

Near-Optimal Sensor Scheduling for Batch State Estimation: Complexity, Algorithms, and Limits

Vasileios Tzoumas¹, Ali Jadbabaie², George J. Pappas¹

Abstract—In this paper, we focus on batch state estimation for linear systems. This problem is important in applications such as environmental field estimation, robotic navigation, and target tracking. Its difficulty lies on that limited operational resources among the sensors, e.g., shared communication bandwidth or battery power, constrain the number of sensors that can be active at each measurement step. As a result, sensor scheduling algorithms must be employed. Notwithstanding, current sensor scheduling algorithms for batch state estimation scale poorly with the system size and the time horizon. In addition, current sensor scheduling algorithms for Kalman filtering, although they scale better, provide no performance guarantees or approximation bounds for the minimization of the batch state estimation error. In this paper, one of our main contributions is to provide an algorithm that enjoys both the estimation accuracy of the batch state scheduling algorithms and the low time complexity of the Kalman filtering scheduling algorithms. In particular: 1) our algorithm is near-optimal: it achieves a solution up to a multiplicative factor $1/2$ from the optimal solution, and this factor is close to the best approximation factor $1/e$ one can achieve in polynomial time for this problem; 2) our algorithm has (polynomial) time complexity that is not only lower than that of the current algorithms for batch state estimation; it is also lower than, or similar to, that of the current algorithms for Kalman filtering. We achieve these results by proving two properties for our batch state estimation error metric, which quantifies the square error of the minimum variance linear estimator of the batch state vector: a) it is supermodular in the choice of the sensors; b) it has a sparsity pattern (it involves matrices that are block tri-diagonal) that facilitates its evaluation at each sensor set.

I. INTRODUCTION

Search and rescue [1], environmental field estimation [2], robotic navigation [3], and target tracking [4] are only a few of the challenging information gathering problems that employ the monitor capabilities of sensor networks [5]. In particular, all these problems face the following three main challenges:

- they involve systems whose evolution is largely unknown, corrupted with noisy inputs [4], and sensors with limited sensor capabilities, corrupted with measurement noise [6].

¹The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6228 USA (email: {vtzoumas, pappasg}@seas.upenn.edu).

²The author is with the Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (email: jadbabai@mit.edu).

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, in part by AFOSR Complex Networks Program and in part by AFOSR MURI CHASE.

- they involve systems that change over time [2], and as a result, necessitate both spacial and temporal deployment of sensors in the environment. At the same time:
- they involve operational constraints, such as limited bandwidth and battery life, which limit the number of sensors that can be simultaneously used (i.e., be switched-on) in the information gathering process [7].

As a result of these challenges, researchers focused on the following question: “How do we select at each measurement step only a few sensors so to minimize the estimation error despite the above challenges?” The effort to answer this question resulted to the problem of sensor scheduling [7]: in particular, sensor scheduling offers a formal methodology to use at each measurement time only a few sensors and obtain an optimal trade-off between the estimation accuracy and the usage of the limited operational resource (e.g., the shared bandwidth). Clearly, sensor scheduling is a combinatorial problem of exponential complexity [5].

In this paper, we focus on the following instance of this problem:

Problem 1 (Sensor Scheduling for Minimum Variance Batch State Estimation): Consider a time-invariant linear system, whose state at time t_k is denoted as $x(t_k)$, a set of m sensors, and a fixed set of K measurement times t_1, t_2, \dots, t_K . In addition, consider that at each t_k at most r_k sensors can be used, where $r_k \leq m$. At each t_k select a set of r_k sensors so to minimize the square estimation error of the minimum variance linear estimator of the batch state vector $(x(t_1), x(t_2), \dots, x(t_K))$.

There are two classes of sensor scheduling algorithms, that trade-off between the estimation accuracy of the batch state vector and their time complexity: these for Kalman filtering, and those for batch state estimation. In more detail:

Kalman filtering algorithms: These algorithms sacrifice estimation accuracy over reduced time complexity. The reason is that they are sequential algorithms: at each t_k , they select the sensors so to minimize the square estimation error of the minimum variance linear estimator of $x(t_k)$ (given the measurements up to t_k). Therefore, their objective is to minimize the sum of the square estimation errors of $x(t_k)$ across the measurement times t_k [8]. However, this sum is only an upper bound to the square estimation error of the batch state vector $(x(t_1), x(t_2), \dots, x(t_K))$. Thus, the Kalman filtering algorithms lack on estimation accuracy with respect to the batch state estimation algorithms.

Batch state estimation algorithms: These algorithms sacrifice time complexity over estimation accuracy. The reason is that they perform global optimization, in accordance to

Problem 1. Therefore, however, they lack on time complexity with respect to the Kalman filtering algorithms.

Notwithstanding, in several recent robotic applications, batch estimation algorithms have been proven competitive in their time complexity to their filtering counterparts [9], [10]. The reason is that sparsity patterns emerge in these applications, that reduce the time complexity of their batch estimation algorithms to an order similar to that of the filtering algorithms [11]. Thereby, the following question on Problem 1 arises:

Question 1: “Is there an algorithm for Problem 1 that enjoys both the estimation accuracy of the batch state algorithms and the low time complexity of the Kalman filtering algorithms?”

Literature review on sensor scheduling algorithms for batch state estimation: The most relevant paper on Problem 1 is [12], where an algorithm based on convex relaxation is provided. This algorithm scales poorly with the system’s size and number of measurement times. In addition, it provides no approximation performance guarantees.

Literature review on sensor scheduling algorithms for Kalman filtering: Several papers in this category have focused on myopic algorithms [13]; such algorithms, however, often perform poorly [14]. Other papers have focused on algorithms that use: tree pruning [15], convex optimization [16], quadratic programming [17], or submodular function maximization [18], [19]. Nevertheless, these algorithms provide no performance guarantees on the batch state estimation error, or have time complexity that scales poorly with the system’s size and number of measurement times [15] [17]. To reduce the time complexity of these algorithms, papers have also proposed periodic sensor schedules [8].

Contributions: We now present our contributions:

- 1) We prove that Problem 1 is NP-hard.
- 2) We provide an algorithm for Problem 1 (Algorithm 1) that answers Question 1 positively. The reasons are two:
 - i) Algorithm 1 is near-optimal: it achieves a solution that is up to a multiplicative factor $1/2$ from the optimal solution. In addition, this multiplicative factor is close to the factor $1/e$ which we prove to be the best approximation factor one can achieve in polynomial time for Problem 1 in the worst-case.
 - ii) Algorithm 1 has (polynomial) time complexity that is not only lower than that of the state of the art scheduling algorithms for batch state estimation; it is also lower than, or similar to, that of the state of the art scheduling algorithms for Kalman filtering. For example, it has similar complexity to the state of the art periodic scheduling algorithm in [8] (in particular: lower for K large enough), and lower than the complexity of the algorithm in [16].

