Automatica 163 (2024) 111550

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Real-time sensor selection for time-varying networks with guaranteed performance☆

Reza Vafaee, Milad Siami *

Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115, USA

ARTICLE INFO

Article history:
Received 26 July 2022
Received in revised form 18 October 2023
Accepted 3 January 2024
Available online 14 February 2024

Keywords:
Network estimation and control
Time-varying dynamics
Sparsification
On-the-fly sensor and actuator selections
Randomization

ABSTRACT

This study addresses the challenge of selecting sensors for linear time-varying (LTV) systems dynamically. We present a framework that designs an online sparse sensor schedule with performance guarantees using randomized algorithms for large-scale LTV systems. Our approach calculates each sensor's contribution at each time in real-time and immediately decides whether to keep or discard the sensor in the schedule, with no possibility of reversal. Additionally, we provide new performance guarantees that approximate the fully-sensed LTV system with a multiplicative approximation factor and an additive one by using a constant average number of active sensors at each time. We demonstrate the validity of our findings through several numerical examples.

1. Introduction

Advancements in high-performance computing processors, high-capacity storage, and efficient algorithms have fueled increasing interest among researchers and scientists in controlling and estimating complex interconnected systems (Chakrabortty & Ilić, 2011; Fitch & Leonard, 2016; Liu & Barabási, 2016; Ruths & Ruths, 2014; Siami & Motee, 2018a, 2018b). Such systems are ubiquitous, with applications ranging from smart grids (Chakrabortty & Ilić, 2011) and social networks (Latora, Nicosia, & Russo, 2017) to statistical physics (Liu & Barabási, 2016; Ruths & Ruths, 2014), multi-robot systems (Fitch & Leonard, 2016; Tian, Khosoussi, & How, 2021), and computational biology (Rajapakse, Groudine, & Mesbahi, 2012). However, in many cases, it is not feasible to obtain individual measurements from all sensors due to either their high cost or computational limitations. To address this challenge, researchers have developed sparse sensor selection techniques that can effectively estimate the system state using a subset of available sensors. In this paper, we address the problem of sparse sensor scheduling for time-varying dynamics and propose a novel online approach that can provide accurate estimates of the overall system state while minimizing the cost of acquiring and processing sensor measurements.

Selecting an optimal set of sensors is crucial for accurately estimating the overall state of a complex system while managing uncertainties. However, identifying this optimal set remains a challenging and mostly unsolved problem. In the simplest scenario, finding the optimal set requires a combinatorial approach, which has been shown to be computationally intractable and NP-hard for all but the simplest cases (Baraniuk, 2007). Therefore, developing efficient algorithms for sparse sensor selection has become a critical research area with numerous practical applications, as demonstrated by recent works such as Tzoumas, Carlone, Pappas, and Jadbabaie (2020), Ye, Roy and Sundaram (2020) and Ye, Woodford, Roy, Sundaram and Shreyas (2020).

Sparse sensor selection involves finding the optimal set of sensors that optimize the performance measures based on observability (Georges, 1995; Müller & Weber, 1972). Several approaches have been proposed to solve this problem, including submodular optimization (Summers, Cortesi, & Lygeros, 2015), nonlinear integer programming (Athans, 1972; Morari & Stephanopoulos, 1980; Müller & Weber, 1972), and convex relaxation (Vafaee & Siami, 2022a). These methods typically rely on Gramian matrices to quantify the observability of the system. Recent advancements in this area have led to the development of systemic metrics that offer a more comprehensive and robust approach to selecting an optimal set of sensors for linear dynamical systems (Siami & Motee, 2018a). These metrics are characterized by their monotonicity, convexity, and homogeneity with respect to the Gramian matrix of the system and include commonly used measures such as the determinant or trace of inverse operators.

The problem of designing a time-varying sparse actuator scheduling for linear dynamical systems has been addressed

☆ The material in this paper was partially presented at the 61st IEEE Conference on Decision and Control, Dec. 6–9, 2022, Cancún, Mexico. This paper was recommended for publication in revised form by Associate Editor Solmaz Kia under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: vafaee.r@northeastern.edu (R. Vafaee), m.siami@northeastern.edu (M. Siami).

https://doi.org/10.1016/j.automatica.2024.111550

in the literature using both deterministic and randomized approaches (Siami, Olshevsky, & Jadbabaie, 2020). In a subsequent study, the authors of Siami and Jadbabaie (2020) investigated the design of joint time-varying sparse sensor and actuator scheduling by leveraging Hankel singular values of the linear system. The performance of the resulting sparse systems was compared to that of fully-actuated and fully-sensed systems. In another recent work, Vafaee and Siami (2022a) used a swapping regret minimization algorithm to round the continuous solution of a relaxed optimization to obtain a $(1+\epsilon)$ approximation of the actual optimal system for all systemic metrics. Other techniques for selecting sensors and actuators to optimize the observability and controllability of the system include balanced model reduction and greedy matrix QR pivoting (Manohar, Kutz, & Brunton, 2021).

The problem of finding the minimal set of sensors or actuators for maintaining observability or controllability of a system is another common challenge in the field (Olshevsky, 2014). This problem, known as the Minimal Control Set problem, has been shown to be NP-hard and cannot be efficiently solved or even approximated in polynomial time (Tzoumas, 2018). Other related challenges include optimal leader selection and control of formation in multi-agent systems (Dong & Huang, 2014; Fitch & Leonard, 2015).

Selecting the appropriate set of sensors or actuators in a system can be challenging due to the limited accuracy of the mathematical model. However, recent studies have sought to address this issue by developing algorithms that enable sensor/actuator selection even when the system model is not known in advance. Such algorithms estimate the model during the design process. For example, Fotiadis and Vamvoudakis (2021) proposed an online actuator selection algorithm for unknown linear time-invariant (LTI) dynamics, and Ye, Chi, Liu, and Gupta (2022) investigated the simultaneous actuator selection and controller design problem for finite-horizon Linear Quadratic Regulation (LQR) when the system matrices are unknown.

Our Contributions: Finding a small representation of sensors becomes challenging when the model of the system changes over time or the t-step observability matrix is large and cannot be stored in memory. To address these challenges, we develop a simple randomized framework for online selection and scheduling of sensors. In our setup, the rows of the observability matrix are considered one-by-one, and we immediately decide to keep or discard each sensor without retracting our decisions. The proposed framework is a Markov chain, meaning the probability of choosing a sensor only depends on previous sensors in the stream. The method is both simple and intuitive, and it approximates fully-sensed LTV systems up to a multiplicative and additive factor in a certain observability sense, while sampling a constant number of active sensors on average.

This paper builds on the findings presented in Vafaee and Siami (2022b) by incorporating new results including Theorem 4, Lemmas 3, 4, 5, and 6. The paper clearly defines the main problem as Problem 1, and provides additional discussions and supplementary materials in Appendix A.1.1 and Remarks 2 through 7. Furthermore, the paper offers more detailed analysis of the proofs for Propositions 1 and 2, Theorem 1, and Lemmas 1 and 2. The simulation results in Section 5 and future work outlined in Section 6 are also presented.

To maintain a clear and focused narrative, certain elements of the theoretical arguments, including proofs and lemmas, have been relegated to the Appendix.

2. Preliminaries and definitions

2.1. Mathematical notations

Indices: Lowercase non-bold letters are used for scalars and indices (e.g. j). The discrete time index is denoted by k throughout this paper.
Sets: Sets of real numbers ($\mathbb{R}$), non-negative real numbers ($\mathbb{R}_+$), and positive real numbers ($\mathbb{R}_{++}$), as well as their integer counterparts ($\mathbb{Z}$, $\mathbb{Z}_+$, and $\mathbb{Z}_{++}$), are represented, respectively. The set of natural numbers $\{i \in \mathbb{Z}_{++} \,:\, i \le n\}$ is denoted by $[n]$.
Vectors: Lowercase bold letters are utilized to denote vectors (e.g., $\boldsymbol{b}$). For a vector $\boldsymbol{x} \in \mathbb{R}^n$, $\mathrm{diag}(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$ is the diagonal matrix with elements of $\boldsymbol{x}$ sitting orderly on its diagonal. The i-th basis vector is denoted by $\boldsymbol{e}_i \in \mathbb{R}^n$, i.e. $\boldsymbol{e}_i(j) = 0$ for $j \neq i$ and $\boldsymbol{e}_i(i) = 1$. Vector norms $\|\boldsymbol{x}\|_0$, $\|\boldsymbol{x}\|_1$, and $\|\boldsymbol{x}\|$ return the total number of non-zero elements, the sum of the absolute values of the elements, and the Euclidean norm of vector $\boldsymbol{x}$, respectively. Both x_i and $\boldsymbol{x}(i)$ are used to denote the i-th entry of vector $\boldsymbol{x}$.
Matrices: Uppercase letters (e.g. A and $\mathcal{A}$), stand for real-valued matrices. For square matrix $X \in \mathbb{R}^{n\times n}$, $\mathrm{diag}(X)$ outputs the diagonal elements of X. Furthermore, $\det X$ and $\mathrm{Trace}\, X$ refer to the determinant and the summation of on-diagonal elements of X, respectively. Let I and $\mathbf{0}$ denote the identity matrix and a matrix of all zeros, respectively, with dimensions specified by context. The transpose of matrix A is represented by $A^\top$, and the Moore–Penrose pseudoinverse of matrix A is denoted by $A^\dagger$, with $A^{-1/2} = (A^\dagger)^{1/2}$. Symbol $\|\cdot\|$ denotes the spectral norm for matrices. Both $A_{i,j}$ and $A(i,j)$ are used to denote the entries located in the i-th row and j-th column of matrix A.
Positive Semidefinite Ordering: For symmetric matrices $A, B \in \mathbb{R}^{n\times n}$, we write $A \preceq B$ to denote the condition that $\boldsymbol{x}^\top A\,\boldsymbol{x} \le \boldsymbol{x}^\top B\,\boldsymbol{x}$, for all $\boldsymbol{x} \in \mathbb{R}^n$. Notations $\succeq$, $\prec$, and $\succ$ can be defined analogously. We say a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive semi-definite if $A \succeq \mathbf{0}$. $\mathbb{S}^n_+$ ($\mathbb{S}^n_{++}$) is the positive semi-definite (positive definite) cone of n-by-n matrices.
Misc: Lowercase non-bold letters are used for function names (e.g., $\rho(\cdot)$). The symbol $\oplus$ denotes the operation of appending the rows of one matrix to another. Given a matrix $Z \in \mathbb{R}^{n\times m}$, the vectorized form of Z is represented as

$$\mathrm{vec}(Z) = [z_{1,1}, \ldots, z_{n,1}, z_{1,2}, \ldots, z_{1,m}, \ldots, z_{n,m}]^\top,$$

while vec^{-1} returns the inverse of this operation.

2.2. Linear systems, controllability and observability

We start with a canonical LTV, discrete-time dynamics as follows:

$$\boldsymbol{x}(k+1) = A(k)\,\boldsymbol{x}(k) \;+\; B(k)\,\boldsymbol{u}(k), \tag{1}$$

$$\boldsymbol{y}(k) = C(k)\,\boldsymbol{x}(k), \tag{2}$$

where $A(k) \in \mathbb{R}^{n\times n}$, $B(k) \in \mathbb{R}^{n\times m}$, $C(k) \in \mathbb{R}^{p\times n}$, and $k \in \mathbb{Z}_+$. The time-varying matrix $A(k)$ describes the underlying structure of the system and the interaction strength between the agents/states at time k. The input matrix $B(k)$ identifies the nodes controlled by an external controller at time k, and the output matrix $C(k)$ shows the relationship between the output vector $\boldsymbol{y}$ and the state vector at time step k. Given the initial condition $\boldsymbol{x}(0)$ of the state variables and sequences of inputs $\boldsymbol{u}(0), \ldots, \boldsymbol{u}(t-1)$, according to (1), we have

$$\begin{aligned}\boldsymbol{x}(t) &= \Phi(t,0)\,\boldsymbol{x}(0) \;+\; \sum_{r=0}^{t-1} \Phi(t, r+1)\,B(r)\,\boldsymbol{u}(r)\\ &= \Phi(t,0)\,\boldsymbol{x}(0) \;+\; \mathcal{R}(t,0)\,\widetilde{\boldsymbol{u}}(t,0),\end{aligned} \tag{3}$$

where $\Phi(t,r)$ is the *state transition matrix*, which relates the state of the undriven system at time t to the state at an earlier time r, i.e., $\boldsymbol{x}(t)=\Phi(t,r)\boldsymbol{x}(r)$ for all $t\geq r$ [1]. The state transition matrix is given by

$$\Phi(t,r)=\begin{cases}A(t-1)A(t-2)\cdots A(r) & , \quad t>r\geq 0\\ I & , \quad t=r.\end{cases}$$

Matrix $\mathcal{R}(t,0)=\begin{bmatrix}\Phi(t,1)B(0) & \Phi(t,2)B(1) & \cdots & B(t-1)\end{bmatrix}$ is the *t-step controllability matrix* of the time-varying dynamics (1)–(2), and $\widetilde{\boldsymbol{u}}(t,0)=\begin{bmatrix}\boldsymbol{u}^\top(0) & \cdots & \boldsymbol{u}^\top(t-1)\end{bmatrix}^\top$. To evaluate the controllability of the system, we are interested in determining whether there are any solutions for $\widetilde{\boldsymbol{u}}(t,0)$ within the context of (3).

