A Complete Greedy Algorithm for Infinite-Horizon Sensor Scheduling [2]

Ahmad Bilal Asghar [a], Syed Talha Jawaid [1], Stephen L. Smith [a]

[a] *Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada*

Abstract

In this paper we study the problem of scheduling sensors to estimate the state of a linear dynamical system. The estimator is a Kalman filter and our objective is to optimize the *a posteriori* error covariance over an infinite time horizon. We focus on the case where a fixed number of sensors are selected at each time step, and we characterize the exact conditions for the existence of a schedule with uniformly bounded estimation error covariance. Using this result, we construct a scheduling algorithm that guarantees that the error covariance will be bounded if the existence conditions are satisfied. We call such an algorithm complete. Finally, we provide simulations to compare the performance of the algorithm against other known techniques.

1 Introduction

One technique for monitoring an environmental process is to deploy a sensor network. Each sensor can be equipped with the ability to make a range of measurements. Sensor networks have been used in various applications including determining a robot's state [5], tracking the position of a target [7], selecting the frequency in radar and sonar applications, or monitoring tasks such as chemical processes [11], seismic activity or toxin levels at a factory. Sensor scheduling techniques can also be applied to problems such as adaptive compressed sensing [12].

The collection of data can be done by operating every sensor continuously; however, the network may be required to have a long life span and so this strategy may not be viable due to energy and communication constraints. To overcome these restrictions, sensors can alternate between awake and asleep modes. Unless the network provides enough redundancy, this method could result in an incomplete picture of the phenomenon of interest. Therefore, a sensing schedule has to be constructed in an intelligent way in order to obtain as much information as possible. This is, in essence, the sensor scheduling problem.

The sensor scheduling problem has received considerable attention in recent years. In the context of linear Gaussian systems, a Kalman filter is the optimal estimator in that it produces an estimate with the least mean square error. Thus, the Kalman filter is commonly used as the basis for the sensor scheduling problem. An exception is [6], where the condition number of the sequence observability matrix is used as a metric to find a sensor schedule. In this paper we will use a metric on the error covariance of the Kalman filter as our objective function. With this setting, the infinite horizon sensor scheduling problem is studied in [23]. Under some mild conditions, it is shown that the optimal infinite horizon schedule is independent of the initial covariance. Also, it is shown that given an optimal schedule, its cost can be estimated arbitrarily closely by a periodic schedule, with a finite period. However, if the optimal schedule is not known, the analysis does not provide a constructive method for efficiently computing an approximate period schedule.

Numerous approaches have been proposed to tackle the sensor scheduling problem. The results in [23] serve as a reason to find optimal periodic schedules for infinite horizon scheduling problem. The authors in [19] find a periodic schedule using a branch-and-bound approach. In [18] the authors find an optimal periodic schedule by approximating the objective function of the sensor scheduling problem. A locally optimal solution to periodic scheduling was proposed in [13] with constraints on the number of times each sensor can be used in a period.

Email addresses: abasghar@uwaterloo.ca (Ahmad Bilal Asghar), stjawaid@uwaterloo.ca (Syed Talha Jawaid), stephen.smith@uwaterloo.ca (Stephen L. Smith).

[1] This work was performed while S. T. Jawaid was at the University of Waterloo.

[2] This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).The material in this paper was partially presented at the 2014 American Control Conference, June 46, 2014, Portland, OR, USA.

Their objective function incorporates both the estimation error and the number of sensors used per time step. A drawback to these approaches is that the optimal period is unknown, and thus the desired period must be given as an input.

Optimal and semi-optimal algorithms for the finite horizon problem that use tree pruning techniques are provided in [20]. In [3], three different approaches (sliding window, greedy thresholding and random selection) are empirically compared. The authors further develop the random selection method in [4], where a strategy for stochastically selecting measurements based on an intelligently constructed probability distribution is described and bounded. In [14], a few different approaches are studied, including a best step look ahead algorithm, an approach based on the Viterbi algorithm and another by casting the problem as a duality problem. The algorithms are described and empirically compared in terms of performance and computation time.

A convex relaxation based approach is discussed in [21] and applied to the monitoring of CO_2 using a wireless sensor network. Another convex relaxation approach is given in [10] along with solution dependent bounds. In [17], this approach is, however, empirically shown to be worse than a greedy algorithm. In [9], authors studied some properties of greedy sensor scheduling algorithms and their relation to submodular set functions.

A general framework for the sensor scheduling problem is presented in [16]. A number of problems can be addressed in this framework such as minimizing the final covariance over a time horizon, the average covariance, the variance of a single state, or even the cost of a finite horizon LQG regulator. A number of network constraints can also be included. The problem is framed as a relaxed quadratic program, and a greedy approach is described although the error bound is not necessarily uniformly bounded for unstable systems. In [15], a continuous time sensor scheduling problem is considered for an objective capturing both estimation error and sensor switching costs.

In this paper we consider infinite-horizon sensor scheduling. Based on the discussion above, existing approaches for this problem are 1) to fix a period and compute a periodic schedule; 2) to repeatedly apply a finite-horizon algorithm; or 3) to greedily select sensors at each time step. For each of these methods, there are no guarantees that the resulting schedule will produce a uniformly bounded sequence of covariance matrices. In fact, we do not know of any results that characterize the exact conditions under which an infinite horizon sensor schedule exists that results in a uniformly bounded sequence of covariance matrices.

