Sparsity-Promoting Sensor Selection for Non-linear Measurement Models

Sundeep Prabhakar Chepuri, Student Member, IEEE, and Geert Leus, Fellow, IEEE

Abstract—Sensor selection is an important design problem in large-scale sensor networks. Sensor selection can be interpreted as the problem of selecting the best subset of sensors that guarantees a certain estimation performance. We focus on observations that are related to a general non-linear model. The proposed framework is valid as long as the observations are independent, and its likelihood satisfies the regularity conditions. We use several functions of the Cramér-Rao bound (CRB) as a performance measure. We formulate the sensor selection problem as the design of a selection vector, which in its original form is a nonconvex \(l_0\)-(quasi) norm optimization problem. We present relaxed sensor selection solvers that can be efficiently solved in polynomial time. We also propose a projected subgradient algorithm that is attractive for large-scale problems and also show how the algorithm can be easily distributed. The proposed framework is illustrated with a number of examples related to sensor placement design for localization.

Index Terms—Sensor selection, sensor placement, Cramér-Rao bound, selection vector, sparsity, non-linear models, statistical inference, projected subgradient algorithm, convex optimization, sensor networks.

I. INTRODUCTION

Advances in sensor technology have enabled a large spectrum of applications and services related to safety and security, surveillance, environmental and climate monitoring, to list a few. The sensor nodes are spatially deployed and operate as a network, with each sensor node capable of sensing, processing, and communicating to other nodes or a central processing unit. As a network, their fundamental task is distributed data sampling (i.e., to sense the environment) from which we seek to extract relevant information. The sensors provide a prohibitively large dataset which is usually gathered at a fusion center. This gathered data has to be optimally processed, rejecting the redundant, identical, or faulty measurements.

Sensor selection is a fundamental design task in sensor networks. The number of sensors is often limited either by economical constraints (hardware costs), or the availability of physical or storage space. In order to reduce the hardware costs, as well as the resulting communications and processing overhead, one would like to smartly deploy the sensors. Sensor selection also enables the design of spatio-temporal sensing patterns that guarantee a certain performance measure such as energy-efficiency, information measure, estimation accuracy, or detection probability. The sensor placement problem can also be interpreted as a sensor selection problem in which the best subset of the available sensor locations are selected subject to a specific performance constraint. Sensor selection is pertinent to various diverse fields, especially to applications dealing with large-scale networks like network monitoring [2], [3], location-aware services like target localization and tracking [4]–[6], field estimation [7], [8], and environment monitoring in general. The fundamental questions of interest are:

q1. Where to deploy the limited sensors available?
q2. Do we need to process all the acquired measurements?

To this end, we focus on processing only the most informative sensors for a general non-linear statistical inference problem.

A. Related prior works

A large volume of literature exists on sensor selection [9, and references therein]. The sensor selection problem is often formulated as an optimization problem based on some well-known performance measures from experimental design [9], [10], [11, Pg. 384]. The sensor selection problem is expressed as the following optimization problem:

\[
\arg\min_{w \in \{0,1\}^M} f(E(w)) \quad \text{s.t.} \quad 1^T_M w = K, \tag{1}
\]

where \(w\) is a selection vector of length \(M\), and \(f(E(w))\) is a scalar cost function related to the mean squared error (MSE) covariance matrix \(E\). The MSE covariance matrix is optimized to select the best subset of \(K\) sensors out of \(M\) available sensors such that \(K \ll M\). Different functions \(f(E(w))\) can be used, and the typical choices for \(f(E(w))\) are related to:

1. \textit{A-optimality:} minimizes the sum of eigenvalues of \(E\) with \(f(w) := \text{tr}\{E(w)\}\).
2. \textit{E-optimality:} minimizes the maximum eigenvalue of \(E\) with \(f(w) := \lambda_{\text{max}}\{E(w)\}\).
3. \textit{D-optimality:} minimizes the determinant of \(E\) with \(f(w) := \ln \det\{E(w)\}\).

This is a combinatorial optimization problem involving \(\binom{K}{M}\) searches, and it is clearly intractable even for small-scale problems with \(K = 10\) and \(M = 100\). To simplify this problem, the nonconvex Boolean constraint \(w \in \{0,1\}^M\) is relaxed to a convex box constraint \(w \in [0,1]^M\). The relaxed optimization problem has been studied in [9] for additive Gaussian linear models, where the matrix \(E\) is available in closed form, and more importantly, where the above listed performance measures are independent of the unknown parameter. Moreover, in practice, the exact number of sensors

This work was supported in part by STW under the FASTCOM project (10551) and in part by NWO-STW under the VICI program (10382).

All the authors are with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. Email: {s.p.chepuri,g.j.t.leus}@tudelft.nl.

A conference precursor of this manuscript has been published in [1].
to select might not be known. However, this number \( K \) can always be tuned to achieve a desired performance.

The above selection problem is applied to sensor placement for power grid monitoring in [12]. Alternative approaches exploiting the submodularity of the objective function [13]–[15], heuristics based on genetic algorithms [16], and greedy algorithms [17] are also proposed to solve the sensor selection problem. Sensor selection for dynamical systems often referred to as sensor polling or scheduling, is studied in [18]–[20]. In [21], the sensor placement problem for linear models is addressed as the design of a sensing matrix that optimizes a measure related to the orthogonality of its rows. All the above literature (in general) deals with measurements that are related to additive Gaussian linear models. Experimental design for non-linear models within the Bayesian and sequential design frameworks is discussed in [22]. In [18], sensor selection for target tracking based on extended Kalman filtering (EKF) has been proposed, in which the selection is performed by designing an appropriate gain matrix. Although a non-linear measurement model in additive Gaussian noise is used in [18], the past state estimate (not the true state) is used to compute the error covariance matrix leading to a suboptimal solution. Sensor selection for detection problems is studied in [23]. In [24], reliable sensor selection based on the actual measurements to identify the outliers is presented. A different problem, yet related to sensor selection, is the problem of identifying source-informative sensors, which is studied in [25].

B. Contributions

The sensor selection problem can be interpreted as the problem to select the best sensors out of \( M \) available sensors. The selected sensors are deemed as the best subset of sensors if they guarantee a certain specified estimation accuracy. We consider general scenarios where the measurements of the unknown parameter follow a non-linear model (unlike [9] for instance). Non-linear measurement models are frequently encountered in applications like source localization, tracking, field estimation, or phase retrieval, to list a few. The non-linear measurement model in additive Gaussian noise is used in [18], the past state estimate (not the true state) is used to compute the error covariance matrix leading to a suboptimal solution. Sensor selection for detection problems is studied in [23].

In [24], reliable sensor selection based on the actual measurements to identify the outliers is presented. A different problem, yet related to sensor selection, is the problem of identifying source-informative sensors, which is studied in [25].

II. NON-LINEAR MEASUREMENT MODEL

In this paper, we consider a generic non-linear measurement model

\[
y_m = h_m(\theta, n_m), \quad m = 1, 2, \ldots, M, \tag{2}
\]

where \( y_m \) is the \( m \)th spatial or temporal sensor measurement, \( \theta \in \mathbb{R}^N \) is the unknown parameter, \( n_m \) for \( m = 1, 2, \ldots, M \), is the noise process, and the regressors \( h_m \) for \( m = 1, 2, \ldots, M \), are (in general) non-linear functionals. Let the vector \( y = [y_1, y_2, \ldots, y_M]^T \in \mathbb{R}^M \) collect the measurements. The likelihood of the measurements \( p(y; \theta) \) is the probability density function (pdf) of \( y \) parameterized by the unknown vector \( \theta \).

We make the following assumptions:
a1. Regularity conditions: The log-likelihood of the measurements satisfies the regularity condition 
\[ \mathbb{E}\{ \frac{\partial \ln p(y; \theta)}{\partial \theta} \} = 0. \] This is a well-known condition for the CRB to exist [26].

a2. Independent observations: The measurements \( y_m \) for \( m = 1, 2, \ldots, M \), are a sequence of independent random variables.

The proposed framework for sensor selection is valid as long as the above two assumptions hold.

Assuming (a1) holds, the covariance of any unbiased estimate \( \hat{\theta} \in \mathbb{R}^N \) of the unknown parameter satisfies the well-known inequality [26]
\[
\mathbb{E}\{(\theta - \hat{\theta})(\theta - \hat{\theta})^T\} \geq C(\theta) = \mathcal{F}^{-1}(\theta),
\]
where the Fisher information matrix (FIM) is given by
\[
\mathcal{F}(\theta) = \mathbb{E}\left\{ \left( \frac{\partial \ln p(y; \theta)}{\partial \theta} \right) \left( \frac{\partial \ln p(y; \theta)}{\partial \theta} \right)^T \right\} \in \mathbb{R}^{N \times N},
\]
and \( C(\theta) \) is the CRB matrix. An important property of the Fisher information is that it is additive for independent observations, which follows from the fact that
\[
\ln p(y; \theta) = \ln \prod_{m=1}^{M} p(y_m; \theta) = \sum_{m=1}^{M} \ln p(y_m; \theta),
\]
where we assume that condition (a2) holds. Using (3), the FIM \( \mathcal{F}(\theta) \) can be alternatively expressed as
\[
\mathcal{F}(\theta) = \sum_{m=1}^{M} \mathbb{E}\left\{ \left( \frac{\partial \ln p(y_m; \theta)}{\partial \theta} \right) \left( \frac{\partial \ln p(y_m; \theta)}{\partial \theta} \right)^T \right\},
\]
which can be further simplified to
\[
\mathcal{F}(\theta) = \sum_{m=1}^{M} \mathcal{F}_m(\theta),
\]
where
\[
\mathcal{F}_m(\theta) = \mathbb{E}\left\{ \left( \frac{\partial \ln p(y_m; \theta)}{\partial \theta} \right) \left( \frac{\partial \ln p(y_m; \theta)}{\partial \theta} \right)^T \right\}
\]
is the \( N \times N \) FIM of the \( m \)th measurement. In other words, (4) means that every independent measurement contributes to the information measure. Note that the FIM for non-linear models depends on the unknown vector \( \theta \).