Moreover, according to (2) for $k=0,1,\ldots,t-1$, we have

$$\widetilde{\boldsymbol{y}}(t,0)=\mathcal{O}(t,0)\boldsymbol{x}(0)\ +\ \mathcal{T}(t,0)\widetilde{\boldsymbol{u}}(t,0), \tag{4}$$

where $\widetilde{\boldsymbol{y}}(t,0)=\begin{bmatrix}\boldsymbol{y}^\top(0) & \cdots & \boldsymbol{y}^\top(t-1)\end{bmatrix}^\top$ is the vector of measurement,

$$\mathcal{O}(t,0)=\begin{bmatrix}\boldsymbol{c}_1^\top(0)\Phi(0,0)\\ \vdots\\ \boldsymbol{c}_p^\top(0)\Phi(0,0)\\ \boldsymbol{c}_1^\top(1)\Phi(1,0)\\ \vdots\\ \boldsymbol{c}_p^\top(1)\Phi(1,0)\\ \vdots\\ \boldsymbol{c}_1^\top(t-1)\Phi(t-1,0)\\ \vdots\\ \boldsymbol{c}_p^\top(t-1)\Phi(t-1,0)\end{bmatrix}\begin{matrix}\left.\begin{matrix}\\ \\ \\ \end{matrix}\right\}C(0)\Phi(0,0)\\ \left.\begin{matrix}\\ \\ \\ \end{matrix}\right\}C(1)\Phi(1,0)\\ \\ \left.\begin{matrix}\\ \\ \\ \end{matrix}\right\}C(t-1)\Phi(t-1,0)\end{matrix}, \tag{5}$$

is the t-step observability matrix, $\boldsymbol{c}_j^\top(k)$'s are the rows of matrix $C(k)\in\mathbb{R}^{p\times n}$, and $\mathcal{T}(t,0)$ maps inputs to outputs and is known, constructed using input, output, and state transition matrices. The second term in (4) is a known quantity and can be subtracted from the vector of measurements to obtain

$$\underline{\boldsymbol{y}}(t,0)=\mathcal{O}(t,0)\boldsymbol{x}(0). \tag{6}$$

We refer to the system (1)–(2) as observable if, over some finite time horizon t, the knowledge of $\widetilde{\boldsymbol{u}}(t,0)$ and $\widetilde{\boldsymbol{y}}(t,0)$ is sufficient to uniquely determine $\boldsymbol{x}(0)$ from (6).

Assumption 1. In this paper, we assume that integer number $t>0$ is the time horizon to control or estimate, also referred to as the time-to-control or time-to-estimate.

From a numerical standpoint it might be better to characterize controllability and observability in terms of the Gramian matrices at time t, respectively defined as follows for the time-varying system (1)–(2): $\mathcal{W}(t,0)=\mathcal{R}(t,0)\mathcal{R}^\top(t,0)$, and

$$\mathcal{X}(t,0)=\mathcal{O}^\top(t,0)\mathcal{O}(t,0). \tag{7}$$

Dynamics in (1)–(2) can also be expressed as follows:

$$\boldsymbol{x}(k+1)=A(k)\boldsymbol{x}(k)\ +\ \sum_{i\in[m]}\boldsymbol{b}_i(k)\boldsymbol{u}_i(k), \tag{8}$$

$$\boldsymbol{y}(k)=\sum_{j\in[p]}\boldsymbol{e}_j\boldsymbol{c}_j^\top(k)\boldsymbol{x}(k), \tag{9}$$

where $\boldsymbol{b}_i(k)$'s are the columns of the time-varying matrix $B(k)\in\mathbb{R}^{n\times m}$ and $\boldsymbol{c}_j^\top(k)$'s are the rows of matrix $C(k)\in\mathbb{R}^{p\times n}$.

For brevity, we avoid discussing the controllability and observability of dynamics (8)–(9) in terms of the defined controllability and observability matrices and their Gramian counterparts. Instead, we directly state the second assumption as follows, including the necessary information from the missing discussion:

Assumption 2. In this paper, we make the assumption that the system described by (8)–(9) is an n-state minimal realization. This implies that the system is controllable (with a controllability matrix of full rank and a positive definite controllability Gramian) and observable (with an observability matrix of full rank and a positive definite observability Gramian).

2.3. Systemic controllability/observability metrics

Similar to the concept of *systemic* introduced in the literature of Siami and Motee (2018b), Siami et al. (2020) and Vafaee and Siami (2022a), we introduce various controllability/observability metrics. These metrics are real-valued functions that quantify various aspects of the energy required in the system when referred to as a controllability metric, or the degree of uncertainties in estimation when designated as an observability metric. They are defined on the set of linear dynamical systems derived from (8)–(9). The metrics rely on the Gramian matrix's controllability/observability, which is a positive definite matrix. Consequently, a systemic performance measure can be formulated as a function that operates on the set of Gramian matrices for all controllable/observable systems with n agents, which we represent by $\mathbb{S}_{++}^n$.

Definition 1 (*Systemic Performance Measure*)**.** A Gramian-based metric $\rho:\mathbb{S}_{++}^n\to\mathbb{R}_+$ is systemic if and only if for all $M,N\in\mathbb{S}_{++}^n$, it satisfies:

- *(Positive) homogeneity* criteria: $\rho(\gamma M)=\gamma^{-1}\rho(M)$, for any $\gamma>0$[2];
- *Monotonicity* criteria: If $N\preceq M$, then $\rho(N)\geq\rho(M)$;
- *Convexity* criteria: $\rho\big(\alpha M+(1-\alpha)N\big)\leq\alpha\cdot\rho(M)+(1-\alpha)\cdot\rho(N)$, for all $\alpha\in[0,1]$.

Several in-depth studies have been conducted in the works of Siami and Motee (2018b) and Siami et al. (2020) regarding this type of performance metrics. It has been demonstrated that the set of criteria outlined in Definition 1 applies to many common measures. To provide an overview of these studies, we discuss some of the well-known measures in Appendix A.1.1. However, for brevity, we do not repeat all of them here and instead, suggest interested readers refer to Siami and Motee (2018a, Table I) and Siami and Motee (2018b, Table I) for a comprehensive list of systemic performance metrics.

While the discussions apply to both sensor and actuator scheduling, our paper exclusively tackles the sensor scheduling problem. We include tools for analogous arguments in actuator scheduling if required.

[1] The undriven system is system (1) when $\boldsymbol{u}(k)=0$ for all $k\in\mathbb{Z}_+$.

[2] A function ρ is considered (positively) homogeneous of degree α if for all $\gamma>0$, $\rho(\gamma M)=\gamma^{-\alpha}\cdot\rho(M)$. In this paper, when we refer to a metric as homogeneous, we mean it is homogeneous of degree 1.

3. Sensor scheduling

The sensor placement problem is to find the optimal sensor locations in an environment to minimize uncertainties and costs. In the sensor scheduling problem, the objective is to determine when and for how long sensors should be active. Energy and cost considerations limit the usage of all sensors at all times. The online sensor scheduling refers to determining sensor activity in real-time based on the current state of the system. Our paper focuses on an online sensor scheduling problem for LTV dynamics.

Online Sensor Scheduling: In online sensor scheduling, the selection of sensors is made dynamically over time, as opposed to being predetermined in advance. At each time step, the sensor set is evaluated based on the current state of the system, and a decision is made on which sensors to keep or discard.

The key characteristic of online sensor scheduling is its causality, meaning the decisions made at each time step are based only on the state knowledge up to the current time, without knowing the future. This makes the problem challenging because the scheduling decisions must account for the uncertainty and changing dynamics of the system without complete future information.

The objective of online sensor scheduling is to minimize the use of sensors while maintaining the observability of the system, as closely as possible to the fully-sensed dynamics.

Our use of the term "causality" is specific to our scheduling problem. This concept differs from the general definition of causality, which involves outputs depending on past and present inputs but not future inputs. In our case, causality means decisions at each time step are based solely on the current knowledge of system matrices without anticipating.

To begin, we will define sensor scheduling, and then provide a clear explanation of online sensor scheduling. Despite the difference in meaning, the terms *selection* and *sampling* might be used interchangeably in the following sections.

3.1. Sensor scheduling problem

The goal of the sparse sensor scheduling problem is to design a schedule for sensor outputs that ensures that the observability performance metrics of the original (fully-sensed) and the sparse systems are similar in an appropriately defined sense, while keeping the number of active sensors much less than a fully-sensed system in the resulting schedule. Specifically, given a canonical discrete-time, LTV dynamics (8)–(9), with p sensors, observability systemic metric $\rho(\cdot)$ that is aligned with the properties addressed in Definition 1, and the t-step observability Gramian matrix $\mathcal{X}(t,0)$, the goal is to find a sensor schedule such that the resulting system with the observability Gramian matrix $\widetilde{\mathcal{X}}(t,0)$ is well-approximated; that is

$$\left|\log\frac{\rho(\widetilde{\mathcal{X}}(t,0))}{\rho(\mathcal{X}(t,0))}\right| \leq \epsilon', \tag{10}$$

where $\epsilon' > 0$ is the approximation factor.

3.2. Weighted sensor scheduling

A weighted sensor schedule can be obtained by scaling the output signal by a non-negative factor while keeping the scales bounded. The scaling introduces an extra degree of freedom that allows us to obtain a sparser set of outputs. With reference to (9), we define a weighted sensor schedule by $S = [s_{j,k+1}]$ with $s_{j,k+1} \geq 0$, where $j \in [p]$ and $k+1 \in [t]$. The resulting output equation with this schedule is

$$\boldsymbol{y}(k) = \sum_{j\in[p]} s_{j,k+1}\cdot \boldsymbol{e}_j \boldsymbol{c}_j^\top(k)\boldsymbol{x}(k), \quad k\in\mathbb{Z}_+, \tag{11}$$

where $s_{j,k+1} \geq 0$ shows the strength of the j-th sensor output at time k. The t-step observability Gramian matrix (7) for the sparse system (11) can be obtained as

$$\widetilde{\mathcal{X}}(t,0) = (\Lambda_s\cdot\mathcal{O}(t,0))^\top \underbrace{(\Lambda_s\cdot\mathcal{O}(t,0))}_{:=\,\widetilde{\mathcal{O}}(t,0)} = \mathcal{O}^\top(t,0)\,\Lambda_s^2\,\mathcal{O}(t,0), \tag{12}$$

where the *sparsification matrix* $\Lambda_s := \mathrm{diag}(\mathrm{vec}(S))$ and $\widetilde{\mathcal{O}}(t,0)$ is the t-step sparse observability matrix.

Our objective is to reduce the average number of active sensors by d, where

$$d := \frac{1}{t}\cdot\|\mathrm{vec}(S)\|_0, \tag{13}$$

with the aim of maintaining close observability Gramian between the fully-sensed and sparse systems. This approximation necessitates horizon lengths that may exceed the state's dimension. The definition below formalizes this approximation.

Definition 2 (*(ϵ,δ,d)-approximation*)**.** Given a time horizon $t \geq n$, system (11) with a sparse weighted sensor schedule S is (ϵ,δ,d)-approximation of system (9), if and only if

$$(1-\epsilon)\mathcal{X}(t,0) - \delta I \preceq \widetilde{\mathcal{X}}(t,0) \preceq (1+\epsilon)\mathcal{X}(t,0) + \delta I, \tag{14}$$

where $\mathcal{X}(t,0)$ and $\widetilde{\mathcal{X}}(t,0)$ are the observability Gramian matrices for the fully-sensed and sparse system defined in (7) and (12), respectively. Parameter d as defined in (13) is the average number of active sensors, and finally $\epsilon \in (0,1)$ and $\delta > 0$ are the approximation factor and the additive approximation factor, respectively. Succinctly, $\widetilde{\mathcal{X}}(t,0) \approx_{\epsilon,\delta} \mathcal{X}(t,0)$ denotes the same condition.

In Siami et al. (2020), a closely related approximation notation was introduced for time-invariant networks referred to as the (ϵ,d)-approximation. This is given by

$$(1-\epsilon)\mathcal{X}(t,0) \preceq \widetilde{\mathcal{X}}(t,0) \preceq (1+\epsilon)\mathcal{X}(t,0), \tag{15}$$

and abbreviated as $\widetilde{\mathcal{X}}(t,0) \approx_\epsilon \mathcal{X}(t,0)$.