Contributions: We give necessary and sufficient conditions for the existence of an infinite horizon sensor schedule with a bounded error covariance (Section 4). We then provide a complete algorithm for sensor scheduling (Section 5): That is, our algorithm outputs a uniformly bounded sensor schedule if one exists. The algorithm has the same runtime as the simple greedy algorithm and we show in simulations (Section 6) that our proposed algorithm outperforms the greedy algorithm, and can be used to efficiently compute schedules for high-dimensional linear systems with a large number of sensors .

A preliminary version of this paper was presented in [8]. Relative to this early version, we now provide a more efficient algorithm along with detail on its implementation. We also extend both the algorithm and analysis to the general problem of k sensors per time step, and provide complete proofs of the correctness of the proposed greedy algorithm. Finally, we present more extensive simulation results on high-dimensional linear systems, including a system obtained by discretizing the heat equation.

2 Preliminaries

Consider the discrete-time linear stochastic system

$$\begin{aligned} x_{t+1} &= Ax_t + w_t, \quad x_t \in \mathbb{R}^n, \\ y_t &= C_t x_t + v_t, \quad y_t \in \mathbb{R}^k \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times n}$. The matrix C_t is a subset of k rows of C. This is the standard sensor selection model, as in [20,16]. The process noise w_t measurement noise v_t are zero mean Gaussian noise vectors with covariance matrices $W, V \in \mathbb{R}^{n\times n}$, respectively, with $W \succeq 0$ and $V \succ 0$. We assume that the noises are independent over time.

For the case $C_t = C$ (LTI system), the system is said to be **observable** if its **observability matrix** $\Theta = \mathrm{col}(C, CA, \ldots, CA^{n-1})$ has rank n.

If the observability matrix is not full rank then a similarity transform T can be used to convert the system into **standard form for unobservable systems.**

$$\bar{A} = T^{-1}AT = \begin{bmatrix} A_{\bar{o}} & A_{12} \\ 0 & A_o \end{bmatrix}, \quad \bar{C} = CT = \begin{bmatrix} 0 & C^o \end{bmatrix}. \tag{2}$$

Here (A_o, C^o) is observable. If $A_{\bar{o}}$ is stable, the system is said to be detectable.

Consider a sequence of measurements $\sigma = (\sigma_0, \sigma_1, \ldots)$, and the corresponding sequence of matrices $(C_0, C_1, \ldots)$. For a given time t and time window k, the **sequence observability matrix** for the given system can be written as

$$B_\sigma(t, t+k) = \mathrm{col}(C_t, C_{t+1}A, \ldots\ C_{t+k}A^k).$$

The following definition follows from the definition of Uniform Detectibility in [1].

Definition 1 (Uniform Detectability) *For the system in* (1)*, the sequence of measurements σ is uniformly detectable if there exists non-negative integers s, r and constants $\alpha \in [0,1)$ and $\beta > 0$, such that for all $\{x \in \mathbb{R}^n | \, \|x\| = 1\}$ and all times t,*

$$\|A^r x\| \geq \alpha \implies \|B(t, t+s)x\| \geq \beta > 0. \quad (3)$$

Additionally, for the given system, the sequence is uniformly observable if there exists integer s and positive constants β_1, β_2 such that

$$\begin{aligned} &0 < \beta_1 \leq \|B(t,t+s)x\| \leq \beta_2 \\ \iff &\operatorname{rank}(B(t,t+s)) = n. \end{aligned} \quad (4)$$

Note that for a general time varying system, the equivalence in (4) holds only in the forward direction. For example, the reverse direction does not necessarily hold when C_t or A_t can take on infinitely many values. This, however, is not the case for the system in (1), where A is fixed and C_t is a subset of rows of a time invariant C.

Finally, we will use the following result for some of our proofs.

Lemma 2 *Suppose (A, C) is observable and A is full rank. Then, letting c_i be the i^{th} row of C,*

$$B \triangleq col(c_1, c_1 A \dots, c_1 A^{n-1}, c_2 A^n, \dots, c_m A^{mn-1}) \quad (5)$$

is full rank.

PROOF. Since $\operatorname{rank}(\Theta) = n$, it suffices to show that each of the rows of Θ can be written as a linear combination of the rows in B. Let $X_i = \{c_i, c_i A, \dots, c_i A^{n-1}\}$ for $i = 1, \dots, m$. Note that the rows of Θ comprise of the vectors in the multiset $\bigcup_{i=1}^m X_i$. Also, note that $x \in X_i \implies x A^{(i-1)n}$ is a row of B. Let $X_i^b = X_i A^{(i-1)n} = \{c_i A^{(i-1)n}, c_i A^{(i-1)n+1}, \dots, c_i A^{in-1}\}$. So the rows of B comprise of elements of the multiset $\bigcup_{i=1}^m X_i^b$.

For any particular $1 \leq i \leq m$, there are $k_i \leq n$ linearly independent (LI) vectors in X_i. Since A is full rank, the set X_i^b contains k_i LI vectors as well.[3] Also, due to the Cayley-Hamilton theorem, $x \in X_i^b \implies x \in \operatorname{span}(X_i)$. Any k LI vectors in the span of X_i will themselves span the space. As a result, every vector in X_i is in $\operatorname{span}(X_i^b)$.

