Scheduling Nonlinear Sensors for Stochastic Process Estimation

Vasileios Tzoumas*, Nikolay A. Atanasov*, Ali Jadbabaie†, George J. Pappas*

***Abstract*—In this paper, we focus on activating only a few sensors, among many available, to estimate the batch state of a stochastic process of interest. This problem is important in applications such as target tracking and simultaneous localization and mapping (SLAM), and in general, in problems where we need to have a good estimate of the trajectory taken so far, e.g., for linearisation purposes. It is challenging since it involves stochastic systems whose evolution is largely unknown, sensors with nonlinear measurements, and limited operational resources that constrain the number of active sensors at each measurement step. We provide an algorithm applicable to general stochastic processes and nonlinear measurements whose time complexity is linear in the planning horizon and whose performance is up to a multiplicative factor $1/2$ away from the optimal performance. This is notable because the algorithm offers a significant computational advantage over the polynomial-time algorithm that achieves the best approximation factor $1/e$. In addition, for important classes of Gaussian processes and nonlinear measurements corrupted with Gaussian noise, our algorithm enjoys the same time complexity as the state-of-the-art algorithms for linear systems and measurements. We achieve our results by proving two properties for the entropy of the batch state vector conditioned on the measurements: a) it is supermodular in the choice of the sensors; b) it has a sparsity pattern (involves block tri-diagonal matrices) that facilitates its evaluation at each sensor set.**

I. Introduction

Adversarial target tracking and capturing [1], [2], robotic navigation and autonomous construction [3], active perception and simultaneous localization and mapping (SLAM) [4] are only a few of the challenging information gathering problems that benefit from the monitoring capabilities of sensor networks [5]. These problems are challenging because:

- they involve systems whose evolution is largely unknown, modeled either as a stochastic process, such as a Gaussian process [6], or as linear or nonlinear system corrupted with process noise [1],
- they involve nonlinear sensors (e.g., cameras, radios) corrupted with noise [7],
- they involve systems that change over time [8], and as a result, necessitate both spatial and temporal deployment of sensors in the environment, increasing the total number of needed sensors, and at the same time,
- they involve operational constraints, such as limited communication bandwidth and battery life, which limit the number of sensors that can simultaneously be active in the information gathering process [9].

Due to these challenges, we focus on the following question: "How do we select, at each time, only a few of the available sensors so as to monitor effectively a system despite the above challenges?" In particular, we focus on the following sensor scheduling problem:

Problem 1. *Consider a stochastic process, whose realization at time t is denoted by $x(t)$ and a set of m sensors, whose measurements are nonlinear functions of $x(t)$, evaluated at a fixed set of K measurement times $t_1, t_2, \ldots, t_K$. In addition, suppose that at each t_k a set of at most $s_k \leq m$ sensors can be used. Select the sensor sets so that the error of the corresponding minimum mean square error estimator of $(x(t_1), x(t_2), \ldots, x(t_K))$ is minimal among all possible sensor sets.*

The reason we focus on estimating the batch state vector $(x(t_1), x(t_2), \ldots, x(t_K))$ is that in many control problems we need to have a good estimate of the trajectory taken so far, e.g., for linearisation purposes.

Literature review: There are two classes of sensor scheduling algorithms, that trade-off between the estimation accuracy of the batch state vector and their time complexity [10]: those used for Kalman filtering, and those for batch state estimation. The most relevant papers on batch state estimation are [10] and [11]. However, both of these papers focus on linear systems and measurements. The most relevant papers for Kalman filtering consider algorithms that use: myopic heuristics [12], tree pruning [13], convex optimization [14]–[17], quadratic programming [18], Monte Carlo methods [19], or submodular function maximization [20], [21]. However, these papers focus similarly on linear or nonlinear systems and measurements, and do not consider unknown dynamics.

At the same time, [22] focuses on sensor selection algorithms for estimating stochastic processes that are, in contrast to the processes in the present paper, spatially correlated and not temporally correlated. In more detail, in [22], $x(t_i)$ represents the value of a parameter of interest at a spatial position t_i, and is constant in time. This is notable since in [22] the proposed algorithms for sensor selection become fast when the covariance matrix of $(x(t_1), x(t_2), \ldots, x(t_K))$ is sparse (or can be approximated by a sparse matrix). Notwithstanding, this is not necessarily the case for dynamic stochastic processes, since $x(t_i)$ may be strongly correlated to the trajectory $(x(t_1), x(t_2), \ldots, x(t_{i-1}))$ taken so far in the state space.

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

†The author is the Associate Director of the Institute for Data, Systems and Society, and the Director of the Sociotechnical Systems Research Center, 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.