Assume for instance that the observations belong to the family of exponential distributions. The log-likelihood of the observations can then be expressed in the form
\[
\ln p(y_m; \theta) = \ln r(y_m) + a_m(\theta)b(y_m) - c(\theta),
\]
where \( r(y_m) \) and \( b(y_m) \) are known functions of the observations only, while \( a_m(\theta) \) and \( c(\theta) \) depend only on the unknown parameter. The regularity conditions in general hold for observations that belong to the family of exponential pdfs, and it already includes a large number of distributions.

One specific example that often occurs in practice is the case where the observations \( y_m, m = 1, 2, \ldots, M \), are related through the following additive Gaussian non-linear model
\[
y_m = h_m(\theta) + n_m, \quad m = 1, 2, \ldots, M,
\]
where \( h_m(\cdot) \) is a non-linear function, and \( n_m \) is a zero-mean Gaussian random variable with variance \( \sigma^2_m \). The log-likelihood of \( y_m \) is then given by (6) with
\[
r(y_m) = \frac{1}{\sqrt{2\pi\sigma^2_m}} \exp\left( -\frac{1}{2\sigma^2_m} y_m^2 \right),
b(y_m) = y_m/\sigma^2_m,
a_m(\theta) = h_m(\theta),
\]
and \( c(\theta) = 1/2\sigma^2_m h_m^2(\theta) \).

Assuming (a2) holds, it is then easy to verify that (5) simplifies to
\[
\mathcal{F}_m(\theta) = \frac{1}{\sigma^2_m} \left( \frac{\partial h_m(\theta)}{\partial \theta} \right) \left( \frac{\partial h_m(\theta)}{\partial \theta} \right)^T.
\]

Remark 1 (Additive Gaussian linear model). As a special case, when the measurement process is linear, we have \( y_m = h_m^T \theta + n_m, m = 1, 2, \ldots, M \), i.e., \( h_m(\theta, n_m) := h_m^T \theta + n_m \) with \( h_m \in \mathbb{R}^N \) being the regressor. The computation of the FIM for a linear model is straightforward, and is given by
\[
\mathcal{F} = \sum_{m=1}^{M} \frac{1}{\sigma^2_m} h_m h_m^T.
\]
The CRB for linear models in additive Gaussian noise is also the MSE, and more importantly it is independent of the unknown vector.

III. PROBLEM FORMULATION

Our goal is now to select the best subset (\( \geq N \)) of the available \( M \) sensor measurements such that a certain accuracy on the estimate \( \hat{\theta} \) is guaranteed. We next mathematically formulate this sensor selection problem.

A. Sensor selection

In order to select the sensors, we introduce a selection vector \( w = [w_1, w_2, \ldots, w_M]^T \in \{0, 1\}^M \), where \( w_m = 1(0) \) indicates that the \( m \)th sensor measurement is (not) selected. The measurement model including the virtual hard selection parameter can be visualized as
\[
y_m = w_m h_m(\theta, n_m), \quad m = 1, 2, \ldots, M.
\]

where the selection vector modifies the log-likelihood of the measurements as \( \ln \prod_{m=1}^{M} p(y_m; \theta)^{w_m} = \sum_{m=1}^{M} w_m \ln p(y_m; \theta) \). The corresponding FIM matrix in (4) can then be expressed as
\[
\mathcal{F}(w, \theta) = \sum_{m=1}^{M} w_m \mathcal{F}_m(\theta).
\]
B. Performance measures

We do not restrict ourselves to any specific estimator, however, we use the CRB as a performance measure. The motivation behind using the CRB is as follows:

1. The CRB is a measure for the (local) identifiability of the problem [27]. More specifically, a non-singular FIM implies (local) solvability and a unique estimate of \( \theta \), however, the converse is not necessarily true. The sensor selection problem presented in this paper seeks a subset of sensors for which the FIM has full rank in some domain such that the solvability of the problem in that domain is always ensured.

2. Typically, the subset of selected sensors that yields a lower CRB also yields a lower MSE, and thus improves the performance of any practical system.

The CRB also has a very attractive mathematical structure resulting in a selection problem that can be efficiently solved using convex optimization techniques.

We next use the consistency assumption of the estimator to derive thresholds for the performance measures. We constrain the estimation error \( e = \theta - \hat{\theta} \) to be within an origin-centered circle of radius \( R_c \) with a probability higher than \( P_c \), i.e.,

\[
\Pr(\|e\| \leq R_c) \geq P_c,
\]

where \( \Pr(\cdot) \) denotes probability, and the values of \( R_c \) and \( P_c \) define the accuracy required and are assumed to be known. A higher accuracy level is obtained by reducing \( R_c \) and/or increasing \( P_c \). This metric is used in several occasions as an accuracy measure (e.g., see [4], [28], [29]). We next discuss two popular performance measures that satisfy the above requirement.

1) Trace constraint: A sufficient condition to satisfy the accuracy requirement in (10) is (see Appendix A)

\[
\text{tr}\{C(w, \theta)\} = \text{tr}\{\sum_{m=1}^{M} w_m F_m(\theta)\}^{-1} \leq \lambda_{tr} = (1 - P_c) R_c^2.
\]

This measure is related to the A-optimality.

2) Minimum eigenvalue constraint: Another popular sufficient condition that also satisfies the accuracy requirement in (10) is

\[
\lambda_{\text{min}}\{F(w, \theta)\} \geq \lambda_{\text{eig}} = \frac{N}{R_c^2} \left( \frac{1}{1 - P_c} \right),
\]

where \( \lambda_{\text{eig}} \) is derived in [29] (see also Appendix A). This measure is related to the E-optimality. The inequality constraint \( \lambda_{\text{min}}\{F\} \geq \lambda_{\text{eig}} \) can be equivalently expressed as the following linear matrix inequality (LMI):

\[
\sum_{m=1}^{M} w_m F_m(\theta) - \lambda_{\text{eig}} I_N \succeq 0_N.
\]

In other words, we put a lower bound on each eigenvalue of the matrix \( F \). The solution set of \( w \) satisfying this LMI is convex as \( F_m(\theta) \in S^N, m = 1, 2, \ldots, M \) and \( \lambda_{\text{eig}} I_N \in S^N \) [11, Pg. 38].

The trace constraint has a larger feasible set as compared to the minimum eigenvalue constraint. However, although the trace constraint is a sufficient condition, the resulting sensor selection problem is computationally less attractive compared to the minimum eigenvalue constraint (as we show later). Moreover, LMIs can be used to also represent the trace constraint. For these reasons, we focus on the minimum eigenvalue (LMI) constraints from now on. However, without loss of generality (w.l.o.g.) either one of the two performance constraints can be used.

The above performance measures depend on the unknown parameter. In practice, the unknown parameter \( \theta \) has a physical meaning and takes values within a certain domain denoted by \( \mathcal{U} \). For example, in the case of direction-of-arrival estimation, \( \mathcal{U} \) is the sector where the source is expected or for target localization it is the surveillance area where the target resides. Since the FIM for non-linear models depends on the unknown \( \theta \), we propose to constrain every point within the domain \( \mathcal{U} \).

Remark 2 (Bayesian CRB constraint). In a Bayesian setting, when prior information of the unknown parameter \( \theta \) is available, this additional knowledge typically yields a lower CRB, and the related information matrix is often called the Bayesian information matrix (BIM). The BIM is given by \( F_B(w, \theta) = F(w, \theta) + J_p \), where \( J_p \) is some prior information matrix \( J_p = -\mathbb{E}_{\theta} \left\{ \frac{\partial}{\partial \theta} \ln p(\theta) \right\}^T \) with the (log) prior \( \ln p(\theta) \). The LMI constraint in (11) for the Bayesian setting will then be

\[
J_p + \sum_{m=1}^{M} w_m F_m(\theta) \succeq \lambda_{\text{eig}} I_N.
\]

The prior information typically comes from the dynamics, previous measurements, or combining other available measurements.

C. Problem statement

Having introduced the selection vector as well as the performance measure we can now formally state the problem.

