On-the-fly Sensor Scheduling with Performance Guarantees

Reza Vafaee[1] and Milad Siami[1]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992391

Abstract— **In this paper, we investigate the problem of time-varying sensor selection for linear time-varying (LTV) dynamical systems. We develop a framework to design an online sparse sensor schedule for a given large-scale LTV system with guaranteed performance bounds using randomized algorithms. In our online setting, the contribution of each sensor at each time is calculated on-the-fly, and we immediately decide to keep the corresponding sensor at each time in the sensor schedule or discard it without ever retracting these decisions. Furthermore, we provide new performance guarantees to approximate fully-sensed LTV systems up to a multiplicative approximation factor and an additive one by choosing on average a constant number of active sensors at each time.**

I. Introduction

Given the increase in the availability of high-performance computing processors, high-capacity storage, and computationally efficient fast algorithms, control and estimation of interconnected complex systems increasingly become of interest to researchers and scientists [1]–[6]. In the past decade, we accordingly observe the rise of efforts that have been investigated for the analyse and estimation of complex networked systems with the broad areas of application such as smart grids [1], social networks [7], statistical physics [2], [3], multi-robot systems [4], [8], computational biology [9], etc. Due to the increasingly large-scale nature of the problems raised by these complex systems, it is often necessary to estimate the overall state of a given large complex system using only a small subset of available sensors either because accruing the individual measurement is expensive, or processing full-state measurements is impossible due to the computational constraints.

A few key sensors should be selected carefully and placed strategically such that we are able to collect the critical information while managing uncertainties. However, developing a framework to find this small optimal set of sensor is an interesting (mostly) unsolved question. In the most straightforward scenario, obtaining the optimal set of sensors in a known task with a well-defined cost function imposes a combinatorial search. Due to the brute-force nature of this combinatorial search, it is proven to be NP-hard and computationally intractable for all but simplest scenarios [10]. The optimal sensor selection problem has mostly been addressed by finding the optimal solution set of the optimization problems built upon some observability performance measures [11], [12]. The performance measures for these optimizations generally are a function of Gramian matrices as it provides a quantitative measure of the energy required for estimation. The authors of [6], [13] introduce the notion of systemic metrics for linear dynamical systems. This class of performance measures is defined such that they are monotone, convex, and homogeneous with respect to the Gramian matrix of the system and includes some frequently used metrics such as determinant or the trace of inverse.

The deterministic and randomized approaches are employed to design a time-varying sparse actuator scheduling for linear dynamical systems in [14]. Later in [15], same authors investigated the problem of designing joint time-varying sparse sensor and actuator scheduling leveraging Hankel singular values of the linear system. In both of these papers, the resulting optimal sparse system is compared to the fully-actuated/sensed one. In contrast, a swapping regret minimization algorithm was exploited recently in [13] to round the continuous solution of the continuous optimization to obtain a $(1+\epsilon)$ approximation of the actual optimal system for all types of systemic metrics. The balanced model reduction and greedy matrix QR pivoting are used to select the sensors and actuators to optimize observability and controllability of the system in [16].

Another similar problem is the problem of finding the minimal (instead of optimal) set of sensors/actuators such that the system remains observable/controllable. The Minimal control set problem was studied in [17] and is shown to be NP-hard. Furthermore, this problem cannot efficiently be solved or even approximated in polynomial time [18]. Some other closely related problems are optimal leader selection and control of the formation in multi-agent systems [19], [20].

Our contributions: Finding the small representation of the sensors becomes more challenging when the model of the system is changing by time, or the *t-step observability* matrix is large and cannot be stored and processed in memory. To tackle these challenges and inspired by a growing body of work on online algorithms for machine learning and big data analysis, in this paper, we develop an extremely simple randomized framework to on-the-fly (online) randomly select sensors and design a time-varying scheduling. In our setup,

This material is based upon work partially supported by the U.S. Department of Homeland Security under Grant Award Number 22STESE00001-01-00. 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. This research was also supported in part by grants ONR N00014-21-1-2431 and NSF 2121121.

[1] Department of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mails: {vafaee.r, m.siami}@northeastern.edu).

the rows of observability matrix of a time-varying system, which are equivalently the contribution of the sensors at the different times, are receiving one-by-one, and we immediately decide to keep the sensor at each time in the schedule or discard it, without ever retracting these decisions. The proposed randomized framework is truly a Markov chain, i.e., the probability of choosing a sensor at a time only depends on the previous sensors in the stream. Moreover, the framework is clean and intuitive and on average samples a constant number of active sensors at each time such that it approximates the fully-sensed LTV systems up to a multiplicative approximation factor and an additive one in the systemic measure sense.