Therefore, if A is full rank, then every row of Θ is in the span of the rows of B, which means that $\operatorname{rank}(B) \geq \operatorname{rank}(\Theta) = n$. Since B is an $mn \times n$ matrix, $\operatorname{rank}(B) \leq n$. So $\operatorname{rank}(B) = n$. □

3 Problem Statement

Consider the dynamical system (1). Each row of C corresponds to a single sensor in the sensor network. For the sensor scheduling problem, we pick a set of k sensors at every time step to make a measurement (i.e., k rows of C). A Kalman filter uses noisy measurements to estimate the state of the system. The *a posteriori* and *a priori* covariances using the Kalman filter are given by

$$\begin{aligned} &\Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1} C_t^\top \left(C_t \Sigma_{t|t-1} C_t^\top + V_t\right)^{-1} C_t \Sigma_{t|t-1} \\ &\Sigma_{t+1|t} = A \Sigma_{t|t} A^\top + W. \end{aligned} \quad (6)$$

where V_t comprises of the k rows and columns of V corresponding to the selected sensors. It is known that the Kalman filter gives the best mean squared error of the state estimate among all linear estimators. An interesting question is under what conditions is the filter is stable? This is answered in the following lemma[4] derived from the results in [1].

Lemma 3 *Assume that the system $(A, W^{\frac{1}{2}})$ is uniformly stabilizable. Then the Kalman filter error covariance, $\Sigma_{t|t}$, and predictor covariance, $\Sigma_{t+1|t}$, are bounded if and only if (A, C_t) is uniformly detectable. Furthermore, the Kalman filter is exponentially stable only if (A, C_t) is uniformly detectable.*

We can represent a sensor schedule as $\sigma = (\sigma_0, \sigma_1, \dots)$, where $\sigma_t \subseteq \{1, \dots, m\}$ gives the indices of the k sensors chosen at time step t. There are no constraints on the number times each sensor can be chosen. The problem that we consider is the following: under what conditions on A and C does there exist a sensor schedule $\sigma = (\sigma_0, \sigma_1, \dots)$ that results in the error covariance being bounded? Moreover, how do we construct such a schedule? The existing scheduling algorithms do not guarantee the boundedness of the resulting error covariance. We define a schedule that results in a bounded error covariance as a bounded sensor schedule. Formally, we seek to keep $F(\sigma)$ bounded, for some metric F such as $\operatorname{trace}(\Sigma_{t|t})$ or $-\log\det(\Sigma_{t|t}^{-1})$ where $\Sigma_{t|t}$ is function of the schedule σ. These covariance metrics are summarized in [9], along with their interpretations. In this paper the function $F(\sigma)$ can refer to any of the metrics unless specified otherwise.

[3] Given a full column rank matrix $A \in \mathbb{R}^{m \times n}$ and $k \leq n$ LI vectors $\{x_i\}_{i=1}^k$, the set of vectors $\{Ax_i\}_{i=1}^k$ is also LI.

[4] The results here are specific to systems in the form of (1).

4 Existence of a Bounded Sensor Schedule

The question we now ask is, given an LTI system, does there exist a sequence of measurements that results in uniformly bounded error covariance? Using Lemma 3, it is equivalent to asking if there exists a uniformly detectable sequence of measurements. In this section we address this question.

Proposition 4 *There exists a sequence of measurements resulting in bounded Kalman filter error covariance if and only if (A, C) is detectable.*

PROOF. Sufficient Condition: To prove this result, we first constructively show in the following lemma that for one sensor per time step ($k = 1$), the periodic sequence that repeats each measurement n times, i.e. each period is $\sigma_C = (1, \ldots, 1, 2, \ldots, 2, \ldots, m, \ldots, m)$, will be uniformly detectable.

Lemma 5 *If (A, C) is detectable, then the periodic sequence of measurements σ_C is uniformly detectable.*

PROOF. Let T be defined as in (2). Let z and l be the number of zero and stable eigenvalues respectively of the observable component A_o, which is a $d \times d$ matrix. Consider the transform

$$Q := T \begin{bmatrix} I & 0 \\ 0 & V_{d\times d} \end{bmatrix}_{n\times n}$$

$$\tilde{A} = Q^{-1}AQ = \begin{bmatrix} A_{\bar{o}} & A_{12}V \\ 0 & \begin{bmatrix} A_{on} & 0 & 0 \\ 0 & A_{os} & 0 \\ 0 & 0 & A_{ou} \end{bmatrix} \end{bmatrix} \qquad (7)$$

$$\tilde{C} = CQ = \begin{bmatrix} 0 & C_1 & C_2 & C_3 \end{bmatrix}$$

where the columns of V are eigenvectors corresponding to stable and unstable eigenvalues of A_o. Here, A_{os} and A_{ou} have stable and unstable eigenvalues respectively, are both full rank and are composed of Jordan blocks. A_{on} is also composed of Jordan blocks and has all zero eigenvalues. Therefore, it is nilpotent and $A_{on}^z = 0$. Also, by definition of detectability, $A_{\bar{o}}$ is stable. For this system, we can define $\tilde{\Theta} = \begin{bmatrix} 0 & \Theta_1 & \Theta_2 & \Theta_3 \end{bmatrix}$ and $\tilde{B}(t, t+s) = \begin{bmatrix} 0 & B_1 & B_2 & B_3 \end{bmatrix}$. Note that $\begin{bmatrix} \Theta_1 & \Theta_2 & \Theta_3 \end{bmatrix}$ is full rank since this part corresponds to the observable subsystem. For detectability of the sequence to hold, there should exist (s, r, α, β) such that (3) is satisfied. Following from (3), we consider only initial states of unit norm ($\|u\| = 1$).

Case 1: $u_3 = 0$ or A_{ou} does not exist (i.e., A is stable): In this case, only the stable modes are active and so the state approaches 0 exponentially. As a result, for any $\alpha \in (0, 1)$, there exists $r > 0$ such that $\left\|\tilde{A}^r u\right\| < \alpha$ for all u.