Main contributions:

1) We prove that Problem 1 is NP-hard.
2) We prove that the best approximation factor one can achieve in polynomial time for Problem 1, in the worst case, is $1/e$.
3) We provide Algorithm 1 for Problem 1 that:
   - for all stochastic processes and nonlinear measurements, achieves a solution that is up to a multiplicative factor $1/2$ from the optimal solution with time complexity that is only linear in the planning horizon K. This is important, since it implies that Algorithm 1 offers a significant computational advantage with negligible loss in performance over the polynomial-time algorithm that achieves the best approximation factor of $1/e$,
   - for important classes of Gaussian processes, and nonlinear measurements corrupted with Gaussian noise, has the same time complexity as state-of-the-art algorithms for linear systems and measurements. For example, for Gaussian process such as those in target tracking, or those generated by linear or nonlinear systems corrupted with Gaussian noise, Algorithm 1 has the same time complexity as the batch state estimation algorithm in [10], and lower than the relevant Kalman filter scheduling algorithms in [14], [17].

   Therefore, Algorithm 1 can enjoy both the estimation accuracy of the batch state scheduling algorithms (compared to the Kalman filtering approach, that only approximates the batch state estimation error with an upper bound [10]) and, surprisingly, even the low time complexity of the Kalman filtering scheduling algorithms for linear systems.

Technical contributions:

1) *Supermodularity in Problem 1*: We achieve the approximation performance of Algorithm 1, and the linear dependence of its time complexity on the planning horizon, by proving that our estimation metric is a supermodular function in the choice of the utilized sensors. This is important, since this is in contrast to the case of multi-step Kalman filtering for linear systems and measurements, where the corresponding estimation metric is neither supermodular nor submodular [20] [21].

2) *Sparsity in Problem 1*: We achieve the reduced time complexity of Algorithm 1 for Gaussian processes by identifying a sparsity pattern in our estimation metric. Specifically, for Gaussian processes the time complexity of each evaluation of our metric is decided by the sparsity pattern of either the covariance of $(x(t_1), x(t_2), \ldots, x(t_K))$, or the inverse of this covariance. This is important since the two matrices are not usually sparse at the same time, even if one of them is [23].

In more detail, we identify that for Gaussian processes such as those in target tracking, the first matrix is block tri-diagonal, whereas for those in SLAM, or those generated by linear or nonlinear systems corrupted with Gaussian noise, the second matrix is block tri-diagonal.

Paper's organization: We organize the rest of the paper as follows: In Section II we formulate Problem 1. In Section III, we present our two main results: first, we prove that our sensor scheduling problem is NP-hard; second, we derive our approximation algorithm, and emphasize on its time complexity. Section IV concludes the paper with our future work. The proofs of our results are found in the Appendix; due to space constraints some of the proofs are omitted, and can be found in the full version of this paper, located in the authors' websites.

Notation: We denote the set of natural numbers $\{1, 2, \ldots\}$ by $\mathbb{N}$, the set of real numbers by $\mathbb{R}$, and the set $\{1, 2, \ldots, n\}$ by $[n]$ ($n \in \mathbb{N}$). The set of real numbers between 0 and 1 is denoted by $[0, 1]$, and the empty set by $\emptyset$. Given a set $\mathcal{X}$, $|\mathcal{X}|$ is its cardinality. In addition, for $n \in \mathbb{N}$, $\mathcal{X}^n$ is the n-times Cartesian product $\mathcal{X} \times \mathcal{X} \times \cdots \times \mathcal{X}$. 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 $\mathcal{X}$; $A^\top$ is its transpose, and $[A]_{ij}$ is its element at the i-th row and j-th column; $\det(A)$ is its determinant. Furthermore, if A is positive definite, we write $A \succ 0$. In the latter case, A^{-1} is its inverse. I is the identity matrix; its dimension is inferred from the context. Similarly for the zero matrix 0. The $\equiv$ denotes equivalence. Moreover, for a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, Ω is the sample space, $\mathcal{F}$ the σ-field, and $\mathbb{P} : \mathcal{F} \mapsto [0, 1]$ the function that assigns probabilities to events in $\mathcal{F}$ [24]. We write $x \sim \mathcal{F}$ to denote a random variable x with probability distribution $\mathcal{F}$; $\mathbb{E}(x)$ is its expected value, and $\Sigma(x)$ its covariance. $x \sim \mathcal{N}(\mu, \Sigma)$ denotes a Gaussian random variable x with mean μ and covariance Σ; with a slight abuse of notation, we equivalently write $x \sim \mathcal{N}(\mathbb{E}(x), \Sigma(x))$. Finally, we write $x|y \sim \mathcal{G}$ to denote that x's probability distribution given y is $\mathcal{G}$.

II. Problem Formulation

This section introduces the system, measurement, and scheduling models and presents the sensor scheduling problem formally.