To make this manuscript more understandable and due to the space limitations, some of the proofs are eliminated in the paper.

II. Preliminaries and Definitions

A. Mathematical Notations

Throughout the paper, the discrete time index is denoted by k. The sets of real (integer), non-negative real (integer), and positive real (integer) numbers are represented by $\mathbb{R}$ ($\mathbb{Z}$), $\mathbb{R}_+$ ($\mathbb{Z}_+$) and $\mathbb{R}_{++}$ ($\mathbb{Z}_{++}$), respectively. The set of natural numbers $\{i \in \mathbb{Z}_{++} \ : \ i \leq n\}$ is denoted by $[n]$. Uppercase letters, such as A or B, stand for real-valued matrices, and lowercase letters denote vectors (e.g., b or c). For a vector $x = [x_i] \in \mathbb{R}^n$, $\operatorname{diag}(x) \in \mathbb{R}^{n\times n}$ is the diagonal matrix with elements of x sitting orderly on its diagonal, and if $X = [x_{i,j}] \in \mathbb{R}^{n\times n}$ is a square matrix, $\operatorname{diag}(X)$ outputs the diagonal elements of X. For a square matrix X, $\det(X)$ and $\operatorname{Trace}(X)$ refer to the determinant and the summation of on-diagonal elements of X, respectively. $\mathbb{S}^n_+$ (resp. $\mathbb{S}^n_{++}$) is the positive semi-definite cone (resp. positive definite cone) of n-by-n matrices. We represent the n-by-n identity matrix and the n-by-1 vector of all ones by I and $\mathbb{1}$, respectively. Notation $A \preceq B$ is equivalent to matrix $B - A$ being positive semi-definite. The transpose of matrix A is denoted by $A^\top$. Non-bold face letters are used for scalars and indices (e.g., j) and function names (e.g., $f(\cdot)$). We denote the Moore-Penrose pseudoinverse of matrix A by $A^\dagger$, and $(A^\dagger)^{1/2} = A^{-1/2}$. Symbol $\lfloor x \rfloor$ represents the floor function of real number $x \in \mathbb{R}$ and it gives as output the greatest integer less than or equal to x. The L_0-norm that counts the total number of nonzero elements of a vector is referred to by $\|\cdot\|_0$. Symbol $\|\cdot\|_1$ denotes L_1-norm of a vector that outputs the sum of the absolute values of the entries of the vector. Moreover, symbol $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. m-by-n matrices of all zeros are represented by $\mathbf{0}_{m\times n}$, and the symbol $\oplus$ denotes appending the rows of one matrix to another.

B. Linear Systems, Controllability and Observability

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

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

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

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 state matrix $A(k)$ describes the underlying structure of the system and the interaction strength between the agents/states at time instant k, the input matrix $B(k)$ identifies the nodes controlled by an outside controller at time k, and the output matrix $C(k)$ shows how output vector y relates to the state vector again at time step k. Given the initial condition $x(0) = x_0$ of the state variables and the input sequence $u(k)$, the goal is to find the state sequence $x(k)$ as well as the output sequence $y(k)$. Referring back to (1), we can write

$$\begin{aligned} x(1) &= A(0)x(0) + B(0)u(0) \\ x(2) &= A(1)x(1) + B(1)u(1) \\ &= A(1)A(0)x(0) + A(1)B(0)u(0) + B(1)u(1) \\ &\cdots \\ x(t) &= \Phi(t,0)x(0) + \sum_{r=0}^{t-1} \Phi(t, r+1)B(r)u(r) \\ &= \Phi(t,0)x(0) + \mathcal{R}(t,0)\mathcal{U}(t,0), \end{aligned} \tag{3}$$

where $\Phi(t,r)$ is the *state transition matrix*, which relates the state of the undriven system[1] at time t to the state at an earlier time r, i.e., $x(t) = \Phi(t,r)x(r)$ for all $t \geq r$. One can formulate the state transition matrix as

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

Furthermore,

$$\mathcal{R}(t,0) = [\Phi(t,1)B(0) \mid \Phi(t,2)B(1) \mid \cdots \mid B(t-1)], \tag{4}$$

is the *t-step controllability matrix* of the time-varying system (1)-(2), and

$$\mathcal{U}(t,0) = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(t-1) \end{bmatrix}.$$

