Robust Localization of Nodes and Time-Recursive Tracking in Sensor Networks Using Noisy Range Measurements

Pınar Oğuz-Ekim, *Student Member, IEEE,* João Gomes, *Member, IEEE,*
João Xavier, *Member, IEEE,* and Paulo Oliveira, *Member, IEEE*

Abstract

Simultaneous localization and tracking (SLAT) in sensor networks aims to determine the positions of sensor nodes and a moving target in a network, given incomplete and inaccurate range measurements between the target and each of the sensors. One of the established methods for achieving this is to iteratively maximize a likelihood function (ML) of positions given the observed ranges, which requires initialization with an approximate solution to avoid convergence towards local extrema. This paper develops methods for handling both Gaussian and Laplacian noise, the latter modeling the presence of outliers in some practical ranging systems that adversely affect the performance of localization algorithms designed for Gaussian noise. A modified Euclidean Distance Matrix (EDM) completion problem is solved for a block of target range measurements to approximately set up initial sensor/target positions, and the likelihood function is then iteratively refined through Majorization-Minimization (MM). To avoid the computational burden of repeatedly solving increasingly large EDM problems in time-recursive operation, an incremental scheme is exploited whereby a new target/node position is estimated from previously available node/target locations to set up the iterative ML initial point for the full spatial configuration. The above methods are first derived under Gaussian noise assumptions, and modifications for Laplacian noise are then considered. Analytically, the main challenges to overcome in the Laplacian case stem from the non-differentiability of ℓ_1 norms that arise in the various cost functions. Simulation results

The authors are with the Institute for Systems and Robotics – Instituto Superior Técnico (ISR/IST), Lisbon, Portugal. e-mails: {poguz,jpg,jxavier,pjcro}@isr.ist.utl.pt.

This research was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through ISR/IST plurianual funding with PIDDAC program funds, projects PTDC/EEA-TEL/71263/2006, PTDC/EEA-CRO/104243/2008, CMU-PT/SIA/0026/2009, and grant SFRH/BD/44771/2008.

show that the proposed algorithms significantly outperform existing localization methods in the presence of outliers, while providing comparable performance for Gaussian noise.

I. Introduction

This work addresses the problem of tracking a single target from distance-like measurements taken by nodes in a sensor network whose positions are not precisely known. The goal is to estimate the positions of all sensors and of the target, given partial or no *a priori* information regarding the spatial configuration of the network. As the ability to track a target is a key component in several scenarios of wireless sensor networks, methods that avoid the need for careful calibration of sensor positions are practically relevant.

In this section we describe the proposed approach, outline the main contributions, and review some relevant literature. The latter includes other SLAT methods and Sensor Network Localization (SNL) methods, which are actually more directly related to our approach.

In [1], [2] SLAT is formulated in a Bayesian framework that exploits the connections with the well-studied problem of simultaneous localization and mapping (SLAM) in robotics. The *a posteriori* probability density function of sensor/target positions and calibration parameters is recursively propagated in time as new target sightings become available. The observations in [1] are true range measurements obtained through a combination of transmitted acoustic and radio pulses. Some alternatives to range include pseudorange and bearing information estimated from camera images [2] or the (somewhat unreliable) Received Signal Strength (RSS) of radio transmissions [3]. In [4] the SLAT problem is also formulated in a Bayesian framework as a general state evolution model under a binary proximity model and solved in a decentralized way using binary sensor networks. Another SLAT-like approach using localization techniques (calibration) is presented in [5], where positions and orientations of unknown sources and sensors are centrally obtained via ML based on time-of-arrival and angle-of-arrival measurements.

In our approach target dynamics are not accounted for; therefore the SLAT problem may be thought of as a special type of SNL, with a limited set of intersensor measurements, for a network comprising the original set of nodes and the sensed target positions. We resort to EDM methods based on Semidefinite Programming (SDP), which were previously adopted for static SNL (see [6] and references therein). EDM completion for SLAT is discussed in [3], although the authors pursued an alternative approximate completion approach based on a variant of Multidimensional Scaling (MDS). Underwater and underground scenarios with uncertainty in anchor positions are considered in [7], and edge-based SDP is proposed to reduce the computational complexity of SNL. In [8] static SNL is formulated as a problem of ML phase retrieval.

In addition to centralized SNL approaches such as [6]–[8], enumerated above, a wealth of results are available on distributed approaches for scenarios where the existence of a central node is inconvenient, e.g., due to congested communications in its vicinity or excessive vulnerability of the whole infrastructure to failure of that single node [3], [9]–[13]. A two-step approach based on second-order cone programming relaxation with inaccurate anchor positions is introduced in [9]. In [11] a weighted least-squares algorithm with successive refinement provides both position estimates and their covariances in partially connected scenarios. A distributed weighted MDS method with majorization approximations is applied in [12]. The cost function and the majorization technique are similar to those used in this paper for ML iterative refinement under Gaussian noise, but initialization relies on prior estimates of sensor positions.

This paper focuses on centralized SLAT based on plain ML estimation. We propose a two-stage approach consisting of a *startup phase* whose main goal is to obtain an outline of the network configuration from a block of measurements (as in [1], we will often use the term *batch* for such a block, or to qualify the associated processing algorithms), followed by an *updating phase* where new target sightings are incrementally assimilated as they become available, while improving all previously determined locations. Each phase consists of an *initialization step* to calculate approximate locations, followed by an iterative *refinement step* of the likelihood function using Majorization-Minimization (MM) [14]. Local convergence to undesirable extrema in ML methods due to poor initialization is thus alleviated.

During startup the initialization step solves an EDM completion problem for range data from multiple target sightings, which requires little *a priori* knowledge of sensor/target positions. The updating phase, carried out for each new target sighting after startup, aims to bypass the need for EDM initialization with increasingly large matrices as time progresses and more range measurements become available. Initialization in this phase uses source localization algorithms that fix all previously estimated positions and attempt to determine the location for the most recently observed target. The computational load of this simplified recursive initialization scheme remains constant with time.

Table I provides a high-level description of our SLAT algorithm. Detailed expressions for the various operations are derived below for Gaussian and Laplacian noise. We highlight the following contributions of our work:

SLAT with modest prior knowledge: We emphasize the development of a SLAT method with modest prior assumptions on the sensor/target positions. This is achieved mainly by casting SLAT as an SNL or source localization problem during initialization in the startup and updating phases, respectively, which admit accurate convex relaxations where local extrema are absent. Anchors are still needed in refinement steps to eliminate fundamental translation and rotation ambiguities in the likelihood function. By contrast,

in the published Bayesian formulations of SLAT a tacit assumption is made that the prior is sufficiently concentrated around the true locations to circumvent local convergence issues.

Coherent ML framework with moderate complexity: Our startup/time-recursive updating approach was proposed in [15] for Gaussian noise, using cost functions for the initialization steps that match *squared* observations with *squared* estimated ranges. These discrepancies with respect to the likelihood function are eliminated in the present paper, allowing for both the initialization and ML refinement steps to operate with cost functions that match plain (non-squared) ranges, and leading to improved robustness under strong measurement noise [16]. This is attained with a similar complexity to the methods in [15].

Localization for Gaussian and Laplacian noise models: In addition to the Gaussian case, this work develops startup and updating algorithms for Laplacian noise, to model the presence of outliers in some practical ranging systems that adversely affect the performance of localization algorithms designed for Gaussian noise [1], [17]. Our proposed methods for the initialization steps are novel and relevant for SNL and source localization applications. In particular, for startup initialization we develop EDM completion methods that depart from related approaches [6], [18] in which squared range measurements are matched. The details of our cost functions are different for Gaussian and Laplacian noise models, but in both cases we gain robustness to range errors relative to more standard EDM methods by matching plain distances. With regard to the updating phase, we use the SLCP method proposed in [19] for the initialization step under Gaussian noise. SLCP is a 2D source localization algorithm which matches plain ranges using a formulation in the complex plane to attain an accurate convex relaxation as an SDP. Under Laplacian noise we present a novel source localization algorithm using ℓ_1 norms that we designate by SLℓ_1.

We address Laplacian noise by replacing ℓ_2 norms with ℓ_1 norms for various optimization subproblems that are initially formulated under the assumption of Gaussian noise, and then performing suitable manipulations to write these in a form that is amenable to general-purpose solvers. In [17] ℓ_1 norms are also used to handle outliers, but the proposed method is very different from the one developed here, as it relies on linear programming to identify the outliers, and then removes them from consideration when computing the source location. In our work all measurements are kept, as the modulus of range differences that appears in cost functions ensures that outlier terms do not overwhelm the remaining ones if the proportion of outliers remains small. Another approach for handling outliers is presented in [20], where the Huber cost function interpolates between ℓ_1 and ℓ_2 norms. This function is minimized via iterative majorization techniques with *a priori* information on sensor positions.

The remainder of the paper is organized as follows. In Section II, the SLAT problem is introduced.

Section III presents estimation methods for range measurements corrupted by Gaussian noise, namely, EDM initialization, iterative likelihood refinement by MM, and time-recursive updating through incremental estimation of target/sensor positions. Section IV develops similar methods for Laplacian noise. Section V provides simulation results for the performances of startup and updating algorithms under both types of measurement noise. Lastly, Section VI summarizes the main conclusions.

Throughout, both scalars and individual position vectors in (2D) space will be represented by lowercase letters. Matrices and vectors of concatenated coordinates will be denoted by boldface uppercase and lowercase letters, respectively. The superscript T (H) stands for the transpose (Hermitian) of the given real (complex) vector or matrix. Below, $\mathbf{I}_m$ is the $m \times m$ identity matrix and $\mathbf{1}_m$ is the vector of m ones. For symmetric matrix $\mathbf{X}$, $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is positive semidefinite.

II. Problem Formulation

The network comprises sensors at unknown positions $\{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^2$, a set of reference sensors (anchors) at known positions $\{a_1, a_2, \ldots, a_l\} \in \mathbb{R}^2$, and unknown target positions $\{e_1, e_2, \ldots, e_m\} \in \mathbb{R}^2$. A central processing node has access to range measurements between target positions and all sensors/anchors, namely,

$$d_{ij} = \|x_i - e_j\| + w_{ij}, \qquad\qquad d_{kj} = \|a_k - e_j\| + w_{kj},$$

where w_{ij} and w_{kj} denote noise terms. A practical system that provides such range measurements is used, e.g., in [1].

a) SLAT Under Gaussian Noise: If disturbances are Gaussian, independent and identically distributed (i.i.d.), then maximizing the likelihood for the full batch of observations is equivalent to minimizing the cost function

$$\Omega_G(\mathbf{x}) = \sum_{i,j} (\|x_i - e_j\| - d_{ij})^2 + \sum_{k,j} (\|a_k - e_j\| - d_{kj})^2. \tag{1}$$

The set of unknown sensor and target positions is concatenated into column vector $\mathbf{x} \in \mathbb{R}^{2(n+m)}$, the argument of Ω_G. The goal of our SLAT approach is to find the set of coordinates in $\mathbf{x}$ which minimizes (1).

b) SLAT Under Laplacian Noise: When the disturbances are Laplacian and i.i.d., thus heavier tailed than Gaussian, maximizing the likelihood amounts to minimizing the cost function

$$\Omega_L(\mathbf{x}) = \sum_{i,j} |\|x_i - e_j\| - d_{ij}| + \sum_{k,j} |\|a_k - e_j\| - d_{kj}|. \tag{2}$$

TABLE I: Summary of the proposed SLAT algorithm

Goal: Given incomplete and inaccurate range measurements, find sensor and target positions which (locally) maximize the likelihood function (1) for Gaussian noise or (2) for Laplacian noise

Startup phase (Batch algorithms)

1) Collect a block of range measurements for target sightings at times $t = 1, \ldots, T$
2) **Initialization step:** Solve EDM completion problem using (8) or (20)
3) Factorize EDM matrix to get spatial coordinates
4) **Refinement step:** Improve the likelihood of sensor/target positions by iterative MM using (12) or (25)

Updating phase (Time-recursive algorithms)

1) Collect range measurements for a new target sighting at time $t > T$
2) **Initialization step:** Solve source localization problem for new target position using (16) or (35)
3) **Refinement step:** Repeat likelihood refinement as in startup

When compared with (1), the absence of squares in the summation terms of (2) renders the function less sensitive to outlier measurements d_{ij} with large deviations from the true ranges.