Remark 1. When ϵ is small enough,[3] we can elaborate (15) to show that the (ϵ,d)-approximation system is in fact a well-approximated system. Identically, if $\widetilde{\mathcal{X}}(t,0) \approx_\epsilon \mathcal{X}(t,0)$, then

$$\left|\log\frac{\rho(\widetilde{\mathcal{X}}(t,0))}{\rho(\mathcal{X}(t,0))}\right| \leq \epsilon, \tag{16}$$

where $\rho(\cdot)$ is some observability systemic measure. To obtain (16), we utilize the facts that $e^{-\beta}$ is almost $1-\beta$ when β is appropriately small, and $1+\beta \leq e^\beta$ for all $\beta \in \mathbb{R}$. Similarly, we can show if $\widetilde{\mathcal{X}}(t,0) \approx_{\epsilon,\delta} \mathcal{X}(t,0)$, then

$$\left|\log\frac{\rho(\widetilde{\mathcal{X}}(t,0)+\lambda I)}{\rho(\mathcal{X}(t,0)+\lambda I)}\right| \leq \epsilon, \tag{17}$$

where $\lambda := \delta/\epsilon$.

[3] This condition almost holds in this paper since we will assume $\epsilon \in (0,1)$.

Finally, since there is a one-to-one correspondence between a sensor at a specific time step and the rows of the t-step observability matrix, row sparsification (sampling) and sensor scheduling address the same process in the subsequent sections. Next, we formulate the online sensor scheduling problem.

3.3. Online sensor scheduling

For the LTV dynamics (8)–(9), consider a *causal* regime where, at each time step t, only the system matrices ($A(k)$ and $C(k)$) for $k = 0, \ldots, t$ are known. The causality assumption is realistic because the system's matrices in real scenarios often undergo changes over time, and access to future matrices may not be guaranteed. The reasons for the changing matrices can vary, including environmental changes, physical wear and tear, external forces, sensor and battery failure, and software updates.

To obtain the observability matrix $\mathcal{O}(t, 0)$, a block matrix is appended to the observability matrix at the previous time step $\mathcal{O}(t-1, 0)$, i.e., $\mathcal{O}(t, 0) = \left[\mathcal{O}^\top(t-1, 0) \,|\, (C(t-1)\Phi(t-1, 0))^\top\right]^\top$. If we are given the sparse matrix $\widetilde{\mathcal{O}}(t-1, 0)$, two methods can be used to obtain the sparse matrix $\widetilde{\mathcal{O}}(t, 0)$. The first method is to reapply the sparsification process used to obtain $\widetilde{\mathcal{O}}(t-1, 0)$ to the new matrix $\mathcal{O}(t, 0)$. However, this can be computationally intensive and impractical for many applications as we progress in time.

The second method, known as online sensor scheduling in this paper, relies on the previous sensor schedule, or the previous sparse matrix $\widetilde{\mathcal{O}}(t-1, 0)$, and only sparsifies the newly appended block matrix. This approach involves making a single decision at each time step t to determine which subset of the newly added p rows should be selected. Equivalently, it requires deciding which subset of the p available sensors at time t should be activated for data collection.

In this paper, this objective, however, is achieved by sequentially evaluating each row of the appended block matrix and individually deciding whether to retain or discard the corresponding sensor. We choose to process the newly appended block matrix sequentially because each row in the t-step observability matrix contributes a simple rank-one matrix to the Gramian matrix of the system, simplifying the analysis considerably. The utility of rank-one matrices for sparsification purposes has been the focus of several recent works (Siami & Jadbabaie, 2020; Siami & Motee, 2018a, 2018b; Siami et al., 2020). For instance in Siami and Motee (2018a), each feedback link's contribution is shown to be rank-one, a property that is used when determining the optimal subset of links to be added to a first-order consensus network under a cardinality constraint.

Another advantage of sequential processing is its flexibility, as it does not require a fixed number of sensors (p) at each time step. This adaptability makes our approach suitable for real-world scenarios where sensor sets and their numbers can change dynamically. In fact, our main goal is to intelligently activate the most informative sensor set at each time step, regardless of the available sensor count. However, for simplicity in notation, we present results for a time-varying system with a fixed p sensors at each time step.

In the online sensor scheduling approach, the goal is to select fewer sensors than are available at each time step. Starting from time step zero and advance in time, the algorithm systematically evaluates the rows of the observability matrix, one-by-one. In particular, it begins with the first row from the initial set of p available rows in the block observability matrix at time zero and continues through to the p-th row of the observability-appended block at time t.

Problem 1 (*Online Sensor Scheduling*)**.** Consider the t-step observability matrix $\mathcal{O}(t, 0)$, as defined in (5), for the discrete-time, linear time-varying dynamics described in (8)–(9). Consider the rows of the matrix one-by-one, and the goal is to make an individual decision for each row to either keep or discard its corresponding sensor (i.e., assigning a positive weight), which cannot be changed afterward. Let $\mathcal{O}_i(t, 0)$ denote the portion of the observability matrix that includes its first i rows. The objective is to find a sensor schedule S such that, for a given approximation factor $\epsilon \in (0, 1)$ as well as an additive approximation factor $\delta > 0$, and for any $i \leq tp$, the approximation

$$\widetilde{\mathcal{O}}_i^\top(t, 0)\widetilde{\mathcal{O}}_i(t, 0) \approx_{\epsilon,\delta} \mathcal{O}_i^\top(t, 0)\mathcal{O}_i(t, 0), \tag{18}$$

is satisfied, where $\widetilde{\mathcal{O}}_i(t, 0) = \text{diag}(\text{vec}(S)(0 : i)) \cdot \mathcal{O}_i(t, 0)$ and $\text{vec}(S)(0 : i)$ is the first i entries of the vector $\text{vec}(S)$. Additionally, the overall number of active sensors should not exceed td (i.e., $\|\text{vec}(S)(0 : tp)\|_0 \leq td$), where d is the desired average number of active sensors as defined in (13).

It is evident that the solution to Problem 1 guarantees an approximation of the system at every time step $i \leq t$ due to (18).

Remark 2. In this setting, at each time step, after the system matrices become available, the sparsification unit determines the set of sensors to be selected (by processing the rows of the newly appended block observability matrix one-by-one). Then, only the selected sensors are activated to collect measurements. This provides advantages such as reduced power usage, decreased bandwidth for communication, and enhanced data privacy.

Remark 3. We distinguish our problem from the issue of dynamically optimizing a schedule, as discussed in Badanidiyuru, Mirzasoleiman, Karbasi, and Krause (2014). While the configuration might appear similar in its sequential data processing approach, our problem seeks a sparse sensor schedule that approximates the observability of the system with a full complement of sensors, rather than trying to get close to the optimal set.

In the remainder of this paper, we will frequently refer to the rows of the observability matrix. To simplify this, we use $\boldsymbol{o}_i^\top$ to denote the i-th row $\boldsymbol{c}_j^\top(k)\Phi(k, 0)$, where $i = kp + j$, $j \in [p]$, and $k + 1 \in [t]$. In a time-invariant system, $\Phi(k, 0)$ is equivalent to A^k, and the variable k representing the time instant in the output matrices will be omitted.

4. Online sensor scheduling result

Randomized algorithms have seen great success in solving subset selection and related problems (Cohen et al., 2015; Cohen, Musco, & Pachocki, 2020; Siami et al., 2020; Vafaee & Siami, 2022b). A sampling scheme for randomly selecting sensors is typically formulated as follows.

Sampling Scheme: For any set of sampling probabilities p_1, $p_2, \ldots, p_{tp}$ include the i-th row, $\boldsymbol{o}_i^\top$, in the sparse observability matrix $\widetilde{\mathcal{O}}(t, 0)$ with probability p_i and re-weight the row by $1/\sqrt{p_i}$, then

$$\mathbb{E}[\underbrace{\widetilde{\mathcal{O}}^\top(t, 0)\widetilde{\mathcal{O}}(t, 0)}_{\widetilde{\mathcal{X}}(t,0)}] = \sum_{i=1}^{tp} p_i \cdot \left(\frac{1}{p_i}\boldsymbol{o}_i\boldsymbol{o}_i^\top\right) = \underbrace{\mathcal{O}^\top(t, 0)\,\mathcal{O}(t, 0)}_{\mathcal{X}(t,0)}. \tag{19}$$

To achieve proper concentration in the sparse observability Gramian matrix, it is important to select unique rows with high

probabilities. As a result, the sensor sampling problem is reduced to determining the *uniqueness* of different rows in the observability matrix.

Our findings on online sensor scheduling, presented in the following section, are based on recent advancements in randomized linear algebra and online sampling (Cohen et al., 2020).

4.1. Leverage score as the uniqueness measure

In Theorem 2 of Siami et al. (2020), the *leverage scores* of the columns in the controllability matrix are used to determine their uniqueness and construct a sparse actuator schedule for LTI systems (see Appendix A.1.3 for comprehensive details and characteristics of the leverage score). By utilizing this definition of uniqueness, it is demonstrated that, on average, choosing a modest amount of actuators randomly at each time results in a sparse schedule with a controllability Gramian matrix $\widetilde{\mathcal{W}}(t,0)$ such that $\widetilde{\mathcal{W}}(t,0) \approx_\epsilon \mathcal{W}(t,0)$, where $\mathcal{W}(t,0)$ represents the controllability of the system in the absence of sparsification.

The dual algorithm of Siami et al. (2020) can be used to sample sensors with probability proportional to their leverage scores to obtain an (ϵ, d)-approximation of the fully-sensed system. However, computing the exact leverage scores is computationally expensive. The following theorem shows that using approximations of the leverage scores is sufficient to obtain the (ϵ, d)-approximation.

Note that $\boldsymbol{\tau}(\mathcal{O}(t,0))$ is the vector of diagonal elements of the projection matrix $\mathcal{O}(t,0)\mathcal{X}^\dagger(t,0)\mathcal{O}^\top(t,0)$, and τ_i refers to the i-th element of this vector.

Theorem 1 (*Overestimate for (ϵ, d)-approximation*)**.** *Given an approximation factor $\epsilon \in (0,1)$, time horizon $t \geq n$, and the dynamics (9), let $\boldsymbol{u} = [u_i]$ be a vector of overestimates of the leverage scores of the rows of the observability matrix, i.e., $u_i \geq \boldsymbol{\tau}(\mathcal{O}(t,0))(i)$ for all $i \in [tp]$. Let $c > 3$ be a fixed constant and let the diagonal elements of the sparsification matrix Λ_s be $\Lambda_s(i,i) = 1/\sqrt{p_i}$ with probability $p_i = \min\left(c \cdot u_i \cdot \log n/\epsilon^2, 1\right)$, and zero otherwise. Then, with probability at least $1 - n^{1-c/3}$, this sparsification matrix Λ_s results in a scheduling S that is an (ϵ, d)-approximation of (9). The average number of active sensors at each time d is at most $c \cdot \|\boldsymbol{u}\|_1 \cdot \log n/t\epsilon^2$.*

Theorem 1 shows that a rough overestimate of the leverage scores of the rows of the observability matrix is sufficient to achieve an (ϵ, d)-approximation of the fully-sensed system. However, these results cannot be easily adapted to more restrictive settings, such as the semi-streaming or online setting, as the entire observability matrix must be available beforehand to obtain these rough overestimates.

In restrictive settings, however, the straightforward approach is to use the existing partial data to estimate the leverage scores. Kelner et al. in Kelner and Levin (2013) exploit a similar concept to obtain a spectral approximation of a graph in a semi-streaming setting. The algorithm receives the rows of the vertex edge incidence matrix (edges) one-by-one and rejects each row based on its leverage score relative to the edges seen so far. As more rows are received, better estimates can be obtained for the leverage scores, so the algorithm adjusts accordingly. The algorithm adds the incoming rows of the vertex edge incidence matrix to a small set of previously sampled rows called the sparsifier set. When the sparsifier set becomes too large, it gets resparsified, considering not only the incoming rows but also the rows already in the sparsifier set. However, as pointed out by Cohen et al. in Cohen et al. (2020), the probability of sampling a row also depends on pruning steps and not just earlier rows in the stream. This dependence seems to break the argument made in Kelner and Levin (2013) that the distribution of their algorithm is the same as one round of sampling by leverage scores. Hence, a spectral approximation may not be guaranteed.

Similar to Cohen et al. (2020), we tackle the issue of dependency by adopting an online sampling approach. When a row (sensor's contribution) is encountered, a decision to either sample or not sample is made and never updated. As a result, our algorithm is a true Markov chain, with the sampling of $\boldsymbol{o}_i^\top$ only dependent on the choices made for $\boldsymbol{o}_j^\top$ with $j < i$, and not on the choices for $\boldsymbol{o}_j^\top$ with $j \geq i$.

4.2. Main result

In this section, we employ the *ridge leverage score* to obtain an (ϵ, δ, d)-approximation. Our approach allows for online sampling and eliminates the need for resparsification, as the sampling probability of a sensor only depends on the previously sampled sensors. The ridge leverage score has already been utilized for purposes such as approximate kernel ridge regression (El Alaoui & Mahoney, 2014), spectral approximation and online sampling (Cohen et al., 2020; Kapralov, Lee, Musco, Musco, & Sidford, 2017), and iterative regular leverage score computation (Li, Miller, & Peng, 2013). We broaden its application to online sensor scheduling.