System Model. *We consider two cases:*

- Continuous time model*: Consider the stochastic process (along with a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$*):*

$$x_\omega(t) : \omega \in \Omega, t \geq t_0 \mapsto \mathbb{R}^n \tag{1}$$

  where $n \in \mathbb{N}$, t_0 *is the initial time, and* $x_\omega(t)$ *the state vector given the sample* ω.
- Discrete time model*: Consider the nonlinear discrete-time system:*

$$x_{k+1} = l_k(x_{1:k}), l_k \sim \mathcal{L}_k, k \in \mathbb{N} \tag{2}$$

  where $x_k \in \mathbb{R}^n$ *is the state vector,* $x_{1:k}$ *the batch vector* $(x_1, x_2, \ldots, x_k)$*, and* $\mathcal{L}_k$ *a probability distribution over functions* $l_k : \mathbb{R}^{nk} \mapsto \mathbb{R}^n$.

Because the system models (1) and (2) assume no characteristic structure, they are appropriate for modeling largely unknown dynamics. For example, an instance of (1) is the time-indexed Gaussian process system model:

$$x(t) \sim \mathcal{GP}(\mu(t), \Sigma(t, t')), \quad t, t' \geq t_0, \tag{3}$$

where $\mu(t)$ is the mean function and $\Sigma(t,t')$ is the covariance function. Similarly, an instance of (2) is the state-indexed Gaussian process system model:

$$x_{k+1} = l(x_k), \quad l \sim \mathcal{GP}(\mu(x), \Sigma(x,x')), x, x' \in \mathbb{R}^n. \quad (4)$$

Measurement Model. *Consider m nonlinear sensors that operate in discrete time:*

$$z_{i,k} = g_i(x_k) + v_{i,k}, \quad i \in [m], k \in \mathbb{N} \quad (5)$$

where for the continuous-time system in (1) we let $x_k := x(t_k)$ at a pre-specified set of measurement times $t_1, t_2, \dots$ and $v_{i,k}$ is the measurement noise of sensor i at time k.

Assumption 1. *$v_{i,k}$ are independent across i and k. In addition, g_i is one-time differentiable.*

Sensor Scheduling Model. *The m sensors in (5) are used at K scheduled measurement times $\{t_1, t_2, \dots, t_K\}$. At each $k \in [K]$, only s_k of the m sensors are used ($s_k \leq m$), resulting in the batch measurement vector y_k:*

$$y_k = S_k z_k, \quad k \in [K], \quad (6)$$

where S_k is a sensor selection matrix, composed of sub-matrices $[S_k]_{ij}$ ($i \in [s_k]$, $j \in [m]$) such that $[S_k]_{ij} = I$ if sensor j is used at time k, and $[S_k]_{ij} = 0$ otherwise. We assume that a sensor can be used at most once at each k, and as a result, for each i there is one j such that $[S_k]_{ij} = I$ while for each j there is at most one i such that $[S_k]_{ij} = I$.

We now present the sensor scheduling problem formally:

Notation: For $i, j \in \mathbb{N}$, $\phi_{i:j} \equiv (\phi_i, \phi_{i+1}, \dots, \phi_j)$. In addition, $\mathcal{S}_k \equiv \{j :$ there exists $i \in [s_k], [S_k]_{ij} = I\}$: $\mathcal{S}_k$ is the set of indices that correspond to utilized sensors at t_k.

Problem 1 (Sensor Scheduling in Stochastic Processes with Nonlinear Observations)**.** *Select at each time k a subset of s_k sensors, out of the m sensors in (5), to use in order to minimize the conditional entropy of $x_{1:K}$ given the measurements $y_{1:K}$:*

$$\begin{aligned} \underset{\mathcal{S}_k \subseteq [m], k \in [K]}{\textit{minimize}} \quad & \mathbb{H}(x_{1:K}|\mathcal{S}_{1:K}) \\ \textit{subject to} \quad & |\mathcal{S}_k| \leq s_k, k \in [K], \end{aligned}$$

where $\mathbb{H}(x_{1:K}|\mathcal{S}_{1:K})$ denotes the conditional entropy $\mathbb{H}(x_{1:K}|y_{1:K})$ of $x_{1:K}$ given the measurements $y_{1:K}$.

The conditional entropy $\mathbb{H}(x_{1:K}|y_{1:K})$ captures the estimation accuracy of $x_{1:K}$ given $y_{1:K}$, as we explain in the following two propositions:

Proposition 1. *$\mathbb{H}(x_{1:K}|y_{1:K})$ is a constant factor away from the mutual information of $x_{1:K}$ and $y_{1:K}$. In particular:*

$$\mathbb{H}(x_{1:K}|y_{1:K}) = -\mathbb{I}(x_{1:K}; y_{1:K}) + \mathbb{H}(x_{1:K}),$$

where $\mathbb{I}(x_{1:K}; y_{1:K})$ is the mutual information of $x_{1:K}$ and $y_{1:K}$, and $\mathbb{H}(x_{1:K})$ is constant.