Overall, in response to Question 1, Algorithm 1 enjoys both the higher estimation accuracy of the batch state estimation approach (compared to the Kalman filtering approach, that only approximates the batch state estimation error with an upper bound) and the low time

complexity of Kalman filtering approach.

- 3) We prove limits on the minimization of the square error of the minimum variance estimator of $(x(t_1), x(t_2), \dots, x(t_K))$ with respect to the scheduled sensors. For example, we prove that the number r_k of used sensors at each measurement time must increase linearly with the system size for fixed estimation error and number of measurement times K ; this is a fundamental limit, especially for large-scale systems.

Our technical contributions: We achieve our aforementioned contributions by proving the following two:

a) *Supermodularity in Problem 1:* We prove that our estimation metric, that quantifies the square error of the minimum variance estimator of $(x(t_1), x(t_2), \dots, x(t_K))$, is a supermodular function in the choice of the used sensors. This result becomes important when we compare it to results on the multi-step Kalman filtering that show that the corresponding estimation metric in this case is neither supermodular nor submodular [18], [19].¹

In addition, this submodularity result cannot be reduced to the batch estimation problem in [21]. The main reasons are two: i) we consider sensors that can measure any linear combination of the element of $x(t_k)$, in contrast to [21], where each sensor measures directly only one element of $x(t_k)$. Nonetheless, the latter assumption is usually infeasible in dynamical systems [22]; ii) our error metric is relevant to estimation problems for dynamical systems and different to the submodular information gain considered in [21].

b) *Sparsity in Problem 1:* We identify a sparsity pattern in our error metric, that facilitates the latter’s evaluation at each sensor set. In particular, we prove that the error covariance of the minimum variance linear estimator of the batch state vector is block tri-diagonal.

We organize the rest of the paper as follows: In Section II we present formally Problem 1. In Section III, we present in three subsections our main results: in Section III-A, we prove that our sensor scheduling problem is NP-hard. In Section III-B, we derive our near-optimal approximation algorithm. In Section III-C, we prove limits on the minimization of the batch state estimation error with respect to the used sensors. Section IV concludes the paper with our future work. *Most of the proofs are omitted due to space constraints, and can be found in the full version of this paper located in the authors’ websites.*²

¹The observation of [19] is also important as it disproves previous results in the literature [20].

²*Standard notation is presented in this footnote:* We denote the set of natural numbers $\{1, 2, \dots\}$ as \mathbb{N} , the set of real numbers as \mathbb{R} , and the set $\{1, 2, \dots, n\}$ as $[n]$ ($n \in \mathbb{N}$). The empty set is denoted as \emptyset . Given a set \mathcal{X} , $|\mathcal{X}|$ is its cardinality. Matrices are represented by capital letters and vectors by lower-case letters. We write $A \in \mathcal{X}^{n_1 \times n_2}$ ($n_1, n_2 \in \mathbb{N}$) to denote a matrix of n_1 rows and n_2 columns whose elements take values in \mathcal{X} . Moreover, for a matrix A , A^\top is its transpose, and $[A]_{ij}$ is its element at the i -th row and j -th column. In addition, $\|A\|_2 \equiv \sqrt{A^\top A}$ is its spectral norm, and $\det(A)$ its determinant. Furthermore, if A is positive semi-definite or positive definite, we write $A \succeq 0$ and $A \succ 0$, respectively. I is the identity matrix; its dimension is inferred from the context. Similarly for the zero matrix 0 . Finally, for a random variable $x \in \mathbb{R}^n$, $\mathbb{E}(x)$ is its expected value, and $\mathbb{C}(x)$ its covariance.

II. PROBLEM FORMULATION

In the following paragraphs, we present our sensor scheduling problem for batch state estimation. To this end, we first build our system and measurement framework. Then, we define our sensor scheduling framework and, finally, present our sensor scheduling problem.

We start in more detail with the system model:

System Model: We consider the linear time-invariant system:

$$\dot{x}(t) = Ax(t) + Bu(t) + Fw(t), t \geq t_0, \quad (1)$$

where t_0 is the initial time, $x(t) \in \mathbb{R}^n$ ($n \in \mathbb{N}$) the state vector, $\dot{x}(t)$ the time derivative of $x(t)$, $u(t)$ the exogenous input, and $w(t)$ the process noise. The system matrices A , B and F are of appropriate dimensions. We consider that $u(t)$, A , B and F are known. Our main assumption on $w(t)$ is found in Assumption 1, that is presented after our measurement model.

Remark 1: Our results extend to continuous and discrete time-variant systems, as explained in detail in Section III (Corollaries 1 and 2).

We introduce the measurement model:

Measurement Model: We consider m sensors:

$$z_i(t) = C_i x(t) + v_i(t), i \in [m], \quad (2)$$

where $z_i(t)$ is the measurement taken by sensor i at time t , $C_i \in \mathbb{R}^{d_i \times n}$ ($d_i \in \mathbb{N}$) is sensor's i measurement matrix, and $v_i(t)$ is its measurement noise.

We make the following assumption on $x(t_0)$, $w(t)$ and $v_i(t)$:

Assumption 1: For all $t, t' \geq t_0$, $t \neq t'$, and all $i \in [m]$: $x(t_0)$, $w(t)$, $w(t')$, $v_i(t)$ and $v_i(t')$ are uncorrelated; in addition, $x(t_0)$, $w(t)$ and $v_i(t)$ have positive definite covariance.

We now introduce the sensor scheduling model:

Sensor Scheduling Model: The m sensors in (2) are used at K scheduled measurement times $\{t_1, t_2, \dots, t_K\}$. Specifically, at each t_k only r_k of these m sensors are used ($r_k \leq m$), resulting in the batch measurement vector $y(t_k)$:

$$y(t_k) = S(t_k)z(t_k), k \in [K], \quad (3)$$

where $z(t_k) \equiv (z_1^\top(t_k), z_2^\top(t_k), \dots, z_m^\top(t_k))^\top$, and $S(t_k)$ is the sensor selection matrix: it is a block matrix, composed of matrices $[S(t_k)]_{ij}$ ($i \in [r_k]$, $j \in [m]$) such that $[S(t_k)]_{ij} = I$ if sensor j is used at t_k , and $[S(t_k)]_{ij} = 0$ otherwise. We consider that each sensor can be used at most once at each t_k , and as a result, for each i there is one j such that $[S(t_k)]_{ij} = I$ while for each j there is at most one i such that $[S(t_k)]_{ij} = I$.