Problem statement (Sensor selection). Given the likelihoods \( p(y_m; \theta), m = 1, 2, \ldots, M \), of the measurements, and assuming (a1) and (a2) hold, find a vector \( w \in \{0, 1\}^M \) that selects the minimum number of most informative sensors satisfying the performance measure \( \sum_{m=1}^{M} w_m F_m(\theta) - \lambda_{\text{eig}} I_N \preceq 0_N, \forall \theta \in \mathcal{U} \).

In order to reduce the hardware costs, storage, processing, and communication overhead, we minimize the number of selected sensors. This can be achieved by minimizing the cardinality of the selection vector, i.e., by minimizing the number of non-zero entries of the selection vector. Mathematically, the sensor selection problem is formulated as the design of a selection vector which can be expressed as the following optimization problem

\[
w^* = \arg \min_w \|w\|_0 \quad (13a)
\]

s.t. \( \sum_{m=1}^{M} w_m F_m(\theta) - \lambda_{\text{eig}} I_N \preceq 0_N, \forall \theta \in \mathcal{U} \),

\( w \in \{0, 1\}^M \),

(13c)
where the $\ell_0$-(quasi) norm refers to the number of non-zero entries in $w$, i.e., $\|w\|_0 := |\{m : w_m \neq 0\}|$. The threshold $\lambda_{\text{eig}}$ imposes the accuracy requirement. The threshold $\lambda_{\text{eig}}$ is also the sparsity-inducing parameter, where $\lambda_{\text{eig}} \rightarrow 0$ implies a sparser solution.

Suppose the domain $\mathcal{U}$ consists of $D$ points, obtained by gridding the entire domain at a certain resolution. The resulting multiple LMI constraints can be stacked together as a single LMI constraint. Let us consider the domain $\mathcal{U} = \{\theta_1, \theta_2, \ldots, \theta_D\}$ with $|\mathcal{U}| = D$. The constraints in (13b) can then be equivalently expressed as a single LMI constraint written as $\sum_{m=1}^M w_m F_m - \lambda_{\text{eig}} I_{DN} \succeq 0_{DN}$, where $F_m = \text{diag}(F_m(\theta_1), F_m(\theta_2), \ldots, F_m(\theta_D)) \in S_{DN}^M$ for $m = 1, 2, \ldots, M$. Note that the FIM after gridding is independent of $\theta$, and we denote this simply by $F_m$ (not explicitly as a function of $\theta$).

**Remark 3** (Worst-case constraints). If there exists some $\tilde{\theta} \in \mathcal{U}_w \subset \mathcal{U}$ such that $\lambda_{\text{min}}(F(\tilde{\theta}, \theta)) \leq \lambda_{\text{min}}(F(w, \theta))$, $\forall w \in \{0, 1\}^M$ and $\forall \theta \in \mathcal{U}_w$, then it is sufficient to constrain the performance for only the worst-case $\theta \in \mathcal{U}_w$ instead of $\forall \theta \in \mathcal{U}_w$. This property can be used as a guideline for gridding.

**IV. SENSOR SELECTION SOLVERS**

It is well known that the $\ell_0$-(quasi) norm optimization is NP-hard and nonconvex. More specifically, the original sensor selection problem in (13) is NP-hard. The Boolean constraint in (13c) is non-convex and incurs a combinatorial complexity. We next present a number of solvers with which the relaxed sensor selection problem.

**A. Convex approximation based on $\ell_1$-norm**

A computationally tractable (suboptimal) solution is to use the traditional best convex surrogate for the $\ell_0$-(quasi) norm namely the $\ell_1$-norm heuristic. The $\ell_1$-norm is known to represent an efficient heuristic for the $\ell_0$-(quasi) norm optimization with convex constraints especially when the solution is sparse [30]. Such relaxations are well-studied for problems with linear constraints in the context of compressed sensing (CS) and sparse signal recovery [31]. The non-convex Boolean constraint in (13c) is further relaxed to the convex box constraint $[0, 1]^M$.

The relaxed optimization problem is given as the following SDP problem

$$
\hat{w} = \arg \min_{w \in \mathbb{R}^M} \|w\|_1 \quad \text{s.t.} \quad \sum_{m=1}^M w_m F_m - \lambda_{\text{eig}} I_{DN} \succeq 0_{DN},
$$

where $\|w\|_1 = \sum_{m=1}^M |w_m|$ denotes the $\ell_1$-norm. Due to the positivity constraint, the objective function $\|w\|_1$ will simply be an affine function $1_T^M w$. The optimization problem in (14) is a standard SDP problem in the inequality form, which can be efficiently solved in polynomial time using interior-point methods [11]. An implementation of the interior-point method for solving SDP problems in the inequality form is typically based on Newton’s method using an approximating barrier function. A brief description of the projected Newton’s method is provided in Appendix C which is used to analyze the computational complexity of the relaxed sensor selection problem.

**Remark 4** (Complexity per iteration). The computational cost involved during each iteration is as follows [11, Pg. 619]. The matrices $F_m, m = 1, 2, \ldots, M$, have a block-diagonal structure with $D$ blocks. Forming the matrix $S = \sum_{m=1}^M w_m F_m - \lambda_{\text{eig}} I_{DN}$ costs $O(\mathcal{M}^2 \mathcal{N}^2)$ flops; computing $S^{-1} F_i$ via Cholesky factorization costs $O(\mathcal{M} \mathcal{D} \mathcal{N}^3)$ flops; the Hessian matrix is computed via the inner product of the matrices $S^{-1} F_i$ and $S^{-1} F_j$, which costs $O(\mathcal{D} \mathcal{M}^2 \mathcal{N}^2)$ via $i,j$. Finally, the Newton step is computed via Cholesky factorization costing $O(\mathcal{M}^3)$ flops, and the projection costs $O(\mathcal{M} \mathcal{N})$ flops. Assuming that $\mathcal{M} \gg \mathcal{N}$, the overall computational complexity per iteration of the projected Newton’s algorithm is then $O(\mathcal{M}^3)$.

Implementations of the interior-point methods are easily available in the form of well-known toolboxes like Yalmip [32], SeDuMi [33], and CVX [34].

**B. Projected subgradient algorithm**

The second-order Newton’s method (cf. Appendix C) is typically intractable when the number of sensors is very large ($\mathcal{M} \gg 1000$ for example). To circumvent this problem, we propose a subgradient based algorithm. The projected subgradient algorithm is a first-order method which is attractive for large-scale problems as each iteration is much cheaper to process.

The subgradient method is typically used for optimizations involving non-differentiable functions [35], [36]. The subgradient method is a generalization of the gradient method for non-smooth and non-differentiable functions, such as, the $\ell_1$-norm and the minimum eigenvalue constraint functions. We next derive the projected subgradient algorithm.

The relaxed sensor selection problem in (14) can be equivalently expressed as

$$
\arg \min_{w} \|w\|_1 \quad \text{s.t.} \quad f_{\text{eig}}(w) \geq \lambda_{\text{eig}}, \quad w \in \mathcal{W},
$$

where $f_{\text{eig}}(w) := \lambda_{\text{min}}(\sum_{m=1}^M w_m F_m)$ is the constraint function in (14b), and the set $\mathcal{W} = \{w \in \mathbb{R}^M | 0 \leq w_m \leq 1, m = 1, 2, \ldots, M\}$ denotes the box constraints in (14c).

The objective $1_T^M w$ is affine, so a subgradient of the objective is the all-one vector $1_M$. Let $g_k^k \in \partial f_{\text{eig}}(w_k)$ denote a subgradient of the constraint function $f_{\text{eig}}(w)$ at $w = w_k$. Here, the set $\partial f_{\text{eig}}(w_k)$ denotes the subdifferential of $f_{\text{eig}}(w)$ evaluated at $w = w_k$. To compute $g_k^k$, we express the constraint function $f_{\text{eig}}(w_k)$ as

$$
f_{\text{eig}}(w_k) = \inf_{\|v\|_1 \leq 1} v^T \left( \sum_{m=1}^M w_m F_m \right) v.
$$
Algorithm 1 Projected subgradient algorithm

1. Initialize iteration counter $k = 0$, $w^k = 1_M$, $g^k = 0$, $k_{\text{max}}$, $\epsilon$, and $\lambda_{\text{eig}}$.
2. for $k = 0$ to $k_{\text{max}}$
3. compute $f_{\text{eig}}(w^k) = \lambda_{\text{min}}\{\sum_{m=1}^{M} w_m^k F_m\}$
4. update
5. if $f_{\text{eig}}(w^k) \geq \lambda_{\text{eig}}$
6. $w^{k+1} = P_W(w^k - (1/\sqrt{k})1_M)$
7. else if $f_{\text{eig}}(w^k) < \lambda_{\text{eig}}$
8. $w^{k+1} = \lambda_{\text{eig}} + \frac{f_{\text{eig}}(w^k) + \epsilon}{\|g\|_2} g$
9. end
10. end
11. $\hat{w} = w^{k_{\text{max}}}$