Proposition 2. *Consider the Gaussian process (3) and suppose that the measurement noise in (5) is Gaussian, $v_{i,k} \sim \mathcal{N}(0, \Sigma(v_{i,k}))$. $\mathbb{H}(x_{1:K}|y_{1:K})$ is a constant factor away from $\log\det(\Sigma(x^\star_{1:K}))$, where $\Sigma(x^\star_{1:K})$ is the error covariance of the minimum mean square estimator $x^\star_{1:K}$ of $x_{1:K}$ given the measurements $y_{1:K}$. In particular:*[1]

$$\mathbb{H}(x_{1:K}|y_{1:K}) = \frac{\log\det(\Sigma(x^\star_{1:K}))}{2} + \frac{nK\log(2\pi e)}{2}.$$

Algorithm 1 Approximation algorithm for Problem 1.

```
Input: Horizon $K$, scheduling constraints $s_1, s_2, \dots, s_K$, error metric $\mathbb{H}(x_{1:K}|\mathcal{S}_{1:K}) : \mathcal{S}_k \subseteq [m], k \in [K] \mapsto \mathbb{R}$
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
         $\min_{\mathcal{S} \subseteq [m]} \{\mathbb{H}(x_{1:K}|\mathcal{S}_{1:k-1}, \mathcal{S}) : |\mathcal{S}| \leq s_k\}$    (7)
    2. Denote by $\mathcal{S}_k$ 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
```

III. Main Results

We first prove that Problem 1 is NP-hard, and then derive for it a provably near-optimal approximation algorithm:

Theorem 1. *The problem of sensor scheduling in stochastic processes with nonlinear observations (Problem 1) is NP hard.*

Due to Theorem 1, we need to appeal to approximation algorithms to obtain a solution to Problem 1 in polynomial-time. To this end, we propose an efficient near-optimal algorithm (Algorithm 1 with a subroutine in Algorithm 2) and quantify its performance and time complexity in the following theorem.

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:*
   i. *satisfy all the feasibility constraints of Problem 1: $|\mathcal{S}_k| \leq s_k, k \in [K]$*
   ii. *achieve an error $\mathbb{H}(x_{1:K}|\mathcal{S}_{1:K})$ such that:*

$$\frac{\mathbb{H}(x_{1:K}|\mathcal{S}_{1:K}) - OPT}{MAX - OPT} \leq \frac{1}{2}, \quad (8)$$

   where OPT is the optimal cost of Problem 1, and $MAX \equiv \max_{\mathcal{S}'_{1:K}} \mathbb{H}(x_{1:K}|\mathcal{S}'_{1:K})$ is the maximum (worst) cost in Problem 1.

2) Time complexity of Algorithm 1: *Algorithm 1 has time complexity $O(\sum_{k=1}^{K} s_k^2 T)$, where T is the time complexity of evaluating $\mathbb{H}(x_{1:K}|\mathcal{S}'_{1:K}) : \mathcal{S}'_k \subseteq [m]$, $k \in [K] \mapsto \mathbb{R}$ at an $\mathcal{S}'_{1:K}$.*

[1]We explain $x^\star_{1:K}$ and $\log\det(\Sigma(x^\star_{1:K}))$: $x^\star_{1:K}$ is the optimal estimator for $x_{1:K}$, since it minimizes among *all* estimators of $x_{1:K}$ the mean square error $\mathbb{E}(\|x_{1:K} - x^\star_{1:K}\|_2^2)$ ($\|\cdot\|_2$ is the euclidean norm), where the expectation is taken with respect to $y_{1:K}$ [25, Appendix E]. $\log\det(\Sigma(x^\star_{1:K}))$ is an estimation error metric related to $\|x_{1:K} - x^\star_{1:K}\|_2^2$, since when it is minimized, the probability that the estimation error $\|x_{1:K} - x^\star_{1:K}\|_2^2$ is small is maximized [10].

In the following paragraphs, we discuss Algorithm 1's approximation quality and time complexity and fully characterize the latter in Theorem 3 and Corollary 1 for Gaussian processes and Gaussian measurement noise.

Supermodularity and monotonicity of $\mathbb{H}(x_{1:K}|y_{1:K})$: We state two properties of $\mathbb{H}(x_{1:K}|y_{1:K})$ that are used to prove Theorem 2. In particular, we show that $\mathbb{H}(x_{1:K}|y_{1:K})$ is a non-increasing and supermodular function with respect to the sequence of selected sensors. Then, Theorem 2 follows by combining these two results with results on submodular functions maximization over matroid constraints [26].

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 (8) is constant. In particular, (8) guarantees that for any instance of Problem 1, the distance of the approximate cost $\mathbb{H}(x_{1:K}|\mathcal{S}_{1:K})$ from OPT is at most $1/2$ the distance of the worst (maximum) cost MAX from OPT. This approximation factor is close to the optimal approximation factor $1/e \cong .38$ one can achieve in the worst-case for Problem 1 in polynomial time [27]; the reason is twofold: first, Problem 1 involves the minimization of a non-increasing and supermodular function [28], and second, as we proved in Theorem 1, Problem 1 is in the worst-case equivalent to the minimal observability problem introduced in [29], which cannot be approximated in polynomial time with a better factor than the $1/e$ [30].