We now present the sensor scheduling problem we study in this paper. To this end, we use two notations:

Notation: First, we set $\mathcal{S}_k \equiv \{j : \text{there exists } i \in [r_k], [S(t_k)]_{ij} = 1\}$; that is, \mathcal{S}_k is the set of indices that correspond to used sensors at t_k . Second, we set $\mathcal{S}_{1:K} \equiv (\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K)$.

Problem 1 (Sensor Scheduling for Minimum Variance Batch State Estimation): Given a set of measurement times t_1, t_2, \dots, t_K , select at each t_k to use a subset of r_k sensors, out of the m sensors in (2), so to minimize the log det of the error covariance of the minimum variance linear estimator of $x_{1:K} \equiv (x(t_1), x(t_2), \dots, x(t_K))$. In mathematical notation:

$$\begin{aligned} & \underset{\mathcal{S}_k \subseteq [m], k \in [K]}{\text{minimize}} && \log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})) \\ & \text{subject to} && |\mathcal{S}_k| \leq r_k, k \in [K], \end{aligned}$$

where $\hat{x}_{1:K}$ is the minimum variance linear estimator of $x_{1:K}$, and $\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})$ its error covariance given $\mathcal{S}_{1:K}$.

Two remarks follow on the definition of Problem 1. In the first remark we explain why we focus on $\hat{x}_{1:K}$, and in the second why we focus on $\log \det(\Sigma(\hat{x}_{1:K}))$.

Notation: For notational simplicity, we use $\Sigma(\hat{x}_{1:K})$ and $\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})$ interchangeably.

Remark 2: We focus on the minimum variance linear estimator $\hat{x}_{1:K}$ because of its optimality: it minimizes among all linear estimators of $x_{1:K}$ the estimation error $\mathbb{E}(\|x_{1:K} - \hat{x}_{1:K}\|_2^2)$, where the expectation is taken with respect to $y(t_1), y(t_2), \dots, y(t_K)$ [6]. Because $\hat{x}_{1:K}$ is also unbiased (that is, $\mathbb{E}(\hat{x}_{1:K}) = x_{1:K}$, where the expectation is taken with respect to $y(t_1), y(t_2), \dots, y(t_K)$), we equivalently say that $\hat{x}_{1:K}$ is the minimum variance estimator of $x_{1:K}$.

We compute the error covariance of $\hat{x}_{1:K}$ in Appendix I.

Remark 3: We focus on the estimation error metric $\log \det(\Sigma(\hat{x}_{1:K}))$ because when it is minimized the probability that the estimation error $\|x_{1:K} - \hat{x}_{1:K}\|_2^2$ is small is maximized. To quantify this statement, we note that this error metric is related to the η -confidence ellipsoid of $x_{1:K} - \hat{x}_{1:K}$ [16]: Specifically, the η -confidence ellipsoid is the minimum volume ellipsoid that contains $x_{1:K} - \hat{x}_{1:K}$ with probability η , that is, it is the $\mathcal{E}_\epsilon \equiv \{x : x^\top \Sigma(\hat{x}_{1:K})x \leq \epsilon\}$, where ϵ is the quantity $F_{\chi_{n(k+1)}^2}^{-1}(\eta)$, and $F_{\chi_{n(k+1)}^2}$ the cumulative distribution function of a χ -squared random variable with $n(k+1)$ degrees of freedom [23]. Thus, its volume

$$\text{vol}(\mathcal{E}_\epsilon) \equiv \frac{(\epsilon\pi)^{n(k+1)/2}}{\Gamma(n(k+1)/2 + 1)} \det\left(\Sigma(\hat{x}_{1:K})^{1/2}\right), \quad (4)$$

where $\Gamma(\cdot)$ denotes the Gamma function [23], quantifies the estimation error of the optimal estimator $\hat{x}_{1:K}$. Therefore, by taking the logarithm of (4), we validate that when the $\log \det(\Sigma(\hat{x}_{1:K}))$ is minimized the probability that the estimation error $\|x_{1:K} - \hat{x}_{1:K}\|_2^2$ is small is maximized.

III. MAIN RESULTS

Our main results are presented in three sections:

- In Section III-A, we prove that Problem 1 is NP-hard.
- In Section III-B, we derive a provably near-optimal approximation algorithm for Problem 1. In addition, we emphasize on its time complexity and compare it to that of existing sensor scheduling algorithms for two categories: batch state estimation, and Kalman filtering.
- In Section III-C, we prove limits on the optimization of the estimation error $\mathbb{E}(\|x_{1:K} - \hat{x}_{1:K}\|_2^2)$ with respect to the scheduled sensors.

Algorithm 1 Approximation algorithm for Problem 1.

Input: Number of measurement times K , scheduling constraints r_1, r_2, \dots, r_K , estimation error function $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})) : \mathcal{S}_k \subseteq [m], k \in [K] \mapsto \mathbb{R}$

Output: Sensor sets $(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K)$ that approximate the solution to Problem 1, as quantified in Theorem 2

$k \leftarrow 1, \mathcal{S}_{1:0} \leftarrow \emptyset$

while $k \leq K$ **do**

1. Apply Algorithm 2 to

$$\min_{\mathcal{S} \subseteq [m]} \{ \log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:k-1}, \mathcal{S})) : |\mathcal{S}| \leq r_k \} \quad (5)$$

2. Denote as \mathcal{S}_k the solution Algorithm 2 returns

3. $\mathcal{S}_{1:k} \leftarrow (\mathcal{S}_{1:k-1}, \mathcal{S}_k)$

4. $k \leftarrow k + 1$

end while

A. Computational Complexity of Sensor Scheduling for Batch State Estimation

In this section, we characterize the computational complexity of Problem 1. In particular, we prove:

Theorem 1: The problem of sensor scheduling for minimum variance batch state estimation (Problem 1) is NP-hard.

Proof: The proof is omitted due to space constraints.