Since the Euclidean distance metric in both problem setups is invariant to global rotation, translation, and reflection, so are the functions Ω_G and Ω_L in the absence of anchors.

To remove most of those ambiguities[1] in the solutions, a minimum of $l = 3$ non collinear anchors must be considered. As in many other ML problems, the functions Ω_G and Ω_L are in general nonconvex and multimodal, hence their (approximate) minimization proceeds in two steps: initialization and refinement. The former provides suitable initial points, through EDM completion (startup) or source localization (updating), for target/sensor positions which tend to avoid convergence towards undesirable local minimizers of the ensuing iterative refinement algorithms based on MM or weighted-MM.

In Sections III and IV we develop algorithms for the operations listed in the algorithm overview of Table I under Gaussian and Laplacian noise, respectively.

III. SLAT UNDER GAUSSIAN NOISE

This section develops algorithms for EDM initialization, MM refinement, and time-recursive estimation in SLAT under the assumption that measurement noise is i.i.d. and Gaussian. First, a basic formulation of EDM completion with squared distances is provided to form the basis for the initialization methods described in Sections III-B and IV-A.

[1]Some geometrical configurations for sensor and target positions have intrinsic rotation/reflection ambiguities for range-based localization that cannot be resolved by anchors.

A. EDM with Squared Distances

The basic EDM completion problem, described below, operates on squared ranges [21], [22]. Even though it is not matched to the likelihood function (1), it is useful for benchmarking in Section V, as its performance is representative of other popular SNL methods [6], [18] and the SLAT approach of [15].

A partial pre-distance matrix $\mathbf{D}$ is a matrix with zero diagonal entries and with certain nonnegative elements equal to the squares of observed distances, $D_{ij} = d_{ij}^2$. The remaining elements are considered free. The nearest EDM problem is to find an EDM $\mathbf{E}$ that is nearest in the least-squares sense to matrix $\mathbf{D}$, when the free variables are not considered and the elements of $\mathbf{E}$ satisfy $E_{ij} = \|y_i - y_j\|^2$ for a set of points y_i. The geometry and properties of EDM (a convex cone) have been extensively studied in the literature [21], [22]. The nearest EDM problem in 2D space is formulated as

$$\begin{aligned} \underset{\mathbf{E}}{\text{minimize}} \quad & \textstyle\sum_{i,j\in\mathcal{O}}(E_{ij} - d_{ij}^2)^2 \\ \text{subject to} \quad & \mathbf{E} \in \mathcal{E} \\ & \mathbf{E}(\mathcal{A}) = \mathbf{A} \\ & \text{rank}(\mathbf{JEJ}) = 2, \end{aligned} \tag{3}$$

where

$$\mathbf{J} = \big(\mathbf{I}_\rho - \frac{1}{\rho}\mathbf{1}_\rho \mathbf{1}_\rho{}^T\big), \quad \rho = m + n + l,$$

is a centering operator which subtracts the mean of a vector from each of its components. In (3), $\mathcal{O}$ is the index set for which range measurements are available. The constraint $\mathbf{E}(\mathcal{A}) = \mathbf{A}$, where $\mathcal{A}$ is the index set of anchor/anchor distances and $\mathbf{A}_{ij} = \|a_i - a_j\|^2$ is the corresponding EDM submatrix, enforces the known *a priori* spatial information. Matrix $\mathbf{E}$ belongs to the EDM cone $\mathcal{E}$ if it satisfies the properties

$$E_{ii} = 0, \qquad E_{ij} \geq 0, \qquad -\mathbf{JEJ} \succeq 0. \tag{4}$$

The rank constraint in (3) ensures that the solution is compatible with a constellation of sensor/anchor/target points in $\mathbb{R}^2$. Extraction of the set y_i from $\mathbf{E}$ is described below. Problem (3) is also known as the penalty function approximation [21] due to the form of the cost function $\varphi_1(\mathbf{E}) = \sum_{i,j}(E_{ij} - d_{ij}^2)^2$. By expressing (3) in terms of full matrices and dropping the rank constraint, a compact relaxed SDP formulation is obtained as

$$\begin{aligned} \underset{\mathbf{E}}{\text{minimize}} \quad & \|\mathbf{W} \odot (\mathbf{E} - \mathbf{D})\|_F^2 \\ \text{subject to} \quad & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A}, \end{aligned} \tag{5}$$

where $\mathbf{W}$ is a mask matrix with zeros in the entries corresponding to free elements of $D_{ij} = d_{ij}^2$ and ones elsewhere. When combined with the Hadamard product $\odot$, the Frobenius norm $\|.\|_F$ replaces the summation in (3) over the observed index set $\mathcal{O}$. From here on, we will call this method EDM with squared ranges (EDM-SR).

B. Startup Initialization: EDM with Plain Distances

Instead of trying to match squared distances, we can apply EDM completion to plain distances as

$$\begin{aligned} \underset{\mathbf{E}}{\text{minimize}} \quad & \textstyle\sum_{i,j}(\sqrt{E_{ij}} - d_{ij})^2 \\ \text{subject to} \quad & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A} \\ & \text{rank}(\mathbf{JEJ}) = 2. \end{aligned} \tag{6}$$

For this method the penalty function is $\varphi_2(\mathbf{E}) = \sum_{i,j}(\sqrt{E_{ij}} - d_{ij})^2$, which more closely resembles the terms in the likelihood function (1), and (6) is thus expected to inherit some of the robustness properties of ML estimation. Expanding the objective function in (6) results in

$$\begin{aligned} \underset{\mathbf{E}}{\text{minimize}} \quad & \textstyle\sum_{i,j}(E_{ij} - 2\sqrt{E_{ij}}d_{ij} + d_{ij}^2) \\ \text{subject to} \quad & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A} \\ & \text{rank}(\mathbf{JEJ}) = 2. \end{aligned} \tag{7}$$

A relaxed SDP is obtained by introducing an epigraph-like variable $\mathbf{T}$ and dropping the rank constraint

$$\begin{aligned} \underset{\mathbf{E},\mathbf{T}}{\text{minimize}} \quad & \textstyle\sum_{i,j}(E_{ij} - 2T_{ij}d_{ij}) \\ \text{subject to} \quad & T_{ij}^2 \leq E_{ij} \\ & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A}. \end{aligned} \tag{8}$$

From here on, we will call this method EDM with plain ranges (EDM-R).

Note that the solutions of the initialization techniques described here and in Sections III-A and IV-A are distance matrices. Detailed explanations of how to estimate the spatial coordinates of the sensors and target positions from EDM and the usage of anchors are given in [15]. The basic idea is to use a linear transformation to obtain the Gram matrix $\mathbf{Y}^T\mathbf{Y}$ from the EDM matrix $\mathbf{E}$, from which spatial coordinates $\mathbf{Y}$ are extracted by singular value decomposition (SVD) up to a unitary matrix. The anchors are then used to estimate the residual unitary matrix by solving a Procrustes problem. As discussed in [15], observation noise can significantly disrupt the estimated sensor/target coordinates through EDM

completion and rank truncation, and it was found that much more accurate results are obtained by using those as a starting point for likelihood maximization. Next, MM algorithms are proposed for iterative likelihood maximization.

C. Refinement Steps: Majorization-Minimization

The key idea of MM is to find, at a certain point $\mathbf{x}^t$, a simpler function that has the same function value at $\mathbf{x}^t$ and anywhere else is larger than or equal to the objective function to be minimized. Such a function is called a majorization function. By minimizing the majorization function we obtain the next point of the algorithm, while decreasing the cost function [14]. The detailed derivation of MM is given in [15], and is summarized below for the reader's convenience. Define two convex functions as

$$f_{ij}(\mathbf{x}) = \|x_i - e_j\|, \qquad\qquad g_{kj}(\mathbf{x}) = \|a_k - e_j\|. \tag{9}$$

Expanding f and g in (1) and using first-order conditions on convexity [21],

$$\begin{aligned}\Omega_G(\mathbf{x}) \leq &\sum_{i,j} \left(f_{ij}^2(\mathbf{x}) - 2d_{ij}\big(f_{ij}(\mathbf{x}^t) + \langle \nabla f_{ij}(\mathbf{x}^t), (\mathbf{x}-\mathbf{x}^t)\rangle\big) + d_{ij}^2\right) \\ &+ \sum_{k,j} \left(g_{kj}^2(\mathbf{x}) - 2d_{kj}\big(g_{kj}(\mathbf{x}^t) + \langle \nabla g_{kj}(\mathbf{x}^t), (\mathbf{x}-\mathbf{x}^t)\rangle\big) + d_{kj}^2\right),\end{aligned} \tag{10}$$

where $\langle u, v\rangle = u^T v$, we get the proposed majorization function on the right side of (10), which is quadratic in $\mathbf{x}$ and easily minimized. The MM iteration

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \sum_{i,j} \left(f_{ij}^2(\mathbf{x}) - 2d_{ij}\langle \nabla f_{ij}(\mathbf{x}^t), \mathbf{x}\rangle\right) + \sum_{k,j} \left(g_{kj}^2(\mathbf{x}) - 2d_{kj}\langle \nabla g_{kj}(\mathbf{x}^t), \mathbf{x}\rangle\right) \tag{11}$$

turns out to be obtained as the solution of the following linear system of equations

$$\left[\sum_{i,j} \mathbf{M}_{ij}^T \mathbf{M}_{ij} + \sum_{k,j} \mathbf{N}_j^T \mathbf{N}_j\right] \mathbf{x}^{t+1} = \sum_{i,j} d_{ij} \nabla f_{ij}(\mathbf{x}^t) + \sum_{k,j} d_{kj} \nabla g_{kj}(\mathbf{x}^t) - \sum_{k,j} \mathbf{N}_j a_k^T, \tag{12}$$

where the selection matrices $\mathbf{M}_{ij}$ and $\mathbf{N}_j$ are defined in Appendix C and

$$\nabla f_{ij}(\mathbf{x}^t) = \frac{\mathbf{M}_{ij}^T \mathbf{M}_{ij} \mathbf{x}^t}{\|\mathbf{M}_{ij}\mathbf{x}^t\|}, \qquad\qquad \nabla g_{kj}(\mathbf{x}^t) = \frac{\mathbf{N}_j^T (a_k + \mathbf{N}_j \mathbf{x}^t)}{\|a_k + \mathbf{N}_j \mathbf{x}^t\|}.$$

D. Updating Initialization: Recursive Position Estimation using SLCP

Suppose that a batch of observations have been processed and a new target position is to be estimated. We could repeat MM refinement with EDM-R initialization acting on an expanded batch that concatenates all previous range measurements and those for the new target sighting. However, this would be computationally expensive due to the EDM completion step. Also, previously estimated positions would

be ignored and could not contribute to computational complexity reduction. To alleviate the load we propose a simple methodology to obtain a good initial point for MM which avoids the EDM step. This consists of fixing the previous positions at their estimated values and only estimating the new target position. More precisely, we minimize

$$\Psi_G(y) = \sum_{i=1}^{n+l} (\|b_i - y\| - d_i)^2, \tag{13}$$

where y is the new target position, b_i denotes the previously estimated position of a sensor or anchor, and d_i is the corresponding range measurement. We propose SLCP [19] to minimize (13), and briefly summarize the method below.

