{w}_{i,k}^{(t)} \bar{\delta}_{i,k}^{(t)} \frac{(\mathbf{y}_i - \mathbf{y}_k)}{\|\mathbf{y}_i - \mathbf{y}_k\|} \right). \tag{29}$$

Substituting (28) and (29) in (27) yields,

$$\begin{aligned}
 \frac{\partial T^{(t)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_i} &= 4 \left(\sum_{j=1, j \neq i}^{n+m} \bar{w}_{i,j}^{(t)} (\mathbf{x}_i - \mathbf{x}_j) - \bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)} \frac{(\mathbf{y}_i - \mathbf{y}_j)}{\|\mathbf{y}_i - \mathbf{y}_j\|} \right) \\
 &+ 2r_i (\mathbf{x}_i - \bar{\mathbf{x}}_i) + \lambda p \frac{(\mathbf{x}_i - \mathbf{x}_i^{(t-1)})}{\|\mathbf{y}_i - \mathbf{x}_i^{(t-1)}\|^{2-p}}.
 \end{aligned} \tag{30}$$

Setting the derivatives to zero yields the following recursive update rule

$$\mathbf{x}_i^k = \frac{1}{a_i} \left(c_i + \left[\mathbf{x}_1^{(k-1)}, \dots, \mathbf{x}_N^{(k-1)} \right] \mathbf{b}_i^{(k-1)} \right), \quad (31)$$

where \mathbf{x}_i^k denotes the location of node i at iteration k . Furthermore, $\mathbf{b}_i^k = [b_1^k, b_2^k, \dots, b_N^k]$ and

$$b_i^k = 4 \left(\sum_{j=1, j \neq i}^{n+m} \frac{\bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)}}{\|\mathbf{x}_i^k - \mathbf{x}_j^k\|} \right), \quad (32)$$

$$b_j^k = 4 \left(\bar{w}_{i,j}^{(t)} - \frac{\bar{w}_{i,j}^{(t)} \bar{\delta}_{i,j}^{(t)}}{\|\mathbf{x}_i^k - \mathbf{x}_j^k\|} \right), \quad j \neq i, \quad (33)$$

$$a_i = 4 \sum_{j=1, j \neq i}^{n+m} \bar{w}_{i,j}^{(t)} + 2r_i + \frac{\lambda p}{\|\mathbf{x}_i^k - \mathbf{x}_i^{t-1}\|^{2-p}}, \quad (34)$$

$$c_i = 2r_i \bar{\mathbf{x}}_i + \frac{\lambda p \mathbf{x}_i^{(t-1)}}{\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^{(t-1)}\|}. \quad (35)$$

The dwMDS algorithm in [9] obtains a recursive update for location \mathbf{x}_i by setting the derivatives of the surrogate to the i^{th} local cost function $(T_i^{(t)}(\mathbf{X}, \mathbf{Y}))$ to zero. This is equivalent to minimizing the global cost function only under anchor free localization ($m = 0$) and no a priori information ($r_i = 0$). However, in our algorithm, we use the local cost functions only to derive a majorizing function for the global cost function and not in the minimization. Moreover, the algorithm is still decentralized in its implementation though we minimize the global cost function with respect to the sensor locations \mathbf{X} .

9 Appendix: Optimal likelihood ratio test

To test the presence of a target on a sensor link i, j , we pose the following hypotheses testing problem

$$\begin{aligned} H_0 &: P_1, \dots, P_M \sim \mathcal{N}(\bar{P}, \sigma_0^2) \\ H_1 &: P_1, \dots, P_M | \hat{P} \sim \mathcal{N}(\hat{P}, \sigma_0^2), \text{ i.i.d.}, \quad \hat{P} \sim \mathcal{N}(\bar{P}, \sigma_1^2), \end{aligned}$$

where P_1, \dots, P_M are the measurements made by a particular link i, j . We leave out the indices i, j in the measurements for brevity. \bar{P} is the mean received power in the sensor link i, j . We assume it can be obtained during the system setup in the absence of targets. Denote the measurements by the M -element vector $\mathbf{p} = [P_1, P_2, \dots, P_M]^T$. Then the hypotheses can be written as

$$\begin{aligned} \text{H}_0 : \mathbf{p} &\sim \mathcal{N}(\bar{P}\mathbf{1}, \sigma_0^2\mathbf{I}) \\ \text{H}_1 : \mathbf{p} &\sim \mathcal{N}(\bar{P}\mathbf{1}, \sigma_1^2\mathbf{1}\mathbf{1}^T + \sigma_0^2\mathbf{I}). \end{aligned}$$