Notwithstanding, we note that the proof is complete by finding an instance of Problem 1 that is equivalent to the NP-hard minimal observability problem introduced in [24] [25]. ■

Due to Theorem 1, for the polynomial time solution of Problem 1 we need to appeal to approximation algorithms. To this end, in Section III-B, we provide an efficient provably near-optimal approximation algorithm:

B. Algorithm for Sensor Scheduling for Minimum Variance Batch State Estimation

We propose Algorithm 1 for Problem 1 (Algorithm 1 uses Algorithm 2 as a subroutine); with the following theorem, we quantify its approximation performance and time complexity.

Theorem 2: The theorem has two parts:

1) Approximation performance of Algorithm 1: *Algorithm 1 returns sensors sets $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K$ that:*

- satisfy all the feasibility constraints of Problem 1: $|\mathcal{S}_k| \leq r_k, k \in [K]$
- achieve an error value $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K}))$, where $\mathcal{S}_{1:K} \equiv (\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K)$, such that:

$$\frac{\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})) - OPT}{MAX - OPT} \leq \frac{1}{2}, \quad (6)$$

where OPT is the (optimal) value to Problem 1, and MAX is the maximum (worst) value to Problem 1 ($MAX \equiv \max_{\mathcal{S}'_{1:K}} \log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}'_{1:K}))$).

2) Time complexity of Algorithm 1: *Algorithm 1 has time complexity of order $O(n^{2.4} K \sum_{k=1}^K r_k^2)$.*

Theorem 2 extends to continuous and discrete time-variant systems as follows:

Corollary 1: Consider the time-variant version of (1):

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + F(t)w(t), t \geq t_0. \quad (7)$$

1) *Part 1 of Theorem 2 holds.*

2) *Part 2 of Theorem 2 holds if the time complexity for computing each transition matrix $\Phi(t_{k+1}, t_k)$ [22], where $k \in [K - 1]$, is $O(n^3)$.*³

Corollary 2: Consider the discrete time version of (7):

$$x[k + 1] = A_k x[k] + B_k u[k] + F_k w[k], k \geq k_0. \quad (8)$$

Similarly, consider the discrete time counterparts of the sensor model (2), Assumption 1, and the sensor scheduling model (3).

1) *Part 1 of Theorem 2 holds.*

2) *Part 2 of Theorem 2 holds if A_k in (8) is full rank for all $k \in [K]$.*

We follow-up with several remarks on Theorem 2:

Remark 4: (Approximation quality of Algorithm 1) Theorem 2 quantifies the worst-case performance of Algorithm 1 across all values of Problem 1's parameters. The reason is that the right-hand side of (6) is constant. In particular, (6) guarantees that for any instance of Problem 1, the distance of the approximate value $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K}))$ from OPT is at most $1/2$ the distance of the worst (maximum) value MAX from OPT . In addition, this approximation factor is near to the optimal approximation factor $1/e \cong .38$ one can achieve in the worst-case for Problem 1 in polynomial time [26]; the reason is twofold: first, as we comment in the next paragraph, we prove that Problem 1 involves the minimization of a non-increasing and supermodular function [27], and second, as we proved in Section III-A, Problem 1 is in the worst-case equivalent to the minimal controllability problem introduced in [24], which cannot be approximated in polynomial time with a better factor than the $1/e$ [28].

Remark 5: (Supermodularity of $\log \det(\Sigma(\hat{x}_{1:K}))$) In the proof of Theorem 2 (Appendix II), we show that $\log \det(\Sigma(\hat{x}_{1:K}))$ is a non-increasing and supermodular function with respect to the sequence of selected sensors. Specifically, the proof of (6) follows by combining these two results and results on the maximization of submodular functions over matroid constraints [29] —we present these three derivations in Appendices II-A, II-B, and II-C, respectively.

We continue with our third remark on Theorem 2:

Remark 6: (Time complexity of Algorithm 1) Algorithm 1's time complexity is broken down into two parts: the first part is the number of evaluations of $\log \det(\Sigma(\hat{x}_{1:K}))$ required by the algorithm, and the second part is the time complexity of each such evaluation. In particular, Algorithm 1 requires at most r_k^2 evaluations of $\log \det(\Sigma(\hat{x}_{1:K}))$ at each t_k . Therefore, Algorithm 1 achieves a time complexity that is only linear in K with respect to the total number of evaluations of $\log \det(\Sigma(\hat{x}_{1:K}))$. The reason is that $\sum_{k=1}^K r_k^2 \leq \max_{k \in [K]} (r_k^2) K$. In addition, for $w(t)$ zero mean and white Gaussian —as commonly assumed in the literature of sensor scheduling— the time complexity of each such evaluation is at most linear in K : the reason is that this $w(t)$ agrees with Assumption 1, in which case we prove that

³The matrices $\Phi(t_{k+1}, t_k)$, where $k \in [K - 1]$, are used in the computation of $\Sigma(\hat{x}_{1:K})$ (cf. proof of Theorem 2 in Appendix II).

Algorithm 2 Single step greedy algorithm (subroutine in Algorithm 1).

Input: Current iteration k (corresponds to t_k), selected sensor sets $(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{k-1})$ up to the current iteration, scheduling constraint r_k , estimation error function $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})) : \mathcal{S}_k \subseteq [m], k \in [K] \mapsto \mathbb{R}$

Output: Sensor set \mathcal{S}_k that approximates the solution to Problem 1 at t_k

$\mathcal{S}^0 \leftarrow \emptyset, \mathcal{X}^0 \leftarrow [m]$, and $t \leftarrow 1$

Iteration t:

1. If $\mathcal{X}^{t-1} = \emptyset$, **return** \mathcal{S}^{t-1}
 2. Select $i(t) \in \mathcal{X}^{t-1}$ for which $\rho_{i(t)}(\mathcal{S}^{t-1}) = \max_{i \in \mathcal{X}^{t-1}} \rho_i(\mathcal{S}^{t-1})$, with ties settled arbitrarily, where:

$$\rho_i(\mathcal{S}^{t-1}) \equiv \log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:k-1}, \mathcal{S}^{t-1})) - \log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:k-1}, \mathcal{S}^{t-1} \cup \{i\}))$$
 and $\mathcal{S}_{1:k-1} \equiv (\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{k-1})$
 - 3.a. If $|\mathcal{S}^{t-1} \cup \{i(t)\}| > r_k$, $\mathcal{X}^{t-1} \leftarrow \mathcal{X}^{t-1} \setminus \{i(t)\}$, and go to Step 1
 - 3.b. If $|\mathcal{S}^{t-1} \cup \{i(t)\}| \leq r_k$, $\mathcal{S}^t \leftarrow \mathcal{S}^{t-1} \cup \{i(t)\}$ and $\mathcal{X}^t \leftarrow \mathcal{X}^{t-1} \setminus \{i(t)\}$
 4. $t \leftarrow t + 1$ and continue
-

the time complexity of each evaluation of $\log \det(\Sigma(\hat{x}_{1:K}))$ is $O(n^{2.4}K)$ (linear in K).⁴