The computation of a subgradient is straightforward, and is given by

$$g^k = (v_{\text{min}}^k)^T F_1 v_{\text{min}}^k, \ldots, (v_{\text{min}}^k)^T F_m v_{\text{min}}^k)^T \in \partial f_{\text{eig}}(w^k),$$

where $v_{\text{min}}^k$ is the eigenvector corresponding to the minimum eigenvalue $\lambda_{\text{min}}\{\sum_{m=1}^{M} w_m^k F_m\}$. The minimum eigenvalue and the corresponding eigenvector can be computed using a low-complexity iterative algorithm called the power method (see Appendix D) or using the standard eigenvalue decomposition [37]. Let the projection of a point onto the set $\mathcal{W}$ be denoted by $P_W(\cdot)$, which can be expressed elementwise as

$$[P_W(w)]_m = \begin{cases} 0 & \text{if } w_m \leq 0, \\ w_m & \text{if } 0 < w_m < 1, \\ 1 & \text{if } w_m \geq 1. \end{cases} \quad (16)$$

The projected subgradient algorithm then proceeds as follows:

$$w^{k+1} = \begin{cases} P_W(w^k - \alpha_k 1_M) & \text{if } f_{\text{eig}}(w^k) \geq \lambda_{\text{eig}}, \\ P_W(w^k + \alpha_k g^k) & \text{if } f_{\text{eig}}(w^k) < \lambda_{\text{eig}}. \end{cases} \quad (17)$$

In other words, if the current iterate $w^k$ is feasible (i.e., $f_{\text{eig}}(w^k) \geq \lambda_{\text{eig}}$), we update $w$ in the direction of a negative objective subgradient, as if the LMI constraints were absent; If the current iterate $w^k$ is infeasible (i.e., $f_{\text{eig}}(w^k) < \lambda_{\text{eig}}$), we update $w$ in the direction of a subgradient $g^k$ associated with the LMI constraints. After the update is computed, the iterate is projected onto the constraint set $\mathcal{W}$ using $P_W(\cdot)$.

When the $k$th iterate is feasible, a diminishing nonsummable step size $\alpha_k = 1/\sqrt{k}$ is used. When the iterate is not feasible, Polyak’s step size $\alpha_k = \frac{f_{\text{eig}}(w^k) + \epsilon}{\|g^k\|_2}$ is used, where we adopt the optimal value for $\epsilon := \frac{1}{10^4} w_{\text{best}}^k$ when $\|w\|_0$ known (i.e., the number of sensors to be selected is known). If this is not known, then we approximate it with $\epsilon := f_{\text{eig}}^k + \gamma$, where $\gamma = 10/(10 + k)$, and $f_{\text{eig}}^k = \min \{f_{\text{eig}}^k - 1, 10^{-4} w_{\text{best}}^k\} [35]$. The algorithm is terminated after a specified maximum number of iterations $k_{\text{max}}$. Finally, the estimate is denoted by $\hat{w} = w^{k_{\text{max}}}$.

The convergence results of the subgradient method for the constrained optimization (i.e., without the projection step) are derived in [35]. Since the projection onto a convex set is non-expansive [36], it does not affect the convergence. The projected subgradient algorithm is summarized as Algorithm 1.

Remark 5 (Complexity per iteration). We first form the matrix $\sum_{m=1}^{M} w_m F_m$, which costs $O(DMN^2)$ flops. The minimum eigenvalue and the corresponding eigenvector can be computed using the power method at a cost of $O(DN^3)$ flops [37]. Forming the vector $g$ costs $O(DMN^2)$ flops, computing its norm costs $O(M)$ flops, and the update and projection together cost $O(M)$ flops. Assuming that $M \gg N$ as earlier, the computational cost of the projected subgradient algorithm is $O(DMN^2)$ which is much lower than the complexity of the projected Newton’s method.

A distributed implementation of the projected subgradient algorithm is very easy. A simple distributed averaging algorithm (e.g., [38]) can be used to compute the sum of matrices $\sum_{m=1}^{M} w_m F_m$. The minimum eigenvalue and the corresponding eigenvector can then be computed using power iterations at each node independently. The update equation (17), the subgradient vector $g$, and the projection are computed coordinatewise and are already distributed.

Subgradient methods are typically very slow compared to the interior-point method involving Newton iterations, and subgradient methods typically require a few hundred iterations. Newton’s method typically requires in the order of ten steps. On the other hand, unlike the projected subgradient method, Newton’s method cannot be easily distributed, and requires a relatively high complexity per iteration due to the computation and storage of up to second-order derivatives. Depending on the scale of the problem and the resources available for processing one could choose between the subgradient or Newton’s algorithm.

C. Concave surrogate: sparsity-enhancing iterative algorithm

The $\ell_1$-norm is customarily used as the best convex relaxation for the $\ell_0$-norm. However, the intersection of the $\ell_1$-norm ball (or an affine subspace) with the positive semi-definite cone (i.e., the LMI constraint) is not always a unique point as shown in the following Theorem.

Theorem 1 (Uniqueness). The projection of a point $w \in [0, 1]^M$ onto a convex LMI constraint set $\sum_{m=1}^{M} w_m F_m - \lambda_{\text{eig}} I_{DN} \succeq 0_{DN}$ under the $\ell_1$-norm is not always unique.

Proof: The proof follows from the fact that the $\ell_1$-norm is not strictly convex, and from the linearity of the constraint set. Let us consider an example with $M = 2$ (w.l.o.g.), and $F_1 = F_2 \succeq 0_{2 \times 2}$. In other words, the observations are identical. In this case, the extreme points of the $\ell_1$-norm ball, i.e., $w_1 = (1, 0)$ and $w_2 = (0, 1)$ are two example solutions. Moreover, since the solution set of a convex minimization problem is convex, $\tau w_1 + (1 - \tau) w_2$ is also a solution for any $0 < \tau < 1$, which gives an infinite number of solutions to the relaxed optimization problem (14). For such cases, the $\ell_1$-norm relaxation will typically not result in a sparse solution.

To improve upon the $\ell_1$-norm solution due to its non-uniqueness following from Theorem 1, we propose an alternative relaxation for the original sensor selection problem which also results in fewer selected sensors. Instead of relaxing the $\ell_0$-(quasi) norm with the $\ell_1$-norm, using a nonconvex surrogate
function can yield a better approximation. It is motivated in [39] that the logarithm of the geometric mean of its elements can be used as an alternative surrogate function for linear inverse problems in CS. Adapting this to our sensor selection problem, we arrive at the optimization problem

\[
\arg\min_{w \in \mathbb{R}^M} \sum_{m=1}^{M} \ln (w_m + \delta) \quad (18a)
\]

subject to

\[
\sum_{m=1}^{M} w_m F_m - \lambda_{\text{eig}} I_{DN} \succeq 0_{DN}, \quad (18b)
\]

\[
0 \leq w_m \leq 1, \quad m = 1, 2, \ldots, M, \quad (18c)
\]

where \( \delta > 0 \) is a small constant that prevents the cost from tending to \(-\infty\). The cost (18a) is concave, but since it is smooth w.r.t. \( w \), iterative linearization can be performed to obtain a local minimum [39]. The first-order approximation of \( \ln (w_m + \delta) \) around \( (w_m[i-1] + \delta) \) results in

\[
\ln (w_m + \delta) \leq \ln (w_m[i-1] + \delta) + \frac{(w_m - w_m[i-1])}{(w_m[i-1] + \delta)}. \]

Instead of minimizing the original cost, the majorizing cost (second term on the right-hand side of the above inequality) can be optimized to attain a local minima. More specifically, the optimization problem (18) can be iteratively driven to a local minimum using the iterations

\[
\hat{w}[i] = \arg\min_{w \in \mathbb{R}^M} \sum_{m=1}^{M} \frac{w_m}{w_m[i-1] + \delta} \quad (19a)
\]

subject to

\[
\sum_{m=1}^{M} w_m F_m - \lambda_{\text{eig}} I_{DN} \succeq 0_{DN}, \quad (19b)
\]

\[
0 \leq w_m \leq 1, \quad m = 1, 2, \ldots, M. \quad (19c)
\]

The iterative algorithm is summarized as Algorithm 2. Each iteration in (19) solves a weighted \( \ell_1 \)-norm optimization problem. The weight updates force the small entries of the vector \( \hat{w}[i] \) to zero and avoid inappropriate suppression of larger entries. The parameter \( \delta \) provides stability, and guarantees that the zero-valued entries of \( \hat{w}[i] \) do not strictly prohibit a nonzero estimate at the next step. Finally, the estimate is given by \( \hat{w} = \hat{w}[i_{\text{max}}] \), where \( i_{\text{max}} \) is the specified maximum number of iterations.

**Remark 6** (Sparsity-enhancing projected subgradient algorithm). The projected subgradient algorithm can be adapted to fit into the sparsity-enhancing iterative algorithm as well. The optimization problem (20) is then replaced with the following update equations:

\[
w^{k+1} = \begin{cases} 
\mathcal{P}_V (w^k[i] - \alpha^k u^k[i]) & \text{if } f_{\text{eig}}(w^k[i]) \geq \lambda_{\text{eig}}, \\
\mathcal{P}_V (w^k[i] + \alpha^k g^k[i]) & \text{if } f_{\text{eig}}(w^k[i]) < \lambda_{\text{eig}},
\end{cases}
\]

where we solve a number of iterations (inner loop) of the projected subgradient algorithm within the \( k \)th iteration (outer loop) of Algorithm 2. Here, the \( k \)th iterate of the inner loop in the \( k \)th iteration is denoted as \((\cdot)^k[i]\).