Remark 1. *We can improve the $1/2$ approximation factor of Algorithm 1 to $1/e$ by utilizing the algorithm introduced in [31]. However, this algorithm has time complexity $O((nK)^{11}T)$, where T is the time complexity of evaluating $\mathbb{H}(x_{1:K}|\mathcal{S}'_{1:K}) : \mathcal{S}'_k \subseteq [m]$, $k \in [K] \mapsto \mathbb{R}$ at an $\mathcal{S}'_{1:K}$.*

Time complexity of Algorithm 1: Algorithm 1's time complexity is broken down into two parts: a) the number of evaluations of $\mathbb{H}(x_{1:K}|y_{1:K})$ required by the algorithm; b) the time complexity of each such evaluation. In more detail:

a) Number of evaluations of $\mathbb{H}(x_{1:K}|y_{1:K})$ required by Algorithm 1: Algorithm 1 requires at most s_k^2 evaluations of $\mathbb{H}(x_{1:K}|y_{1:K})$ at each $k \in [K]$. Therefore, Algorithm 1 achieves a time complexity that is only linear in K with respect to the number of evaluations of $\mathbb{H}(x_{1:K}|y_{1:K})$; the reason is that $\sum_{k=1}^{K} s_k^2 \leq \max_{k\in[K]}(s_k^2)K$. This is in contrast to the algorithm in Remark 1, that obtains the best approximation factor $1/e$, whose time complexity is of the order $O((nK)^{11})$ with respect to the number of evaluations of $\mathbb{H}(x_{1:K}|y_{1:K})$.[2]

b) Time complexity of each evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$: This time complexity depends on the properties of both the stochastic process (1) (similarly, (2)) and the measurement noise $v_{i,k}$ in (5). For the case of Gaussian stochastic processes and measurement noises:

[2]We can also speed up Algorithm 1 by implementing in Algorithm 2 the method of lazy evaluations [32]: this method avoids in Step 2 of Algorithm 2 the computation of $\rho_i(\mathcal{S}^{t-1})$ for unnecessary choices of i.

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

```
Input: Current iteration $k$, selected sensor sets $(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{k-1})$ up to the current iteration, constraint $s_k$, error metric $\mathbb{H}(x_{1:K}|\mathcal{S}_{1:K}) : \mathcal{S}_k \subseteq [m], k \in [K] \mapsto \mathbb{R}$
Output: Sensor set $\mathcal{S}_k$ that approximates the solution to Problem 1 at time $k$
  $\mathcal{S}^0 \leftarrow \emptyset$, $\mathcal{X}^0 \leftarrow [m]$, 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:
        $\rho_i(\mathcal{S}^{t-1}) \equiv \mathbb{H}(x_{1:K}|\mathcal{S}_{1:k-1}, \mathcal{S}^{t-1}) - \mathbb{H}(x_{1:K}|\mathcal{S}_{1:k-1}, \mathcal{S}^{t-1} \cup \{i\})$
 3.a. If $|\mathcal{S}^{t-1} \cup \{i(t)\}| > s_k$, $\mathcal{X}^{t-1} \leftarrow \mathcal{X}^{t-1} \setminus \{i(t)\}$, and go to Step 1
 3.b. If $|\mathcal{S}^{t-1} \cup \{i(t)\}| \leq s_k$, $\mathcal{S}^t \leftarrow \mathcal{S}^{t-1} \cup \{i(t)\}$ and $\mathcal{X}^t \leftarrow \mathcal{X}^{t-1} \setminus \{i(t)\}$
   4. $t \leftarrow t + 1$ and continue
```

Theorem 3. *Consider the Gaussian process model (3) and suppose that the measurement noise is Guassian: $v_{i,k} \sim \mathcal{N}(0, \Sigma(v_{i,k}))$ such that $\Sigma(v_{i,k}) \succ 0$. The time complexity of evaluating $\mathbb{H}(x_{1:K}|y_{1:K})$ depends on the sparsity pattern of $\Sigma(x_{1:K})$ and $\Sigma(x_{1:K})^{-1}$ as follows.*

- *Each evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$ has time complexity $O(n^{2.4}K)$, when* either *$\Sigma(x_{1:K})$* or *$\Sigma(x_{1:K})^{-1}$ is exactly sparse (that is, block tri-diagonal).*
- *Each evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$ has time complexity $O(n^{2.4}K^{2.4})$, when* both *$\Sigma(x_{1:K})$* and *$\Sigma(x_{1:K})^{-1}$ are dense.*