In ridge leverage scores, the focus is on computing leverage scores for $Q^\top Q + \lambda I$ rather than just $Q^\top Q$, where $\lambda > 0$ is a small constant. These scores are also referred to as λ-ridge leverage scores in machine learning literature (Alaoui & Mahoney, 2015) and defined as

$$\tau_i^\lambda := \boldsymbol{q}_i^\top (Q^\top Q + \lambda I)^{-1} \boldsymbol{q}_i, \tag{20}$$

for the i-th row, $\boldsymbol{q}_i^\top$, of matrix $Q \in \mathbb{R}^{r \times n}$.

One can modify Theorem 1 to work with λ-ridge leverage scores to achieve an (ϵ, δ, d)-approximation of the dynamics (9). This is formally stated in the following theorem.

Theorem 2 (*Overestimate for (ϵ, δ, d)-approximation*)**.** *Consider the dynamics in (9) and an approximation factor $\epsilon \in (0,1)$, additive approximation factor $\delta > 0$, $\lambda := \delta/\epsilon$, and time horizon $t \geq n$. Let $\boldsymbol{\ell} = [\ell_i]$ be a vector of overestimates of the λ-ridge leverage scores of the rows of the observability matrix, i.e., $\ell_i \geq \boldsymbol{o}_i^\top (\mathcal{X}(t,0) + \lambda I)^{-1} \boldsymbol{o}_i$ for all $i \in [tp]$. Let $c > 3$ be a constant and the diagonal elements of the sparsification matrix Λ_s be $\Lambda_s(i,i) = 1/\sqrt{p_i}$ with probability $p_i = \min(c \cdot \ell_i \cdot \log n/\epsilon^2, 1)$, and zero otherwise. Then, this sparsification matrix Λ_s specifies scheduling (11) that is an (ϵ, δ, d)-approximation of (9) with probability at least $1 - n^{1-c/3}$, i.e., $\mathcal{O}^\top(t,0)\Lambda_s^2\mathcal{O}(t,0) \approx_{\epsilon,\delta} \mathcal{O}^\top(t,0)\mathcal{O}(t,0)$. Additionally, the average number of active sensors at each time d is at most $c \cdot \|\boldsymbol{\ell}\|_1 \cdot \log n/t\epsilon^2$.*

Proof. Theorem 1 states that if we sample the rows of the observability matrix with probabilities proportional to their overestimated leverage scores, we can obtain an (ϵ, d)-approximation with high probability. This means

$$(1-\epsilon)\mathcal{O}^\top(t,0)\mathcal{O}(t,0) \preceq \widetilde{\mathcal{O}}^\top(t,0)\widetilde{\mathcal{O}}(t,0) \preceq (1+\epsilon)\mathcal{O}^\top(t,0)\mathcal{O}(t,0).$$

We define $\mathcal{O}^\lambda(t,0) = \mathcal{O}(t,0) \oplus \sqrt{\lambda} \cdot I$, so

$$\mathcal{O}^{\lambda\top}(t,0)\mathcal{O}^\lambda(t,0) = \mathcal{O}^\top(t,0)\mathcal{O}(t,0) + \lambda I.$$

If we sample the rows of $\mathcal{O}^\lambda(t,0)$ with their overestimated leverage scores, we get

$$(1-\epsilon)(\mathcal{X}(t,0) + \lambda I) \preceq \widetilde{\mathcal{X}}(t,0) + \lambda I \preceq (1+\epsilon)(\mathcal{X}(t,0) + \lambda I). \tag{21}$$

It is worth noting that all the rows of $\sqrt{\lambda} \cdot I$ are sampled, as their leverage scores are one. Finally, subtracting λI from the sides and substituting $\lambda = \delta/\epsilon$ completes the proof of the theorem. □

Theorem 2 demonstrates that using λ-ridge leverage scores will not result in significantly different performance bounds. However, its implications cannot yet be applied to sensor sampling. This is due to two reasons: the method for computing the overestimates of the λ-ridge leverage scores remains unknown, and the sum of these overestimate scores, $\|\boldsymbol{\ell}\|_1$, is also unknown. We will address these issues one at a time and demonstrate how the proposed solutions provide a framework for designing an online sensor scheduling.

The current definition of λ-ridge leverage scores given in (20) cannot be used in an online setting to obtain even the exact values of the scores, as the entire observability matrix is not available beforehand. To align this definition with the online objective, we modify and redefine it in an online fashion as the following:

Definition 3 (*Online λ-ridge Leverage Score,* $\overline{\tau}_i^{\lambda}$**).** Let $\mathcal{O}_{i-1}(t,0)$ represent the fraction of the observability matrix including its first $i-1$ rows for $i \in [tp]$. The *online λ-ridge leverage score* is defined as

$$\overline{\tau}_i^{\lambda} = \min\left(\boldsymbol{o}_i^{\top}(\mathcal{X}_{i-1}(t,0)+\lambda I)^{-1}\boldsymbol{o}_i,\, 1\right), \tag{22}$$

where $\mathcal{X}_{i-1}(t,0) = \mathcal{O}_{i-1}^{\top}(t,0)\mathcal{O}_{i-1}(t,0)$ is the observability Gramian matrix of the first $i-1$ rows considered.

Lemma 1 demonstrates that the online λ-ridge leverage score is precisely what is required to determine the overestimates.

Lemma 1. *The online λ-ridge leverage scores* ($\overline{\tau}_i^{\lambda}$) *overestimate the regular λ-ridge leverage scores* (τ_i^{λ}) *for all* $i = 1, \ldots, tp$*, meaning:*

$$\overline{\tau}_i^{\lambda} \geq \tau_i^{\lambda}. \tag{23}$$

Lemma 1 gives us the overestimates, so we only need to approximate the sum of these scores to apply Theorem 2 in the online setting. Lemma 2 provides this approximation and bounds the sum.

Lemma 2. *Let* $\boldsymbol{\ell} = [\ell_i]$ *be a vector of λ-ridge leverage score overestimates obtained by computing the online λ-ridge leverage scores* $\overline{\tau}_i^{\lambda}$ *for the rows of the observability matrix. The sum of these overestimates can be bounded as* $\|\boldsymbol{\ell}\|_1 = \sum_{i\in[tp]} \ell_i \leq 2n \cdot \log(1 + \|\mathcal{O}(t,0)\|^2/\lambda)$.

Finally, in the last theorem of this paper, we demonstrate how to design a sampling strategy that utilizes online λ-ridge leverage scores to construct a sensor schedule, S, on-the-fly. This results in rigorous guarantees on the quality of the approximation achieved for the dynamics described by (9).

Theorem 3 (*Online Sampling*)**.** *Assuming the dynamics of (8)–(9), a time horizon* $t \geq n$*, an approximation factor* $\epsilon \in (0,1)$*, an additive approximation factor* $\delta > 0$*, and a fixed positive constant* c *are given, Algorithm 1 produces a scheduling of (11) that solves Problem 1 with a probability of at least* $1 - n^{-c/3}$ *for* $d \leq (2cn/(t\epsilon^2)) \cdot \log n \cdot \log(\epsilon \cdot \|\mathcal{O}(t,0)\|^2/\delta + 1)$*, where* d *is the average number of active sensors.*

Proof. This theorem combines the results in Theorem 2, Lemmas 1, and 2. □

Remark 4. To guarantee that Theorem 3 holds with high probability, the positive constant c should be chosen sufficiently large to make the term $n^{-c/3}$ negligible. If we have an estimate of the spectral norm of $\mathcal{O}(t,0)$, c can be adjusted such that Algorithm 1 solves Problem 1 for an adjustable average number of active sensors d.

Algorithm 1 `OnTheFly-Schedule`$(\{\boldsymbol{o}_i^{\top}\}_{i=1}^{tp}, \epsilon, \delta, c)$

Input: Rows $\{\boldsymbol{o}_i^{\top}\}_{i=1}^{tp}$ of the observability matrix (5), an approximation factor $\epsilon \in (0,1)$, additive approximation factor $\delta > 0$, and a positive constant c.

Output: Weighted sparse sensor schedule S such that $\|\text{vec}(S)\|_0 = O(n \log n \cdot \log(\epsilon \cdot \|\mathcal{O}(t,0)\|^2/\delta)/\epsilon^2)$.

Initialization: $\text{vec}(S) = \mathbf{0}$;
$\lambda = \delta/\epsilon$;
$\mathcal{X}_0(t,0) = \mathbf{0}$;
$\mathcal{O}_0(t,0) = [\,]$;
for $i = 1$ to tp **do**
  $\overline{\tau}_i^{\lambda} = \min(\boldsymbol{o}_i^{\top}(\mathcal{X}_{i-1}(t,0) + \lambda I)^{-1}\boldsymbol{o}_i, 1)$;
  $p_i = \min(c \cdot \overline{\tau}_i^{\lambda} \cdot \log n/\epsilon^2, 1)$;
  $\text{vec}(S)(i) = \begin{cases} 1/\sqrt{p_i} & \text{with probability } p_i, \\ 0 & \text{otherwise;} \end{cases}$
  $\mathcal{O}_i(t,0) = \begin{bmatrix} \mathcal{O}_{i-1}(t,0) \\ \boldsymbol{o}_i^{\top} \end{bmatrix}$;
  $\mathcal{X}_i(t,0) = \mathcal{O}_i^{\top}(t,0)\mathcal{O}_i(t,0)$;
end for
return S.

Remark 5. The upper bound on the average number of active sensors simplifies to $O(\log n \cdot \log(1+\kappa(\mathcal{X}(n,0))))$, assuming $t = n$ and $\lambda = \sigma_{\min}^2(\mathcal{O}(n,0))$ in Algorithm 1. Here, $\kappa(\cdot)$ is the condition number, $\sigma_{\min}(\cdot)$ finds the minimum singular value, and $\mathcal{X}(n,0)$ is the n-step observability Gramian matrix. To achieve a $O(\log n)$ bound, the condition number $\kappa(\mathcal{X}(n,0))$ must remain bounded as n increases, preventing certain states from becoming nearly unobservable as the system size grows. In practice, when $t \geq n$ and λ does not precisely equal $\sigma_{\min}^2(\mathcal{O}(t,0))$, we can introduce two positive constant factors to the sample complexity, but the conclusion remains the same. In summary, with a bounded condition number, only $O(\log n)$ sensors are required for a reliable approximation.

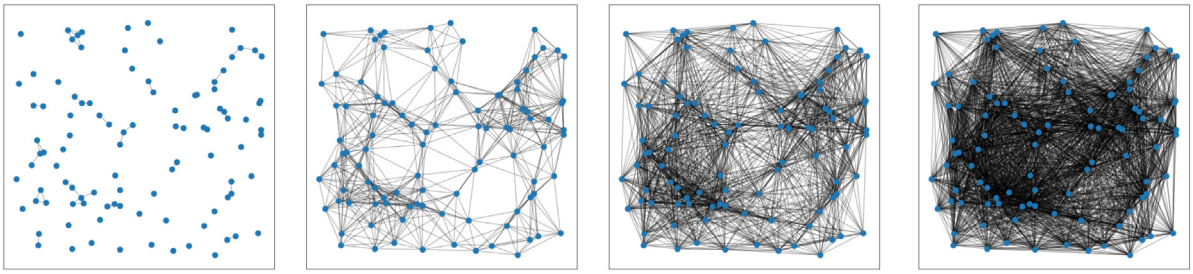
Remark 6. Theorem 3 shows a correlation between the average number of active sensors d and the time horizon to estimate t in terms of approximation factors ϵ and δ. To achieve the same approximation factors, a decrease in d results in an increase in t, and vice versa. Increasing d requires more active sensors, while increasing t requires longer time horizon.

Remark 7. As stated in Siami et al. (2020, Theorem 2), to achieve an (ϵ, d)-approximation of an LTI system, we need to sample $O(n \log n/\epsilon^2)$ rows of the observability matrix. In contrast, in the online setting, we demonstrate that we need to sample $O(n \log n \cdot \log(\epsilon \cdot \|\mathcal{O}(t,0)\|^2/\delta + 1)/\epsilon^2)$ rows to achieve an (ϵ, δ, d)-approximation of an LTV system. The factor $\log(\epsilon \cdot \|\mathcal{O}(t,0)\|^2/\delta + 1)$ represents the cost of online row sampling and is not an artifact of our analysis.