Let repeat the steps this time for the output equation (2). Take $k = 0, 1, \ldots, t-1$ in (2), i.e.,

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(t-1) \end{bmatrix} = \begin{bmatrix} C(0)\Phi(0,0) \\ C(1)\Phi(1,0) \\ C(2)\Phi(2,0) \\ \vdots \\ C(t-1)\Phi(t-1,0) \end{bmatrix} x(0) +$$

$$\begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ C(1)B(0) & 0 & \ldots & 0 & 0 \\ C(2)\Phi(2,1)B(0) & C(2)B(1) & \ldots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ C(t-1)\Phi(t-1,1)B(0) & C(t-1)\Phi(t-1,2)B(1) & \ldots & C(t-1)B(t-2) & 0 \end{bmatrix} \mathcal{U}(t,0). \tag{5}$$

The second term on the right-hand side of (5) is known (we assume the input sequence $u(k)$ is given); therefore, one can subtract it from the vector of measurements on the left

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

to obtain

$$\underset{\sim}{\mathcal{Y}}(t,0)=\begin{bmatrix} C(0)\Phi(0,0) \\ C(1)\Phi(1,0) \\ C(2)\Phi(2,0) \\ \vdots \\ C(t-1)\Phi(t-1,0) \end{bmatrix} x(0)=\mathcal{O}(t,0)x(0), \quad (6)$$

where we have made the obvious definitions for $\underset{\sim}{\mathcal{Y}}(t,0)$ and the *t-step observability matrix* $\mathcal{O}(t,0)$. We can see that it is only needed to check the observability when $\mathcal{U}(t,0)$ is zero based on (6); the effect of a non-zero input is just to change what $\underset{\sim}{\mathcal{Y}}(t,0)$ is, but in either case $\underset{\sim}{\mathcal{Y}}(t,0)$ is a known vector.

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

It is well-known that from a numerical standpoint it is better to characterize controllability and observability in terms of the Gramian matrices at time t, defined as follows for the time-varying system (1)-(2):

$$\mathcal{W}(t,t_0=0)=\mathcal{R}(t,0)\mathcal{R}^\top(t,0)=\sum_{r=t_0}^{t-1}\Phi(t,r+1)B(r)B^\top(r)\Phi^\top(t,r+1), \quad (7)$$

and

$$\mathcal{X}(t,t_0=0)=\mathcal{O}^\top(t,0)\mathcal{O}(t,0)=\sum_{r=t_0}^{t-1}\Phi^\top(r,t_0)C^\top(r)C(r)\Phi(r,t_0). \quad (8)$$

For notational convenience, we will drop writing t_0 in controllability and observability Gramian matrices and only write $\mathcal{W}(t,0)$ and $\mathcal{X}(t,0)$. It is worth to highlight that the dynamics presented in (1)-(2) can be also written in the following form

$$x(k+1)=A(k)x(k)+\sum_{i\in[m]}b_i(k)u_i(k), \quad (9)$$

$$y(k)=\sum_{j\in[p]}e_jc_j^\top(k)x(k), \quad (10)$$

where $b_i(k)$'s are the columns of the time-varying matrix $B(k)\in\mathbb{R}^{n\times m}$, $c_j^\top(k)$'s are rows of matrix $C(k)\in\mathbb{R}^{p\times n}$, and e_j's are the standard basis for $\mathbb{R}^p$.

Assumption 2: Throughout the paper, we assume that the system (9)-(10) is an n-state minimal realization (i.e., it is reachable and observable).

C. Systemic Controllability/Observability Metrics

Similar to the *systemic* notion introduced in [5], [6], [13], [14], [21], we define various controllability/observability metrics. These measures are real-valued operators defined on the set of all linear dynamical systems derived from (9)-(10) and quantify various measures of the required energy in the system. All of the metrics depend on the controllability/observability Gramian matrix of the system, which is a positive definite matrix. Therefore, one can define a systemic performance measure as an operator on the set of Gramian matrices of all controllable/observable systems over 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 $A,B\in\mathbb{S}^n_{++}$, it satisfies: (i) *(positively) Homogeneity* criteria: $\rho(tA)=t^{-1}\rho(A)$, for $t>0$ [2]; (ii) *Monotonicity* criteria: if $B\preceq A$, then $\rho(B)\geq\rho(A)$; (iii) *Convexity* criteria: $\rho\big(\alpha A+(1-\alpha)B\big)\leq\alpha\rho(A)+(1-\alpha)\rho(B)$, for all $\alpha\in[0,1]$.