Minimization of (13) can be expressed equivalently as (see Appendix D)

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i=1}^{n+l} \|y - y_i\|^2 \\ y, y_i & \\ \text{subject to} \quad & \|b_i - y_i\| = d_i \quad i = 1, \ldots, n+l. \end{aligned} \tag{14}$$

Geometrically, y is a point that minimizes the sum of squared distances to a set of circles with center b_i and radius d_i. If we fix y_i, (14) becomes an unconstrained optimization problem in y whose solution is readily obtained as the center of mass of the constellation. The reduced problem can be compactly described in the complex plane as

$$\begin{aligned} \text{minimize} \quad & \|\tfrac{1}{n+l}\mathbf{1}\mathbf{1}^T\mathbf{y} - \mathbf{y}\|^2 \\ \mathbf{y}, \boldsymbol{\theta} & \\ \text{subject to} \quad & \mathbf{y} = \mathbf{b} + \mathbf{R}\boldsymbol{\theta}, \end{aligned} \tag{15}$$

where $\boldsymbol{\theta} = \begin{bmatrix} e^{j\phi_1} & \ldots & e^{j\phi_{n+l}} \end{bmatrix}^T \in \mathbb{C}^{n+l}$, $\mathbf{R} = \text{diag}(d_1, \ldots, d_{n+l}) \in \mathbb{R}^{(n+l)\times(n+l)}$, and $\mathbf{b} \in \mathbb{C}^{n+l}$ holds the 2D coordinates $b_1, \ldots, b_{n+l}$, expressed as complex numbers.

The primary goal of SLCP is to find the vector of phases $\boldsymbol{\theta}$. To this end, the cost function in (15) is first expanded as a quadratic form in $\boldsymbol{\theta}$ under the constraints $|\theta_i| = 1$, and the non-constant terms then manipulated into a sum of complex magnitudes $-2|\mathbf{c}^H\boldsymbol{\theta}| + \frac{1}{n+l}|\mathbf{d}^T\boldsymbol{\theta}|^2$ for constant $\mathbf{c}$, $\mathbf{d}$. A relaxed SDP formulation is obtained by introducing the new variable $\boldsymbol{\Phi} = \boldsymbol{\theta}\boldsymbol{\theta}^H$ and neglecting the associated rank-1 constraint. In standard epigraph form (variable t) this is written as

$$\begin{aligned} \text{maximize} \quad & t + \tfrac{1}{n+l}\mathbf{d}^T\boldsymbol{\Phi}\mathbf{d} \\ \boldsymbol{\Phi}, t & \\ \text{subject to} \quad & \boldsymbol{\Phi} \succeq 0, \;\; \phi_{ii} = 1 \\ & 4\mathbf{c}^H\boldsymbol{\Phi}\mathbf{c} \geq t^2. \end{aligned} \tag{16}$$

The solution of the optimization problem (16) is a positive semidefinite matrix from which the target coordinates are estimated by SVD as described in [15]. Although $\mathbf{\Phi}$ is not guaranteed to have rank 1, in practice it very often has a clearly dominating singular value.

After an optimal target position is obtained, we return to the cost function (1) and iteratively refine all estimates using (12). In SNL this incremental procedure could also be applied as new sensors become available.

In a previous paper [15], the source localization method derived in [23], termed Squared-Range Least Squares (SR-LS), was proposed for initialization during the updating phase. Note that in [23] *squared* distances are matched, leading to a Trust Region optimization problem. However, as demonstrated in [19], SLCP is a more accurate source localization method and its cost function (13) is better matched to the likelihood function (1). This makes it more convenient for initialization of iterative refinement algorithms, which will then require fewer iterations to converge and/or will be less likely to get trapped in undesirable local extrema.

IV. SLAT UNDER LAPLACIAN NOISE

A. Startup Initialization: EDM with Ranges and ℓ_1-norm

Among the penalty function approximation methods, the ℓ_1-norm approximation is known to be robust to outliers [21]. Therefore, the penalty function of the third SLAT startup initialization method is chosen as $\varphi_3(\mathbf{E}) = \sum_{i,j}|\sqrt{E_{ij}} - d_{ij}|$, and the associated optimization problem becomes

$$\begin{aligned} \underset{\mathbf{E}}{\text{minimize}} \quad & \textstyle\sum_{i,j}|\sqrt{E_{ij}} - d_{ij}| \\ \text{subject to} \quad & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A} \\ & \text{rank}(\mathbf{JEJ}) = 2. \end{aligned} \tag{17}$$

The terms $|\sqrt{E_{ij}} - d_{ij}|$ in the objective function for this problem are convex when $\sqrt{E_{ij}} - d_{ij} < 0$, but concave for $\sqrt{E_{ij}} - d_{ij} > 0$. To obtain a convex approximation each of those terms is replaced by a linear approximation

$$a_{ij}E_{ij} + b_{ij}, \qquad a_{ij} = \frac{1}{\sqrt{E_{\max}} + d_{ij}}, \qquad b_{ij} = -\frac{d_{ij}^2}{\sqrt{E_{\max}} + d_{ij}} \tag{18}$$

in part of the domain where it is concave, as shown in Figure 1. The two functions coincide for $E_{ij} = d_{ij}^2$ and $E_{ij} = E_{\max}$, where the constant $E_{\max}$ is a practical upper bound on (squared) range measurements. Thus, we replace $|\sqrt{E_{ij}} - d_{ij}|$ by its convex envelope $\max\{d_{ij} - \sqrt{E_{ij}}, a_{ij}E_{ij} + b_{ij}\}$ and use the epigraph

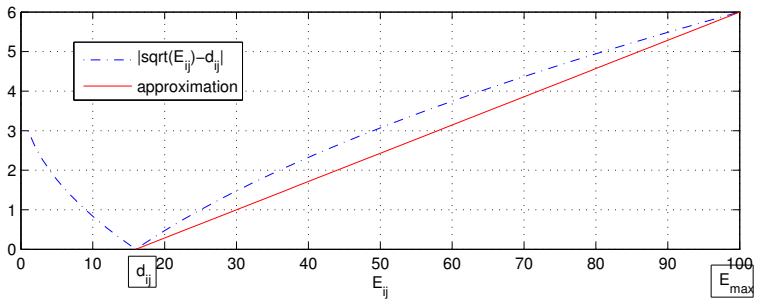


Fig. 1: The value of $|\sqrt{E_{ij}} - d_{ij}|$ vs E_{ij}, and the linear approximation of the concave part.

variable $\mathbf{T}$ to obtain

$$\begin{aligned} \underset{\mathbf{E},\,\mathbf{T}}{\text{minimize}} \quad & \textstyle\sum_{i,j} T_{ij} \\ \text{subject to} \quad & \max\{d_{ij} - \sqrt{E_{ij}}, a_{ij}E_{ij} + b_{ij}\} \leq T_{ij} \\ & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A} \\ & \text{rank}(\mathbf{JEJ}) = 2. \end{aligned} \tag{19}$$

A relaxation of (19) after dropping the rank constraint is

$$\begin{aligned} \underset{\mathbf{E},\,\mathbf{T}}{\text{minimize}} \quad & \textstyle\sum_{i,j} T_{ij} \\ \text{subject to} \quad & (d_{ij} - T_{ij})^2 \leq E_{ij} \\ & a_{ij}E_{ij} + b_{ij} \leq T_{ij} \\ & \mathbf{E} \in \mathcal{E},\ \mathbf{E}(\mathcal{A}) = \mathbf{A}. \end{aligned} \tag{20}$$

Note that the first constraint in (20) is not equivalent to $d_{ij} - \sqrt{E_{ij}} \leq T_{ij}$, but rather to $-\sqrt{E_{ij}} \leq d_{ij} - T_{ij} \leq \sqrt{E_{ij}}$, which amounts to intersecting the original epigraph with the parabolic hypograph $d_{ij} + \sqrt{E_{ij}} \geq T_{ij}$. This preserves the convexity of the feasible set and does not change its lower boundary for $E_{ij} \in [0,\, E_{\text{max}}]$, where the optimal point will be found. The constraint can now be readily expressed in standard form without introducing additional variables, e.g., as an LMI or a second-order cone constraint [24]

$$\begin{bmatrix} 1 & d_{ij} - T_{ij} \\ d_{ij} - T_{ij} & E_{ij} \end{bmatrix} \succeq 0 \qquad \text{or} \qquad \left\| \begin{matrix} 2(d_{ij} - T_{ij}) \\ E_{ij} - 1 \end{matrix} \right\| \leq E_{ij} + 1. \tag{21}$$

This technique will be called EDM with ranges and ℓ_1-norm (EDM-R-ℓ_1).

B. Refinement Steps: Weighted Majorization Minimization

Robustness to outliers in the cost function (2) for Laplacian noise is gained at the expense of differentiability. To circumvent this shortcoming we resort to the well-known re-weighted least squares approach [25], which replaces the minimization of (2) with a sequence of minimizations of smooth approximation functions that converge to $\Omega_L(\mathbf{x})$. Specifically, (2) is first written as

$$\Omega_L(\mathbf{x}) = \sum_{i,j} u_{ij}(\|x_i - e_j\| - d_{ij})^2 + \sum_{k,j} v_{kj}(\|a_k - e_j\| - d_{kj})^2, \tag{22}$$

with

$$u_{ij} = \frac{1}{|\|x_i - e_j\| - d_{ij}|}, \qquad\qquad v_{kj} = \frac{1}{|\|a_k - e_j\| - d_{kj}|}.$$

At time t the minimization function becomes $\Omega_L^t(\mathbf{x})$, which has the same form of (22) but the *functions* u_{ij}, v_{kj} above are now replaced by *constants* based on the estimated positions after the previous iteration

$$u_{ij}^t = \frac{1}{|\|x_i^t - e_j^t\| - d_{ij}|}, \qquad\qquad v_{kj}^t = \frac{1}{|\|a_k - e_j^t\| - d_{kj}|}. \tag{23}$$

An inner optimization loop could now be used to minimize $\Omega_L^t(\mathbf{x})$ for every t but, as shown in Appendix A, a single iteration suffices to ensure convergence. With fixed u_{ij}^t, v_{kj}^t the same majorization technique of Section III-C yields the weighted-MM iteration

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \sum_{i,j} u_{ij}^t \left(f_{ij}^2(\mathbf{x}) - 2d_{ij}\langle \nabla f_{ij}(\mathbf{x}^t), \mathbf{x}\rangle\right) + \sum_{k,j} v_{kj}^t \left(g_{kj}^2(\mathbf{x}) - 2d_{kj}\langle \nabla g_{kj}(\mathbf{x}^t), \mathbf{x}\rangle\right). \tag{24}$$

Thus, the new point is obtained by solving the linear system

$$\left[\sum_{i,j} u_{ij}^t \mathbf{M}_{ij}^T \mathbf{M}_{ij} + \sum_{k,j} v_{kj}^t \mathbf{N}_j^T \mathbf{N}_j\right] \mathbf{x}^{t+1} = \sum_{i,j} u_{ij}^t d_{ij} \nabla f_{ij}(\mathbf{x}^t) + \sum_{k,j} v_{kj}^t d_{kj} \nabla g_{kj}(\mathbf{x}^t) - \sum_{k,j} v_{kj}^t \mathbf{N}_j a_k^T, \tag{25}$$

where $\nabla f_{ij}(\mathbf{x}^t)$, $\nabla g_{kj}(\mathbf{x}^t)$, $\mathbf{M}_{ij}$ and $\mathbf{N}_j$ are the same as in (12).

In practice the weights u_{ij}^t and v_{ij}^t must be modified to avoid the possibility of division by zero [26], which in our case is achieved by saturating them at 10^5 when computing (24). Hence, truncating the weights, which are the Huber thresholds, is equivalent to using the Huber function.