Case 2: $\|u_3\| > 0$: In this case, (s, β) can be chosen so that $\left\|\tilde{B}(t, t+s)u\right\| \geq \beta$ irrespective of the values of (r, α).

Take $s = 2mn$ so that $\tilde{B}(t, t+s)$ always contains the full sequence σ_C. As a result, the *sequence observability matrix* for s time steps, $\tilde{B}(t, t+s)$, will always contain the rows of $\tilde{B}_\sigma \tilde{A}^p$ for some $p > 0$, where $\tilde{B}_\sigma$ is defined as

$$\begin{aligned} \tilde{B}_\sigma &= \mathrm{col}(\tilde{c}^1, \ldots, \tilde{c}^1 \tilde{A}^{n-1}, \ldots, \tilde{c}^m \tilde{A}^{mn-1}) \\ &=: \begin{bmatrix} 0 & B_1 & B_2 & B_3 \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} B_1 &= \mathrm{col}(c_1^1, \ldots, c_1^1 A_{on}^{z-1}, 0, \ldots, 0) \\ B_2 &= \mathrm{col}(c_2^1, \ldots, c_2^1 A_{os}^{n-1}, \ldots, c_2^m A_{os}^{mn-1}) \\ B_3 &= \mathrm{col}(c_3^1, \ldots, c_3^1 A_{ou}^{n-1}, \ldots, c_3^m A_{ou}^{mn-1}) \end{aligned}$$

and $\tilde{c}^i$ is the i^{th} row of $\tilde{C}$. Note that

$$\tilde{B}_\sigma \tilde{A}^p = \begin{bmatrix} 0 & B_1 A_{on}^p & B_2 A_{os}^p & B_3 A_{ou}^p \end{bmatrix}.$$

Also, $\begin{bmatrix} \Theta_2 & \Theta_3 \end{bmatrix}$ is full rank and, since both A_{os} and A_{ou} are full rank, $\begin{bmatrix} B_2 & B_3 \end{bmatrix}$ is also full rank (by Lemma 2) and so is $\begin{bmatrix} B_2 A_{os}^p & B_3 A_{ou}^p \end{bmatrix}$. Note that $B_1 A_{on}^p = 0$ for $p \geq z$.

Now, without loss of generality, assume that the sequence σ_C starts at time t. So, since $p = mn \geq z$,

$$\begin{aligned} \tilde{B}(t, t+s)u &= \begin{bmatrix} \tilde{B}_\sigma \\ \tilde{B}_\sigma \tilde{A}^{mn} \end{bmatrix} u \\ &= \begin{bmatrix} B_1 u_1 + B_2 u_2 + B_3 u_3 \\ B_2 A_{os}^{mn} u_2 + B_3 A_{ou}^{mn} u_3 \end{bmatrix} =: \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}. \end{aligned}$$

Now, $d_2 = 0$ if and only if $\begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = 0$. Given that $u_3 \neq 0$, it follows that $\left\|\tilde{B}(t, t+s)u\right\| \geq \|d_2\| > 0$. Therefore,

$$\beta \leq \min_{\{u | u_1 = 0, \|u\| = 1\}} \left\| B_2 A_{os}^{mn} u_2 + B_3 A_{ou}^{mn} u_3 \right\|,$$

is an appropriate choice to obtain detectability. □

For k sensors per time step, consider the periodic schedule σ_C in which the first k sensors are selected for the first n time steps, the next k for the next n time steps, and so on. For example, if $m = 5$ and $k = n = 2$, then $\sigma_C = (\{1,2\},\{1,2\},\{3,4\},\{3,4\},\{5,1\},\{5,1\})$. This sequence has a size of $\lceil \frac{m}{k} \rceil n \leq mn$. This sequence is uniformly detectable by a similar argument as used in Lemma 5. The only difference from the proof of Lemma 5 is the formulation of $\tilde{B}_\sigma$. The matrix $\begin{bmatrix} B_2 A^p_{os} & B_3 A^p_{ou} \end{bmatrix}$ will remain full rank and the rest of proof will follow.

Necessary Condition: If (A, C) is not detectable, then there exists an eigenvalue-eigenvector pair, (λ, v), of A such that $|\lambda| \geq 1$ and $Cv = 0$. For any pair (α, r), assuming $\|v\| = 1$, we must have $\|A^r v\| = |\lambda|^r \geq 1 > \alpha$. So, there has to exist (s, β) such that $\|B(t, t+s)v\| \geq \beta$ for uniform detectability. However, the rows of $B(t, t+s)$ consist of vectors of the form $c_i A^k$, for some k, where c_i is a row of C. Now $c_i A^k v = \lambda^k c_i v = 0$ and so $B(t, t+s)v = 0$ no matter what the actual schedule is. Therefore, no sensor schedule can be uniformly detectable.

We have shown that (A, C) is detectable if and only if there exists a sequence of uniformly detectable sequence of measurements. Using Lemma 3 with this result completes the proof. □

It is reasonable to expect that if (A, C) is observable, then a sequence of measurements exists such that the system is uniformly observable through that sequence. This, however, is not the case. Consider the trivial example where $A = 0$ and $k = 1$. Here, it is obvious that $\text{rank}(B(t, t+s)) = 1$ since the second row onwards of B will be 0. As another example, consider $A = \mathbf{1}_{3\times 3}$ and C as a 3×3 lower triangular matrix where all non-zero entries are 1. It is easy to show that $\text{rank}(B(t, t+s))$ will be either 1 or 2 for every schedule. The following result gives the conditions for the system to be uniformly observable.