Remark 7: (Sparsity of $\Sigma(\hat{x}_{1:K})$) We state the three properties of $\log \det(\Sigma(\hat{x}_{1:K}))$ we prove to obtain the time complexity for Algorithm 1. The first two properties were mentioned in Remark 5: the monotonicity and supermodularity of $\log \det(\Sigma(\hat{x}_{1:K}))$. These two properties are responsible for that Algorithm 1 requires at most r_k^2 evaluations at each t_k . The third property, which follows, is responsible for the low time complexity for each evaluation of $\log \det(\Sigma(\hat{x}_{1:K}))$:

- $\Sigma(\hat{x}_{1:K})$ is the sum of two $nK \times nK$ sparse matrices: the first matrix is block diagonal, and the second one is block tri-diagonal. As a result, given that both of these matrices are known, each evaluation of $\log \det(\Sigma(\hat{x}_{1:K}))$ has time complexity $O(n^{2.4}K)$, linear in K (using the results in [31] —cf. Theorem 2 therein).

We show in Appendix II-C that after we include at each evaluation step of $\log \det(\Sigma(\hat{x}_{1:K}))$ the complexity to compute the two sparse matrices in $\Sigma(\hat{x}_{1:K})$, the total time complexity of Algorithm 1 is as given in Theorem 2.

Our final remark on Theorem 2 follows:

Remark 8: (Comparison of Algorithm 1's time complexity to that of existing scheduling algorithms) We do the comparison for two cases: batch state estimation, and Kalman filtering. In particular, we show that the time complexity of our algorithm is lower than that of existing sensor scheduling algorithms for batch state estimation, and of the similar order, or lower, of existing algorithms for Kalman filtering.

⁴We can also speed up Algorithm 1 by implementing in Algorithm 2 the method of lazy evaluations [30]: this method avoids in Step 2 of Algorithm 2 the computation of $\rho_i(\mathcal{S}^{t-1})$ for unnecessary choices of i .

Comparison with algorithms for batch state estimation: In [12], Problem 1 is considered, and a semi-definite programming (SDP) algorithm is proposed; its time complexity is of the order $O(\max_{k \in [K]}(r_k)K(nK)^{3.5} + (\max_{k \in [K]}(r_k^2)K^2(nK)^{2.5})$ [32]. Clearly, this time complexity is higher than that of Algorithm 1, whose complexity is $O(\max_{k \in [K]}(r_k)^2K^2n^{2.4})$. In addition, the algorithm presented in [12] provides no worst-case approximation guarantees (6), in contrast to Algorithm 1 that provides (6).

Comparison with algorithms for Kalman filtering: We do the comparison in two steps: first, we consider algorithms based on the maximization of submodular functions, and second, algorithms based on convex relaxation techniques or the alternating direction method of multipliers (ADMM):

- *Algorithms based on the maximization of submodular functions:* In [19], an algorithm is provided that is valid for a restricted class of linear systems: its time complexity is $O(\max_{k \in [K]}(r_k)mn^2K + n^{2.4}K)$. This time complexity is of similar order to that of Algorithm 1, whose complexity is of the order $O(\max_{k \in [K]}(r_k)^2K n^{2.4}K)$, since $\max_{k \in [K]}(r_k) < m$. Specifically, we observe in Algorithm 1's time complexity the additional multiplicative factor K (linear in K); this difference emanates from that Algorithm 1 offers a near-optimal guarantee over the whole time horizon (t_1, t_2, \dots, t_K) whereas the algorithm in [19] offers a near-optimal guarantee only for the last time step t_K . In addition, Algorithm 1 holds for any linear continuous time-invariant system (no restrictions are necessary), in contrast to the algorithm in [19], and it holds for any discrete time-variant systems where A_k in (8) is full rank; the latter assumption is one of the four restrictive conditions in [19] (Theorem 13).
- *Algorithms based on convex relaxation techniques or ADMM:* In [16], the authors assume a single sensor ($r_k = 1$ across t_k), and their objective is to achieve a minimal estimation error by minimizing the number of times this sensor will be used over the horizon t_1, t_2, \dots, t_K . The time complexity of the proposed algorithm is $O(n^{2.5}K^2 + n^{3.5}K)$. This time complexity is higher than that of Algorithm 1, whose complexity for $r_k = 1$ is of the order $O(n^{2.4}K^2)$. In [8], the authors employ ADMM techniques to solve a periodic sensor scheduling problem. They consider a zero mean and white Gaussian $w(t)$. The time complexity of the proposed algorithm is $O((nK)^3 + (\max_{k \in [K]}(r_k)K)n^2K^2 + \max(r_k)^2nK^3)$. This time complexity is of similar order to that of Algorithm 1, whose complexity in this case is $O(\max_{k \in [K]}(r_k^2)n^{2.4}K^2)$, since $\max_{k \in [K]}(r_k) \leq K$; in particular, for $K > n^{0.4} \max_{k \in [K]}(r_k)$, Algorithm 1 has lower time complexity.⁵

⁵More algorithms exist in the literature, that also use convex relaxation [33] or randomization techniques [34], and have similar time complexity to Algorithm 1. They achieve this complexity using additional approximation methods: e.g., they optimize instead an upper bound to the involved estimation error metric.

With the above remarks we conclude: Algorithm 1 enjoys both the estimation accuracy of the batch state scheduling algorithms and the low time complexity of the Kalman filtering scheduling algorithms, since:

- Algorithm 1 offers a near-optimal worst-case approximation guarantee for the batch state estimation error. This estimation error is only approximated by the Kalman filtering sensor scheduling algorithms: the reason is that they aim instead to minimize the sum of each of the estimation errors for $x(t_k)$ (across t_k). However, this sum only upper bounds the batch state estimation error.
- Algorithm 1 has time complexity lower than the state of the art batch estimation algorithms, and at the same time, lower than, or similar to, the time complexity of the corresponding Kalman filtering algorithms.