Theorem 3 implies that when $\Sigma(x_{1:K})$ or $\Sigma(x_{1:K})^{-1}$ is exactly sparse, the time complexity of each evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$ is only linear in K. This is important because $\Sigma(x_{1:K})$ or $\Sigma(x_{1:K})^{-1}$ is exactly sparse for several applications and system models [33]. For example, in adversarial target tracking applications, where the target wants to avoid capture and randomizes its motion in the environment (by uncorrelating its movements), $\Sigma(x_{1:K})$ can be considered tri-diagonal (since this implies $x(t_k)$ and $x(t_{k'})$ are uncorrelated for $|k - k'| > 2$). Similarly, in SLAM, or in system models where the Gaussian process in (3) is generated by a linear or nonlinear system corrupted with Gaussian noise, $\Sigma(x_{1:K})^{-1}$ is block tri-diagonal [23]. In particular, for linear systems, $\Sigma(x_{1:K})^{-1}$ is block tri-diagonal [23, Section 3.1], and for nonlinear systems, $\Sigma(x_{1:K})^{-1}$ is efficiently approximated by a block tri-diagonal matrix as follows: for each k, before the k-th iteration of Step 1 in Algorithm 1, we first compute $\tilde{\mu}_{1:K}$ given $y_{1:(k-1)}$ up to k. This step has complexity $O(n^{2.4}K)$ when $\Sigma(x_{1:K})^{-1}$ is sparse [23, Eq. (5)] [34, Section 3.8], and it does not increase the total time complexity of Algorithm 1. Then, we continue as in [23, Section 3.2].

Sparsity in $\mathbb{H}(x_{1:K}|y_{1:K})$: We state the two properties of $\mathbb{H}(x_{1:K}|y_{1:K})$ that result to Theorem 3. In particular, we prove that $\mathbb{H}(x_{1:K}|y_{1:K})$ is expressed in closed form with two different formulas such that the time complexity for the evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$ using the first formula is decided by the sparsity pattern of $\Sigma(x_{1:K})$, whereas using the second formula is decided by the sparsity pattern of $\Sigma(x_{1:K})^{-1}$. The reason for this dependence is that the rest of the matrices in these formulas are sparser than $\Sigma(x_{1:K})$ or $\Sigma(x_{1:K})^{-1}$; in particular, they are block diagonal.

The full characterization of Algorithm 1's time complexity for Gaussian processes and Gaussian measurement noises follows.

Corollary 1. *Consider the Gaussian process model (3) and suppose that the measurement noise is Gaussian: $v_{i,k} \sim \mathcal{N}(0, \Sigma(v_{i,k}))$ such that $\Sigma(v_{i,k}) \succ 0$. The time complexity of Algorithm 1 depends on the sparsity pattern of $\Sigma(x_{1:K})$ and $\Sigma(x_{1:K})^{-1}$ as follows.*

- *Algorithm 1 has time complexity $O(n^{2.4}K\sum_{k=1}^{K} s_k^2)$, when* either $\Sigma(x_{1:K})$ or $\Sigma(x_{1:K})^{-1}$ *is exactly sparse (that is, block tri-diagonal).*
- *Algorithm 1 has time complexity $O(n^{2.4}K^{2.4}\sum_{k=1}^{K} s_k^2)$, when* both $\Sigma(x_{1:K})$ and $\Sigma(x_{1:K})^{-1}$ *are dense.*

Comparison of Algorithm 1's time complexity for Gaussian processes and Gaussian measurement noises, per Corollary 1, to that of existing scheduling algorithms: The most relevant algorithm to Algorithm 1 is the one provided in [10], where linear systems with additive process noise and measurement noises with *any* distribution are assumed. Algorithm 1 generalizes [10] from linear systems and measurements to Gaussian processes and nonlinear measurements. At the same time, it achieves the same time complexity as the algorithm in [10] when $\Sigma(x_{1:K})$ or $\Sigma(x_{1:K})^{-1}$ is exactly sparse. This is important since the algorithm in [10] has time complexity lower than the-state-of-the-art batch estimation sensor scheduling algorithms, such as the algorithm proposed in [11], and similar to that of the state of the art Kalman filter scheduling algorithms, such as those proposed in [14], [17], [21] (in particular, lower for large K).

IV. Conclusion

In this paper, we proposed Algorithm 1 for the NP-hard problem of sensor scheduling for stochastic process estimation. Exploiting the supermodularity and monotonicity of conditional entropy, we proved that the algorithm has an approximation factor $1/2$ and linear complexity in the scheduling horizon. It achieves both the accuracy of batch estimation scheduling algorithms and, surprisingly, when the information structure of the problem is sparse, the low time complexity of Kalman filter scheduling algorithms for linear systems. This is the case, for example, in applications such as SLAM and target tracking, and for processes generated by linear or nonlinear systems corrupted with Gaussian noise. Future work will focus on an event-triggered version of the scheduling problem, in which the measurement times are decided online based on the available measurements, and on a decentralized version, in which information is exchanged only among neighboring sensors.