Proposition 6 *If (A, C) is observable and A is full rank, then the system is uniformly observable through the periodic sequence σ_C.*

PROOF. In order for the sequence to be observable, we require the existence of (s, β) such that $\text{rank}(B(t, t+s)) = n$ for all t. Let B be defined as in equation (5) and take $s = 2mn$ so that $B(t, t+s)$ will always contain the rows of BA^p for some p. Since A is full rank, and B is full rank by Lemma 2, BA^p is full rank and hence $\text{rank}(B(t, t+s)) = n$. □

In the following section we present an algorithm that produces a uniformly detectable schedule.

5 A Complete Sensor Selection Algorithm

We define a complete scheduling algorithm as follows.

Definition 7 (Complete Scheduling Algorithm) *A sensor scheduling algorithm is complete if for every detectable LTI system (A, C), the resulting sequence of error covariance matrices are uniformly bounded for all time.*

A complete sensing schedule can be naively constructed using the periodic sequence σ_C. it will in general result in very large values of the covariance metric $F(\sigma_C)$. The greedy schedule presented in [17] chooses the sensors at each step that minimize the covariance over a single time step. However, establishing the boundedness of greedy algorithm has proved difficult. We present a modified greedy algorithm in this section such that it is complete, and empirically it obtains better performance than the greedy algorithm.

The following example shows that even if the greedy algorithm produces a bounded sensor schedule, it may perform quite poorly.

Example 8 *Consider the pathological system*

$$A = \Sigma_0 = I_{3\times 3}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} =: \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix},$$

$$W = \begin{bmatrix} 0.1 & 0.13 & 0.13 \\ 0.13 & 0.41 & 0.36 \\ 0.13 & 0.36 & 0.33 \end{bmatrix}, \quad V = I_{3\times 3}.$$

Running the greedy algorithm for one sensor per time step, the resulting value of the objective function is plotted in Figure 1. The first time measurement 3 is selected in the greedy schedule is at $t = 8576$ and is then repeated approximately every 73 time steps. Eventually every measurement is made and so the schedule is uniformly detectable though the time window needed is over 8000 time steps.

We now present a modified greedy algorithm that ensures that the output sequence is uniformly detectable.

The DETECTABLEGREEDY algorithm is given in Algorithm 1. The idea is to make a greedy choice at each iteration such that the measurement increases the rank of the matrix M. This is done in lines 8 to 14 where we mark the row rA^s_{su} corresponding to sensor r as valid if

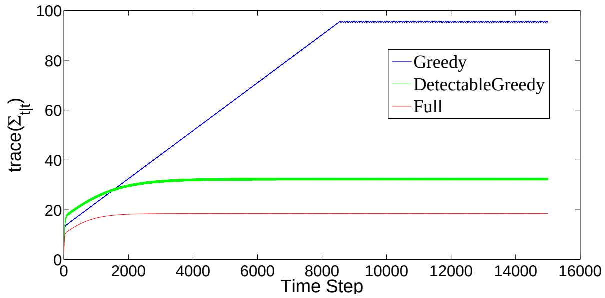

Fig. 1. The value of the greedy sequence compared with the DETECTABLEGREEDY schedule and "full" schedule. The full schedule means that every measurement is made at all times.

it increases the rank of M and then greedily pick the best valid row in line 13. Matrix M emulates the sequence observability matrix $B(t, t+s)$. Once M becomes full rank, it is reset in line 19. As a result, M acts as a sliding window and the algorithm attempts to keep this window fully observable. We now show that the algorithm does, in fact, result in a uniformly detectable schedule. We will reuse notation from the proof of Theorem 5:

$$A_{su} := \begin{bmatrix} A_{os} & 0 \\ 0 & A_{ou} \end{bmatrix}, \quad C_{su} := \begin{bmatrix} C_2 & C_3 \end{bmatrix},$$
$$\Theta_{su} := \begin{bmatrix} \Theta_2 & \Theta_3 \end{bmatrix}, B_{su} := \begin{bmatrix} B_2 & B_3 \end{bmatrix} \quad \text{and} \quad p = \operatorname{rank}(A_{su}).$$

Lemma 9 *The rank of the matrix M will increase within p steps. In other words, consider the matrix at any time such that* $\operatorname{rank}(M(t, t+q)) < p$, *then* $\operatorname{rank}(M(t, t+q+p)) \geq \operatorname{rank}(M(t, t+q)) + 1$.

PROOF. Given the matrix $M(t, t+q)$ with rank less than p, the next p measurements will be chosen sequentially from the sequence of matrices $\{C_{su}A_{su}^{q+i}\}_{i=1}^{p}$. Stacking these on top of each other we get $\Theta_{su}A_{su}^{q+1}$. Since both Θ_{su} and A_{su} are full rank, $\operatorname{rank}(\Theta_{su}A_{su}^{q+1}) = p$. Therefore, at least one of the rows in $\Theta_{su}A_{su}^{q+1}$ will be LI from the rows of $M(t, t+q)$ and can be added to the sequence. So the rank of M will increase by at least 1 after p more time steps. □

Theorem 10 *Algorithm 1 is complete and thus it produces a bounded sensor schedule.*

PROOF. By Lemma 9, a rank increasing row will be selected within p time steps and so the maximum size of the M matrix is $p^2 - p + 1$.