The following theorem confirms that the number of sampled rows suggested by our online result is nearly tight, up to a constant and a logarithmic factor

Theorem 4 (*Optimal Row Size*)**.** *Suppose* $\epsilon \cdot \|\mathcal{O}(t,0)\|^2 \geq c_1\delta$ *and* $\epsilon \in [c_2/\sqrt{n}, 1)$*, where* c_1 *and* c_2 *are fixed constants. Then, in order to achieve an* (ϵ, δ, d)*-approximation for the LTV dynamics (8)–(9) with probability at least 0.5, any online algorithm for sparse sensor scheduling must include at least* $\Omega\left(\frac{n \log(\epsilon \|\mathcal{O}(t,0)\|^2/\delta)}{\epsilon^2}\right)$ *sensors in expectation.*

We would like to emphasize that the lower bounds in Theorem 4 on $\epsilon\|\mathcal{O}(t,0)\|^2$ and ϵ are very minor. They simply ensure



(a) $k = 1 : r = 0.06$ (b) $k = 4 : r = 0.24$ (c) $k = 7 : r = 0.42$ (d) $k = 10 : r = 0.6$

Fig. 1. These plots depict snapshots of the evolving underlying graph for a dynamic network of 100 randomly distributed agents within a unit square space, connected via a proximity graph. Every agent is connected to all of its spatial neighbors within a closed ball of radius $r = 0.06$ (Plot (a)), $r = 0.24$ (Plot (b)), $r = 0.42$ (Plot (c)), and $r = 0.6$ (Plot (d)). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that $\log(\epsilon\|\mathcal{O}(t,0)\|^2/\delta) \geq 1$, and that ϵ is not so small that all sensors in space and time are essentially sampled.

Due to space limitations, we only discuss the sketch of the proof for Theorem 4. To obtain the result, we apply *Yao's minimax principle* and construct a distribution on inputs $\mathcal{O}(t,0)$ with $\|\mathcal{O}(t,0)\|^2 \leq \varphi$ for any large enough φ. This distribution ensures that any deterministic online sensor scheduling algorithm that succeeds with probability at least 0.5 must include $\Omega\left(n\log(\epsilon\cdot\varphi/\delta)/\epsilon^2\right)$ active sensors on average. Consequently, the best possible randomized algorithm that works with probability 0.5 on any input matrix with $\|\mathcal{O}(t,0)\|^2 \leq \varphi$ must contain at least $\Omega\left(n\log(\epsilon\cdot\varphi/\delta)/\epsilon^2\right)$ sensors in expectation on the worst-case input.

By using Algorithm 1 for online sensor scheduling, the pruning routines described in Kelner and Levin (2013) are avoided, thereby eliminating the dependency issues since the sampling probability of a sensor depends only on earlier sensors considered so far. However, we may miss the opportunity to have a lower number of sampled sensors, as seen in the streaming setup described in Kelner and Levin (2013). To ensure that the sampling probabilities are bounded in the online setting, we add λI to the process, but this modification adds an additive approximation factor, δ, to the performance guarantee compared to Siami et al. (2020).

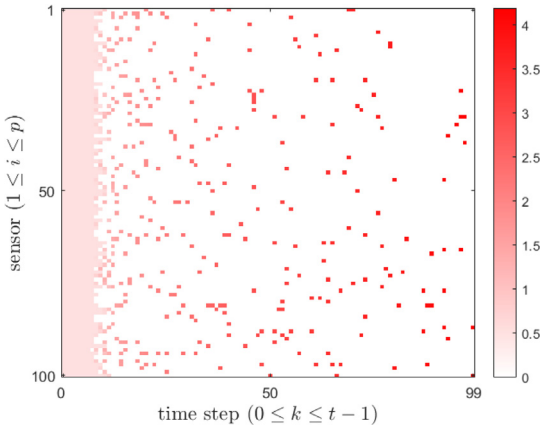


Fig. 2. This plot shows the weighted sparse sensor schedule based on Algorithm 1 for the time-varying dynamics (8)–(9) with the system matrices (24) where only ≈13% of the sensors on the average are active at each time between 0 to $n-1 = 99$. The network has $p = 100$ sensors, and the color of element (j,k) demonstrates the normalized weight $\bar{s}_{j,k+1}$ obtained by normalizing the resulting schedule $S = [s_{j,k+1}]$ of Algorithm 1 such that $\sum_k \sum_j \bar{s}_{j,k+1} = nd$, where $j \in [100]$, $k+1 \in [100]$, and d is the average number of nonzero elements in the schedule S. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Numerical examples

We demonstrate the results of the `OnTheFly-Schedule` algorithm through several numerical examples to showcase its efficiency.

Consider a dynamic networks including $n = 100$ agents/nodes, which are randomly distributed in a 1×1 square-shape area and are coupled over a proximity graph. Every agent will be connected to all of its spatial neighbors within a closed ball of radius $r = 0.06k$ where $k \in \mathbb{Z}_+$. The radius increases with time, causing the graph to become increasingly dense as we move forward in time. This phenomenon is illustrated in Fig. 1 which depicts the graphs of the network for the early time steps $k = 1, 4, 7$, and 10. It can be observed that the graph rapidly becomes connected and dense even in these early stages of k. The time-varying state matrix $A(k)$ and the output matrix $C(k)$ for this system are given by

$$A(k) = I - \vartheta L(k), \text{ and } C(k) := I, \tag{24}$$

where $L(k)$ represents the Laplacian matrix of the underlying graph during the k-th time step, and ϑ denotes the time resolution. To ensure the (marginal) stability of the networks, which are almost connected over time, we set the time resolution to $1/n$.

Let us consider the online sensor scheduling problem discussed in Section 3.3 for the network (24). The `OnTheFly-Schedule` algorithm is applied to sequentially pick both sensor outputs and activation times in an online setting. For the purpose of this experiment, the time-to-estimate t is set to n without loss of generality. This problem, in a static sense, appears similar to problems in undirected consensus networks, where a predetermined number of active agents are selected as leaders to minimize certain controllability measures (cf. Rahmani, Ji, Mesbahi, and Egerstedt (2009)). The `OnTheFly-Schedule` algorithm simultaneously uses both non-structural information (i.e., sensor outputs $\boldsymbol{c}_j^\top(k)$, where $j \in [p]$ and $k+1 \in [t]$) and structural information (i.e., $A(k)$) of the network to design a weighted sparse sensor schedule S. This design leads to an (ϵ, δ, d)-approximation of the fully-sensed dynamics (8)–(9). The sparse schedule produced by Algorithm 1 with $\epsilon = 0.5$, $\delta = 0.2$, and $c = 8$ is shown in Fig. 2.

Table 1 presents the comparative results for the A-optimality performance measure or Trace $\mathcal{X}^{-1}(t,0)$. The offline randomized algorithm, (Siami et al., 2020, Algorithm 6), is used to obtain the offline results. We assume that the complete set of system matrices is accessible during the offline computation to calculate the leverage scores, with an approximation factor of 0.5 and a constant of 0.94 set in the offline algorithm. One may expect slightly better performance and fewer sampled sensors for the offline algorithm because it provides an (ϵ, d)-approximation of the system and uses exact values for the leverage scores. To

Table 1
Observability performance measure results.

	Alg. 1 (Fig. 2)	Offline alg.	Uniform sampling	Fully sensed
Trace $\mathcal{X}^{-1}(t,0)$	78.47	75.80	1360.3	20.23
Average number of active sensors (d)	12.53	10.30	25	100

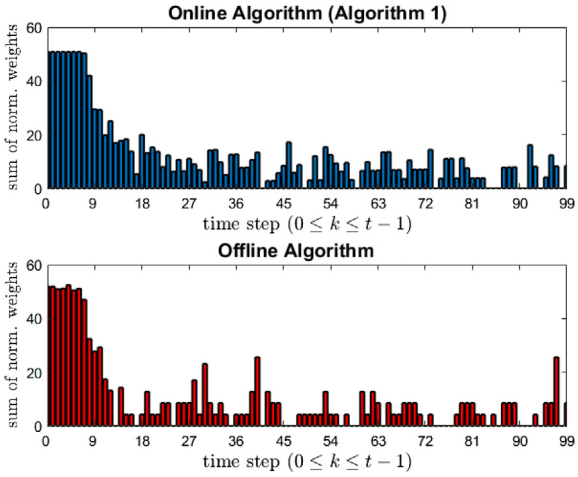


Fig. 3. This figure shows a comparison between Algorithm 1 (in blue) and the offline algorithm (in red) in terms of the sum of normalized weights of activated sensors over time. The weights used in this analysis are the same as those used in Table 1 which were normalized for the resulting average number of active sensors. Both algorithms exhibit a similar front-loaded behavior, but the offline algorithm shows higher variance due to repeated sampling of certain sensors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

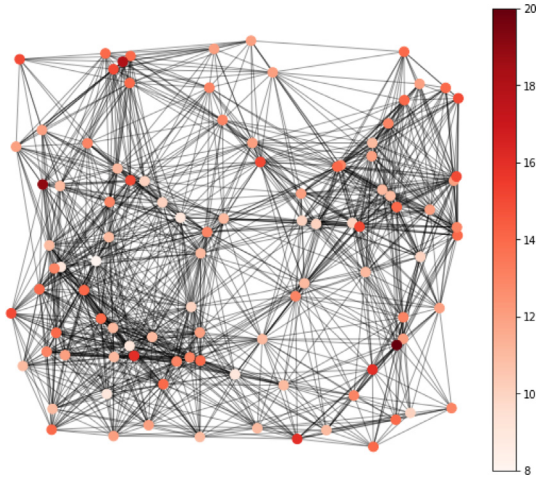


Fig. 4. For clarity and ease of visual interpretation, the underlying graph for the network in Fig. 2 is displayed at time $k = 6$ here, as the edge set of the graph at time $k = 99$ is too dense and obscures the nodes. Node colors indicate the sum of the normalized weights of Fig. 2 (the total number of active steps) for each node, i.e., $\sum_{k=0}^{99} \bar{s}_{j,k+1}$, from least (white) to greatest (red). Note that the weights are extracted by Algorithm 1 and normalized for the resulting d. According to the simulation, on average, only around 13% of the nodes/agents are sensed at each time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ensure a fair comparison, we normalize the resulting schedules of both our algorithm and the offline algorithm such that the sum of the non-zero elements in the resulting sparse schedule S equals nd, where d is the average number of non-zero elements as reported in the table. The performance of our proposed algorithm and the randomized offline algorithm is comparable to that of a fully-sensed system, with the offline algorithm expectedly performing slightly better. For uniform sampling, we randomly select d sensors from the total of p available sensors at each time step k between 0 and 99. To obtain comparable results, we perform the entire sampling process 50 times and report the one with the minimum objective value in Table 1.

Fig. 3 illustrates the sum of the normalized weights of the activated sensors at each time step for both our algorithm (Algorithm 1) in the top and the offline algorithm in the bottom. These weights are the same ones used to obtain the results in Table 1. We observe that the two algorithms behave almost similarly. Both demonstrate a "front-loaded" behavior, with more active sensors early in the time horizon, followed by sampling of only those that are significantly different later on. However, due to the nature of sampling in the offline algorithm that allows for replacement, one can observe that the algorithm is interested in repeating the sampling of particular sensors, leading to zero sums as well as several spikes in various places of the figure for the offline algorithm. These phenomena lead to a variance of 189.97 in the sums obtained by the offline algorithm, which is higher than the variance of 174.28 for our algorithm. The averages of the sums are listed in Table 1.

To gain some visual insights about which agents are sensed more and which less, we color the nodes of the underlying graph in Fig. 4 based on the total number of active steps during time steps 0 to 99 from least (white) to greatest (red). Remark that the active steps are generated based on Algorithm 1 and normalized for resulting d, the average number of nonzero entries in the output schedule (same as Fig. 2). Additionally, in order to maintain the clarity of the figure, we do not plot the underlying graph in the final step ($k = 99$), and instead show the edges generated for the network at an earlier step, $k = 6$.

To demonstrate the practicality of the bounds obtained in Theorem 3, we consider a time-invariant version of the network (24) where the underlying structure does not change over time. We examine three different scenarios for this time-invariant network. In the first scenario, we assume that the underlying graph only consists of unweighted self-loops for each node and set the time resolution to $(n-1)/n$. In the second scenario, we consider a connected underlying graph for the network and set $\vartheta = 1/n$. Finally, in the third scenario, we set the time resolution for the connected structure such that the dynamics become unstable. We use the same values for ϵ, δ, and c as we used previously.

In the first scenario, setting the estimation time horizon to n yields a spectral norm of the t-step observability matrix equal to $\frac{1-n^{-2n}}{1-n^{-2}}$, which remains close to one when n is greater than or equal to six. According to Theorem 3, this implies a theoretical upper bound of $34.8 \cdot \log n$ on the average number of active sensors. In the second scenario, the spectral norm is always equal to n when $t = n$, regardless of the underlying connected graph. Therefore, the theoretical upper bound is $64 \cdot \log(2.5 \cdot n + 1) \cdot \log n$. In the third scenario, the spectral norm is greater than n and is determined by the maximum eigenvalue of the Laplacian matrix of the graph and the time resolution. For simplicity, let us assume that the spectral norm is $\gamma \cdot n$ for $\gamma > 1$. Therefore, the theoretical upper bound is $64 \cdot \log(2.5 \cdot \gamma \cdot n + 1) \cdot \log n$.