Several comprehensive studies have been done in [5], [14] on this class of performance metrics for dynamical networks. It was shown that the set of criteria listed in Definition 1 holds for many popular choices of systemic metric ρ. However, for the sake of brevity, we do not repeat them here and refer the interested readers to [14, Table I] and [5, Table I] for a complete list of systemic measures.

III. Online Sparse Sensor Schedule

A. Sparse Sensor Selection Problems

Suppose the linear, time-varying system (9)-(10) is given; the goal in the *sparse* sensor selection problem is to design an sparse sensor outputs schedule such that the observability performance metrics of the original (fully-sensed) and the sparse systems are similar in an appropriately defined sense. Moreover, we try to retain the number of active sensors much less than fully-sensed system in the resulting schedule. Particularly, given a canonical linear, discrete-time, time-varying system (9)-(10) that has p sensors, observability systemic metric $\rho(\cdot)$ that is aligned with the properties addressed in Definition 1, and the observability Gramian matrix $\mathcal{X}(t,0)$ at time t, the goal is to find a sensor schedule such that the resulting system with the observability Gramian matrix $\mathcal{X}_s(t,0)$ is *well-approximated*, i.e.,

$$\left|\log\frac{\rho(\mathcal{X}_s(t,0))}{\rho(\mathcal{X}(t,0))}\right|\leq\epsilon', \quad (11)$$

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

B. A Weighted Sparse Sensor Schedule

To develop a framework to online sample the sensors, we first show how a sparse system looks like in terms of dynamical equations. Then, we show that a weighted schedule can be obtained if we scale the output signal by a non-negative factor while keeping the scales bounded. Indeed, the scaling introduces an extra degree of freedom that allows us to may obtain a sparser set of outputs. Considering (10), we define a weighted sensor schedule by $S=[s_{j,k+1}]$ and scaling $s_{j,k+1}\geq 0$ where $j\in[p]$ and $k+1\in[t]$. The resulting output equation with this schedule is

$$y(k)=\sum_{j\in[p]}s_{j,k+1}\cdot e_jc_j^\top(k)x(k),\ k\in\mathbb{Z}_+, \quad (12)$$

where $s_{j,k+1}\geq 0$ shows the strength of the j-th sensor output at time k. Let denote the vectorization of the p-by-t weighted sensor schedule S by $\mathrm{vec}(S)$ and define it as

$$\mathrm{vec}(S):=[s_{1,1},\dots,s_{p,1},s_{1,2},\dots,s_{p,2},\dots,s_{1,t},\dots,s_{p,t}]^\top, \quad (13)$$

[2] A function ρ is (positively) homogeneous if $\rho(tA)=t^{-n}\cdot\rho(A)$, for $t>0$, where n is the degree of homogeneity. However, throughout this paper, when we say a metric is homogeneous, we mean it is (positively) homogeneous of degree 1.

where $\mathrm{vec}(S)$ is the tp-by-1 column vector obtained by stacking the columns of the schedule S on top of one another. Using this vectorization notation, the observability Gramian matrix (8) at time t for the sparse system (12) can be reformulated as

$$\begin{aligned}\mathcal{X}_s(t,0) &= \left(\Lambda_s \cdot \mathcal{O}(t,0)\right)^\top \underbrace{\left(\Lambda_s \cdot \mathcal{O}(t,0)\right)}_{:=\,\mathcal{O}_s(t,0)} \\ &= \mathcal{O}^\top(t,0)\ \Lambda_s^2\ \mathcal{O}(t,0), \end{aligned} \qquad (14)$$

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

Our goal is to decrease the number of active sensors on average d, where

$$d := \frac{1}{t} \cdot \|\mathrm{vec}(S)\|_0, \qquad (15)$$

such that the observability Gramian of the fully-sensed and the sparse system are close. One might show that obtaining this approximate sparse system needs horizon length that is potentially longer than the dimension of the state.