From the solution of the relaxed optimization problem, the approximate Boolean solution to \( w \in \{0, 1\}^M \) can be obtained using randomization techniques, as described next.

**Algorithm 2** Sparsity-enhancing iterative algorithm

1. Initialize the iteration counter \( i = 0 \), the weight vector \( u[0] = [u_1[0], u_2[0], \ldots, u_M[0]]^T = 1_M, \delta \), and \( i_{\text{max}} \).
2. for \( i = 0 \) to \( i_{\text{max}} \)
3. solve the weighted \( \ell_1 \)-norm minimization problem

\[
\hat{w}[i] = \arg\min_{w \in \mathbb{R}^M} u[i]^T w \quad (20a)
\]

subject to

\[
\sum_{m=1}^{M} w_m F_m - \lambda_{\text{eig}} I_{DN} \succeq 0_{DN}, \quad (20b)
\]

\[
0 \leq w_m \leq 1, \quad m = 1, 2, \ldots, M. \quad (20c)
\]

4. update the weight vector \( u_m[i+1] = \frac{1}{\delta + u_m[i]} \), for each \( m = 1, 2, \ldots, M. \)
5. end
6. \( \hat{w} = \hat{w}[i_{\text{max}}] \).

**Algorithm 3** Randomized rounding algorithm

1. Generate \( l = 1, 2, \ldots, L \), candidate estimates of the form \( \hat{w}[i, l] = 0 \) with a probability \( 1 - \hat{w}[i, l] \) for \( m = 1, 2, \ldots, M. \)
2. Define \( \hat{w}_{l} = [\hat{w}_1,l, \ldots, \hat{w}_{M,l}]^T \) and the index set of the candidate estimates satisfying the constraints as

\[
\Omega \triangleq \{ l \mid \lambda_{\text{min}}[F(w_l, \theta)] \geq \lambda_{\text{eig}}, \forall \theta \in \mathcal{U}, l = 1, 2, \ldots, L \}.
\]
3. if \( \Omega = \{ \} \) is empty, go back to step 1.
4. The suboptimal Boolean estimate is the solution to the optimization problem

\[
\hat{w}_{bp} = \arg\min_{l \in \Omega} \|\hat{w}_l\|_1.
\]

**D. Randomized rounding**

The solution of the relaxed optimization problem is used to compute the suboptimal Boolean solution for the selection problem. A straightforward technique that is often used is the simple rounding technique, in which the Boolean estimate is given by round(\( \hat{w}_m \)), \( m = 1, 2, \ldots, M \), where we define \( \hat{w} = [\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_M]^T \), and the round(.) operator rounds its arguments towards the nearest integer. However, there is no guarantee that the Boolean estimates obtained from the rounding technique always satisfy the LMI constraint. Hence, we propose a randomized rounding technique, where the suboptimal Boolean estimates are computed based on random experiments guided by the solution from the SDP problem in (14) or the iterative version in (19). The randomized rounding technique is summarized as Algorithm 3.

**V. Extensions**

**A. The dual problem**

The dual of the relaxed primal optimization problem has an interesting relation to the diameter of the confidence ellipsoids, and is closely related to the dual of the \( E \)-optimal design [11, }
Pg. 388]. The dual problem of (14) is given as follows

\[
(Z, \mu) = \arg \max_{Z, \mu} \lambda_{\text{eig}} \text{tr}\{Z\} - 1_{M}^{T}\mu
\]

\[
s.t. \quad \mathbb{E}\{s_{m}^{T}Zs_{m}\} \leq 1 + \mu_{m}, m = 1, 2, \ldots, M,
\]

\[
Z \succeq 0, \mu_{m} \geq 0, m = 1, 2, \ldots, M,
\]

where \(Z \in \mathbb{S}_{+}^{D} \) and \(\mu \in \mathbb{R}^{N} \) are the dual variables, and we use \(\mathbb{E}\{s_{m}^{T}Zs_{m}\} = \text{tr}\{F_{m}Z\} \) with \(s_{m} = [\partial_{\theta_{1}} \{s_{m}(\theta, n)\}, \ldots, \partial_{\theta_{q}} \{s_{m}(\theta, n)\}]^{T} \in \mathbb{R}^{DN} \). For a detailed derivation of the dual problem, see Appendix B. The dual problem can be interpreted as the problem of maximizing the (average) diameter of the confidence ellipsoid. If we set \(\mu_{m} = 0, m = 1, 2, \ldots, M \), the optimal solution \(Z \) to the problem (21) is also the solution to the dual of the E-optimal design problem [11, Pg. 388], which maximizes the diameter of the confidence ellipsoid centered around the origin.

The dual formulation is often easier to solve and has only \(M \) inequality constraints. The dual problem can be solved using Yalmip, SeDuMi, or CVX as earlier. Suppose \(Z \) and \(\mu \) are dual feasible, and \(w \) is primal feasible, then the dual problem yields the following bound on the primal SDP problem: \(\lambda_{\text{eig}} \text{tr}\{Z\} - 1_{M}^{T}\mu \leq 1_{M}^{T}w\).

B. Scalar constraints

Trace constraint: The relaxed sensor selection problem with the scalar trace constraint is given as follows

\[
\arg \min_{w \in \mathbb{R}^{M}} \|w\|_{1}
\]

\[
s.t. \quad \text{tr}\left\{\left(\sum_{m=1}^{M} w_{m}F_{m}(\theta)\right)^{-1}\right\} \leq \lambda_{10}, \forall \theta \in \mathcal{U},
\]

\[
0 \leq w_{m} \leq 1, \quad m = 1, 2, \ldots, M.
\]

The trace constraint in (22) is convex in \(w\); this is easier to verify when the above trace constraint is expressed as an LMI [11, Pg. 387]. The optimization problem in (22) is a convex problem, and can be cast as an SDP:

\[
\arg \min_{w \in \mathbb{R}^{M}, x \in \mathbb{R}^{N}} \|w\|_{1}
\]

\[
s.t. \quad \left[\sum_{m=1}^{M} w_{m}F_{m}(\theta) \delta_{n}\right]^{T}x_{n} \geq 0, n = 1, 2, \ldots, N,
\]

\[
0 \leq w_{m} \leq 1, \quad m = 1, 2, \ldots, M,
\]

\[
1_{N}^{T}x \leq \lambda_{10}, \quad x_{n} \geq 0, n = 1, 2, \ldots, N, \forall \theta \in \mathcal{U},
\]

where \(x = [x_{1}, x_{2}, \ldots, x_{N}]^{T} \in \mathbb{R}^{N} \) is a variable, and \(\delta_{n}\) is the \(n\)th unit vector in \(\mathbb{R}^{N} \). The optimization problem in (23) has \(N\) LMI constraints for every point in \(\mathcal{U} \) and \(N + 1\) inequality constraints (in addition to the box constraint), while the optimization problem in (14) has only one LMI constraint for every point in \(\mathcal{U} \) (in addition to the box constraint). Hence, solving (23) is computationally more intense than solving (14).

Determinant constraint: Another popular scalar performance measure for the quality of the estimate is the determinant (product of eigenvalues) constraint. This measure is related to the D-optimality. The relaxed sensor selection problem with the determinant constraint is given as follows

\[
\arg \min_{w \in \mathbb{R}^{M}} \|w\|_{1}
\]

\[
s.t. \quad \ln \det \left\{\sum_{m=1}^{M} w_{m}F_{m}(\theta)\right\} \geq \lambda_{\text{det}}, \forall \theta \in \mathcal{U},
\]

\[
0 \leq w_{m} \leq 1, \quad m = 1, 2, \ldots, M,
\]

where the threshold \(\lambda_{\text{det}}\) specifies the mean radius a confidence ellipsoid (see Appendix A). In other words, although it is an indication of the performance of the estimator, it is not a sufficient condition for (10). The log-determinant constraint is a concave function of \(w\) for \(w_{m} \geq 0\), for \(m = 1, 2, \ldots, M\).

The relaxed sensor selection problem with the scalar (trace or determinant) constraints can be solved with either one of the two proposed cost functions, i.e., the \(\ell_{1}\)-norm or the log-based concave surrogate.

VI. EXAMPLES: SENSOR PLACEMENT FOR LOCALIZATION

Localization is an important and extensively studied topic in wireless sensor networks (WSNs). Target localization can be performed using a plethora of algorithms [4]–[6] (and references therein), which exploit inter-sensor measurements like time-of-arrival (TOA), time-difference-of-arrival (TDOA), angle-of-arrival (AOA), or received signal strength (RSS). The performance of any location estimator depends not only on the algorithm but also on the placement of the anchors (sensors with known locations). Sensor placement is a key challenge in localization system design, as certain sensor constellations not only deteriorate the performance but also result in ambiguity or identifiability issues [1].

The sensor placement problem can be interpreted as the problem where we divide a specific sensor area \(S\) into \(M\) grid points and select the best subset from these grid points. Here, the selected sensors are deemed the best, if they guarantee a certain minimal accuracy on the location estimates within a specific target area \(\mathcal{U}\). We consider a two-dimensional network with one target located in the target area \(\mathcal{U}\) and \(M\) possible sensors located at the \(M\) grid points.