To construct the LRT, we first compute the log likelihood ratio as

$$\begin{aligned} \Lambda &= \log \left(\frac{f(\mathbf{p}|\text{H}_1)}{f(\mathbf{p}|\text{H}_0)} \right) \\ &= \frac{1}{2}(\mathbf{p} - \bar{P}\mathbf{1})^T (\mathbf{C}_0^{-1} - \mathbf{C}_1^{-1})(\mathbf{p} - \bar{P}\mathbf{1}) + \frac{1}{2} \log \left(\frac{|\mathbf{C}_0|}{|\mathbf{C}_1|} \right), \end{aligned} \quad (36)$$

where $\mathbf{C}_0 = \sigma_0^2\mathbf{I}$, $\mathbf{C}_1 = \sigma_1^2\mathbf{1}\mathbf{1}^T + \sigma_0^2\mathbf{I}$ and $|\mathbf{C}|$ denotes the determinant of a matrix \mathbf{C} . The eigendecompositions of the covariance matrices \mathbf{C}_0 and \mathbf{C}_1 can be written as

$$\begin{aligned} \mathbf{C}_0 &= \mathbf{V}_0\mathbf{D}_0\mathbf{V}_0^T, \\ \mathbf{C}_1 &= \mathbf{V}_1\mathbf{D}_1\mathbf{V}_1^T, \end{aligned}$$

where \mathbf{D}_i is a diagonal matrix composed of the eigenvalues $\lambda_1^i, \dots, \lambda_M^i$ and \mathbf{V}_i is the matrix of corresponding eigenvectors. The eigenvalues of the covariance matrix \mathbf{C}_1 are given by $\lambda_1^1 = \sigma_1^2 M + \sigma_0^2$ and $\lambda_i^1 = \sigma_0^2$, $i = 2, \dots, M$. The corresponding eigenvectors are $\mathbf{v}_1 = \mathbf{1}/\sqrt{M}$, $\mathbf{v}_2, \dots, \mathbf{v}_M$, where $\{\mathbf{v}_i\}_{i=1}^M$ are a set of orthogonal unit norm vectors. The eigenvalues of \mathbf{C}_0 are all σ_0^2 and it is easy to verify that $\mathbf{v}_1, \dots, \mathbf{v}_M$ are eigenvectors to \mathbf{C}_0 , i.e., $\mathbf{V}_0 = \mathbf{V}_1$. Thus

$$\mathbf{C}_0^{-1} - \mathbf{C}_1^{-1} = \mathbf{V}_0 \text{diag} \left(\frac{M\sigma_1^2}{M\sigma_1^2 + \sigma_0^2}, 0, \dots, 0 \right) \mathbf{V}_0^T = \frac{\sigma_1^2 M}{\sigma_1^2 M + \sigma_0^2} \frac{\mathbf{1}\mathbf{1}^T}{M}. \quad (37)$$

Substituting (37) in (36) and collecting constant terms at the right hand side yields the optimal LRT as

$$|\bar{p} - \bar{P}| \underset{\text{H}_0}{\overset{\text{H}_1}{\geq}} \gamma, \quad (38)$$

where $\bar{p} = \sum_{i=1}^M P_i/M$ is the minimal sufficient statistics of this test. Under H_0 , \bar{p} is distributed as $\mathcal{N}(\bar{P}, \sigma_0^2/M)$ and under H_1 , \bar{p} is $\mathcal{N}(\bar{P}, \sigma_0^2/M + \sigma_1^2)$. We find γ to satisfy a false alarm of level α , i.e.,

$$P(|\bar{p} - \bar{P}| > \gamma | \text{H}_0) = 2Q \left(\frac{\sqrt{M}\gamma}{\sigma_0} \right) = \alpha, \quad (39)$$

which implies $\gamma = (\sigma_0/\sqrt{M})Q^{-1}(\alpha/2)$. The probability of correct decision, β is then given by

$$\begin{aligned} \beta &= P(|\bar{p} - \bar{P}| > \gamma | \text{H}_1) \\ &= 2Q \left(\frac{\gamma}{\sqrt{\sigma_0^2/M + \sigma_1^2}} \right) \\ &= 2Q \left(Q^{-1}(\alpha/2) \sqrt{\frac{\sigma_0^2}{\sigma_0^2 + M\sigma_1^2}} \right). \end{aligned} \quad (40)$$

References

- [1] D. L. Alspach and H. W. Sorensen. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. Automat. Contr.*, 82:1032–1063, 1987.
- [2] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, and S. Kulkarni. A line in the sand: a wireless sensor network for detection, classification, and tracking. *Computer Networks*, 46(5):605–634, 2004.
- [3] Y. Bar-Shalom. *Multitarget Multisensor Tracking: Advanced Applications*. Artech House, 1990.
- [4] R. R. Brooks, P. Ramanathan, and A. M. Sayeed. Distributed target classification and tracking in sensor networks. *Proc. IEEE*, 91(8):1163–1171, 2003.
- [5] S. Capkun, L. Buttyan, and J. Hubaux. SECTOR: Secure tracking of node encounters in multi-hop wireless networks. In *Proc. 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 21–32, 2003.
- [6] V. Cevher and J. H. McClellan. Sensor array calibration via tracking with the extended Kalman filter. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, volume 5, pages 2817–2820, 2001.
- [7] J. Chamberland and V. V. Veeravalli. Decentralized detection in sensor networks. *IEEE Trans. Signal Processing*, 51(2):407–416, 2003.
- [8] W. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Am. Statist. Assoc.*, 74(368):829–836, 1979.
- [9] J. Costa, N. Patwari, and A. O. Hero III. Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. *ACM J. Sensor Networking*, 2(1):39–64, 2006.
- [10] A. J. Coulson, A. G. Williamson, and R. G. Vaughan. A statistical basis for lognormal shadowing effects in multipath fading channels. *IEEE Trans. on Veh. Tech.*, 46(4):494–502, 1998.
- [11] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.
- [12] M. L. Davidson. *Multidimensional scaling*. Wiley, New York, NY, 1983.
- [13] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003.
- [14] L. Doherty, K. S. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1655–1663, 2001.
- [15] D. L. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. on Inform. Theory*, 52(1):6–18, 2006.