Appendix A: Proof of Proposition 2

Proof. We first show that the conditional probability distribution of $x_{1:K}$ given $y_{1:K}$ is Gaussian with covariance $\Sigma(x^\star_{1:K})$, and then apply the following lemma:

Lemma 1 (Ref. [35]). *Let $x \sim \mathcal{N}(\mu, \Sigma)$ and $x \in \mathbb{R}^m$:*

$$\mathbb{H}(x) = \frac{1}{2}\log[(2\pi e)^m \det(\Sigma)].$$

Specifically, due to Assumption 1, $(x_{1:K}, y_{1:K})$ are jointly Gaussian. This has a twofold implication: first, the minimum mean square estimator of $x_{1:K}$ given $y_{1:K}$ is linear in $y_{1:K}$ [25, Proposition E.2]; second, the conditional probability distribution of $x_{1:K}$ given $y_{1:K}$ is Gaussian [36], with covariance $\Sigma(x^\star_{1:K})$. Therefore, due to [25, Proposition E.3], this is also the covariance of the minimum mean square estimator of $x_{1:K}$ given $y_{1:K}$. As a result, due to Lemma 1:

$$\begin{aligned}\mathbb{H}(x_{1:K}|y_{1:K}) &= \mathbb{E}_{y_{1:K}=y'_{1:K}}\left(\mathbb{H}(x_{1:K}|y_{1:K}=y'_{1:K})\right)\\ &= \mathbb{E}_{y_{1:K}=y'_{1:K}}\left(\frac{1}{2}\log[(2\pi e)^{nK}\det(\Sigma(x^\star_{1:K}))\right)\\ &= \frac{nK\log(2\pi e)+\log\det(\Sigma(x^\star_{1:K}))}{2}. \qquad (9)\end{aligned}$$

We derive a formula for $\Sigma(x^\star_{1:K})$ in the proof of Lemma 2. □

Appendix D: Proof of Theorem 3

Notations: We introduce four notations: first, $S_{1:K}$ is the block diagonal matrix with diagonal elements the sensor selection matrices $S_1, S_2, \ldots, S_K$; second, $C(x_{1:K})$ is the block diagonal matrix with diagonal elements the matrices $S_1C_1, S_2C_2, \ldots, S_KC_K$, where $C_k \equiv G(x_k)$ and $G(x(t)) \equiv \partial g(x(t))/\partial x(t)$; third, v_k is the batch measurement noise vector $(v_{1,k}^\top, v_{2,k}^\top, \ldots, v_{m,k}^\top)^\top$; and fourth, $\mu_{1:K} \equiv (\mu(t_1)^\top, \mu(t_2)^\top, \ldots, \mu(t_K)^\top)^\top$.

Proof. We first derive the two formulas for $\mathbb{H}(x_{1:K}|y_{1:K})$: the first formula is expressed in terms of $\Sigma(x_{1:K})^{-1}$, and the second formula is expressed in terms of $\Sigma(x_{1:K})$.

Lemma 2 (Formula of $\mathbb{H}(x_{1:K}|y_{1:K})$ in terms of $\Sigma(x_{1:K})^{-1}$). *Consider the start of the k-th iteration in Algorithm 1. Given the measurements $y_{1:(k-1)}$ up to k, $\mathbb{H}(x_{1:K}|y_{1:K})$ is given by $-T'_1 + nK\log(2\pi e)/2$, where:*

$$\begin{aligned}T'_1 &\equiv \frac{1}{2}\log\det(\Xi + \Sigma(x_{1:K})^{-1})\\ \Xi &\equiv C(\tilde{\mu}_{1:K})^\top S_{1:K}\Sigma(v_{1:K})^{-1}S_{1:K}^\top C(\tilde{\mu}_{1:K})\end{aligned}$$

and $\tilde{\mu}_{1:K}$ is the maximum a posteriori (MAP) estimate of $x_{1:K}$ given the measurements $y_{1:(k-1)}$ up to k.

Lemma 3 (Formula of $\mathbb{H}(x_{1:K}|y_{1:K})$ in terms of $\Sigma(x_{1:K})$). *Consider the start of the k-th iteration in Algorithm 1. Given *the measurements $y_{1:(k-1)}$ up to k, $\mathbb{H}(x_{1:K}|y_{1:K})$ is given by $\mathbb{H}(x_{1:K}|y_{1:K}) = T_1 - T_2 + \mathbb{H}(x_{1:K})$, where:*

$$T_1 \equiv \frac{1}{2}\sum_{k=1}^{K} \log[(2\pi e)^{s_k} \det(S_k \Sigma(v_k) S_k^\top)] \tag{10}$$

$$T_2 \equiv \frac{1}{2}\log[(2\pi e)^{\sum_{k=1}^{K} s_k} \det(\Sigma(y_{1:K}))] \tag{11}$$

$$\Sigma(y_{1:K}) = S_{1:K}\Sigma(v_{1:K})S_{1:K}^\top + C(\tilde{\mu}_{1:K})\Sigma(x_{1:K})C(\tilde{\mu}_{1:K})^\top,$$

and $\tilde{\mu}_{1:K}$ is the maximum a posteriori (MAP) estimate of $x_{1:K}$ given the measurements $y_{1:(k-1)}$ up to k.