C. Updating Initialization: Recursive Position Estimation using SLℓ_1

The ML source localization problem under Laplacian noise is equivalent to

$$\begin{aligned} &\underset{y}{\text{minimize}} \quad \Psi_L(y) = \textstyle\sum_{i=1}^{n+l} |\|y - b_i\| - d_i| \end{aligned} \tag{26}$$

or

$$\begin{array}{ll} \text{minimize} & (\sum_{i=1}^{n+l} |\|y-b_i\|-d_i|)^2, \\ y & \end{array} \tag{27}$$

where y, b_i and d_i are defined in Section III-D. We use ideas from [27] to express the minimization of Ψ_L^2 as a weighted sum of squares.

Lemma IV-C.1. *The following problem is equivalent to* (27)

$$\begin{array}{lll} \text{minimize} & \text{minimize} & \sum_{i=1}^{n+l} \frac{(\|y-b_i\|-d_i)^2}{\lambda_i} \\ y & \boldsymbol{\lambda} \in \mathbb{R}^{n+l} & \\ & \text{subject to} & \lambda_i > 0,\ \mathbf{1}^T\boldsymbol{\lambda} = 1. \end{array} \tag{28}$$

A proof is given in Appendix B. As claimed in Section III-D and shown in Appendix D, the difference between the true range and observed range is actually equivalent to the distance between the source position and the closest point on the circle with center b_i and radius d_i. An equivalent formulation is therefore

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n+l} \frac{\|y-y_i\|^2}{\lambda_i} \\ y, y_i, \boldsymbol{\lambda} & \\ \text{subject to} & \|y_i - b_i\| = d_i \\ & \lambda_i > 0,\ \mathbf{1}^T\boldsymbol{\lambda} = 1. \end{array} \tag{29}$$

If we fix the y_i and λ_i, the solution of (29) with respect to y is an unconstrained optimization problem whose solution is readily obtained by invoking the optimality conditions

$$\sum_{i=1}^{n+l} \frac{(y-y_i)}{\lambda_i} = 0 \ \Rightarrow\ y^* = \frac{\sum_{i=1}^{n+l} \frac{y_i}{\lambda_i}}{\sum_{i=1}^{n+l} \frac{1}{\lambda_i}}. \tag{30}$$

Geometrically, the first constraint of (29) defines circle equations, which can be compactly described in the complex plane as $y_i = b_i + d_i e^{j\phi_i}$. We collect these into a vector $\mathbf{y} = \mathbf{b} + \mathbf{R}\mathbf{u}$, where $\mathbf{b}$ and $\mathbf{R}$ are defined as in Section III-D and $\mathbf{u} = \begin{bmatrix} e^{j\phi_1} & \dots & e^{j\phi_{n+l}} \end{bmatrix}^T \in \mathbb{C}^{n+l}$. Using the optimal y, we get

$$\begin{array}{ll} \text{minimize} & \mathbf{y}^H\boldsymbol{\Pi}\mathbf{y} = (\mathbf{b}+\mathbf{R}\mathbf{u})^H\boldsymbol{\Pi}(\mathbf{b}+\mathbf{R}\mathbf{u}) \\ \boldsymbol{\lambda}, \mathbf{u} & \\ \text{subject to} & \lambda_i > 0,\ \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ & |u_i| = 1, \end{array} \tag{31}$$

where

$$\begin{aligned}\mathbf{\Pi} &= \begin{bmatrix} \frac{1}{\lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\lambda_{n+l}} \end{bmatrix} - \frac{1}{\sum_{i=1}^{n+l} \frac{1}{\lambda_i}} \begin{bmatrix} \frac{1}{\lambda_1} \\ \vdots \\ \frac{1}{\lambda_{n+l}} \end{bmatrix} \begin{bmatrix} \frac{1}{\lambda_1} & \cdots & \frac{1}{\lambda_{n+l}} \end{bmatrix} \\ &= \mathbf{\Lambda}^{-1} - \mathbf{\Lambda}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{\Lambda}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{\Lambda}^{-1},\end{aligned} \tag{32}$$

with $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_{n+l})$.

Matrix $\mathbf{\Pi}$ resembles an orthogonal projector. Using the matrix inversion lemma[2], it is seen to be the limiting case $\mathbf{\Pi} = \lim_{\sigma\to\infty}(\mathbf{\Lambda} + \sigma\mathbf{1}\mathbf{1}^T)^{-1}$ and thus positive semidefinite. This format is more amenable to analytic manipulations in optimization problems and will be used throughout this paper. The parameter σ is taken as a sufficiently large constant (see Appendix B), although it could also be regarded as an additional optimization variable to ensure adequate approximation accuracy.

We now introduce an epigraph variable $t \in \mathbb{R}$ in (31), i.e., we minimize over t and add the constraint $t - (\mathbf{b} + \mathbf{R}\mathbf{u})^H\mathbf{\Pi}(\mathbf{b} + \mathbf{R}\mathbf{u}) \geq 0$. Applying Schur complements the constraint may be successively written as

$$\begin{bmatrix} t & (\mathbf{b}+\mathbf{R}\mathbf{u})^H \\ \mathbf{b}+\mathbf{R}\mathbf{u} & \mathbf{\Pi}^{-1} \end{bmatrix} \succeq 0 \qquad \Leftrightarrow \qquad \mathbf{\Pi}^{-1} - \frac{(\mathbf{b}+\mathbf{R}\mathbf{u})(\mathbf{b}+\mathbf{R}\mathbf{u})^H}{t} \succeq 0. \tag{33}$$

The formulation becomes

$$\begin{aligned} \text{minimize} \quad & t \\ \boldsymbol{\lambda}, \mathbf{u}, t & \\ \text{subject to} \quad & \lambda_i > 0,\ \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ & |u_i| = 1 \\ & t\mathbf{\Lambda} + t\sigma\mathbf{1}\mathbf{1}^T \succeq (\mathbf{b}+\mathbf{R}\mathbf{u})(\mathbf{b}+\mathbf{R}\mathbf{u})^H. \end{aligned} \tag{34}$$

Finally, we define $\mathbf{B} = [\mathbf{b}\ \ \mathbf{R}]$, $\mathbf{v}^H = [1\ \ \mathbf{u}^H]$, $\mathbf{V} = \mathbf{v}\mathbf{v}^H$, and we drop the rank-1 constraint on the new variable $\mathbf{V}$ to obtain the relaxed SDP

$$\begin{aligned} \text{minimize} \quad & t \\ \boldsymbol{\beta}, \mathbf{V}, t & \\ \text{subject to} \quad & \beta_i > 0,\ \mathbf{1}^T\boldsymbol{\beta} = t \\ & V_{ii} = 1, \mathbf{V} \succeq 0 \\ & \text{diag}(\boldsymbol{\beta}) + t\sigma\mathbf{1}\mathbf{1}^T \succeq \mathbf{B}\mathbf{V}\mathbf{B}^H. \end{aligned} \tag{35}$$

[2] $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$.

The solution of the optimization problem (35) includes the positive semidefinite matrix $\mathbf{V}$ from whose first row or column $\mathbf{u}$ can be extracted directly[3] to obtain $\mathbf{y} = \mathbf{b} + \mathbf{R}\mathbf{u}$ and the target coordinates from (30).

As in Section III-D, after an optimal target position is obtained we return to the cost function (2) and iteratively refine all the estimates using (25). Section V demonstrates in simulation that SLℓ_1 is more robust to outliers than the SLCP algorithm of Section III-D, as its cost function (26) is better matched to the likelihood function (2).

V. Numerical Results

Example 1 [Comparison of Initialization Methods for the Startup Phase (EDM Completion)]: To investigate the accuracy of the methods, we set a physical scenario containing four anchors, five unknown sensors, and six target positions in a $[0, 2] \times [0, 2]$ area. Range measurements are corrupted by additive spatio-temporally white noise with standard deviation $\sigma_{\text{gaussian}} \in [0.005, 0.03]$. This noisy observation model may lead to near-zero or negative range measurements, in which case we follow normal practice [6] and set them equal to a small positive constant (10^{-5} in our simulations). With the chosen noise variances this occurs sufficiently seldom (up to 0.04% of measurements) for its impact on estimation accuracy to be unimportant. Several algorithms are tested (EDM-SR, EDM-R, EDM-R-l_1, MM initialized by EDM-SR (EDM-SR+MM), MM initialized by EDM-R (EDM-R+MM) and MM initialized by EDM-R-l_1 (EDM-R-l_1+MM)), and their performances are compared according to the total root mean square error (RMSE)

$$\sqrt{\frac{1}{K}\frac{1}{n+m}\sum_{k=1}^{K}\sum_{i=1}^{n+m}\|x_i - \hat{x}_i^k\|^2}, \tag{36}$$

where $\hat{x}_i^k$ denotes the i-th estimated sensor or target position in the k-th Monte Carlo run for the specific noise realization. In each of $K = 150$ Monte Carlo runs, a random network is generated according to the physical scenario described above. To assess the fundamental hardness of position estimation, error plots for Gaussian noise also show the total Cramér-Rao Lower Bound (CRLB), calculated as

$$\sqrt{\frac{1}{K}\frac{1}{n+m}\sum_{k=1}^{K}\text{trace}(\text{CRLB}_k)} \tag{37}$$

for each noise variance, where CRLB_k denotes the k-th diagonal element of the matrix lower bound. The CRLB for anchored and anchor-free localization using ranging information has been studied in

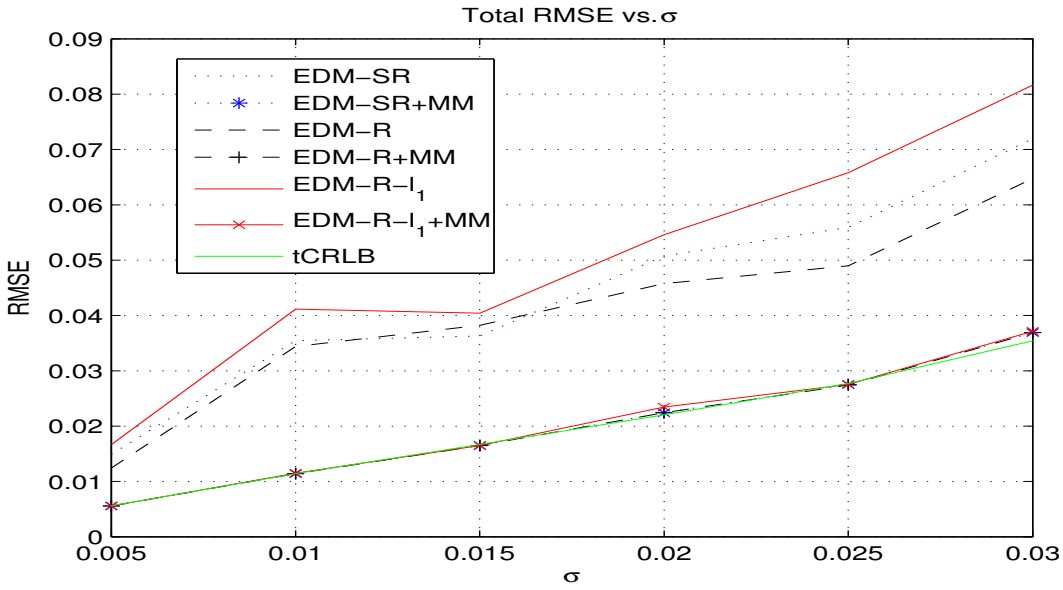
[3]Alternatively, $\mathbf{u}$ can be obtained by rank 1 factorization of the lower right submatrix of $\mathbf{V}$ corresponding to $\mathbf{u}\mathbf{u}^H$, as in [6], [18].