The absolute positions of the sensor grid points are known, hence, the considered sensors are commonly referred to as anchor nodes. Let the coordinates of the target and the \(m\)th anchor be denoted by the \(2 \times 1\) vectors \(\theta = [\theta_{1}, \theta_{2}]^{T}\) and \(a_{m} = [a_{m,1}, a_{m,2}]^{T}\), respectively, where \(\theta\) is assumed to be unknown but known to be within \(\mathcal{U}\). We next illustrate the proposed framework with a number of examples all related to localization.

A. Distance measurements

Let the pairwise distance between the target and the \(m\)th anchor be denoted by \(d_{m} = \|\theta - a_{m}\|_{2}\). In practice, the pairwise distances are obtained by ranging and they are generally noisy. The range measurements generally follow an additive Gaussian non-linear model, as given by

\[
y_{m} = d_{m} + n_{m}, \quad m = 1, 2, \ldots, M,
\]

where \(n_{m} \sim \mathcal{N}(0, \sigma_{m}^{2})\) is the noise with \(\sigma_{m}^{2} = \frac{\sigma_{\eta}^{2}}{d_{m}}\). Here, \(\sigma_{\eta}\) is the nominal noise variance, and \(\eta\) is the path-loss exponent.
Using (9), we can now write the FIM for the localization problem as \( \mathbf{C}^{-1} = \mathbf{F}(\mathbf{w}, \mathbf{\theta}) = \sum_{m=1}^{M} w_m \mathbf{F}_m(\mathbf{\theta}) \), where using (5) we can compute
\[
\mathbf{F}_m(\mathbf{\theta}) := \frac{(\mathbf{\theta} - \mathbf{a}_m)(\mathbf{\theta} - \mathbf{a}_m)^T}{\sigma_m^2 \| \mathbf{\theta} - \mathbf{a}_m \|_2^2}.
\]

B. Received signal strength (RSS)

RSS is the voltage measured by a sensor’s received signal strength indicator (RSSI) circuit. RSS is often reported as the measured power. The ensemble mean received power at the mth sensor can be expressed as
\[
\bar{y}_m = y_0 - 10\eta \ln \frac{d_m}{d_0}, m = 1, 2, \ldots, M,
\]
where \( y_0 \) is the received power (dBm) at a reference distance \( d_0 \). However, due to shadowing, the difference between a measured received power and its ensemble average is random.

C. Bearing measurements

Another popular target localization technique is based on bearing measurements from a set of direction finding (DF) sensors [40]. The bearing measurement of the mth DF sensor is given by
\[
y_m = \arctan \left( \frac{\Delta a_{m,2}}{\Delta a_{m,1}} \right) + n_m, m = 1, 2, \ldots, M,
\]
where \( \Delta a_{m,2} = \theta_2 - \theta_2, \Delta a_{m,1} = \theta_1 - \theta_1, \) and \( n_m \sim \mathcal{N}(0, \sigma_n^2) \) is the noise. Defining a \( 2 \times 2 \) permutation matrix
\[
\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},
\]
we can then compute the FIM contribution from the mth DF sensor as
\[
\mathbf{F}_m = \frac{1}{\sigma_n^2 d_m^2} \mathbf{P}(\mathbf{\theta} - \mathbf{a}_m)(\mathbf{\theta} - \mathbf{a}_m)^T \mathbf{P}^T.
\]

D. Energy measurements

Another popular localization scheme relevant to field estimation, (active/passive) radar, and sonar is to estimate the location of a point source that emits or reflects energy. Suppose the sensors measure the energy generated by a point source. The measurements are given as
\[
y_m = \sqrt{2} h_m(\mathbf{\theta}) + n_m, m = 1, 2, \ldots, M,
\]
where \( e \) is the known energy emitted or reflected by the source, the known propagation function for some gain \( \beta \geq 0 \) is modeled as an isotropic exponential attenuation \( h_m(\mathbf{\theta}) = e^{-d_m^\beta} \), and \( n_m \sim \mathcal{N}(0, \sigma_n^2) \) is the noise. The FIM related to the mth measurement is then given by
\[
\mathbf{F}_m(\mathbf{\theta}) := \frac{4e^{\beta^2}}{\sigma_n^2 (\beta + d_m^2)^2} (\mathbf{\theta} - \mathbf{a}_m)(\mathbf{\theta} - \mathbf{a}_m)^T.
\]

**Remark 7** (Active sensor selection). The sensor selection problem can also be formulated for active sensing. In active sensing, the sensors transmit probing signals (e.g., radar, sonar). The selection parameter \( w_m \) for active sensing is a soft parameter used for joint selection and resource allocation [I], i.e., \( w_m \in [0, 1] \) is a resource (e.g., transmit energy) normalized to the maximum prescribed value, and hence, it is dimensionless. The relaxed active sensor selection problem takes the same form as in (14). In fact, minimizing the \( \ell_1 \)-norm in active sensor selection minimizes the overall network resources (e.g., overall transmit energy).

E. Simulations

We apply the proposed sensor selection problem to sensor placement design for target localization. To test the proposed algorithms, we use CVX [34], CVX internally calls SeDuMi [33], a MATLAB implementation of the second-order interior-point methods.

We consider the scenario shown in Fig. 1a with \( M = 80 \) sensors to illustrate the sensor selection problem. Recall that the problem here is to choose the best sensor positions out of \( M = 80 \) available ones, such that a certain specified localization accuracy is achieved. The domain \( \mathcal{U} \) for this example will be the target (or surveillance) area where the target resides, and to avoid having infinitely many constraints the area \( \mathcal{U} \) consists of grid points at a certain resolution. We grid the target area of \( 15 \times 15 \) m\(^2\) uniformly with a resolution of 1.75 m along both horizontal and vertical directions as shown in Fig. 1a.

The original non-convex sensor selection problem is relaxed to an \( \ell_1 \)-norm optimization problem. Alternatively, a concave surrogate function can be used to enhance the sparsity. The optimization problem with the concave surrogate cost function is iteratively solved by affinely scaling the objective based on the solution from the previous iteration. For the sparsity-enhancing iterative Algorithm 2, we use \( i_{\text{max}} = 10 \) and \( \delta = 10^{-8} \). The number of candidates used in the randomized rounding Algorithm 3 is \( L = 100 \). As observed in the simulations, a solution is typically found in the first batch itself, and a few tens of candidate entries are sufficient. We use the following parameters for the simulations: \( \eta = 2, \sigma^2 = 2 \times 10^{-5}, \sigma_b^2 = 2 \times 10^{-5} \) square-degrees, \( \sigma_{e, \text{dB}} = 2 \) dB, \( \sigma_e^2 = -20 \) dBm, \( e = 1, \beta = 1, \) and \( P_e = 0.9 \) (except in Fig. 1d).

Fig. 1 shows the sensor selection for the distance (range) measurement model. The thresholds are computed with \( R_e = 20 \) cm and \( P_e = 0.9 \). The selection shown in Fig. 1a is based on Algorithm 2 with randomized rounding to recover the approximate Boolean solution. The selection results based on the \( \ell_1 \)-norm cost with the minimum eigenvalue constraint is shown in Fig. 1b. Fig. 1b also shows that the solution based on the concave surrogate cost function with the minimum eigenvalue constraint leads to a sparser solution. The selection results based on the trace constraint obtained by solving (22) are illustrated in Fig. 1c. The sensors from the same region (close to the red filled boxes in Fig. 1a) are selected with both constraints. Fig. 1d shows a zero-duality gap (gap between the
Fig. 1: Sensor placement for target localization with \( M = 80 \) available sensors. The thresholds are computed using \( R_e = 20 \) cm, and \( P_e = 0.9 \) (expect for \( d \)). (a) Selection based on sparsity-enhancing iterations with minimum eigenvalue constraints. The Boolean solution is recovered using randomized rounding. (b) Minimum eigenvalue constraints with \( \ell_1 \)-norm and concave surrogate based relaxations. Randomized rounding is applied to the concave surrogate based solution. (c) \( \ell_1 \)-norm based selection with the trace constraints. (d) \( \ell_1 \)-norm cost function for different \( P_e \) with \( R_e = 20 \) cm.

The optimization problem \( (14) \) is also solved using the projected subgradient method summarized in Algorithm 1 with \( k_{\text{max}} = 1000 \) iterations. The solution of the projected subgradient algorithm is compared to the solution of the interior-point methods (implemented using SeDuMi) denoted by \( f_{\text{opt}} \) (obtained via SeDuMi), i.e., \( (f^k_{\text{best}} - f_{\text{opt}})/f_{\text{opt}} \) is shown in Fig. 2b. Even though the convergence of the projected subgradient algorithm is very slow, the estimated support after a few hundred iterations can be used along with randomized rounding to further refine the solution. The computation time on the same computer for the projected subgradient algorithm that solves \( (14) \) is around 8.84 seconds for 1000 iterations while SeDuMi takes around 4.03 seconds to solve the SDP problem in \( (14) \).