Fig. 5 shows the three theoretical upper bounds and the number of available sensors as functions of the number of nodes n. These bounds are practical and effective when they are less than

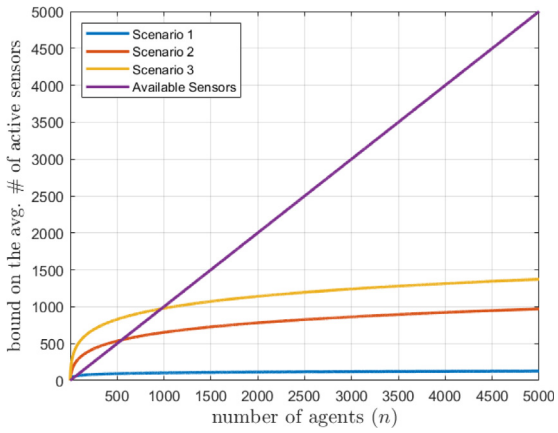


Fig. 5. Comparison of theoretical upper bounds on the average number of sampled sensors for three different network scenarios and the available sensor set, plotted as a function of the number of nodes n. Scenario 1 corresponds to unweighted self-loops with time resolution $(n-1)/n$, Scenario 2 corresponds to connected underlying graphs with time resolution $1/n$, and Scenario 3 corresponds to connected graphs with unstable dynamics and spectral norm $\gamma \cdot n$ with $\gamma = 50$. It can be observed that when the system is stable (Scenario 1), the bound becomes effective sooner in terms of the number of nodes than for marginally stable and unstable network scenarios (Scenarios 2 and 3). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the number of available sensors. The practicality of the bounds appears to be directly related to the stability of the system. Specifically, in Scenario 1 (stable), the bound is effective for a network size less than 100, while in the marginally and unstable scenarios (Scenarios 2 & 3), it is effective for network sizes greater than 600 and 1000, respectively.

6. Future steps

This work represents one of the first endeavors to design a sparse schedule for LTV dynamics in an online setting. To further advance this novel field of research and assist the community, we provide insights into some of the open questions related to our work as follows:

Batch Processing: Algorithm 1 processes the input stream of row vectors from the observability matrix one-by-one. However, the full set of sensors is available at each time step of the dynamic system. To improve performance, a possible extension of the `OnTheFly-Schedule` algorithm is to modify it for batch processing of rows/sensors. Specifically, at time step k, the sensor set $C(k)$ or the observability rows $\mathcal{O}_{kp+1:(k+1)p}(t,0) = C(k)\Phi(k,0)$ is accessible. To speed up computation, one could calculate the sampling probabilities for these rows simultaneously using a system solver for $(\mathcal{O}^{\top}_{kp+1:(k+1)p}(t,0)\mathcal{O}_{kp+1:(k+1)p}(t,0) + \lambda I)$.

Semi-streaming Algorithm: We observe that a less restrictive setting, semi-streaming, would potentially offer to sample fewer sensors and even provide better performance guarantees than our online algorithm. Furthermore, as the nature of the network dynamics confirms, the semi-streaming algorithm might be better suited to the structure of our on-the-fly sensor sparsification problem. Therefore, an interesting extension of our work would be developing an algorithm that is capable of designing a sensor schedule in a streaming setting. A streaming algorithm typically divides data into small chunks for efficient processing, while an online algorithm processes data one item at a time in real-time.

Memory Reduction: It is clear from Algorithm 1 that to calculate $\overline{\tau}_i^{\lambda}$, we must store all previous rows of the observability matrix that have been processed so far in memory. This might not be ideal for many applications that aim to reduce memory usage through sensor sampling. This raises two important questions: is it possible to use the contribution of the sample sensors stored so far as an approximation of $\mathcal{O}_{i-1}(t,0)$ while computing $\overline{\tau}_i^{\lambda}$ and still obtain the same or similar performance guarantees and row sample size? Can we go further and instead of storing all previous sampled rows in memory, only retain a smaller subset of rows from the observability matrix, while still achieving acceptable guarantees and row sample size?

Our ongoing research is devoted to answering some of these open questions.

7. Concluding remarks

In this paper, we present a new framework for randomly selecting a constant number of active sensors on-the-fly in order to approximate certain observability measures. The framework is inspired by recent advancements in online algorithms for machine learning and big data analysis. In the proposed algorithm, each row of the observability matrix of a given large-scale LTV system considers one-by-one, and we irrevocably decide whether to keep the corresponding sensor at each time in the sensor scheduling or not. The selected sensor is added to the schedule by assigning it a weight and does not discard or re-weight later. Our framework is simple and intuitive, and it represents new theoretical properties of the leverage score. Similar results can be developed for the actuator selection problem.

Acknowledgments

The first author of this paper would like to express gratitude to Navid Nouri from EPFL university for providing insightful discussions. This research was supported in part by grants NSF, United States 2208182, ONR, United States N00014-21-1-2431, NSF, United States 2121121, and DHS, United States 22STESE00001-02-00. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

Appendix

A.1. Definition of terms and further discussions

In this section, we have the objective of gathering and explaining the definitions of key concepts that are central to understanding the paper. Additionally, we will present new findings and results that add further depth to the ideas and arguments presented.

A.1.1. Why systemic performance measure

Here, we will discuss the reasons for using the systemic performance metrics introduced in Section 2.3 to characterize the performance of sparse sensor scheduling.

In real-world scenarios, it is probable that measurements will be corrupted by noise. To account for this, the updated output dynamics $\boldsymbol{y}(k) = C(k)\boldsymbol{x}(k) + \boldsymbol{\xi}(k)$, will be studied where $\boldsymbol{\xi}$ is the vector of *sensor noise* or *error*.

The system of Eqs. (6) can then be updated to $\underline{\boldsymbol{y}}(t,0) = \mathcal{O}(t,0)\boldsymbol{x}(0) + \underline{\boldsymbol{\xi}}(t,0)$ where $\underline{\boldsymbol{\xi}}(t,0) := \begin{bmatrix}\boldsymbol{\xi}^{\top}(0) & \boldsymbol{\xi}^{\top}(1) & \cdots & \boldsymbol{\xi}^{\top}(t-1)\end{bmatrix}^{\top}$.

Suppose we have independent and identically distributed random variables $\boldsymbol{\xi}(0), \ldots, \boldsymbol{\xi}(t-1)$ with a normal distribution $\mathcal{N}(\mathbf{0}, \sigma^2 I)$. In an estimation problem, the goal is to estimate the initial state vector $\boldsymbol{x}(0)$. If $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{tp}$ span $\mathbb{R}^n$, indicating

observability of the system's dynamics, the maximum-likelihood estimate of $\boldsymbol{x}(0)$, which is the same as the minimum variance estimate, is given by the least-squares solution

$$\hat{\boldsymbol{x}}(0) = \Big(\sum_{i=1}^{tp} \boldsymbol{o}_i \boldsymbol{o}_i^\top\Big)^{-1} \sum_{i=1}^{tp} \underline{\boldsymbol{y}}(i)\, \boldsymbol{o}_i,$$

where $\underline{\boldsymbol{y}}(i)$ is the i-th entry of the vector of measurement $\underline{\boldsymbol{y}}(t,0)$. The associated estimation error $\boldsymbol{e} = \boldsymbol{x}(0) - \hat{\boldsymbol{x}}(0)$ has zero mean and covariance matrix

$$\Sigma = \mathbb{E}\, \boldsymbol{e}\boldsymbol{e}^\top = \sigma^2 \Big(\sum_{i=1}^{tp} \boldsymbol{o}_i \boldsymbol{o}_i^\top\Big)^{-1} = \sigma^2 \mathcal{X}^{-1}(t,0), \tag{A.1}$$

where $\mathcal{X}(t,0)$ is the t-step observability Gramian matrix. The matrix Σ characterizes the precision of the estimation or the informative value of the measurements. As indicated by A.1.1, the covariance matrix is inversely proportional to the observability Gramian matrix. The objective of sparse sensor scheduling in this study is to select a limited number of vectors $\boldsymbol{o}_i$ from the available options, in order to approximate the error covariance matrix Σ as closely as possible to the scenario in which all vectors are utilized.

The β-confidence ellipsoid for $\boldsymbol{e}$ is the minimum volume ellipsoid that contains $\boldsymbol{e}$ with a probability of β. This ellipsoid represents the region in which $\boldsymbol{x}(0) - \hat{\boldsymbol{x}}(0)$ lies with β confidence and is given by

$$\mathcal{E}_\alpha = \left\{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{z}^\top \mathcal{X}(t,0)\, \boldsymbol{z} \le \alpha\right\}, \tag{A.2}$$

where $\alpha = F^{-1}_{\chi^2_n}(\beta)$, with $F_{\chi^2_n}$ being the cumulative distribution function of a χ-squared random variable with n degrees of freedom, and for simplicity, we assume that the variance σ is equal to one.

A metric that quantifies the accuracy of an estimation according to the covariance matrix is the size of the β-confidence ellipsoid

$$\text{Volume}(\mathcal{E}_\alpha) = \frac{(\alpha\pi)^{n/2}}{\Gamma(\frac{n}{2}+1)} \det \mathcal{X}^{-1/2}(t,0), \tag{A.3}$$

where $\Gamma(\cdot)$ is the Gamma function. We are generally interested in volume ratios, so it is more convenient to work with the logarithm of the volume

$$\log \text{Volume}(\mathcal{E}_\alpha) = \eta - \left(\frac{1}{2}\right) \log\det \mathcal{X}(t,0), \tag{A.4}$$

where constant η only depends on n and β. The logarithm of the volume of the confidence ellipsoid, as stated in (A.4), provides a quantitative measure of the information content of the set of measurements (i.e., the set of the observability matrix rows). This metric is referred to as D-optimality (Joshi & Boyd, 2008).

Another commonly used metric is the norm of the error covariance matrix, which is equal to the smallest eigenvalue of the t-step observability Gramian matrix $\mathcal{X}(t,0)$. The size of the confidence ellipsoid $\mathcal{E}_\alpha$ is proportional to $\|\Sigma\|^{1/2}$, thus controlling the value of $\|\Sigma\|$ can be understood as adjusting the size of the ellipsoid. This approach is referred to as E-optimality, as outlined in Boyd and Vandenberghe (2004).

In contrast, A-optimality focuses on controlling the trace of the error covariance matrix, Trace Σ. The trace represents the sum of the eigenvalues of the covariance matrix, which determine the lengths of the semi-axes of the confidence ellipsoid. As a result, an A-optimal design can be interpreted as controlling the volume of the ellipsoid. This objective is equivalent to the expected squared norm of the error, as expressed by the equation: $\mathbb{E}\,\|\boldsymbol{e}\|^2 = \mathbb{E}\,\text{Trace}(\boldsymbol{e}\boldsymbol{e}^\top) = \text{Trace}\,\Sigma$.

There are additional optimality criteria such as T-, V-, and G-optimality, which we will not discuss here. It can be seen that all the optimality criteria discussed so far are ways to quantify different aspects of the error covariance matrix and, by extension, the observability Gramian matrix. In the context of sensor selection, the objective is to choose a subset of rows from the observability matrix that results in a system that is as close as possible to the system with the complete set of rows, based on one of the discussed metrics for the confidence ellipsoid. The paper introduces two approximations, (ϵ, d)-approximation and (ϵ, δ, d)-approximation, to quantify the proximity of the sparse system to the fully sensed system based on the resulting covariance (observability Gramian) matrix.

It can be easily demonstrated that all the properties listed in the definition of the systemic performance measure, Definition 1, hold for these optimality measures. As a result, these optimality measures are commonly used examples of the larger class of systemic performance measures. We encourage interested readers to review the comprehensive works (Siami & Motee, 2018a, 2018b) for a complete list of widely recognized systemic performance measures.

A.1.2. (ϵ, δ, d) to (ϵ, d) approximation

The following result demonstrates how an (ϵ, δ, d)-approximation can be simplified to an (ϵ, d)-approximation.

Proposition 1. *Taking the additive approximation factor $\delta = \epsilon \cdot \sigma^2_{\min}(\mathcal{O}(t,0))$ transforms the (ϵ, δ, d)-approximation into an (ϵ, d)-approximation, with $\sigma_{\min}(\mathcal{O}(t,0))$ denoting the minimum singular value of the observability matrix $\mathcal{O}(t,0)$.*

Proof. By setting $\delta = \epsilon \cdot \sigma^2_{\min}(\mathcal{O}(t,0))$, we have the following:

$$\delta \cdot I \preceq \epsilon \cdot \mathcal{X}(t,0). \tag{A.5}$$

As a result, we obtain $(1+\epsilon)\mathcal{X}(t,0) + \delta I \preceq (1+2\epsilon)\mathcal{X}(t,0)$. Halving ϵ (which only affects the bounds by a constant factor) yields $(1+\epsilon/2)\mathcal{X}(t,0) + \delta I \preceq (1+\epsilon)\mathcal{X}(t,0)$. To complete the proof, we can also show that $(1-\epsilon)\mathcal{X}(t,0) \preceq (1-\epsilon/2)\mathcal{X}(t,0) - \delta I$. □

Proposition 1 requires some estimate of $\sigma_{\min}(\mathcal{O}(t,0))$ beforehand which is not available in an online setup.