[28]–[30] for different variance models of range estimation noise. For convenience, the CRLB for our SLAT problem under Gaussian noise is rederived in Appendix C in terms of the notation adopted in this paper. We do not prove the unbiasedness of our estimators, a mathematically challenging endeavor that would be required to fully justify benchmarking against the CRLB. In our experimental results, however, we found no clear evidence of bias for small noise levels, where convergence to undesirable extrema of the cost functions is avoided. Figure 2a shows that plain EDM-R has better accuracy than EDM-SR and EDM-R-l_1, although the performance gap closes after iterative refinement by MM. Moreover, MM initialized by the various methods nearly touches the CRLB except when the noise variance is large.

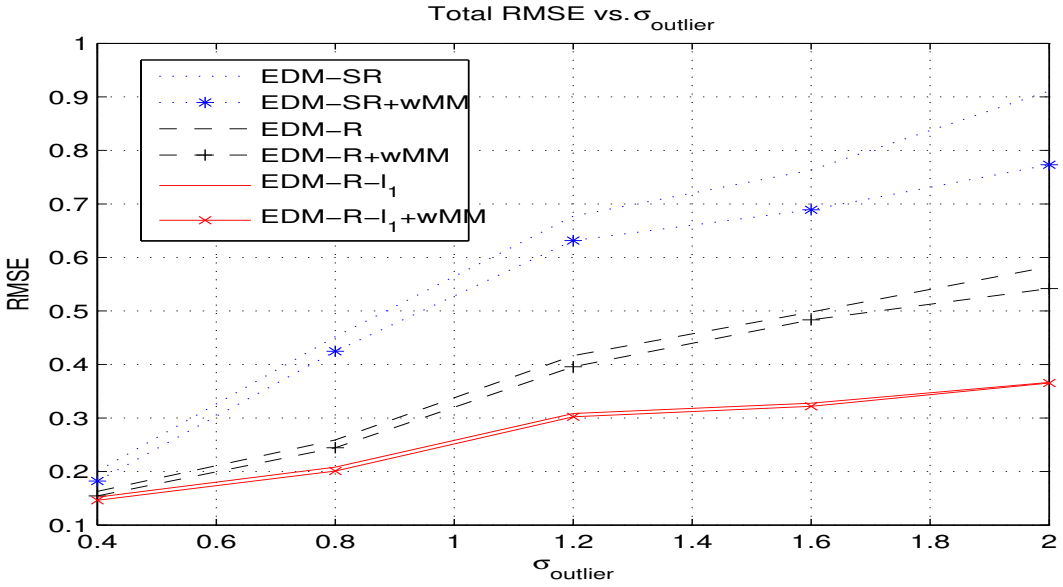
To compare the total RMSE of the algorithms in the presence of outliers, modified range measurements are created according to a "selective Gaussian" model $d_i = \|\cdot\| + w_i + |\epsilon_i|$, where ϵ_i is a white Gaussian noise term with standard deviation $\sigma_{\text{outlier}} \in [0.4, 2]$. The disturbance ϵ_i randomly affects only two range measurements, whereas w_i with $\sigma_{\text{gaussian}} = 0.01$ is present in all observations. This outlier generating model deviates from the earlier Laplacian asssumption, but it is arguably representative of observed range measurements in practical systems [8]. Numerical results under a pure Laplacian model will be presented in Examples 3 and 4. In the presence of high noise and/or outliers, Fig. 2b shows that weighted-MM refinement does not close the performance gap between EDM-R-l_1, EDM-R and EDM-SR initialization because in the latter cases the algorithms converge more often to local minima, thus producing a larger total RMSE.

Example 2 [Uncertainty Ellipsoids]: To further examine the accuracy of MM and weighted-MM with different initialization methods in the startup phase, we randomly generated two networks of 10 sensors, 4 anchors and 11 target positions. 100 Monte Carlo runs were used to find the mean and (1σ) uncertainity ellipsoids of the positions estimated by the methods. The mean and uncertainity ellipsoids for $\sigma_{\text{gaussian}} = 0.025$ and $\sigma_{\text{gaussian}} = 0.02/\sigma_{\text{outlier}} = 0.8$ are shown in Figs. 3 and 4, respectively. Again, outliers are randomly added to two range measurements in Fig. 4.

Without outliers (Fig. 3) using EDM-SR, EDM-R, or EDM-R-l_1 as an initialization to MM makes the uncertainity ellipsoids shrink dramatically after refinement, yielding very similar means and covariances. These are only displayed in the detail view of Fig. 3b, as they are too small to be shown in Fig. 3a. In the presence of outliers (Fig. 4), the uncertainity ellipsoids of EDM-SR+wMM are bigger than for other methods and the means of the estimated positions are shifted. Since EDM-R-l_1 and EDM-R initializations converge to global extrema most of the time, the means of the positions estimated by weighted MM still approach the true positions and their uncertainity ellipsoids are much smaller than for EDM-SR+wMM. In the presence of outliers this example shows that EDM-R-l_1+wMM is clearly superior to the other



(a) No outliers



(b) Selective Gaussian outliers, $\sigma_{\text{gaussian}} = 0.01$

Fig. 2: Comparison of initialization and refinement methods in the startup phase of SLAT.

TABLE II: RMSE of SLℓ_1, SLCP, and SR-LS under Gaussian noise.

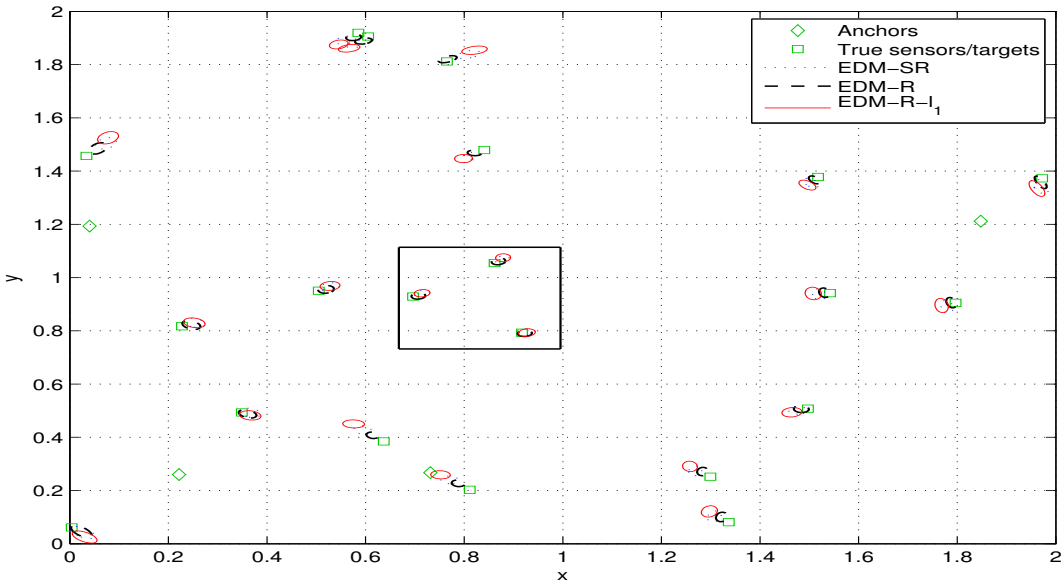
σ_{gaussian}	SLℓ_1	SLCP	SR-LS
1e-3	1.5e-3	1.3e-3	1.8e-3
1e-2	1.31e-2	1.1e-2	1.32e-2
1e-1	0.1608	0.1247	0.1479
1	1.3778	1.2672	1.5351

methods.

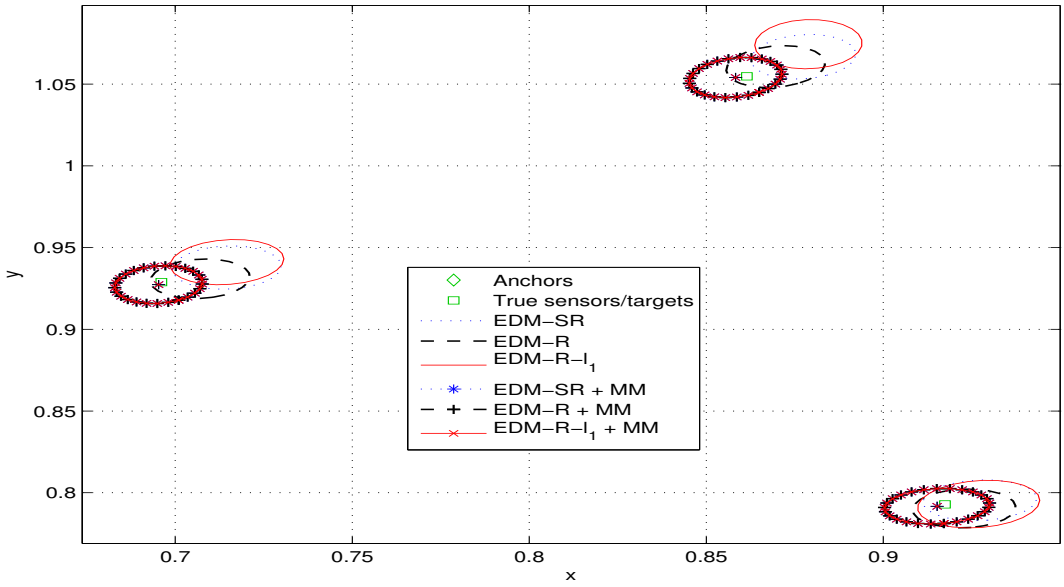
A Note on Practical Computational Complexity of EDM Initialization: Our experiments were conducted on a machine with an Intel Xeon 2.93 GHz Quad-Core CPU and 8 GB of RAM, using Matlab 7.1, CVX 1.2 and Yalmip 3/SeDuMi 1.1 as a general-purpose SDP solver. CPU times are similar for EDM-SR, EDM-R and EDM-R-ℓ_1, under 5 seconds for the example described above with $n = 25$ unknown positions and empirically increasing with $n^{4.5}$ for larger values of n (< 100). This gives a notion of what network sizes are currently practical for the EDM initialization methods, while keeping in mind that CPU times are known to be unreliable surrogates for intrinsic computational complexity due to dependencies on factors such as machine hardware architecture, operating system, efficiency of numerical libraries, and solver preprocessing. We did not attempt to formulate the EDM completion problems in the most efficient way possible for the SDP solver. For MM-type iterative algorithms, extremely large problem sizes can be efficiently handled using contemporary numerical algorithms and computing platforms. In our experiments each iteration takes up to about 1 millisecond.

Example 3 [Comparison of Initialization Methods for the Updating Phase (Source Localization)]: In this example several source localization algorithms are compared using five anchors. We performed 100 Monte Carlo runs, where in each run the anchor and source positions were randomly generated from a uniform distribution over the square $[-10, 10] \times [-10, 10]$. Table II lists the RMSE of source positions under Gaussian noise, with standard deviations $\sigma_{\text{gaussian}} = 10^{-3}$, 10^{-2}, 10^{-1}, and 1 for the methods SLℓ_1, SLCP, and SR-LS [23]. SLℓ_1 uses $\sigma = 10^6$ for the "projector" $\mathbf{\Pi} = (\mathbf{\Lambda} + \sigma \mathbf{1}\mathbf{1}^T)^{-1}$, which is a very conservative value (see Appendix B).