Definition 2 ((ϵ,δ,d)-approximation): Given a time horizon $t \geq n$, system (12) with a sparse weighted sensor schedule S is (ϵ,δ,d)-approximation of system (10), if and only if

$$(1-\epsilon)\mathcal{X}(t,0) - \delta I \preceq \mathcal{X}_s(t,0) \preceq (1+\epsilon)\mathcal{X}(t,0) + \delta I, \quad (16)$$

where $\mathcal{X}(t,0)$ and $\mathcal{X}_s(t,0)$ are the observability Gramian matrices for the fully-sensed and sparse system defined in (8) and (14), respectively. Parameter d as defined in (15) 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, $\mathcal{X}_s(t,0) \approx_{\epsilon,\delta} \mathcal{X}(t,0)$ denotes the same condition.

We note that a closely related approximation notation was developed in [14] for static time-invariant networks. It is called (ϵ,d)-approximation and formulated as

$$(1-\epsilon)\mathcal{X}(t,0) \preceq \mathcal{X}_s(t,0) \preceq (1+\epsilon)\mathcal{X}(t,0), \qquad (17)$$

and addressed succinctly as $\mathcal{X}_s(t,0) \approx_{\epsilon} \mathcal{X}(t,0)$ here. One might find out that once $\delta = 0$, both (ϵ,δ,d) and (ϵ,d) approximations are same.

Remark 1: When ϵ is small enough [3], we can elaborate (17) to show that the (ϵ,d)-approximation system is in fact a well-approximated system. Identically, if $\mathcal{X}_s(t,0) \approx_{\epsilon} \mathcal{X}(t,0)$, then

$$\left|\log \frac{\rho(\mathcal{X}_s(t,0))}{\rho(\mathcal{X}(t,0))}\right| \leq \epsilon, \qquad (18)$$

where $\rho(\cdot)$ is some observability systemic measure. To obtain (18), 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 $\mathcal{X}_s(t,0) \approx_{\epsilon,\delta} \mathcal{X}(t,0)$, then

$$\left|\log \frac{\rho(\mathcal{X}_s(t,0)+\lambda I)}{\rho(\mathcal{X}(t,0)+\lambda I)}\right| \leq \epsilon, \qquad (19)$$

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

Proposition 1: Taking the additive approximation factor $\delta = \epsilon \cdot \sigma^2_{\min}(\mathcal{O}(t,0))$ reduces (ϵ,δ,d)-approximation to (ϵ,d)-approximation, where $\sigma_{\min}(\mathcal{O}(t,0))$ denotes the minimum singular value of the observability matrix $\mathcal{O}(t,0)$ as defined in (6).

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

In the next section, we try to formulate the online problem and establish a bridge between prior solved problems and the new online problem to put forward a solution to on-the-fly design a sensor schedule.

C. Online Problem

We begin with formally define some concepts that we will frequently use in this paper. We will later use these definitions to formulate our online problem. Then, we close this section by revisiting prior work and discussing some new results.

Online setting: For system (9)-(10), the observability matrix can be rewritten in the more expanded form as

$$\mathcal{O}(t,0) = \begin{bmatrix} c_1^\top(0)\Phi(0,0) \\ \vdots \\ c_p^\top(0)\Phi(0,0) \\ c_1^\top(1)\Phi(1,0) \\ \vdots \\ c_p^\top(1)\Phi(1,0) \\ \vdots \\ c_1^\top(t-1)\Phi(t-1,0) \\ \vdots \\ c_p^\top(t-1)\Phi(t-1,0) \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ \vdots \\ a \end{matrix}}\right\} C(0)\Phi(0,0) \\ \left.\vphantom{\begin{matrix} a \\ \vdots \\ a \end{matrix}}\right\} C(1)\Phi(1,0) \\ \vphantom{\vdots} \\ \left.\vphantom{\begin{matrix} a \\ \vdots \\ a \end{matrix}}\right\} C(t-1)\Phi(t-1,0) \end{matrix} \qquad (20)$$

where each row of this matrix can be interpreted as the contribution of a sensor at a time in the observability Gramian matrix. More precisely, each row of this matrix is constructed by the j-th sensor of the sensor set of time k, or equivalently $c_j^\top(k)$, where $j \in [p]$ and $k+1 \in [t]$. Furthermore, the additional term $\Phi(k,0)$ in each row only transfers the state at time k to the initial state $x(0)$. In an *online setting*, the observability matrix $\mathcal{O}(t,0)$ is not given to us in advance, and instead, we are receiving the rows (sensor contributions) of this matrix one-by-one. Each time a row arrives, we decide irrevocably whether to keep its corresponding sensor in our schedule or discard it, without ever retracting these decisions.