A.1.3. Leverage score

Definition 4 (*Leverage Score, τ_i*)**.** The leverage score of the i-th row of matrix $Q \in \mathbb{R}^{r\times n}$ is the solution of the following optimization problem

$$\begin{aligned} \tau_i = \tau(\boldsymbol{q}_i^\top) = \underset{\boldsymbol{w}\in\mathbb{R}^r}{\text{minimize}} \quad & \|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & Q^\top \boldsymbol{w} = \boldsymbol{q}_i \end{aligned} \tag{A.6}$$

where $\boldsymbol{q}_i^\top$ is the i-th row of matrix Q. τ_i measures how important $\boldsymbol{q}_i$ is in composing range of $Q^\top$.

Optimization (A.6) is a least norm optimization, where the unique optimal solution can be obtained by introducing Lagrange multipliers. For a full-row rank matrix Q, the solution is in the form of $\hat{\boldsymbol{w}} = Q(Q^\top Q)^{-1}\boldsymbol{q}_i$, and hence $\tau_i = \boldsymbol{q}_i^\top (Q^\top Q)^{-1}\boldsymbol{q}_i$. If Q is not a full-rank matrix, then $\tau_i = \boldsymbol{q}_i^\top (Q^\top Q)^\dagger \boldsymbol{q}_i$.

Remark 8. The maximum value of τ_i is one, which can be achieved by selecting $\boldsymbol{w}$ as the i-th basis vector in $\mathbb{R}^r$. However, τ_i will be less than one if other rows have a similar alignment with $\boldsymbol{q}_i^\top$ or if $\|\boldsymbol{q}_i\|$ is small.

Proposition 2. *For all matrices $Q \in \mathbb{R}^{r\times n}$ and for every $i \in [r]$, the leverage score $\tau(\boldsymbol{q}_i^\top)$ is defined as the smallest α satisfying:*

$$\boldsymbol{q}_i \boldsymbol{q}_i^\top \preceq \alpha \cdot Q^\top Q. \tag{A.7}$$

Proof. This proposition can be established through two steps. Firstly, we demonstrate that the outer product of any row $\boldsymbol{q}_i^\top$ by itself is bounded above by a factor of α times $Q^\top Q$ in the semi-definite sense. Secondly, we prove that the minimum value of this factor is precisely the leverage score of the row. To prove the first step, we begin by utilizing the definition of Loewner ordering as follows:

$$\forall\, \boldsymbol{x} \in \mathbb{R}^n \;:\; \boldsymbol{x}^\top \boldsymbol{q}_i \boldsymbol{q}_i^\top \boldsymbol{x} \le \alpha \cdot \boldsymbol{x}^\top Q^\top Q \boldsymbol{x}. \tag{A.8}$$

We claim that, without loss of generality, we can assume $\boldsymbol{x}$ lies in the range space of $Q^\top Q$. This can be justified by noting that if $\tilde{\boldsymbol{x}}$ is the component of $\boldsymbol{x}$ that belongs to the null space of $Q^\top Q$, it can be disregarded as $Q^\top Q\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}^\top Q^\top Q\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}^\top \boldsymbol{q}_i^\top \boldsymbol{q}_i \tilde{\boldsymbol{x}} = 0$. If $\boldsymbol{x}$ lies in the range space, it can be represented as $\boldsymbol{x} = (Q^\top Q)^{-1/2}\boldsymbol{y}$ for some $\boldsymbol{y}$, where $(Q^\top Q)^{-1/2} = V\Sigma^{-1}V^\top$ if $U\Sigma V^\top$ is the SVD decomposition of the matrix Q. Therefore, the left-hand side of (A.8) can be rewritten as $\boldsymbol{y}^\top (Q^\top Q)^{-1/2}\boldsymbol{q}_i\boldsymbol{q}_i^\top (Q^\top Q)^{-1/2}\boldsymbol{y}$. The middle term $M = (Q^\top Q)^{-1/2}\boldsymbol{q}_i\boldsymbol{q}_i^\top (Q^\top Q)^{-1/2}$ is a rank-one matrix, so its Trace and its only eigenvalue are equal. Therefore,

$$\lambda_{\max}(M) = \text{Trace}\;\; \boldsymbol{q}_i^\top (Q^\top Q)^{-1}\boldsymbol{q}_i = \tau(\boldsymbol{q}_i^\top),$$

or equivalently we have $(Q^\top Q)^{-1/2}\boldsymbol{q}_i\boldsymbol{q}_i^\top (Q^\top Q)^{-1/2} \preceq \alpha \cdot I$, for some $\alpha \ge \lambda_{\max}(M)$. This implies for any $\boldsymbol{y} \in \mathbb{R}^n$,

$$\boldsymbol{y}^\top (Q^\top Q)^{-1/2}\boldsymbol{q}_i\boldsymbol{q}_i^\top (Q^\top Q)^{-1/2}\boldsymbol{y} \le \alpha \cdot \|\boldsymbol{y}\|^2,$$

which gives the proof for the first step. Since $\tau(\boldsymbol{q}_i^\top) = \lambda_{\max}(M)$, the second step will be proven automatically. □

A.1.4. Essential lemmas

This subsection provides a compilation of essential lemmas necessary for the paper.

Lemma 3. *For any symmetric matrix $Z \in \mathbb{R}^{n\times n}$ satisfying $Z \succeq I$, it follows that $Z^{-1} \preceq I$.*

Proof. Suppose the eigenvalues of $Z \in \mathbb{R}^{n\times n}$ are $\lambda_1, \dots, \lambda_n$. Then, the eigenvalues of $Z - I$ are $\lambda_1 - 1, \dots, \lambda_n - 1$ because if $\boldsymbol{v}_i$ is the eigenvector associated with λ_i, then $(Z-I)\boldsymbol{v}_i = Z\boldsymbol{v}_i - \boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i - \boldsymbol{v}_i = (\lambda_i - 1)\boldsymbol{v}_i$. The condition $Z \succeq I$ implies that $Z - I \succeq 0$, which means all eigenvalues of $Z-I$ are non-negative, i.e., $\lambda_i \ge 1$ for all $i \in [n]$. The eigenvalues of Z^{-1} are $\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n}$. Given that $\lambda_i \ge 1$, it follows that $\frac{1}{\lambda_i} \le 1$, which further implies that $Z^{-1} \preceq I$. □

To prove Lemma 2 (one of the main results of the paper), we will utilize two mathematical concepts: the relationship between the determinant of a matrix and its rank-one perturbation and an upper bound for $\det Q^\top Q$. These concepts are addressed in Lemmas 4 and 5, respectively.

Lemma 4 (*Lemma 1.1 of Ding and Zhou (2007)*)**.** *Suppose $P \in \mathbb{R}^{n\times n}$ is an invertible matrix and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ are column vectors, then*

$$\det(P + \boldsymbol{u}\boldsymbol{v}^\top) = (1 + \boldsymbol{v}^\top P^{-1}\boldsymbol{u})\det P. \tag{A.9}$$

Lemma 5. *Suppose $Q \in \mathbb{R}^{r\times n}$ such that $Q^\top Q \ge 0$, then $\det Q^\top Q \le (\|Q\|^2)^n$.*

A.2. Missing proofs

The following sections present supplementary proofs for the theoretical findings discussed in the main text of this paper.

A.2.1. Proof of Theorem 1

To prove this theorem, we need the following generalization of the *Chernoff* bound for matrices (which is a variant of Tropp (2012, Corollary 5.2)).

Lemma 6 (*Matrix Chernoff Bound*)**.** *Suppose a sequence of independent random matrices $M_i \in \mathbb{S}_+^n$ is given. Define $M = \sum_i M_i$ and $D = \mathbb{E}[M]$. If $M_i \preceq R \cdot D$, then*

$$\mathbb{P}\Big[M \preceq (1-\epsilon)D\Big] \le n \cdot e^{\frac{-\epsilon^2}{2R}}, \tag{A.10}$$

and

$$\mathbb{P}\Big[M \succeq (1+\epsilon)D\Big] \le n \cdot e^{\frac{-\epsilon^2}{3R}}. \tag{A.11}$$

Now, let us prove Theorem 1.

Proof. To utilize the outcome of Lemma 6, we assign matrix $M_i = \frac{1}{p_i}\boldsymbol{o}_i\boldsymbol{o}_i^\top$ to each row of the observability matrix with a probability of p_i, and zero otherwise. According to 4, $D = \mathbb{E}[\sum_{i=1}^{tp} M_i] = \mathcal{X}(t,0)$. Additionally, we need to calculate R in order to apply the Chernoff lemma. It is evident that when $p_i < 1$, R can be obtained easily through the properties of Loewner ordering:

$$M_i = \frac{\boldsymbol{o}_i\boldsymbol{o}_i^\top}{c \cdot u_i \cdot \log n/\epsilon^2} \preceq \frac{\boldsymbol{o}_i\boldsymbol{o}_i^\top}{c \cdot \tau_i \cdot \log n/\epsilon^2}, \quad \forall i \in [tp], \tag{A.12}$$

(this is correct since $\tau_i \le u_i$), and applying the results of Proposition 2, i.e., $\frac{\boldsymbol{o}_i\boldsymbol{o}_i^\top}{\tau_i} \preceq \mathcal{X}(t,0)$, so

$$M_i \preceq \underbrace{\frac{1}{c \cdot \log n/\epsilon^2}}_{R}\, \mathcal{X}(t,0). \tag{A.13}$$

If $p_i = 1$, $M_i = \boldsymbol{o}_i\boldsymbol{o}_i^\top$ with certainty, and therefore (A.13) does not apply in this case. However, choosing a random matrix M_i with a probability of one is equivalent to selecting and summing several new random matrices, each with a probability of one, or

$$M_i = \boldsymbol{o}_i\boldsymbol{o}_i^\top = \sum_{j=1}^{c\cdot\log n/\epsilon^2} \frac{\boldsymbol{o}_i\boldsymbol{o}_i^\top}{c\cdot\log n/\epsilon^2} = \sum_{j=1}^{c\cdot\log n/\epsilon^2} M_i^{(j)},$$

where $M_i^{(j)} \preceq \frac{\tau_i}{c\cdot\log n/\epsilon^2}\mathcal{X}(t,0) \preceq \frac{1}{c\cdot\log n/\epsilon^2}\mathcal{X}(t,0)$. In fact, viewing M_i as the sum of these new random matrices does not alter the expected value of M (i.e., D), so for the purpose of proof, one can assume that M_i has been replaced by the sum of these smaller random matrices. This assumption gives the concentration with $R = \frac{1}{c\cdot\log n/\epsilon^2}$, and thus

$$(1-\epsilon)\mathcal{X}(t,0) \preceq \sum_i M_i \preceq (1+\epsilon)\mathcal{X}(t,0), \tag{A.14}$$

with probability at least $1 - n\cdot e^{-\frac{\epsilon^2}{3R}} = 1 - n\cdot e^{-\frac{c\cdot\log n}{3}} = 1 - n^{1-c/3}$. Equivalently (A.14) can be written as

$$(1-\epsilon)\mathcal{X}(t,0) \preceq \underbrace{\mathcal{O}^\top(t,0)\,\Lambda_s^2\,\mathcal{O}(t,0)}_{\tilde{\mathcal{X}}(t,0)} \preceq (1+\epsilon)\mathcal{X}(t,0),$$

where Λ_s is a diagonal matrix whose diagonal entries

$$\Lambda_s(i,i) = \begin{cases} 1/\sqrt{p_i} & \text{with probability } p_i, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, using the standard Chernoff bound, it can be demonstrated that the sparsification matrix Λ_s has, at most, $\sum \min(c \cdot u_i \cdot \log n/\epsilon^2, 1) \le c \cdot \|\boldsymbol{u}\|_1 \cdot \log n/\epsilon^2$ non-zero elements with high probability. □

A.2.2. Proof of Lemma 1

We begin the proof by considering the fact that for any observable time-varying system (8)–(9) (see Assumption 2), and for a time horizon $t \geq n$, we have $0 \preceq \mathcal{X}_{i-1}(t,0) \preceq \mathcal{X}(t,0)$; therefore,

$$0 \prec \underbrace{\mathcal{X}_{i-1}(t,0)+\lambda I}_{Q} \preceq \underbrace{\mathcal{X}(t,0)+\lambda I}_{P},$$

where $\mathcal{X}_{i-1}(t,0)$ represents the observability Gramian matrix calculated using the first $i-1$ rows. Suppose that $Q = U\Lambda U^\top$ is the eigen decomposition of the symmetric matrix Q. Since $Q \succ 0$, then $\Lambda \succ 0$, and therefore $Q^{-1/2} = Q^{-\top/2} = U\Lambda^{-1/2}U^\top$ exists. By congruence, $P - Q \succeq 0$ implies that $Q^{-\top/2}(P-Q)Q^{-1/2} \succeq 0$, or equivalently $Q^{-\top/2}PQ^{-1/2} - I \succeq 0$. Taking $Z = Q^{-\top/2}PQ^{-1/2}$, one can use the result of Lemma 3 to get $(Q^{-\top/2}PQ^{-1/2})^{-1} - I \preceq 0$, or $Q^{1/2}P^{-1}Q^{\top/2} - I \preceq 0$. Again by congruence this implies

$$Q^{-1/2}(Q^{1/2}P^{-1}Q^{\top/2} - I)Q^{-\top/2} \preceq 0,$$

or equivalently $P^{-1} \preceq Q^{-1}$. Substituting the original expressions for Q and P, we get

$$(\mathcal{X}_{i-1}(t,0)+\lambda I)^{-1} \geq (\mathcal{X}(t,0)+\lambda I)^{-1}. \tag{A.15}$$

By definition, (A.15) means that for any real vector $\boldsymbol{a} \in \mathbb{R}^n$ the following inequality holds

$$\boldsymbol{a}^\top(\mathcal{X}_{i-1}(t,0)+\lambda I)^{-1}\boldsymbol{a} \geq \boldsymbol{a}^\top(\mathcal{X}(t,0)+\lambda I)^{-1}\boldsymbol{a}.$$

By setting $\boldsymbol{a} = \boldsymbol{o}_i$ for any $i \in [tp]$, the proof is completed.