In addition: Algorithm 1's approximation guarantee holds for any linear system (continuous or discrete time). Moreover, Algorithm 1's time complexity guarantee holds for any continuous time system, and for discrete time systems where A_k in (8) is full rank across k .

The proof of Theorem 2 can be found in Appendix II.

C. Limits on Sensor Scheduling for Minimum Variance Batch State Estimation

In this section, we derive two trade-offs between three important parameters of our sensor scheduling problem:⁶

- the number of measurements times (t_1, t_2, \dots, t_K)
- the number r_k of sensors that can be used at each t_k
- the value of the estimation error $\mathbb{E}(\|x_{1:K} - \hat{x}_{1:K}\|_2^2)$.

The first of the two trade-offs is captured in the next theorem:

Theorem 3: Let $\sigma_w^{(-1)} \equiv \max_{i \in [nK]} [\mathbb{C}(x_{1:K})^{-1}]_{ii}$ and $\sigma_v^{(-1)} \equiv \|\mathbb{C}(v_{1:K})^{-1}\|_2$. Also, let $C_{1:K}$ be the block diagonal matrix where each of its K diagonal elements is equal to C , where C is the matrix $[C_1^\top, C_2^\top, \dots, C_m^\top]^\top$. For the variance of the error of the minimum variance estimator $\hat{x}_{1:K}$:

$$\mathbb{E}(\|x_{1:K} - \hat{x}_{1:K}\|_2^2) \geq \frac{n}{\sigma_v^{(-1)} \max_{k \in [K]} (r_k) \|C_{1:K}\|_2^2 + \sigma_w^{(-1)} / K}. \quad (9)$$

The lower bound in (9) decreases as the number of used sensors for scheduling r_k increases or the number measurement times K increases, and increases as the system's size increases. Since these qualitative relationships were expected, the importance of this theorem lies on the quantification of these relationships (that also includes the dependence on the noise parameters $\sigma_w^{(-1)}$ and $\sigma_v^{(-1)}$): for example, (9) decreases only inversely proportional with the number of sensors for scheduling; that is, increasing the number r_k so to reduce the variance of the error of the minimum variance estimator is ineffective, a fundamental limit. In addition, this bound increases linearly with the system's size; this is another limit for large-scale systems.

⁶We recall from Section II that the objective of Problem 1 is related to $\mathbb{E}(\|x_{1:K} - \hat{x}_{1:K}\|_2^2)$ in that when $\log \det(\Sigma(\hat{x}_{1:K}))$ is minimized the probability that the estimation error $\|x_{1:K} - \hat{x}_{1:K}\|_2^2$ is small is maximized.

Similar results are proved in [35] for the steady state error covariance of scalar systems in the case that the number of sensors goes to infinity. In more detail, the authors in [35] account for different types of multi-access schemes, as well as, for fading channels between the sensors and the fusion centre that combines the sensor measurements.

The next corollary presents our last trade-off:

Corollary 3: Consider that the desired value for $\mathbb{E}(\|x_{1:K} - \hat{x}_{1:K}\|_2^2)$ is α . Any set of scheduled sensors at t_1, t_2, \dots, t_K that achieves this error satisfies:

$$\max_{k \in [K]} (r_k) \geq \frac{n/\alpha - \sigma_w^{(-1)}/K}{\sigma_v^{(-1)} \|C_{1:K}\|_2^2}. \quad (10)$$

Eq. (10) implies that the number of sensors used for scheduling at each t_k increases as the error of the minimum variance estimator or the number of measurements times K decreases. More importantly, it quantifies that this number increases linearly with the system's size for fixed error variance. This is again a fundamental limit, meaningful for large-scale systems.

IV. FUTURE WORK

We work on extending the results of this paper to largely unknown systems, under the presence of non-linear measurements. The first of these extensions allows systems whose evolution is captured by, e.g., Gaussian processes or random networks (the former example is a widely used assumption for motion models; cf. [10] and references therein). The second of these extensions allows complex measurement environments, such as camera-sensor environments, that can enable the application of our results in domains such as robotics and the automotive sector.

V. ACKNOWLEDGEMENTS

Vasileios Tzoumas thanks Nikolay A. Atanasov for bringing into his attention paper [10].

APPENDIX I

CLOSED FORMULA FOR THE ERROR COVARIANCE OF $\hat{x}_{1:K}$

We compute the error covariance of $\hat{x}_{1:K}$: Denote as $S_{1:K}$ the block diagonal matrix with diagonal elements the sensor selection matrices $S(t_1), S(t_2), \dots, S(t_K)$. Moreover, denote as C the matrix $[C_1^\top, C_2^\top, \dots, C_m^\top]^\top$. Finally, denote $y_{1:K} \equiv (y(t_1)^\top, y(t_2)^\top, \dots, y(t_K)^\top)^\top$, $w_{1:K} \equiv (w(t_1)^\top, w(t_2)^\top, \dots, w(t_K)^\top)^\top$, and $v_{1:K} \equiv (v(t_1)^\top, v(t_2)^\top, \dots, v(t_K)^\top)^\top$, where $v(t_k) \equiv (v_1(t_k)^\top, v_2(t_k)^\top, \dots, v_m(t_k)^\top)^\top$. Then, from (1), (2) and (3):

$$y_{1:K} = O_{1:K} x_{1:K} + S_{1:K} v_{1:K}, \quad (11)$$

where $O_{1:K}$ is the $\sum_{k=1}^K r_k \times nK$ block diagonal matrix with diagonal elements the matrices $S(t_1)C, S(t_2)C, \dots, S(t_K)C$. $\hat{x}_{1:K}$ has the error covariance $\Sigma(\hat{x}_{1:K}) = \mathbb{E}((x_{1:K} - \hat{x}_{1:K})(x_{1:K} - \hat{x}_{1:K})^\top)$ [6]:

$$\Sigma(\hat{x}_{1:K}) = \mathbb{C}(x_{1:K}) - \mathbb{C}(x_{1:K}) O_{1:K}^\top \Xi O_{1:K} \mathbb{C}(x_{1:K}), \quad (12)$$

where $\Xi \equiv (O_{1:K} \mathbb{C}(x_{1:K}) O_{1:K}^\top + S_{1:K} \mathbb{C}(v_{1:K}) S_{1:K}^\top)^{-1}$.