To compare the algorithms in the presence of outliers, range measurements are created according to $d_i = \| \cdot \| + v_i$, where v_i is a Laplacian noise term with standard deviation $\sigma_{\text{laplace}} \in [0.2, 1.6]$. Results are also presented for the alternative selective Gaussian outlier generation model of Examples 1–2 with $\sigma_{\text{outlier}} \in [0.5, 2]$ and $\sigma_{\text{gaussian}} = 0.04$, where outliers only affect measured ranges between the second

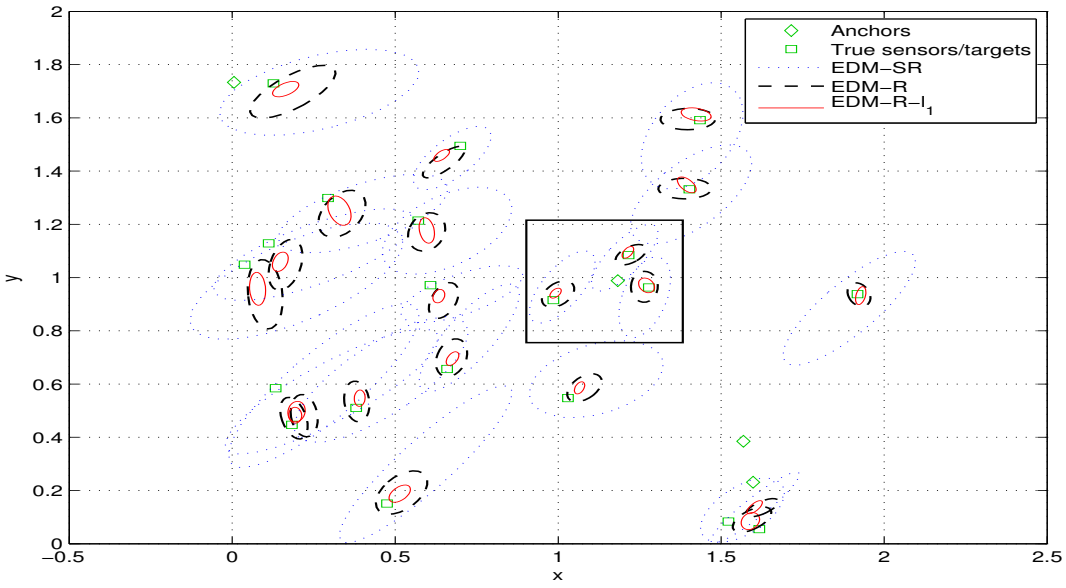


(a) Whole constellation

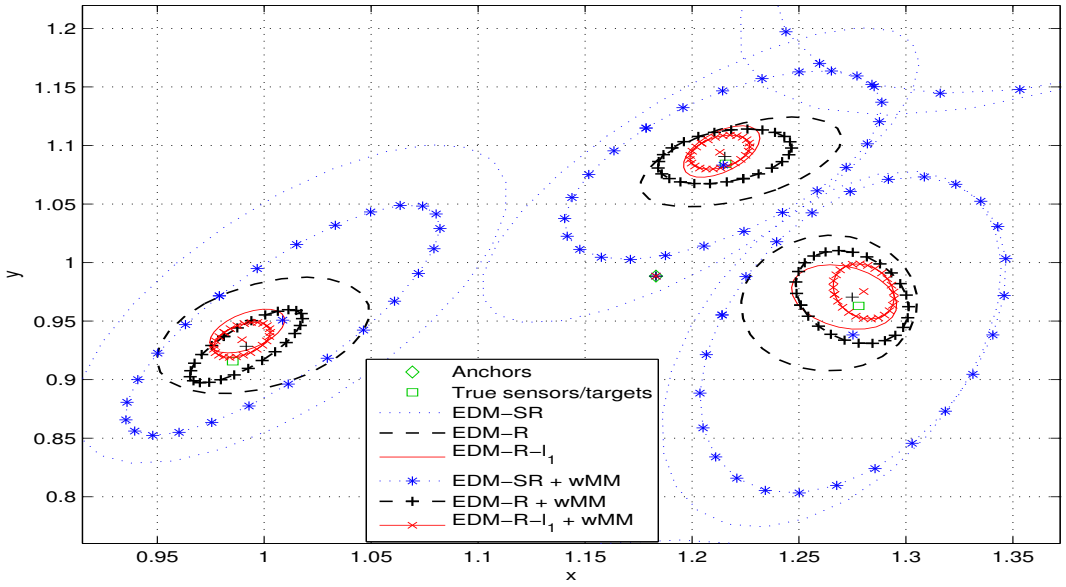


(b) Detail view

Fig. 3: Mean and uncertainity ellipsoids in the startup phase with different initialization methods. No outliers, $\sigma_{\text{gaussian}} = 0.025$.



(a) Whole constellation



(b) Detail view (including an anchor at $(1.18; 0.99)$)

Fig. 4: Mean and uncertainity ellipsoids in the startup phase with different initialization methods. Selective Gaussian outliers, $\sigma_{\text{outlier}} = 0.8/\sigma_{\text{gaussian}} = 0.02$.

TABLE III: RMSE of SLℓ_1, SLCP, SR-LS, and USR-LSO in the presence of outliers.

σ_{laplace}	SLℓ_1	SLCP	SR-LS	USR-LSO
0.2	0.2959	0.3246	0.3433	0.5007
0.4	0.4894	0.5562	0.5637	0.9123
0.8	1.3439	1.4453	1.4481	2.3643
1.6	2.5807	2.7221	3.4486	4.1156

(a) Laplacian noise

σ_{outlier}	SLℓ_1	SLCP	SR-LS	USR-LSO
0.5	0.2352	0.2870	0.3800	0.5125
1.0	0.3773	0.4507	0.5812	0.9686
1.5	0.9427	1.0555	1.2429	2.0687
2.0	1.0471	1.0900	1.3681	2.3707

(b) Selective Gaussian noise

anchor and the source. Table III lists the RMSE of source positions for the algorithms SLℓ_1, SLCP, SR-LS, and USR-LSO (Unconstrained Squared-Range LS with Outliers) [17]. The latter uses basis pursuit methods to detect the presence of outliers, subtracts them from measurements, and then estimates the source position using unconstrained LS operating on squared ranges.

We conclude from Tables II and III that the relative accuracies of SLℓ_1, SLCP, SR-LS, and USR-LSO depend on the data generation model. For Gaussian noise the RMSE of SLℓ_1 and SR-LS are about 20–30% higher than that of SLCP, whereas in the presence of outliers the situation is reversed, and SLℓ_1 becomes more accurate by at least 10%, relative to SLCP. This is because SLCP and SLℓ_1 are better matched with Gaussian and Laplacian modeling assumptions, respectively.

Note that SR-LS was shown in [23] to be more accurate than competing localization methods based on semidefinite relaxation under Gaussian noise. The fact that both SLℓ_1 and SLCP outperform it here under a similar scenario indicates that these algorithms provide state-of-the-art localization accuracy in their class. In the presence of outliers both our algorithms also outperform SR-LS and USR-LSO. The somewhat disappointing results for USR-LSO are due to a combination of factors; the outlier detection method does not always find spurious measurements, particularly under Laplacian noise; subtraction of outliers sometimes fails to produce reasonable range estimates; the simple USR-LS algorithm used to compute the source position was shown in [23] to be less accurate than other methods which retain more constraints of the localization problem. Interestingly, the performance gap between SLCP and SLℓ_1 is actually larger for selective Gaussian outliers, whose generation model does not match the assumptions of SLℓ_1. Similarly to Fig. 2b, the differences in initialization accuracy using SLCP or SLℓ_1 are large enough to prevent closing of the RMSE gap after weighted MM refinement due to convergence to undesirable extrema of the likelihood function.

Example 4 [Global Assessment of the Updating Phase (Time-Recursive Algorithms)]: This example assesses the performance of the full time-recursive procedure (updating phase), comprising SLCP or SLℓ_1 initialization followed by refinement. The network scenario has 16 unknown sensors, 4 anchors and 10 target locations, all randomly positioned. A new target sighting (the 11th one) becomes available and is processed incrementally, i.e., the position is estimated through SLCP or SLℓ_1 by fixing all the remaining ones, then all estimates are jointly refined. Results are benchmarked against refinement with full batch initialization, which makes a fresh start to the process without using any previous knowledge at every new target position to be estimated, solving different and increasingly large EDM completion problems for ML initialization.

This type of incremental approach was used in [15] with the SR-LS algorithm of [23] and MM refinement for Gaussian noise. SLCP is used here instead of SR-LS because, as shown in [19], it increases the convergence speed of subsequent iterative methods and also alleviates the problem of convergence to local extrema of the ML cost function by providing better initial points than SR-LS does. Figure 5 shows the evolution of the Gaussian cost function $\Omega_G(\mathbf{x})$ during refinement after ranges to the 11th target position are sensed ($\sigma_{\text{gaussian}} = 0.04$). The time-recursive (SLCP)+MM approach takes advantage of previously estimated positions to start with a lower cost than batch (EDM-R)+MM, but it reaches the same final error value.

The same network scenario is adopted in the presence of outliers. Figure 6 shows the evolution of cost function $\Omega_L(\mathbf{x})$ during refinement for Laplacian outliers ($\sigma_{\text{laplacian}} = 0.1$), whose behavior is similar to the Gaussian case of Fig. 5. In both Gaussian and Laplacian settings refinement yields similar accuracy and convergence speed after batch or time-recursive initializations. Therefore, time-recursive updating is seen to retain the essential features of our EDM-based approach to SLAT, namely, a very limited need for *a priori* spatial information and fast convergence, at a fraction of the computational cost.

VI. Conclusion

In this paper, we have presented a ML-based technique to solve a SLAT problem using a two-phase approach under Gaussian or Laplacian noise. A MM method is proposed to iteratively maximize the non-convex likelihood function, for which a good initial point is required. To that end, we have investigated two initialization schemes based on EDM completion and source localization (SLCP/SLℓ_1) that bypass the need for strong priors on sensor/target positions. After acquiring an initial block of range measurements for the startup phase, a SNL method based on EDM completion was used to estimate the node positions and some of the target locations. In our experiments this was accomplished reasonably fast (a few seconds)

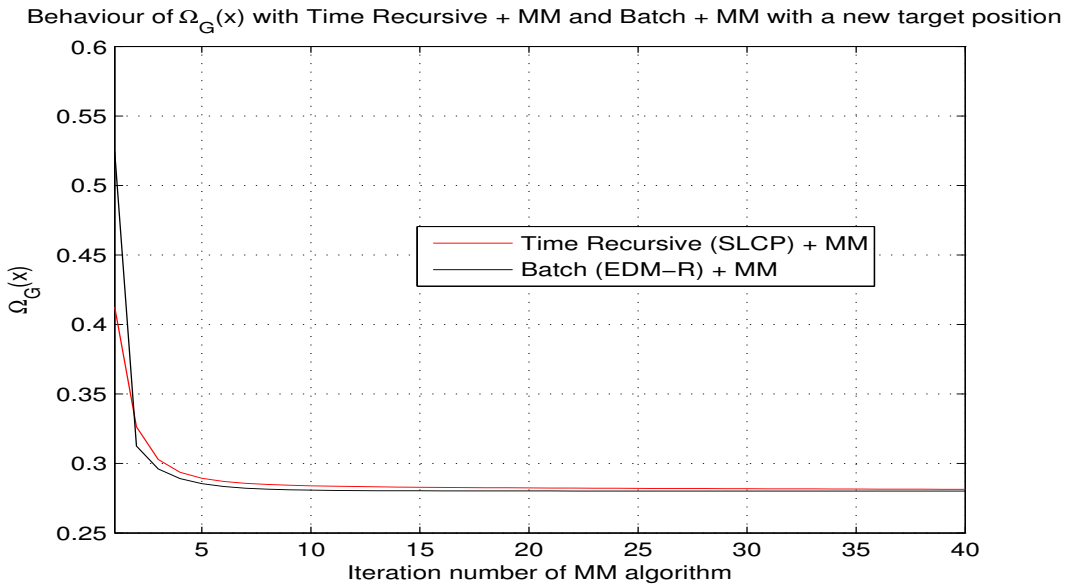


Fig. 5: Evolution of Gaussian cost function $\Omega_G(\mathbf{x})$ during refinement for EDM-R+MM and SLCP+MM approaches, with $\sigma_{\text{gaussian}} = 0.04$.