A.2.3. Proof of Lemma 2

Suppose the process has just begun, and the first row (corresponding to a sensor) of the expanded observability matrix (5) has just been considered. Using (A.9), one can write

$$\det(\lambda I + \boldsymbol{c}_1(0)\boldsymbol{c}_1^\top(0)) = \det(\lambda I)\cdot(1+\underbrace{\boldsymbol{c}_1^\top(0)(\lambda I)^{-1}\boldsymbol{c}_1(0)}_{\overline{\tau}_1^\lambda}). \tag{A.16}$$

Using the fact that for any $x \in [0,1]$, $1+x \geq e^{\frac{x}{2}}$, we can lower bound the right-hand side of (A.16) by

$$\det(\lambda I)\cdot(1+\overline{\tau}_1^\lambda) \geq \det(\lambda I)\cdot e^{\overline{\tau}_1^\lambda/2}. \tag{A.17}$$

Now, suppose the second row, $\boldsymbol{c}_2^\top(0)$, is considered, then,

$$\begin{aligned}\det(\lambda I + \underbrace{\begin{bmatrix}\boldsymbol{c}_1^\top(0)\\ \boldsymbol{c}_2^\top(0)\end{bmatrix}^\top\begin{bmatrix}\boldsymbol{c}_1^\top(0)\\ \boldsymbol{c}_2^\top(0)\end{bmatrix}}_{\mathcal{O}_2^\top(t,0)\cdot\mathcal{O}_2(t,0)}) &= \det(\lambda I + \boldsymbol{c}_1(0)\boldsymbol{c}_1^\top(0) + \boldsymbol{c}_2(0)\boldsymbol{c}_2^\top(0))\\ &= \det(\lambda I + \boldsymbol{c}_1(0)\boldsymbol{c}_1^\top(0))\cdot(1+\overline{\tau}_2^\lambda)\\ &\geq \det(\lambda I)\cdot e^{\frac{1}{2}(\overline{\tau}_1^\lambda+\overline{\tau}_2^\lambda)}.\end{aligned}$$

Upon receipt of all tp rows, the observation shows that we have:

$$\begin{aligned}\det(\lambda I + \mathcal{O}^\top(t,0)\mathcal{O}(t,0)) &\geq \det(\lambda I)\cdot e^{\frac{1}{2}\sum_{i\in[tp]}\overline{\tau}_i^\lambda}\\ &= \lambda^n\cdot e^{\frac{1}{2}\sum_{i\in[tp]}\overline{\tau}_i^\lambda}.\end{aligned} \tag{A.18}$$

Eq. (A.18) shows that the lower bound of $\det(\lambda I+\mathcal{O}^\top(t,0)\mathcal{O}(t,0))$ is directly related to $\sum_{i\in[tp]}\overline{\tau}_i^\lambda$. Then, applying Lemma 5 will complete the proof as

$$\lambda^n\cdot e^{\frac{1}{2}\sum_{i\in[tp]}\overline{\tau}_i^\lambda} \leq \det(\lambda I + \mathcal{O}^\top(t,0)\mathcal{O}(t,0)) \leq (\|\mathcal{O}(t,0)\|^2+\lambda)^n$$

so

$$\lambda^n\cdot e^{\frac{1}{2}\sum_{i\in[tp]}\overline{\tau}_i^\lambda} \leq (\|\mathcal{O}(t,0)\|^2+\lambda)^n.$$

Taking the logarithm of both sides results in

$$n\log\lambda + \frac{1}{2}\sum_{i\in[tp]}\overline{\tau}_i^\lambda \leq n\log(\|\mathcal{O}(t,0)\|^2+\lambda),$$

or

$$\|\boldsymbol{\ell}\|_1 = \sum_{i\in[tp]}\overline{\tau}_i^\lambda \leq 2n\log(\|\mathcal{O}(t,0)\|^2/\lambda+1).$$

References

Alaoui, Ahmed, & Mahoney, Michael W. (2015). Fast randomized kernel ridge regression with statistical guarantees. *Advances in Neural Information Processing Systems*, *28*.

Athans, Michael (1972). On the determination of optimal costly measurement strategies for linear stochastic systems. *IFAC Proceedings Volumes*, *5*(1), 303–313.

Badanidiyuru, Ashwinkumar, Mirzasoleiman, Baharan, Karbasi, Amin, & Krause, Andreas (2014). Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 671–680).

Baraniuk, Richard G. (2007). Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, *24*(4), 118–121.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chakrabortty, Aranya, & Ilić, Marija D. (2011). *Vol. 3, Control and optimization methods for electric smart grids*. Springer.

Cohen, Michael B, Lee, Yin Tat, Musco, Cameron, Musco, Christopher, Peng, Richard, & Sidford, Aaron (2015). Uniform sampling for matrix approximation. In *Proceedings of the 2015 conference on innovations in theoretical computer science* (pp. 181–190).

Cohen, Michael B., Musco, Cameron, & Pachocki, Jakub (2020). Online row sampling. *Theory of Computing*, *16*(1), 1–25.

Ding, Jiu, & Zhou, Aihui (2007). Eigenvalues of rank-one updated matrices with some applications. *Applied Mathematics Letters*, *20*(12), 1223–1226.

Dong, Yi, & Huang, Jie (2014). Leader-following connectivity preservation rendezvous of multiple double integrator systems based on position measurement only. *IEEE Transactions on Automatic Control*, *59*(9), 2598–2603.

El Alaoui, Ahmed, & Mahoney, Michael W. (2014). Fast randomized kernel methods with statistical guarantees. *Stat*, *1050*, 2.

Fitch, Katherine, & Leonard, Naomi Ehrich (2015). Joint centrality distinguishes optimal leaders in noisy networks. *IEEE Transactions on Control of Network Systems*, *3*(4), 366–378.

Fitch, Katherine, & Leonard, Naomi Ehrich (2016). Optimal leader selection for controllability and robustness in multi-agent networks. In *2016 European control conference* (pp. 1550–1555). IEEE.

Fotiadis, Filippos, & Vamvoudakis, Kyriakos G. (2021). Learning-based actuator placement for uncertain systems. In *2021 60th IEEE conference on decision and control* (pp. 90–95). IEEE.

Georges, Didier (1995). The use of observability and controllability gramians or functions for optimal sensor and actuator location in finite-dimensional systems. *Vol. 4*, In *Proceedings of 1995 34th IEEE conference on decision and control* (pp. 3319–3324). IEEE.

Joshi, Siddharth, & Boyd, Stephen (2008). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, *57*(2), 451–462.

Kapralov, Michael, Lee, Yin Tat, Musco, CN, Musco, Christopher Paul, & Sidford, Aaron (2017). Single pass spectral sparsification in dynamic streams. *SIAM Journal on Computing*, *46*(1), 456–477.

Kelner, Jonathan A., & Levin, Alex (2013). Spectral sparsification in the semi-streaming setting. *Theory of Computing Systems*, *53*(2), 243–262.

Latora, Vito, Nicosia, Vincenzo, & Russo, Giovanni (2017). *Complex networks: principles, methods and applications*. Cambridge University Press.

Li, Mu, Miller, Gary L., & Peng, Richard (2013). Iterative row sampling. In *2013 IEEE 54th annual symposium on foundations of computer science* (pp. 127–136). IEEE.

Liu, Yang-Yu, & Barabási, Albert-László (2016). Control principles of complex systems. *Reviews of Modern Physics*, *88*(3), Article 035006.

Manohar, Krithika, Kutz, J. Nathan, & Brunton, Steven L. (2021). Optimal sensor and actuator selection using balanced model reduction. *IEEE Transactions on Automatic Control*, *67*(4), 2108–2115.

Morari, Manfred, & Stephanopoulos, George (1980). Minimizing unobservability in inferential control schemes. *International Journal of Control*, *31*(2), 367–377.

Müller, P. C., & Weber, H. I. (1972). Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems. *Automatica*, *8*(3), 237–246.

Olshevsky, Alex (2014). Minimal controllability problems. *IEEE Transactions on Control of Network Systems*, *1*(3), 249–258.

Rahmani, Amirreza, Ji, Meng, Mesbahi, Mehran, & Egerstedt, Magnus (2009). Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, *48*(1), 162–186.

Rajapakse, Indika, Groudine, Mark, & Mesbahi, Mehran (2012). What can systems theory of networks offer to biology? *PLoS Computational Biology*, *8*(6), Article e1002543.

Ruths, Justin, & Ruths, Derek (2014). Control profiles of complex networks. *Science*, *343*(6177), 1373–1376.
Siami, Milad, & Jadbabaie, Ali (2020). A separation theorem for joint sensor and actuator scheduling with guaranteed performance bounds. *Automatica*, *119*, Article 109054.
Siami, Milad, & Motee, Nader (2018a). Growing linear dynamical networks endowed by spectral systemic performance measures. *IEEE Transactions on Automatic Control*, *63*(8).
Siami, Milad, & Motee, Nader (2018b). Network abstraction with guaranteed performance bounds. *IEEE Transactions on Automatic Control*, *63*(11).
Siami, Milad, Olshevsky, Alexander, & Jadbabaie, Ali (2020). Deterministic and randomized actuator scheduling with guaranteed performance bounds. *IEEE Transactions on Automatic Control*, *66*(4), 1686–1701.
Summers, Tyler H., Cortesi, Fabrizio L., & Lygeros, John (2015). On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, *3*(1), 91–101.
Tian, Yulun, Khosoussi, Kasra, & How, Jonathan P. (2021). A resource-aware approach to collaborative loop-closure detection with provable performance guarantees. *International Journal of Robotics Research*, *40*(10–11), 1212–1233.
Tropp, Joel A. (2012). User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, *12*(4), 389–434.
Tzoumas, Vasileios (2018). *Resilient submodular maximization for control and sensing* (Ph.D. thesis), University of Pennsylvania.
Tzoumas, Vasileios, Carlone, Luca, Pappas, George J, & Jadbabaie, Ali (2020). Lqg control and sensing co-design. *IEEE Transactions on Automatic Control*, *66*(4), 1468–1483.
Vafaee, Reza, & Siami, Milad (2022a). Learning-based sensor selection with guaranteed performance bounds. In *2022 American control conference* (pp. 1459–1465). IEEE.
Vafaee, Reza, & Siami, Milad (2022b). On-the-fly sensor scheduling with performance guarantees. In *2022 IEEE 61st conference on decision and control* (pp. 6018–6025). IEEE.
Ye, Lintao, Chi, Ming, Liu, Zhi-Wei, & Gupta, Vijay (2022). Online actuator selection and controller design for linear quadratic regulation over a finite horizon. arXiv preprint arXiv:2201.10197.
Ye, Lintao, Roy, Sandip, & Sundaram, Shreyas (2020). Resilient sensor placement for Kalman filtering in networked systems: Complexity and algorithms. *IEEE Transactions on Control of Network Systems*, *7*(4), 1870–1881.
Ye, Lintao, Woodford, Nathaniel, Roy, Sandip, & Sundaram, Shreyas (2020). On the complexity and approximability of optimal sensor selection and attack for Kalman filtering. *IEEE Transactions on Automatic Control*, *66*(5), 2146–2161.

Reza Vafaee is a Ph.D. candidate in the Department of Electrical and Computer Engineering at Northeastern University (NEU) in Boston, MA. His research focuses on the mathematical and algorithmic foundations of network optimization and control, with applications in robotics and power systems. Prior to joining NEU, he earned his second master's degree with highest distinction in pure and applied mathematics from Montclair State University, NJ, in 2020.

Milad Siami received his dual B.Sc. degrees in electrical engineering and pure mathematics from Sharif University of Technology in 2009, M.Sc. degree in electrical engineering from Sharif University of Technology in 2011. He received his M.Sc. and Ph.D. degrees in mechanical engineering from Lehigh University in 2014 and 2017 respectively. He was a postdoctoral associate in the Institute for Data, Systems, and Society at MIT, from 2017 to 2019. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA. His research interests include distributed control systems, distributed optimization, and applications of fractional calculus in engineering. Dr. Siami has received several awards and fellowships, including a Gold Medal at the National Mathematics Olympiad in Iran, the Best Student Paper Award at the 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems, and the Rossin College Doctoral Fellowship at Lehigh University.