We simplify (12) in the following lemma:

Lemma 1: The error covariance of $\hat{x}_{1:K}$ has the equivalent form:

$$\Sigma(\hat{x}_{1:K}) = \left(\sum_{k=1}^K \sum_{i=1}^m s_i(t_k) U^{(ki)} + \mathbb{C}(x_{1:K})^{-1} \right)^{-1}, \quad (13)$$

where $s_i(t_k)$ is a zero-one function, equal to 1 if and only if sensor i is used at t_k , and $U^{(ki)}$ is the block diagonal matrix $C_{1:K}^\top I^{(ki)} \mathbb{C}(v_{1:K})^{-1} I^{(ki)} C_{1:K}$; $C_{1:K}$ is the block diagonal matrix where each of its K diagonal elements is equal to C , and $I^{(ki)}$ is the block diagonal matrix with mK diagonal elements such that: the $((k-1)m+i)$ -th element is the $d_i \times d_i$ identity matrix I , and the rest of the elements are equal to zero.

APPENDIX II PROOF OF THEOREM 2

We prove Theorem 2 in three steps:⁷ we first show that $\log \det(\Sigma(\hat{x}_{1:K}))$ is a non-increasing function in the choice of the sensors; we then show that $\log \det(\Sigma(\hat{x}_{1:K}))$ is a supermodular function in the choice of the sensors; finally, we prove Theorem 2 by combining the aforementioned two results and results on the maximization of submodular functions over matroid constraints [29].

Notation: We recall that any collection (x_1, x_2, \dots, x_k) is denoted as $x_{1:k}$ ($k \in \mathbb{N}$).

A. Monotonicity in Sensor Scheduling for Minimum Variance Batch State Estimation

We first provide two notations, and then the definition of non-increasing and non-decreasing set functions. Afterwards, we present the main result of this subsection.

Notation: Given K disjoint finite sets $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_K$ and $A_i, B_i \in \mathcal{X}_i$, we write $A_{1:K} \preceq B_{1:K}$ to denote that for all $i \in [K]$, $A_i \subseteq B_i$ (A_i is a subset of B_i). Moreover, we denote that $A_i \in \mathcal{X}_i$ for all $i \in [K]$ as $A_{1:K} \in \mathcal{X}_{1:K}$.

Definition 1: Consider K disjoint finite sets $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_K$. A function $h : \mathcal{X}_{1:K} \mapsto \mathbb{R}$ is non-decreasing if and only if for all $A, B \in \mathcal{X}_{1:K}$ such that $A \preceq B$, $h(A) \leq h(B)$; $h : \mathcal{X}_{1:K} \mapsto \mathbb{R}$ is non-increasing if $-h$ is non-decreasing.

The main result of this subsection follows:

Proposition 1: For any finite $K \in \mathbb{N}$, consider K distinct copies of $[m]$, denoted as $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_K$. The estimation error metric $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})) : \mathcal{M}_{1:K} \mapsto \mathbb{R}$ is a non-increasing function in the choice of the sensors $\mathcal{S}_{1:K}$.

We next show that $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K}))$ is a supermodular function with respect to the selected sensors $\mathcal{S}_{1:K}$.

B. Submodularity in Sensor Scheduling for Minimum Variance Batch State Estimation

We first provide a notation, and then the definition of submodular and supermodular set functions. Afterwards, we present the main result of this subsection.

⁷Due to space constraints we have omitted the proof of the monotonicity and supermodularity of $\log \det(\Sigma(\hat{x}_{1:K}))$ from this proof; they can be found in the full version of this paper located in the authors websites.

Notation: Given K disjoint finite sets $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_K$ and $A_{1:K}, B_{1:K} \in \mathcal{X}_{1:K}$, we write $A_{1:K} \uplus B_{1:K}$ to denote that for all $i \in [K]$, $A_i \cup B_i$ (A_i union B_i).

Definition 2: Consider K disjoint finite sets $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_K$. A function $h : \mathcal{X}_{1:K} \mapsto \mathbb{R}$ is submodular if and only if for all $A, B, C \in \mathcal{X}_{1:K}$ such that $A \preceq B$, $h(A \uplus C) - h(A) \geq h(B \uplus C) - h(B)$; $h : \mathcal{X}_{1:K} \mapsto \mathbb{R}$ is supermodular if $-h$ is submodular.

According to Definition 2, set submodularity is a diminishing returns property: a function $h : \mathcal{X}_{1:K} \mapsto \mathbb{R}$ is set submodular if and only if for all $C \in \mathcal{X}_{1:K}$, the function $h_C : \mathcal{X}_{1:K} \mapsto \mathbb{R}$ defined for all $A \in \mathcal{X}_{1:K}$ as $h_C(A) \equiv h(A \uplus C) - h(C)$ is non-increasing.

The main result of this subsection follows:

Proposition 2: For any finite $K \in \mathbb{N}$, consider K distinct copies of $[m]$, denoted as $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_K$; the estimation error metric $\log \det(\Sigma(\hat{x}_{1:K} | \mathcal{S}_{1:K})) : \mathcal{M}_{1:K} \mapsto \mathbb{R}$ is a set supermodular function in the choice of the sensors $\mathcal{S}_{1:K}$.

Proposition 2 implies that as we increase at each t_k the number of sensors used, the marginal improvement we get on the estimation error of $x_{1:K}$ diminishes.

We are now ready for the proof of Theorem 2.

C. Proof of Theorem 2

We first provide the definition of a matroid, and then continue with the main proof:

Definition 3: Consider a finite set \mathcal{X} and a collection \mathcal{C} of subsets of \mathcal{X} . $(\mathcal{X}, \mathcal{C})$ is:

- an independent system if and only if:
 - $\emptyset \in \mathcal{C}$, where \emptyset denotes the empty set
 - for all $X' \subseteq X \subseteq \mathcal{X}$, if $X \in \mathcal{C}$, $X' \in \mathcal{C}$.
- a matroid if and only if in addition to the previous two properties:
 - for all $X', X \in \mathcal{C}$ where $|X'| < |X|$, there exists $x \notin X'$ and $x \in X$ such that $X' \cup \{x\} \in \mathcal{C}$.