We complete the proof for each case of Theorem 3:

- *Time complexity of each evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$ when either $\Sigma(x_{1:K})$ or $\Sigma(x_{1:K})^{-1}$ is exactly sparse (that is, block tri-diagonal)*: We present the proof only for the case where $\Sigma(x_{1:K})^{-1}$ is exactly sparse since the proof for the case where $\Sigma(x_{1:K})$ is exactly sparse is similar. In particular, consider the formula of $\mathbb{H}(x_{1:K}|y_{1:K})$ in Lemma 2: T_1' involves the log determinant of a matrix that is the sum of two $nK \times nK$ sparse matrices: the first matrix is block diagonal, and the second one is block tri-diagonal. The block diagonal matrix is evaluated in $O(n^{2.4}K)$ time, since the determinant of an $n \times n$ matrix is computed in $O(n^{2.4})$ time using the Coppersmith-Winograd algorithm [37]. Then, T_1' is evaluated in $O(n^{2.4}K)$ [38, Theorem 2].
- *Time complexity of each evaluation of $\mathbb{H}(x_{1:K}|y_{1:K})$ when both $\Sigma(x_{1:K})$ and $\Sigma(x_{1:K})^{-1}$ are dense*: In this case, T_1' (and similarly T_2 in Lemma 3) is the log determinant of a dense $nK \times nK$ matrix. Therefore, it is evaluated in $O((nK)^{2.4})$ time, since the determinant of an $n \times n$ matrix is computed in $O(n^{2.4})$ time using the Coppersmith-Winograd algorithm [37]. □

References

[1] 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.

[2] N. Karnad, *Robot Motion Planning for Tracking and Capturing Adversarial, Cooperative and Independent Targets*. U. of Minnesota, 2015.

[3] M. P. Vitus, "Sensor placement for improved robotic navigation," *Robotics: Science and Systems VI*, p. 217, 2011.

[4] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[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] S. Karlin, *A first course in stochastic processes*. Academic press, 2014.

[7] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall Upper Saddle River, NJ, 2000, vol. 1.

[8] 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.

[9] A. O. Hero III and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, 2011.

[10] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Near-Optimal Sensor Scheduling for Batch State Estimation: Complexity, Algorithms, and Limits," in *55th IEEE Conference on Decision and Control (CDC)*, 2016.

[11] V. Roy, A. Simonetto, and G. Leus, "Spatio-temporal sensor management for environmental field estimation," *Signal Processing*, vol. 128, pp. 369 – 381, 2016.

[12] 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.

[13] 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.

[14] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[15] J. Le Ny, E. Feron, and M. A. Dahleh, "Scheduling continuous-time kalman filters," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1381–1394, 2011.

[16] X. Shen, S. Liu, and P. K. Varshney, "Sensor selection for nonlinear systems in large sensor networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2664–2678, 2014.

[17] S. Liu, M. Fardad, P. K. Varshney, and E. Masazade, "Optimal periodic sensor scheduling in networks of dynamical systems," *Signal Processing, IEEE Transactions on*, vol. 62, no. 12, pp. 3055–3068.

[18] 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.

[19] Y. He and E. K. P. Chong, "Sensor scheduling for target tracking: A monte carlo sampling approach," *Digital Signal Processing*, vol. 16, no. 5, pp. 533–545, 2006.

[20] 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.

[21] 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.

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

[23] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch nonlinear continuous-time trajectory estimation as exactly sparse gaussian process regression," *Autonomous Robots*, vol. 39, no. 3, pp. 221–238, 2015.

[24] R. Durrett, *Probability: theory and examples*. Cambridge University Press, 2010.

[25] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I*, 3rd ed. Athena Scientific, 2005.

[26] M. L. Fisher, G. L. Nemhauser, and L. Wolsey, *An analysis of approximations for maximizing submodular functions–II*. Springer, 1978, 1978.

[27] J. Vondrák, "Submodularity and curvature: the optimal algorithm," *RIMS Kokyuroku Bessatsu B*, vol. 23, pp. 253–266, 2010.

[28] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience, 1988.

[29] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[30] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998.

[31] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM, 2008, pp. 67–74.

[32] M. Minoux, "Accelerated greedy algorithms for maximizing submodular functions," in *Optimization Techniques*. Springer, 1978, pp. 234–243.

[33] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[34] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical mathematics*. Springer Science & Business Media, 2010, vol. 37.

[35] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[36] S. Venkatesh, *The Theory of Probability: Explorations and Applications*. Cambridge University Press, 2012.

[37] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987, pp. 1–6.

[38] L. G. Molinari, "Determinants of block tridiagonal matrices," *Linear algebra and its applications*, vol. 429, no. 8, pp. 2221–2226, 2008.