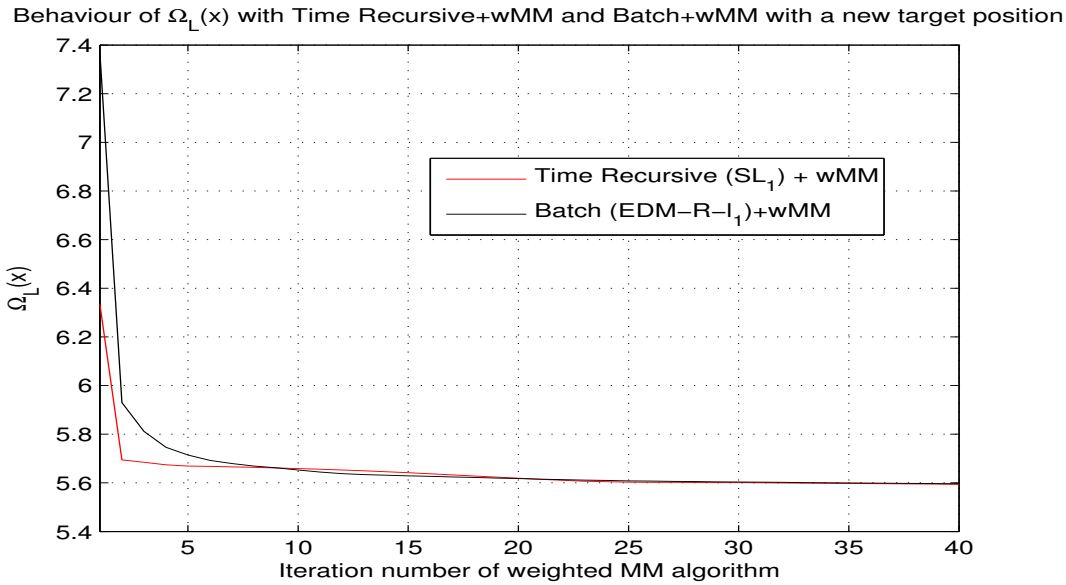


Fig. 6: Evolution of Laplacian cost function $\Omega_L(\mathbf{x})$ during refinement for EDM-R-ℓ_1+wMM and SLℓ_1+wMM approaches, with $\sigma_{\text{laplacian}} = 0.1$.

for scenarios with up to about 30 unknown positions. As EDM completion is not scalable, after startup we resorted to an alternative, lightweight, incremental initialization scheme as additional target range measurements become available. The SLCP or SLℓ_1 time-recursive methods fix the already estimated positions whenever a new one is to be determined; afterwards all positions are given as initialization to the likelihood refinement methods.

Simulation results showed that our method nearly attains the Cramér-Rao lower bound under moderate Gaussian noise. In the presence of outliers, both EDM-R-ℓ_1 and SLℓ_1 provide more accurate initial position estimates than other existing methods. Moreover, when used as input to iterative refinement methods they provide a good starting point that reduces the probability of convergence to undesirable extrema, yielding improved overall estimation performance. Hence, with this methodology, we obtained a processing structure that is robust to outliers and provides a scalable and accurate solution to the SLAT problem. Importantly, the algorithms based on ℓ_1 norm optimization exhibited robust behavior in simulation not only for Laplacian outliers, but also for an alternative outlier generation technique that did not match the underlying Laplacian modeling assumptions.

APPENDIX A
CONVERGENCE OF WEIGHTED MAJORIZATION MINIMIZATION

To prove (local) convergence of the weighted MM iteration (24) the Laplacian cost function (2) is first majorized at time t by

$$\Gamma_L^t(\mathbf{x}) = \frac{1}{2}\sum_{i,j}\Big\{u_{ij}^t(f_{ij}(\mathbf{x})-d_{ij})^2 + \frac{1}{u_{ij}^t}\Big\} + \frac{1}{2}\sum_{k,j}\Big\{v_{kj}^t(g_{kj}(\mathbf{x})-d_{kj})^2 + \frac{1}{v_{kj}^t}\Big\}, \tag{38}$$

where f_{ij}, g_{kj} and u_{ij}^t, v_{kj}^t are defined in (9) and (23). The inequality $\Omega_L(\mathbf{x}) \le \Gamma_L^t(\mathbf{x})$ follows from

$$\begin{aligned}\Gamma_L^t(\mathbf{x}) - \Omega_L(\mathbf{x}) &= \frac{1}{2}\sum_{i,j}\Big\{u_{ij}^t(f_{ij}(\mathbf{x})-d_{ij})^2 + \frac{1}{u_{ij}^t} - 2|f_{ij}(\mathbf{x})-d_{ij}|\Big\} \\ &\quad + \frac{1}{2}\sum_{k,j}\Big\{v_{kj}^t(g_{kj}(\mathbf{x})-d_{kj})^2 + \frac{1}{v_{kj}^t} - 2|g_{kj}(\mathbf{x})-d_{kj}|\Big\} \\ &= \frac{1}{2}\sum_{i,j}\Big\{\sqrt{u_{ij}^t}|f_{ij}(\mathbf{x})-d_{ij}| - \frac{1}{\sqrt{u_{ij}^t}}\Big\}^2 \quad + \frac{1}{2}\sum_{k,j}\Big\{\sqrt{v_{kj}^t}|g_{kj}(\mathbf{x})-d_{kj}| - \frac{1}{\sqrt{v_{kj}^t}}\Big\}^2 \ge 0.\end{aligned} \tag{39}$$

It is easy to check that $\Omega_L(\mathbf{x}^t) = \Gamma_L^t(\mathbf{x}^t)$, so $\Gamma_L^t(\mathbf{x})$ has the properties of a true majorization function for the iterate $\mathbf{x}^t$. Now the same technique used in (10) is applied to majorize (38) by a convex quadratic

function of $\mathbf{x}$, yielding

$$\begin{aligned}\Omega_L(\mathbf{x}) \leq \frac{1}{2}\sum_{i,j}\Big\{u_{ij}^t\big(f_{ij}^2(\mathbf{x}) - 2d_{ij}f_{ij}(\mathbf{x}^t) - 2d_{ij}\langle\nabla f_{ij}(\mathbf{x}^t),(\mathbf{x}-\mathbf{x}^t)\rangle + d_{ij}^2\big) + \frac{1}{u_{ij}^t}\Big\} \\ +\frac{1}{2}\sum_{k,j}\Big\{v_{ij}^t\big(g_{kj}^2(\mathbf{x}) - 2d_{kj}g_{kj}(\mathbf{x}^t) - 2d_{kj}\langle\nabla g_{kj}(\mathbf{x}^t),(\mathbf{x}-\mathbf{x}^t)\rangle + d_{kj}^2\big) + \frac{1}{v_{ij}^t}\Big\}.\end{aligned} \tag{40}$$

As before, equality holds for $\mathbf{x} = \mathbf{x}^t$, so the right-hand side of (40) is still a valid majorization function. Discarding constant terms the weighted MM iteration (24) results.

APPENDIX B
PROPERTIES OF SINGLE-SOURCE LOCALIZATION USING SLℓ_1

Proof of Lemma IV-C.1: To streamline the notation we define $K_i = |\|y - b_i\| - d_i|$, and apply the KKT condition to the inner optimization problem in (28) while fixing y. The Lagrangian function is

$$L(\boldsymbol{\lambda},\gamma) = \sum_{i=1}^{n+l}\frac{K_i^2}{\lambda_i} + \gamma(\mathbf{1}^T\boldsymbol{\lambda} - 1). \tag{41}$$

The KKT conditions are

$$\frac{d\,L}{d\lambda_i} = -\frac{K_i^2}{\lambda_i^2} + \gamma^* = 0, \qquad\qquad \mathbf{1}^T\boldsymbol{\lambda} = 1. \tag{42}$$

Using (42), we find $\lambda_i^* = \frac{K_i}{\sum_{i=1}^{n+l}K_i}$ as a solution of the inner optimization problem. Plugging the optimal $\boldsymbol{\lambda}$ in the cost function of (28) yields $(\sum_i K_i)^2$, thus establishing the equivalence with (27). ■

Approximation accuracy of $\boldsymbol{\Pi} = \lim_{\sigma\to\infty}(\boldsymbol{\Lambda} + \sigma\mathbf{1}\mathbf{1}^T)^{-1}$*:* To decide how large σ should be, let us first define $\boldsymbol{\Pi}(\sigma) = (\boldsymbol{\Lambda} + \sigma\mathbf{1}\mathbf{1}^T)^{-1}$. The norm of the difference to the original definition of $\boldsymbol{\Pi}$ in (32) is given by

$$\begin{aligned}\|\boldsymbol{\Pi} - \boldsymbol{\Pi}(\sigma)\|_F &= \|\boldsymbol{\Lambda}^{-1}\mathbf{1}[(\mathbf{1}^T\boldsymbol{\Lambda}^{-1}\mathbf{1})^{-1} - (\mathbf{1}^T\boldsymbol{\Lambda}^{-1}\mathbf{1} + \sigma^{-1})^{-1}]\mathbf{1}^T\boldsymbol{\Lambda}^{-1}\|_F \\ &= \frac{\mathbf{1}^T\boldsymbol{\Lambda}^{-2}\mathbf{1}}{(\mathbf{1}^T\boldsymbol{\Lambda}^{-1}\mathbf{1})(\sigma\mathbf{1}^T\boldsymbol{\Lambda}^{-1}\mathbf{1} + 1)}.\end{aligned} \tag{43}$$

Now assume the most unfavorable case with identical $\lambda_i = \frac{1}{n+l}$, such that

$$\|\boldsymbol{\Pi} - \boldsymbol{\Pi}(\sigma)\|_F = \frac{n+l}{\sigma(n+l)^2 + 1} \leq \epsilon \;\Rightarrow\; \sigma \geq \frac{1}{(n+l)\epsilon} - \frac{1}{(n+l)^2}. \tag{44}$$

For $\epsilon = 10^{-4}$ and $n + l = 100$, for example, this yields $\sigma \geq 10^2 - 10^{-4} \approx 10^2$, which is quite low and does not raise any numerical issues in commonly available convex optimization solvers. ■

APPENDIX C
DERIVATION OF CRLB FOR GAUSSIAN NOISE

The log of the joint conditional pdf for the SLAT problem is (up to an additive constant)

$$\log f(\mathbf{d}|\mathbf{x}) = -\frac{1}{2\sigma^2}\left\{\sum_{i,j}(\|x_i - e_j\| - d_{ij})^2 + \sum_{k,j}(\|a_k - e_j\| - d_{kj})^2\right\}, \tag{45}$$

where, similarly to $\mathbf{x}$, $\mathbf{d}$ denotes the concatenation of all range measurements. Let us define matrices $\mathbf{M}_{ij}$ and $\mathbf{N}_j$ that extract individual positions or their differences from the vector of concatenated coordinates[4] $\mathbf{x}$ as follows

$$\mathbf{M}_{ij}\mathbf{x} = x_i - e_j, \qquad \mathbf{N}_j\mathbf{x} = -e_j. \tag{46}$$

Thus, (45) is rewritten as

$$\log f(\mathbf{d}|\mathbf{x}) = -\frac{1}{2\sigma^2}\left\{\sum_{i,j}(\|\mathbf{M}_{ij}\mathbf{x}\| - d_{ij})^2 + \sum_{k,j}(\|a_k + \mathbf{N}_j\mathbf{x}\| - d_{kj})^2\right\}. \tag{47}$$