We need to show that the sequence is uniformly detectable for the sensor schedule to be bounded. To show the sequence is uniformly detectable, we can follow the same argument as in the proof of Theorem 5, i.e., we need to show that $\left\|\tilde{B}(t, t+s)u\right\| > 0$ for some s and for any vector that has $u_3 \neq 0$. Note that the matrix M in the algorithm corresponds to sections of $B_{su}(t, t+s)$.

Algorithm 1 DETECTABLEGREEDY

Input: Value function F, system parameters (A, C, W, V), sensors per time step k
Output: Sensor schedule with k sensors per time step

```
1:  $A_{su} := \begin{bmatrix} A_{os} & 0 \\ 0 & A_{ou} \end{bmatrix}$ and $C_{su} := \begin{bmatrix} C_2 & C_3 \end{bmatrix}$ (cf. (7)).
2:  $M \leftarrow$ empty matrix
3:  $\sigma \leftarrow$ empty schedule
4:  $s \leftarrow 0$
5:  for each $t$ :
6:      $\sigma_t \leftarrow \emptyset$
7:      while $|\sigma_t| < k$ :
            // Use $(A_{su}, C_{su})$ to construct $M$
8:          for each row (sensor) $r$ of $C_{su}$ :
9:              if $rA_{su}^s$ increases rank$(M)$ :
10:                 Mark $r$ as valid.
            // Use $(A, C)$ for selection
11:         if No rows are valid :
12:             Mark all rows valid.
13:         Greedily select the valid row $r$ of $C$ that minimizes $F\big((\sigma, \sigma_t \cup \{r\})\big)$
14:         Update $M$ with new measurement
15:         $\sigma_t \leftarrow \sigma_t \cup \{r\}$
16:     Add $\sigma_t$ to end of $\sigma$
17:     $s \leftarrow s + 1$
18:     if rank$(M) =$ rank$(A_{su})$ :
19:         $M \leftarrow$ empty matrix
20:         $s \leftarrow 0$
```

One difference with the proof of Theorem 5 is that here we have a sequence of full rank matrices instead of just one being repeated. Assume, without loss of generality, that line 19 is executed at time $t-1$. So after $k_1 \geq z$ time steps, the matrix M_1 will be constructed.

Take $s = z + 2p^2$. Looking at $\tilde{B}(t, t+s)$, the rows $z+1$ through to $z + 2p^2$ will have to contain $M_1A_{su}^{k_1}$. Expanding $\tilde{B}(t, t+s)u$,

$$\begin{bmatrix} \tilde{B}(t, t+k_1-1) \\ \tilde{B}(t+k_1, t+s)\tilde{A}^{k_1} \end{bmatrix} u = \begin{bmatrix} 0 & B_1 & X \\ 0 & 0 & M_1A_{su}^{k1} \\ 0 & 0 & Y \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_{23} \end{bmatrix}$$
$$= \begin{bmatrix} B_1u_1 + Xu_{23} \\ M_1A_{su}^{k_1}u_{23} \\ Yu_{23} \end{bmatrix} =: \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}.$$

Again, $d_2 \neq 0$ since $u_{23} > 0$ (as we are considering only states with unstable modes). Therefore,

$$\beta \leq \min_{\substack{\{u | u_1 = 0, \|u\| = 1\} \\ j = 1, \ldots, s \\ M}} \left\| M A_{su}^{j} u_{23} \right\|,$$

where the minimum is taken over all possible full rank M. The minimum exists since M has only a finite number of choices as the number of possible measurements is finite. □

Implementation Details:

- The matrix M is useful to prove the completeness of the algorithm, but in implementation, the QR factorization of M can be used instead of M.
- Matrix Q gives the null space basis of M. Given a row v, if the projection of v on the null space of M is non zero, then it will increase the rank of M. This can be used to check for valid rows in line 9.
- We can update the QR factorization after appending a row in M in line 14 by using Givens rotations [2] instead of factorizing M at each step.

With these implementation details, we can characterize the runtime of the DetectableGreedy algorithm. We make the assumption that $n \leq Tk$, which is justified since we are interested in infinite, or very large time horizons T. Recall that the simple greedy algorithm, chooses the row r of C at each iteration that minimizes $F\big((\sigma, \sigma_t \cup \{r\})\big)$. That is, all rows are valid, and no rank calculations are required.

Proposition 11 (Complexity of algorithm) *If $n \leq Tk$, then the (asymptotic) runtime of the* detectableGreedy *algorithm is equal to that of the simple greedy algorithm.*

In the following, we denote the runtime for computing the multiplication of two $n \times n$ matrices as μ. In the naive implementation of matrix multiplication, $\mu = O(n^3)$. State-of-the-art algorithms have a complexity of $\mu = O(n^{2.3728639})$ [22].

PROOF. The detectableGreedy algorithm is a greedy algorithm with two extra steps: Calculate A_{su}, C_{su}, and check which of the measurements increase rank of M.

Given a time horizon T, there will be mT iterations. For each of the T time steps, k measurements need to be selected from a set of m. For any of the metrics $F(\sigma)$, the runtime to greedily pick the k best measurements is $O(\mu mk)$, since we must perform mk measurement updates (6), each of which requires μ time. For the metric $F(\sigma) = -\log\det(\Sigma_{t|t}^{-1})$, the greedy selection can be cleverly implemented in a runtime of $O(n^2 mk)$ per time step [17]. In addition to this, a time update (6) requires two matrix multiplications, which require μ time. Therefore, the greedy part has a runtime of $O(Tmn^2k + Tk\mu)$ for the log det metric and $O(Tmk\mu)$ otherwise.