Proof: [of Part 1 of Theorem 2] We use the next result from the literature of maximization of submodular functions over matroid constraints:

Lemma 2 (Ref. [29]): Consider K independence systems $\{(\mathcal{X}_k, \mathcal{C}_k)\}_{k \in [K]}$, each the intersection of at most P matroids, and a submodular and non-decreasing function $h : \mathcal{X}_{1:K} \mapsto \mathbb{R}$. There exist a polynomial time greedy algorithm that returns an (approximate) solution $\mathcal{S}_{1:K}$ to:

$$\begin{aligned} & \underset{\mathcal{S}_{1:K} \preceq \mathcal{X}_{1:K}}{\text{maximize}} && h(\mathcal{S}_{1:K}) \\ & \text{subject to} && \mathcal{S}_k \cap \mathcal{X}_k \in \mathcal{C}_k, k \in [K], \end{aligned} \quad (14)$$

that satisfies:

$$\frac{h(\mathcal{O}) - h(\mathcal{S}_{1:K})}{h(\mathcal{O}) - h(\emptyset)} \leq \frac{P}{1+P}, \quad (15)$$

where \mathcal{O} is an (optimal) solution to (14).

In particular, we prove:

Lemma 3: Problem 1 is an instance of (14) with $P = 1$. This observation, along with Lemmas 2 and 3 complete the proof of (6), since the adaptation to Problem 1 of the greedy algorithm in [29] (Theorem 4.1) results to Algorithm 1. ■

Proof: [of Part 2 of Theorem 2] In Lemma 1 in Appendix I we prove that $\Sigma(\hat{x}_{1:K})$ is the sum of two matrices: the first matrix is a block diagonal matrix, and the second one is the inverse of the covariance of $x_{1:K}$, $\mathbb{C}(x_{1:K})$. The block diagonal matrix is computed in $O(n^{2.4}K)$ time. Moreover, by extending the result in [10] (Theorem 1), we get that $\mathbb{C}(x_{1:K})^{-1}$ is a block tri-diagonal matrix, that is described by the $(K - 1)$ transition matrices $\Phi(t_{k+1}, t_k)$ [22], where $k \in [K - 1]$, and K identity matrices. For continuous time systems, the time complexity to compute all the block elements in $\mathbb{C}(x_{1:K})^{-1}$ is $O(n^3K)$ [36]; for discrete time systems, it is $O(n^{2.4}K)$ [22]. This computation of $\mathbb{C}(x_{1:K})^{-1}$ is made only once. Finally, from Theorem 2 in [31], we can now compute the $\det(\Sigma(\hat{x}_{1:K}))$ in $O(n^{2.4}K)$ time, since $\Sigma(\hat{x}_{1:K})$ is block tri-diagonal. Therefore, the overall time complexity of Algorithm 1 is: $O(n^3K) + O(2n^{2.4}K \sum_{k=1}^K r_k^2) = O(n^{2.4}K \sum_{k=1}^K r_k^2)$ for K large, since $\mathbb{C}(x_{1:K})^{-1}$ is computed only once, and Algorithm 1 requests at most $\sum_{k=1}^K r_k^2$ evaluations of $\Sigma(\hat{x}_{1:K})$.

The proof is complete. ■

REFERENCES

- [1] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive computing*, vol. 3, no. 4, pp. 24–33, 2004.
- [2] R. Nowak, U. Mitra, and R. Willett, "Estimating inhomogeneous fields using wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 999–1006, 2004.
- [3] M. P. Vitus, "Sensor placement for improved robotic navigation," *Robotics: Science and Systems VI*, p. 217, 2011.
- [4] E. Masazade, M. Fardad, and P. K. Varshney, "Sparsity-promoting extended kalman filtering for target tracking in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 845–848, 2012.
- [5] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. La Porta, "A survey of sensor selection schemes in wireless sensor networks," in *Defense and Security Symposium*. International Society for Optics and Photonics, 2007, pp. 65 621A–65 621A.
- [6] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall Upper Saddle River, NJ, 2000, vol. 1.
- [7] A. O. Hero III and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, 2011.
- [8] S. Liu, M. Fardad, E. Masazade, and P. K. Varshney, "Optimal periodic sensor scheduling in networks of dynamical systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3055–3068.
- [9] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [10] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch non-linear continuous-time trajectory estimation as exactly sparse gaussian process regression," *Autonomous Robots*, vol. 39, no. 3, pp. 221–238, 2015.
- [11] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2657–2664.
- [12] V. Roy, A. Simonetto, and G. Leus, "Spatio-temporal sensor management for environmental field estimation," *Signal Processing*, vol. 128, pp. 369 – 381, 2016.
- [13] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 2572–2577.
- [14] J. Liu, D. Petrovic, and F. Zhao, "Multi-step information-directed sensor querying in distributed sensor networks," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 5. IEEE, 2003, pp. V–145.
- [15] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.
- [16] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.
- [17] Y. Mo, R. Ambrosino, and B. Sinopoli, "Sensor selection strategies for state estimation in energy constrained wireless sensor networks," *Automatica*, vol. 47, no. 7, pp. 1330–1338, 2011.
- [18] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for optimal filtering of linear dynamical systems: Complexity and approximation," in *IEEE Conference on Decision and Control (CDC)*, 2015.
- [19] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.
- [20] M. F. Huber, A. Kuwertz, F. Sawo, and U. D. Hanebeck, "Distributed greedy sensor scheduling for model-based reconstruction of space-time continuous physical phenomena," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 102–109.
- [21] 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.
- [22] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. New York, NY, USA: Oxford University Press, Inc., 1998.
- [23] S. Venkatesh, *The Theory of Probability: Explorations and Applications*. Cambridge University Press, 2012.
- [24] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.
- [25] S. Pequito, S. Kar, and A. Aguiar, "A framework for structural input/output and control configuration selection in large-scale systems," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2015.
- [26] J. Vondrák, "Submodularity and curvature: the optimal algorithm," *RIMS Kokyuroku Bessatsu B*, vol. 23, pp. 253–266, 2010.
- [27] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience, 1988.
- [28] U. Feige, "A threshold of $\ln n$ for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998.
- [29] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, *An analysis of approximations for maximizing submodular set functionsII*. Springer, 1978.
- [30] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*. Springer, 1978, pp. 234–243.
- [31] L. G. Molinari, "Determinants of block tridiagonal matrices," *Linear algebra and its applications*, vol. 429, no. 8, pp. 2221–2226, 2008.
- [32] A. Nemirovski, "Interior point polynomial time methods in convex programming."
- [33] J. E. Weimer, B. Sinopoli, and B. H. Krogh, "A relaxation approach to dynamic sensor selection in large-scale wireless networks," in *28th International Conference on Distributed Computing Systems Workshops (ICDCS)*, 2008, pp. 501–506.
- [34] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.
- [35] A. S. Leong, S. Dey, and J. S. Evans, "Asymptotics and power allocation for state estimation over fading channels," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 611–633, January 2011.
- [36] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003.