A practical estimator does not meet the CRB in some cases (for instance at low SNRs or finite data records). Therefore, the sensors obtained with a specific \( R_e \) would lead to an underestimate of the desired MSE. We can account for this gap by choosing \( R_e \) appropriately. To this end, we give the entire solution path of the selected sensors for different values of \( R_e \) in Fig. 3a. The solution path can be efficiently computed by increasing \( R_e \). The sensors corresponding to some \( R_e \) can then be used to meet the desired MSE requirement. The non-linear model in \( (25) \) is solved in the least-squares sense iteratively using Gauss-Newton’s method with 10 iterations [26]. The maximum root-MSE (RMSE), maximum root-CRB, average RMSE, and average CRB of the location estimates of a target within the target area using the selected sensors (as shown in the solution path) for different values of \( R_e \) are shown in Fig. 3b. For the considered scenario, both the maximum and average root-CRB satisfy the performance constraint which is
Fig. 2: (a) The projected subgradient algorithm used to solve (14). (b) Performance of the projected subgradient algorithm. The thresholds are computed using $R_e = 20$ cm, and $P_e = 0.9$.

Fig. 3: (a) Solution path of the sensors selected for different values of $R_e$ and $P_e = 0.9$. Maximum RMSE based on selected sensors can be seen on the top of this plot. (b) Maximum and average RMSE of the location estimates based on Gauss-Newton’s method, the corresponding maximum and average root-CRB, and the performance constraint in (27) for different values of $R_e$, and $P_e = 0.9$.

given by the inequality in (27). The performance constraint is shown as a red solid line in Fig. 3b. The maximum RMSE does not satisfy the accuracy requirement specified by a certain $R_e$, and this can be corrected by using an appropriate (lower) $R_e$. Moreover, for the considered scenario, the gap between the average RMSE and the performance constraint is still reasonable. We also show the maximum RMSE on top of Fig. 3a.

The proposed framework is very general, and can be applied to a variety of data models as long as (a1) and (a2) are valid. To illustrate this we next consider a few more measurement models. The sensor selection based on bearing measurements is illustrated in Fig. 4. Here, we use $R_e = 25$ cm, and $P_e = 0.9$. The selection results for the RSS based measurement model is shown in Fig. 5. We use $\sigma_r^2_{dB} = 2$ dB, $P_e = 0.9$, and $R_e = 5$ m. Sensor selection results based on energy measurements are shown in Fig. 6, where we use $R_e = 10$ cm, and $P_e = 0.9$. An illustration of the field generated by a point source at location $[25, 25]^T$ m with unit amplitude is also shown here.

The FIM for all the considered measurement models has a common structure, and it decreases as the distance $d_m$ increases. However, the rate at which it decreases is different for different models. Anyway, as a result of this decrease, the optimization problem leads to a sensor selection that is close to the target area (in the Euclidean distance sense) for all the considered models.

VII. CONCLUSIONS

Sensor selection is an important design problem in sensor networks. The sensor selection problem can be described as the problem of selecting the best subset of sensors that guarantees a certain specified performance measure. The sensor selection enables deployment of the sensors with guarantees on the
resulting estimation accuracy. This also minimizes the hardware, communications, and resulting processing costs in large-scale networks. We focus on observations that follow a nonlinear model. The proposed framework is valid as long as the observations are independent, and their pdfs are regular. We use a number of functions related to the FIM as a performance measure. The original nonconvex optimization problem is relaxed using convex relaxation techniques which can then be efficiently solved in polynomial time. To handle large-scale problems, we have also presented a projected subgradient algorithm. This also enables easy distributed implementations. The proposed framework is applied to sensor placement design for a number of different models related to localization.

**APPENDIX A**

**PERFORMANCE THRESHOLDS**

*Trace and minimum eigenvalue constraints*

We can relate the accuracy requirement and the CRB using Chebyshev’s inequality [28]

$$
\Pr(\|\epsilon\|_2 \geq R_e) \leq \frac{\operatorname{tr}\{C\}}{R_e^2}
$$

which can be equivalently expressed as

$$
\Pr(\|\epsilon\|_2 \leq R_e) \leq 1 - \frac{\operatorname{tr}\{C\}}{R_e^2}.
$$

Combining this inequality together with

$$
\Pr(\|\epsilon\|_2 \leq R_e) \geq P_e
$$

in (10) results in the following sufficient condition

$$
\operatorname{tr}\{C\} \leq \lambda_{\operatorname{tr}} = (1 - P_e)R_e^2.
$$

Each eigenvalue of $C^{-1}$ is greater than $\lambda_{\min}(F)$, and as a result, $\operatorname{tr}\{C\} \leq N\lambda_{\min}^{-1}(F)$. Hence, a stronger sufficient condition (with a smaller feasible set) is $N\lambda_{\min}^{-1}(F) \leq (1 - P_e)R_e^2$. 

---

**Fig. 4:** (a) Sensor selection based on bearing measurements with $M = 80$ available sensors. (b) Sensor selection solved with minimum eigenvalue constraints using $\ell_1$-norm and log-based heuristics. The thresholds are computed using $R_e = 25$ cm, and $P_e = 0.9$. The noise variance is $\sigma_n^2 = 2 \times 10^{-5}$ square-degrees.

**Fig. 5:** (a) Sensor selection based on RSS with $M = 80$ available sensors. (b) Sensor selection solved with minimum eigenvalue constraints using $\ell_1$-norm and log-based heuristics. The thresholds are computed using $R_e = 5$ m, and $P_e = 0.9$. We use $\sigma_r, \text{dB} = 2$ dB.
or equivalently [29]
\[
\lambda_{\text{min}}(\mathbf{F}) \geq \lambda_{\text{eig}} = \frac{N}{R_e^2} \left( \frac{1}{1 - P_e} \right).
\]

**Determinant constraint**

The determinant constraint is related to the volume or the mean radius of the confidence ellipsoid that contains \( e = \theta - \bar{\theta} \) with probability \( P_e \). Such a confidence ellipsoid can be expressed as
\[
\mathcal{E} = \{ \epsilon \mid \epsilon^T \mathbf{F}^{-1} \epsilon \leq \xi \},
\]
where \( \xi \) is a constant that depends on \( P_e \). Assuming \( \mathbf{F} \) has ordered eigenvalues \( \lambda_{\text{max}} \geq \lambda_2 \cdots \geq \lambda_{\text{min}} \), the length of the \( n \)th semi-axis of the ellipsoid \( \mathcal{E} \) will be \( \sqrt{\xi/\lambda_n} \). The geometric mean radius of the confidence ellipsoid \( \mathcal{E} \) given by
\[
\bar{R}_e = \sqrt{\xi/(\det(\mathbf{F}))^{1/2N}},
\]
gives a quantitative measure of how informative the observations are. For the estimates to be within the confidence ellipsoid \( \mathcal{E} \), we use the constraint
\[
\ln \det(\mathbf{F}) \geq 2N \ln \frac{\sqrt{\xi}}{\bar{R}_e} = \lambda_{\text{det}},
\]
where \( \bar{R}_e \) and \( \sqrt{\xi} \) specify the required accuracy, and are assumed to be known. A typical choice for \( \xi \) is constant chi-squared values, i.e., \( \xi = F^{-1}_{\chi^2}(P_e) \). Here, \( F^{-1}_{\chi^2} \) is the cumulative distribution function of a chi-squared random variable with \( N \) degrees of freedom. This performance measure is related to the D-optimality.

**APPENDIX B**

**DERIVATION OF THE DUAL PROBLEM**

Consider the optimization problem (14) as the primal problem. To this problem, we then associate the following dual variables or Lagrangian multipliers: \( \mathbf{Z} \in \mathbb{S}^{DN} \) with the LMI constraint; \( \nu_m \in \mathbb{R} \) and \( \mu_m \in \mathbb{R} \) with the \( w_m \geq 0 \) and \( w_m \leq 1 \) constraints, respectively. The Lagrangian is
\[
\mathcal{L}(\mathbf{w}, \mathbf{Z}, \mu, \nu) = 1^T_M \mathbf{w} - \sum_{m=1}^{M} w_m \text{tr} \{\mathbf{F}_m \mathbf{Z}\} - \text{tr} \{ -\lambda_{\text{eig}} \mathbf{Z} \} + \mu^T (\mathbf{w} - 1_M) - \nu^T \mathbf{w}
\]
\[
= \sum_{m=1}^{M} w_m (1 - \text{tr} \{\mathbf{F}_m \mathbf{Z}\} + \mu_m - \nu_m) - 1^T_M \mu - \text{tr} \{ -\lambda_{\text{eig}} \mathbf{Z} \}.
\]