Table 1
Algorithm performance (ratio relative to sliding window) and runtime for different problem sizes.

n	G		DG		SW	
	$F(\sigma)$	Time	$F(\sigma)$	Time	Time	Win.
4	1.29	<1	1.16	<1	136	7
16	1.51	2	1.37	2	990	4
36	1.68	5	1.58	5	1785	3
64	1.53	16	1.46	17	571	2
100	1.55	48	1.49	49	3328	2

To calculate A_{su} and C_{su}, the observability matrix of the system has to be evaluated at the start of the algorithm. It requires A^x to be calculated for $x = \{1, \ldots, n-1\}$. So calculation of A_{su} requires $O(n\mu)$ time.

For determining valid rows to choose, a matrix multiplication has to be performed for each row of C ($O(mn^2)$). Finding the projection of a row on the null space of M requires $O(n^2)$ time and updating the QR factorization of M also requires $O(n^2)$ time [2]. That is because in the implementation, M will only have rank increasing rows and hence its maximum size is $n \times n$. The power of A_{su} needs to be calculated once per time step and hence requires $O(T\mu)$ time.

Therefore, the total complexity becomes $O(Tk(mn^2 + \mu) + n\mu)$ for the log det metric and $O(Tmk\mu + n\mu)$ for other metrics. If $n \leq Tk$, our modification in the greedy algorithm does not increase the complexity of the algorithm. □

In the following section we compare the performance of Algorithm 1 with some other known scheduling algorithms.

6 Simulations

We perform several simulation experiments to investigate the performance of Algorithm 1. In these simulations, we run the algorithms for T time steps. We let the covariance metric F be the average trace of the *a posteriori* error covariance, so at each time step t,

$$F(\sigma) = \frac{1}{t} \sum_{i=1}^{t} \operatorname{trace}(\Sigma_{i|i}).$$

For this section, we refer to the greedy algorithm as G and the detectableGreedy algorithm as DG. We use a sliding window approximation for comparison with these algorithms. The slidingWindow (SW) algorithm is essentially an extended greedy algorithm such that the optimal is calculated over a certain time window by considering every possible sequence. The final error covariance of a window serves as the initial covariance for the next window. Two drawbacks of the sliding window approach are that although the optimal is achieved over the window, the algorithm's runtime grows exponentially with window size, and there is no guarantee on performance over the entire time horizon.

Linear system generation: We performed a discretization of the heat equation in 2D to form the system matrix A for our simulations. A similar formulation of the discrete time system is used for the scheduling algorithm proposed in [13]. The matrix C was taken to be identity to realize sensors that measure temperatures on the 2D surface in the form of a grid. The W matrix was set to a random positive definite matrix in which each entry is uniformly distributed in $[0, 5]$. The matrix V was set to a diagonal matrix with each entry uniformly randomly distributed in $[0.5, 2]$.

One sensor per time step: In our first experiment, we compare three algorithms: G, DG and SW. The performance of the three algorithms for the case of one sensor per time step is summarized in Table 1. In the first column, n records the number of states in the system. For each system, each algorithm was run 10 times for different W and V, and the average run time and average covariance metric value at the end of time horizon T was recorded. In the table, the times are rounded off to the nearest second and the function values are normalized with respect to the values for SW algorithm. The time horizon was taken to be $T = 500$ for these experiments. For the SW algorithm, the window size is stated under the "Win." column, and is decreased for larger systems in order to maintain tractable runtimes.

In Table 1 we see that the SW algorithm outperforms the Greedy algorithm by between 30% and 60%, and the detectableGreedy by between 15% and 50%. The runtimes, on the other hand, are approximately 70 to 100 times faster for the greedy algorithms than for slidingWindow. The run times of DG and G algorithms are essentially equal, agreeing with Proposition 11. However, the detectableGreedy algorithm gives consistently lower values of $F(\sigma)$ than the greedy algorithm. Also note that the detectableGreedy algorithm is scalable to large systems — the last row of Table 1 represents a system with 100 states and 100 sensors, and the computation time is still less than 0.1 seconds per time step.

An important feature of the detectableGreedy algorithm is that it guarantees uniformly bounded error covariance. However, it is not guaranteed that the DG algorithm will always outperform the greedy algorithm.

Table 2
Performance (number of wins) of Greedy versus DG algorithm for 500 randomly generated W and V.

$n = m$	DG	G	decrease in average $F(\sigma)$
2	388	75	10 %
5	484	16	20 %
10	500	0	27 %

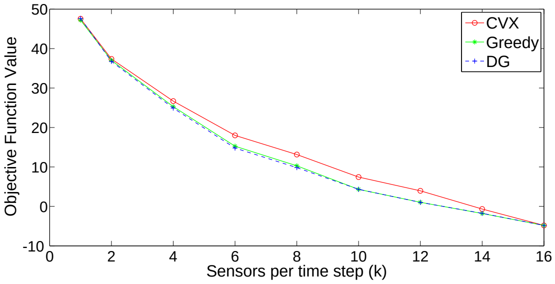

Fig. 2. Plot of log determinant metric versus k. The value of objective function is averaged over 5 runs of the algorithms.

For example, if $A = C = I$ for a 2 state system and the state with more system noise is being measured by the noisy sensor, then the greedy algorithm performs better. But for all the other combinations of system and sensor noise covariances, either DG outperforms greedy or they give the same result. For 500 randomly generated W and V, the number of times one of the algorithms performed better than the other is given in Table 2 for different system dimensions. The last column shows the decrease in the average value of objective function for DG as compared to greedy algorithm. For randomly generated C, a similar trend is observed. In the case of randomly generated A, each row of C_{su} serves as a rank increasing row and hence the two algorithms give the same performance.