- [16] C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *ACM SIGCOMM*, pages 137–148, 2003.
- [17] J. C. Gower and G .B. Dijkstra. *Procrustes Problems*. Oxford University Press, 2004.
- [18] M. J. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press Inc., London, UK, 1984.
- [19] C. J. Gregory, R. R. Carthy, and L. G. Pearlstine. Survey and monitoring of species at risk at camp blanding training site, northeastern florida. *Southeastern Naturalist*, 5(3):473–498, 2006.
- [20] P. Groenen. *The majorization approach to multidimensional scaling: some problems and extensions*. DSWO Press, 1993.
- [21] A. O. Hero III. Geometric entropy minimization (GEM) for anomaly detection and localization. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2006.
- [22] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proc. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1976–1986, 2003.
- [23] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In *Proc. IEEE Infocom*, pages 2652–2661, 2004.
- [24] S. J. Julier and J. K. Uhlman. A new extension of the kalman filter to nonlinear systems. In *Proc. AeroSense: Eleventh Intl. Symp. on Aerospace/Defense Sensing, Simulations and Control Multi Sensor Fusion, Tracking and Resource Management II*, volume 3068, pages 182–193, 1997.
- [25] W. Kim, K. Mechitov, J. Y. Choi, and S. Ham. On target tracking with binary proximity sensors. In *Proc. Fourth Intl Symposium Information Processing in Sensor Networks*, pages 301–308, 2005.
- [26] J. Kumagai and S. Cherry. Sensors and sensibility. *IEEE Spectrum*, 41(7):22–28, 2004.
- [27] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.
- [28] M. S. Lee and Y. H. Kim. An efficient multitarget tracking for car applications. *IEEE Trans. Industrial Electronics*, 50(2):397–399, 2003.
- [29] M. Leoncini, G. Resta, and P. Santi. Analysis of a wireless sensor dropping problem for wide-area environmental monitoring. In *Fourth Intl. Symp. on Inform. Processing Sensor Networks (IPSN)*, pages 239–245, 2005.
- [30] R. L. Moses, D. Krishanmurthy, and R. Patterson. A self-localization method for wireless sensor networks. *EURASIP J. Applied Signal Processing*, 4:348–358, 2003.
- [31] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proc. 2nd*

- Intl Workshop on Inform. Processing Sensor Networks (IPSN), Lecture Notes in Computer Science*, volume 2634, pages 333–348, 2003.
- [32] D. Niculescu and B. Nath. Ad hoc positioning systems. In *Proc. IEEE Global Communications Conference (GLOBECOM)*, volume 5, pages 2926–2931, 2001.
- [33] R. Niu, P. K. Varshney, and Q. Cheng. Distributed detection in a large wireless sensor network. *Information Fusion*, 7:380–394, 2006.
- [34] P. Pathirana, N. Bulusu, S. Jha, and A. Savkin. Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Trans. Mobile Computing*, 4(3):285–296, 2005.
- [35] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O’Dea. Relative location estimation in wireless sensor networks. *IEEE Trans. Signal Processing*, 51(8):2137–2148, 2003.
- [36] N. Patwari, A. O. Hero III, and A. Pacholski. Manifold learning visualization of network traffic data. In *Proc. Workshop on Mining Network Data, Philadelphia, PA*, pages 191–196, 2005.
- [37] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, New York, N.Y., 1988.
- [38] R. Rangarajan, R. Raich, and A. O. Hero III. Single-stage waveform selection for adaptive resource constrained state estimation. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, volume 3, pages 672–675, 2006.
- [39] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proc. Intl Workshop on Sensor Nets & Apps.*, pages 112–121, 2002.
- [40] Y. Shang, W. Rumi, Y. Zhang, and M. P. Fromherz. Localization from mere connectivity. In *Proc. 4th ACM Intl. Symp. on Mobile ad hoc networking and computing*, pages 201–212, 2003.
- [41] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In *Proc. ACM 4th Intl. Conf. Embedded networked sensor systems (SenSys)*, pages 251–264, 2006.
- [42] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. In *Proc. 5th Intl. Conf. Proc. Information Processing in Sensor Networks (IPSN)*, pages 27–33, 2006.
- [43] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *IEEE Symp. Adaptive Systems for Signal Processing, Communications, and Control*, pages 153–158, 2000.
- [44] N. Wang, M. H. Wang, and N. Q. Zhang. Wireless sensors in agriculture and food industry: Recent developments and future perspective. *Computers and electronics in agriculture*, 50(1):1–14, 2006.
- [45] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *Proc. ACM Workshop on wireless security*, pages 51–60, 2004.