The Lagrange dual function
\[
\phi(\mathbf{Z}, \mu, \nu) = \inf_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mathbf{Z}, \mu, \nu)
\]
is given as (29). The dual problem which is also an SDP can therefore be expressed as
\[
\arg \max_{\mathbf{Z}, \mu, \nu} \lambda_{\text{eig}} \text{tr} \{ \mathbf{Z} \} - 1^T_M \mu
\]
s.t. \( \text{tr} \{ \mathbf{F}_m \mathbf{Z}\} + \nu_m = 1 + \mu_m, \ m = 1, 2, \ldots, M, \ Z \succeq 0, \ \mu_m \geq 0, \ \nu_m \geq 0, \ m = 1, 2, \ldots, M. \)

where \( \mathbf{Z} \in \mathbb{S}^{DN}_+ \) (we use the fact that \( \mathbb{S}^{DN}_+ \) is self-dual), \( \mu = [\mu_1, \mu_2, \ldots, \mu_M]^T \in \mathbb{R}^M \) and \( \nu = [\nu_1, \nu_2, \ldots, \nu_M]^T \in \mathbb{R}^M \) are the dual variables. By eliminating \( \nu_m \), the dual problem is simplified to
\[
\arg \max_{\mathbf{Z}, \mu} \lambda_{\text{eig}} \text{tr} \{ \mathbf{Z} \} - 1^T_M \mu
\]
s.t. \( \text{tr} \{ \mathbf{F}_m \mathbf{Z}\} \leq 1 + \mu_m, \ m = 1, 2, \ldots, M, \)
\[
Z \succeq 0, \ \mu_m \geq 0, \ m = 1, 2, \ldots, M. \)

**APPENDIX C**

**PROJECTED NEWTON’S METHOD**

In order to analyze the complexity of the interior point methods, we briefly describe the projected Newton’s method.
\[ \phi(Z, \mu, \nu) = \inf_{w} \mathcal{L}(w, Z, \mu, \nu) = \begin{cases} -\operatorname{tr}\{-\lambda_{\text{eig}} Z\} - 1_T^T \mu & \text{if } \operatorname{tr}(F_m Z) + \nu_m = 1 + \mu_m, m = 1, 2, \ldots, M, \text{ otherwise}, \\ -\infty & \end{cases} \]  

The Newton’s method for an SDP problem in the inequality form is adapted to suit our problem [11, Pg. 619]. The optimization problem in (14) can be approximated using the log-determinant barrier function which is given as

\[ \arg \min_{w \in [0,1]^N} \psi(w) = t \mathbf{1}_T^T w - \ln \det \left\{ \sum_{m=1}^{M} w_m F_m - \lambda_{\text{eig}} I_{DN} \right\}, \]

where \( t > 0 \) is a parameter to tune the approximation. The projected Newton’s update equation is given by

\[ w^{k+1} = P_W \left( w^k - \alpha_k \left( \frac{\partial^2 \psi(w^k)}{\partial w_i^2} - \mathbf{1}_T \right) \right), \]

where the entries of the Hessian matrix are given by

\[ \frac{\partial^2 \psi(w)}{\partial w_i^2} \bigg|_{w=w^k} = \operatorname{tr}(S^{-1} F_i S^{-1} F_j), i, j = 1, 2, \ldots, M, \]

and the entries of the gradient vector are given by

\[ \frac{\partial \psi(w)}{\partial w_i} \bigg|_{w=w^k} = t + \operatorname{tr}(S^{-1} F_i), i = 1, 2, \ldots, M. \]

Here, we have introduced the matrix \( S = \sum_{m=1}^{M} w_m F_m - \lambda_{\text{eig}} I_{DN} \), and recall the projector operator \( P_W() \) defined in (16). The step-length \( \alpha_k \) is chosen by line-search.

**APPENDIX D**

**POWER ITERATIONS FOR COMPUTING THE MINIMUM EIGENVALUE**

We briefly describe the power iterations [37] to compute the minimum eigenvalue of a matrix \( F \in \mathbb{S}^N \). Assuming \( F \) has ordered eigenvalues \( \lambda_{\text{max}} \geq \lambda_2 \cdots \geq \lambda_{\text{min}} \), the power iterations

\[ w^{k+1} = \frac{F w^k}{\|F w^k\|_2}, \quad \text{and} \quad \lambda^{k+1} = \frac{(w^{k+1})^T F w^{k+1}}{\|w^{k+1}\|_2}, \]

converge to the eigenvector corresponding to the maximum eigenvalue \( \lambda_{\text{max}} \) and the maximum eigenvalue \( \lambda_{\text{max}} \), respectively, as \( k \to \infty \). Here, we use \( v^0 = [1, 0, \ldots, 0]^T \). By forming a matrix \( F = \lambda_{\text{max}} I_N - F \) which has the dominant eigenvalue \( \lambda_{\text{max}} - \lambda_{\text{min}} \), we can apply the above power iterations on \( F \) to compute \( \lambda_{\text{max}} - \lambda_{\text{min}} \) and \( v_{\text{min}} \), and thus the minimum eigenvalue of \( F \) and it’s corresponding eigenvector.

REFERENCES

[1] S. P. Chepuri, G. Leus, and A.-J. van der Veen, “Sparsity-exploiting anchor placement for localization in sensor networks,” in *Proc. of the 21st European Signal Processing Conference (EUSIPCO)*, Sept. 2013.

[2] G. Mateos and K. Rajawat, “Dynamic network cartography: Advances in network health monitoring,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 129–143, 2013.

[3] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-I. Kim, H. Zhu, and B. F. Wollenberg, “Monitoring and optimization for power grids: A signal processing perspective,” *IEEE Signal Process. Mag.*, 2013.

[4] F. Gustafsson and F. Gunnarsson, “Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, July 2005.

[5] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, July 2005.

[6] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R.L. Moses, and N.S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54 – 69, Jul. 2005.

[7] T. van Waterschoot and G. Leus, “Static field estimation using a wireless sensor network based on the finite element method,” in *Proc. of 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMPAS)*, 2011, pp. 369–372.

[8] Haotian Zhang, J.M.F. Moura, and B. Krog, “Dynamic field estimation using wireless sensor networks: Tradeoffs between estimation error and communication cost,” *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2383–2395, 2009.

[9] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[10] F. Pukelsheim, *Optimal design of experiments*, vol. 50, SIAM, 1993.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.

[12] V. Kekatos, G.B. Giannakis, and B. Wollenberg, “Optimal placement of phasor measurement units via convex relaxation,” *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1521–1530, 2012.

[13] A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies,” *The Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[14] A. Krause and C. Guestrin, “Near-optimal observation selection using submodular functions,” in *AAAI*, vol. 7, pp. 1650–1654, 2007.

[15] A. Krause, H.B. McMahan, C. Guestrin, and A. Gupta, “Robust submodular observation selection,” *Journal of Machine Learning Research*, 2008.

[16] L. Yao, W.A. Sethares, and D.C. Kammer, “Sensor placement for on-orbit modal identification via a genetic algorithm,” *AIAA journal*, vol. 31, no. 10, pp. 1922–1928, 1993.

[17] M. Shamaih, S. Banerjee, and H. Vikalo, “Greedy sensor selection: Leveraging submodularity,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 2572–2577.

[18] E. Masazade, M. Fardad, and P.K. Varshney, “Sparsity-promoting extended kalman filtering for target tracking in wireless sensor networks,” *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 845–848, 2012.

[19] A. Caru, “Sensor scheduling via compressed sensing,” in *Proc. of 13th Conference on Information Fusion (FUSION)*, 2010, pp. 1–8.

[20] Yinfei Fu, Qing Ling, and Zhi Tian, “Distributed sensor allocation for multi-target tracking in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1135–1146, March 2014.

[21] I. Ford, D.M. Titterington, and Christos P. Kitsos, “Recent advances in nonlinear experimental design,” *Technometrics*, vol. 31, no. 1, pp. 49–60x, 1989.

[22] D. Bajovic, B. Sinopoli, and J. Xavier, “Sensor selection for event detection in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4938–4953, Oct 2011.

[23] V. Kekatos and G.B. Giannakis, “From sparse signals to sparse residuals for robust sensing,” *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3355–3368, 2011.

[24] I.D. Schizas, “Distributed informative-sensor identification via sparsity-aware matrix decomposition,” *IEEE Trans. Signal Process.*, vol. PP, no. 99, pp. 1–1, 2013.

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

[26] T.J. Rothenberg, “Identification in parametric models,” *Econometrica: Journal of the Econometric Society*, pp. 577–591, 1971.
[28] Thomas M Cover and Joy A Thomas, *Elements of information theory*, John Wiley & Sons, 2012.
[29] T. Wang, G. Leus, and L. Huang, “Ranging energy optimization for robust sensor positioning based on semidefinite programming,” *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4777–4787, Dec. 2009.
[30] B. Polyak, M. Khlebnikov, and P. Shcherbakov, “An lmi approach to structured sparse feedback design in linear control systems,” in *The European Control Conference*, 2013.
[31] D.L. Donoho, “For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution,” *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.
[32] J. Lfberg, “Yalmip : A toolbox for modeling and optimization in MATLAB,” in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
[33] Jos F. Sturm, “Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones,” 1998.
[34] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.0 beta,” http://cvxr.com/cvx, Sept. 2012.
[35] S. Boyd, L. Xiao, and A. Mutapcic, “Subgradient methods,” http://www.stanford.edu/class/ee364b/notes/subgrad_method_notes.pdf, 2003.
[36] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific optimization and computation series, Athena Scientific, 1999.
[37] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.
[38] Lin Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
[39] E. Candès, M. Wakin, and S. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.
[40] M. Gavish and A.J. Weiss, “Performance analysis of bearing-only target location algorithms,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, pp. 817–828, Jul 1992.