Multiple sensors per time step: In [10] the metric $-\log\det(\Sigma_{t|t}^{-1})$ is optimized over a single time step using a convex relaxation to choose the best k sensors for one time step. The process is repeated at each time step to get a sensor schedule. We will refer to this algorithm as CVX. For multiple sensors per time step, we compare our algorithm to CVX and G. The system used was the discretization of heat equation for $n = m = 16$ with randomly chosen W and V and the algorithms were run for $T = 50$ time steps. Figure 2 shows a plot of the objective function $\frac{-1}{T}\sum_{i=1}^{T}\log\det(\Sigma_{i|i}^{-1})$ for each algorithm plotted against number of sensors chosen per time step. In this case, the greedy and detectableGreedy algorithms achieve similar performance, and both outperform the CVX algorithm. These results agree with [17] where the greedy algorithm is shown to be empirically better than the CVX.

7 Conclusions and Future Directions

We gave conditions for the existence of a bounded sensor schedule and then presented an algorithm that outputs a bounded sensor schedule if one exists. The algorithm attains the same asymptotic runtime as the greedy algorithm, but we show empirically that it obtains better performance.

We are interested in quantifying the quality of the schedule given by the algorithm relative to the optimal. Another problem to consider is determining the minimum number of sensors required per time step that achieve the same bound as selecting all m sensors.

References

[1] BDO Anderson and John B Moore. Detectability and stabilizability of time-varying discrete-time linear systems. *SIAM Journal on Control and Optimization*, 19(1):20–32, 1981.

[2] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[3] Vijay Gupta, Timothy Chung, Babak Hassibi, and Richard M Murray. Sensor scheduling algorithms requiring limited computation [vehicle sonar range-finder example]. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 3, pages 825–828, 2004.

[4] Vijay Gupta, Timothy H. Chung, Babak Hassibi, and Richard M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, 2006.

[5] G.E. Hovland and B.J. McCarragher. Dynamic sensor selection for robotic systems. In *IEEE Int. Conf. Robotics and Automation*, pages 272–277, 1997.

[6] Utku Ilkturk. *Observability Methods in Sensor Scheduling*. PhD thesis, ARIZONA STATE UNIVERSITY, 2015.

[7] Volkan Isler and Ruzena Bajcsy. The sensor selection problem for bounded uncertainty sensing models. In *International Symposium on Information Processing in Sensor Networks*, page 20, 2005.

[8] Syed Talha Jawaid and Stephen L Smith. A complete algorithm for the infinite horizon sensor scheduling problem. In *American Control Conference*, pages 437–442. IEEE, 2014.

[9] Syed Talha Jawaid and Stephen L. Smith. Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. *Automatica*, 61:282 – 288, 2015.

[10] Siddharth Joshi and Stephen Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.

[11] Ioannis K Kookos and John D Perkins. A systematic method for optimum sensor selection in inferential control systems. *Industrial & engineering chemistry research*, 38(11):4299–4308, 1999.

[12] Entao Liu, Edwin K. P. Chong, and Louis L. Scharf. Greedy Adaptive Compression in Signal-Plus-Noise Models. *CoRR*, abs/1202.3913v5, 2012.

[13] Sijia Liu, Makan Fardad, Pramod K. Varshney, and Engin Masazade. Optimal periodic sensor scheduling in networks of dynamical systems. *IEEE Transactions on Signal Processing*, 62(12):3055–3068, 2014.

[14] Suhinthan Maheswararajah, Saman K Halgamuge, and Malin Premaratne. Sensor scheduling for target tracking by suboptimal algorithms. *IEEE Trans. on Vehicular Technology*, 58(3):1467–1479, 2009.

[15] Dipankar Maity and John S Baras. Dynamic, optimal sensor scheduling and value of information. In *International Conference on Information Fusion*, pages 239–244, 2015.

[16] Yilin Mo, Roberto Ambrosino, and Bruno Sinopoli. Sensor selection strategies for state estimation in energy constrained wireless sensor networks. *Automatica*, 47(7):1330–1338, 2011.

[17] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. Greedy sensor selection: Leveraging submodularity. In *IEEE Conf. on Decision and Control*, pages 2572 –2577, 2010.

[18] Dawei Shi and Tongwen Chen. Approximate optimal periodic scheduling of multiple sensors with constraints. *Automatica*, 49:993 – 1000, 2013.

[19] Dawei Shi and Tongwen Chen. Optimal periodic scheduling of sensor networks: A branch and bound approach. *Systems & Control Letters*, 62:732 – 738, 2013.

[20] Michael P. Vitus, Wei Zhang, Alessandro Abate, Jianghai Hu, and Claire J. Tomlin. On efficient sensor scheduling for linear dynamical systems. *Automatica*, 48(10):2482–2493, 2012.

[21] James E. Weimer, Bruno Sinopoli, and Bruce H. Krogh. A relaxation approach to dynamic sensor selection in large-scale wireless networks. In *International Conf. on Distributed Computing Systems*, pages 501–506, 2008.

[22] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, pages 887–898. ACM, 2012.

[23] Wei Zhang, Michael P. Vitus, Jianghai Hu, A. Abate, and C.J. Tomlin. On the optimal solutions of the infinite-horizon linear sensor scheduling problem. In *IEEE Conf. on Decision and Control*, pages 396–401, 2010.