The first derivative of (47) with respect to $\mathbf{x}$ is

$$\nabla_{\mathbf{x}} \log f(\mathbf{d}|\mathbf{x}) = -\frac{1}{\sigma^2}\left\{\sum_{i,j}(\|\mathbf{M}_{ij}\mathbf{x}\| - d_{ij})\frac{\mathbf{M}_{ij}^T\mathbf{M}_{ij}\mathbf{x}}{\|\mathbf{M}_{ij}\mathbf{x}\|} + \sum_{k,j}(\|a_k + \mathbf{N}_j\mathbf{x}\| - d_{kj})\frac{\mathbf{N}_j^T(a_k + \mathbf{N}_j\mathbf{x})}{\|a_k + \mathbf{N}_j\mathbf{x}\|}\right\}. \tag{48}$$

The second derivative of (47) with respect to $\mathbf{x}$ is

$$\nabla_{\mathbf{x}}^2 \log f(\mathbf{d}|\mathbf{x}) = -\frac{1}{\sigma^2}\left\{\sum_{i,j}\left\{\frac{\mathbf{M}_{ij}^T\mathbf{M}_{ij}\mathbf{x}\mathbf{x}^T\mathbf{M}_{ij}^T\mathbf{M}_{ij}}{\|\mathbf{M}_{ij}\mathbf{x}\|^2} + \frac{\|\mathbf{M}_{ij}\mathbf{x}\| - d_{ij}}{\|\mathbf{M}_{ij}\mathbf{x}\|}\left(\mathbf{M}_{ij}^T\mathbf{M}_{ij} - \frac{\mathbf{M}_{ij}^T\mathbf{M}_{ij}\mathbf{x}\mathbf{x}^T\mathbf{M}_{ij}^T\mathbf{M}_{ij}}{\|\mathbf{M}_{ij}\mathbf{x}\|^2}\right)\right\}\right.$$
$$\left. + \sum_{k,j}\left\{\frac{\mathbf{N}_j^T(a_k + \mathbf{N}_j\mathbf{x})(a_k + \mathbf{N}_j\mathbf{x})^T\mathbf{N}_j^T}{\|a_k + \mathbf{N}_j\mathbf{x}\|^2} + \frac{\|a_k + \mathbf{N}_j\mathbf{x}\| - d_{kj}}{\|a_k + \mathbf{N}_j\mathbf{x}\|}\left(\mathbf{N}_j^T\mathbf{N}_j - \frac{\mathbf{N}_j^T(a_k + \mathbf{N}_j\mathbf{x})(a_k + \mathbf{N}_j\mathbf{x})^T\mathbf{N}_j^T}{\|a_k + \mathbf{N}_j\mathbf{x}\|^2}\right)\right\}\right\} \tag{49}$$

[4]If sensor positions x_i and target positions e_j are concatenated into vector $\mathbf{x}$ according to the order $x_1, \ldots, x_n, e_1, \ldots, e_m$, the selection matrices are explicitly given by

$$\mathbf{M}_{ij} = \begin{bmatrix}\mathbf{z}_i^T \otimes \mathbf{I}_2 & -\mathbf{v}_j^T \otimes \mathbf{I}_2\end{bmatrix}, \qquad \mathbf{N}_j = \begin{bmatrix}\mathbf{0}_{2\times 2n} & -\mathbf{v}_j^T \otimes \mathbf{I}_2\end{bmatrix},$$

where $\otimes$ denotes kronecker product, vector $\mathbf{z}_i \in \mathbb{R}^n$ has 1 in the i-th component and zeros elsewhere, and similarly for $\mathbf{v}_j \in \mathbb{R}^m$.

The Fisher information matrix, $F_{\mathbf{x}}$, is obtained by taking the negative expected value of (49) with respect to ranges as [16]

$$F_{\mathbf{x}} = -\mathrm{E}_{\mathbf{d}}\{\nabla^2_{\mathbf{x}} \log f(\mathbf{d}|\mathbf{x})\} = \frac{1}{\sigma^2}\left\{\sum_{i,j} \frac{\mathbf{M}_{ij}^T\mathbf{M}_{ij}\mathbf{x}\mathbf{x}^T\mathbf{M}_{ij}^T\mathbf{M}_{ij}}{\|\mathbf{M}_{ij}\mathbf{x}\|^2} + \sum_{k,j} \frac{\mathbf{N}_j^T(a_k+\mathbf{N}_j\mathbf{x})(a_k+\mathbf{N}_j\mathbf{x})^T\mathbf{N}_j^T}{\|a_k+\mathbf{N}_j\mathbf{x}\|^2}\right\}. \tag{50}$$

The CRLB matrix in (37) is taken as the inverse of $F_{\mathbf{x}}$.

APPENDIX D
EQUIVALENCE OF (13) AND (14)

We write (14) as

$$\begin{array}{lll} \text{minimize} & \text{minimize} & \sum_{i=1}^{n+l} \|y - y_i\|^2 \\ y & y_i & \\ & \text{subject to} & \|b_i - y_i\| = d_i \quad i = 1, \ldots, n+l. \end{array} \tag{51}$$

Given y, the inner optimization problem is separable. Defining $\eta_i = \frac{y_i - b_i}{d_i}$ it can be solved for y_1, …, y_{n+l} by individually solving the subproblems

$$\begin{array}{ll} \text{minimize} & \|y - d_i\eta_i - b_i\|^2 = \|y - b_i\|^2 + d_i^2\|\eta_i\|^2 - 2d_i\eta_i^T(y - b_i) \\ \eta_i & \\ \text{subject to} & \|\eta_i\| = 1, \end{array} \tag{52}$$

or, equivalently,

$$\begin{array}{ll} \text{maximize} & \eta_i^T(y - b_i) \\ \eta_i & \\ \text{subject to} & \|\eta_i\| = 1, \end{array} \tag{53}$$

The optimal solution of (53) is clearly given by $\eta_i = \frac{y - b_i}{\|y - b_i\|}$, leading to an optimal cost in (52) $\|y - b_i\|^2 + d_i^2 - 2d_i\|y - b_i\| = (\|y - b_i\| - d_i)^2$. Substituting the sum of these optimal costs for $i = 1$, …, $n + l$ back into (51) yields an unconstrained problem whose cost function is given by (13).

REFERENCES

[1] C. Taylor, H. Shrobe, A. Rahimi, J. Bachrach, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN'06)*, Nashville, USA, April 2006, pp. 27–33.

[2] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed localization of networked cameras," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN'06)*, Nashville, USA, April 2006, pp. 34–42.

[3] R. Rangarajan, R. Raich, and A. Hero, III, "Euclidean matrix completion problems in tracking and geo-localization," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'08)*, Las Vegas, Nevada, USA, March 2008, pp. 5324–5327.

[4] J. Teng, H. Snoussi, and C. Richard, "Decentralized variational filtering for simultaneous sensor localization and target tracking in binary sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'09)*, Taipei, Taiwan, April 2009, pp. 2233–2236.

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

[6] P. Biswas, T.-C. Liang, K.-C. Toh, and Y. Ye, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, pp. 360–371, 2006.

[7] W. K. Lui, W. K. Ma, H. C. So, and F. K. W. Chan, "Semi-definite programming approach to sensor network node localization with anchor position uncertainty," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'09)*, Taipei, Taiwan, April 2009, pp. 2245–2248.

[8] A. J. Weiss and J. Picard, "Maximum-likelihood position estimation of network nodes using range measurements," *IET Signal Process.*, vol. 2, no. 4, pp. 394–404, 2008.

[9] S. Srirangarajan, A. H. Tewfik, and Z.-Q. Luo, "Distributed sensor network localization with inaccurate anchor positions and noisy distance information," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'07)*, vol. 3, Honolulu, Hawaii, USA, April 2007, pp. 521–524.

[10] M. Vemula, M. F. Bugallo, and P. M. Djurić, "Sensor self-localization with beacon position uncertainty," *Elsevier Signal Process.*, vol. 89, no. 6, pp. 1144–1154, 2009.

[11] F. K. W. Chan and H. C. So, "Accurate distributed range-based positioning algorithm for wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4100–4105, 2009.

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

[13] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, 2009.

[14] D. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[15] P. Oğuz-Ekim, J. Gomes, J. Xavier, and P. Oliveira, "ML-based sensor network localization and tracking: Batch and time-recursive approaches," in *Proc. European Signal Process. Conf. (EUSIPCO'09)*, Glasgow, Scotland, August 2009.

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

[17] J. Picard and A. J. Weiss, "Accurate geolocation in the presence of outliers using linear programming," in *Proc. European Signal Process. Conf. (EUSIPCO'09)*, Glasgow, Scotland, August 2009, pp. 2077–2081.

[18] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, "Sensor network localization, Euclidean distance matrix completions, and graph realization," in *Research report CORR 2006-23*, University of Waterloo, Waterloo, ONN2L 3G1, Canada, 2008.

[19] P. Oğuz-Ekim, J. Gomes, J. Xavier, and P. Oliveira, "A convex relaxation for approximate maximum-likelihood 2D source localization from range measurements," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'10)*, Dallas, Texas, USA, March 2010.

[20] S. Korkmaz and A. J. van der Veen, "Robust localization in sensor networks with iterative majorization techniques," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'09)*, Taipei, Taiwan, April 2009, pp. 2049–2052.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[22] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo publishers, 2005.

[23] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, May 2008.

[24] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[25] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, 2000.

[26] P. Rodriguez and B. Wohlberg, "Efficient minimization method for a generalized total variational functional," *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 322–332, 2009.

[27] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

[28] N. Patwari, A. Hero, III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, 2003.

[29] C. Chang and A. Sahai, "Cramer-Rao-type bounds for localization," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–13, 2006.

[30] T. Jia and R. M. Buehrer, "A new Cramer-Rao lower bound for TOA-based localization," in *Proc. Military Commun. Conf. (MILCOM'08)*, November 2008, pp. 1–5.

Pınar Oğuz-Ekim (S'10) received the B.S. and M.S. degrees in electrical and electronics engineering from Middle East Technical University (METU), Ankara, Turkey, in 2003 and 2006, respectively. She is currently pursuing her Ph.D. in the Department of Electrical and Computer Engineering of Instituto Superior Técnico (IST), Lisbon, Portugal, since 2007. Her research interests include localization and target tracking in wireless sensor networks.

João Pedro Gomes (S'95–M'03) received the Diploma, M.S. and Ph.D. degrees in electrical and computer engineering from Instituto Superior Técnico (IST), Lisbon, Portugal, in 1993, 1996 and 2002, respectively. He joined the Department of Electrical and Computer Engineering of IST in 1995, where he is currently an Assistant Professor. Since 1994 he has also been a Researcher in the Signal and Image Processing Group of the Institute for Systems and Robotics, in Lisbon. His current research interests include channel identification and equalization in wireless communications, underwater communications and acoustics, fast algorithms for adaptive filtering, and sensor networks.

João Xavier (S'97-M'03) received the Ph.D. degree in electrical and computer engineering from Instituto Superior Técnico (IST), Lisbon, Portugal, in 2002. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, IST. He is also a Researcher at the Institute for Systems and Robotics (ISR), Lisbon, Portugal. His current research interests are in the area of optimization, sensor networks, and signal processing on manifolds.

Paulo Oliveira (S'91-M'95) received the "Licenciatura," M.S., and Ph.D. degrees in electrical and computer engineering in 1987, 1991, and 2002, respectively, from Instituto Superior Técnico (IST), Lisbon, Portugal. He is an Associate Professor in the Department of Mechanical Engineering of IST and Senior Researcher in the Institute for Systems and Robotics – Associated Laboratory. His research interests are in the area of Autonomous Robotic Vehicles with focus on the fields of Estimation, Sensor Fusion, Navigation, Positioning, and Industrial Automation. He is author or co-author of more than 20 journal papers and 100 conference communications. He participated in more than 20 European and Portuguese research projects, over the last 25 years.