Although the online setting just discussed requires to receive the rows one-by-one, in a real system, we outright get access to a bunch of rows each time that the time instant k is increased by one. In the other words, at time instant $k=0$, the entire of output matrix $C(0)$ is available to us. Right after the time step has changed to 1, the entire matrix $C(1)\Phi(1,0)$ becomes available. One can observe that the same story repeats each time that the time step changes. However, this property arising from the nature of the system does not affect what we define as *Online Setting* (we still can assume that the rows arrive one at a time).

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

Recently, randomized algorithms have achieved a lot of successes in the theoretical computer science in subset selection and similar problems. For example, probabilistic method was employed by Marcus, Spielman, and Srivastava to propose a solution to the so-called Kadison-Singer (KS) conjecture [22]. Getting inspired by the solution of KS, authors of [14] proposed a framework to randomly select actuators to construct a sparse schedule for a linear, time-invariant, discrete-time systems. Similar to us, their goal was picking a fairly small number of actuators at each time such that the controllability systemic measures of the original and sparse system are close in some sense. Since the entire of the controllability/observability matrix for the time-invariant system is available at time step t if we know the matrices of the system, i.e., A, B and C, the whole of the controllability matrix was used to obtain the sampling probabilities for the columns of the controllability matrix (inputs' contributions), and then the columns (actuators at different times) were randomly picked using this probability distribution to design the scheduling in their work.

Note: Since in the rest of this paper, we will frequently address the rows of the observability matrix, we use $\mathfrak{o}_i^\top$ to denote the i-th row $c_j^\top(k)\Phi(k,0)$ where $i = kp+j$, $j \in [p]$, and $k+1 \in [t]$. In the time-invariant system, $\Phi(k,0)$ is A^k, and the letter k representing the time instant in the output matrices will be dropped.

In general, a *sampling scheme* for randomly selecting actuators/sensors is formulated and formalized as the following. Since the focus of this paper is on sensor scheduling, we state the scheme accordingly. However, one can write the dual scheme for the actuator setup.

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

$$\begin{aligned}\mathbb{E}[\underbrace{\mathcal{O}_s^\top(t,0)\mathcal{O}_s(t,0)}_{\mathcal{X}_s(t,0)}] &= \sum_{i=1}^{tp} p_i \cdot \big(\frac{1}{p_i}\mathfrak{o}_i\mathfrak{o}_i^\top\big) \\ &= \sum_{i=1}^{tp} \mathfrak{o}_i\mathfrak{o}_i^\top = \underbrace{\mathcal{O}^\top(t,0)\mathcal{O}(t,0)}_{\mathcal{X}(t,0)}. \quad (21)\end{aligned}$$

To obtain good concentration in the sparse observability Gramian matrix, we need to select more unique rows with the higher probabilities. Thus, the sensor sampling problem reduces to this simple question that how to measure the *uniqueness* of different rows in the observability matrix.

In [14, Theorem 2], the *leverage scores* of the columns of the controllability matrix are used as the measure of uniqueness of the columns to construct a sparse weighted actuator schedule for the time-invariant systems. Leveraging this uniqueness definition, it was shown that randomly selecting on average $d = \frac{9c^2 n\log n}{t\epsilon^2}$ actuators at each time results in a sparse schedule with controllability Gramian matrix $\mathcal{W}_s(t,0)$ such that $\mathcal{W}_s(t,0) \approx_\epsilon \mathcal{W}(t,0)$, where $\mathcal{W}(t,0)$ is the controllability of the fully-actuated system. Constant c in their algorithm was supposed same as the constant c in [23, Theorem 3.1]. The leverage score (which is same as effective resistance in the algebraic graph theory [24]) is defined as follows.

Definition 3 (Leverage Score, τ): 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

$$\begin{aligned}\tau_i = \tau(q_i^\top) = \underset{w\in\mathbb{R}^r}{\text{minimize}} \quad & \|w\|^2 \\ \text{subject to} \quad & Q^\top w = q_i\end{aligned} \qquad (22)$$

where $q_i^\top$ is the i-th row of matrix Q. τ_i measures how important q_i is in composing range of $Q^\top$.

Remark 2: At most, τ_i is one, since we can simply choose w to be the i-th basis vector in $\mathbb{R}^r$. Moreover, τ_i is less than one when other rows approximately align with $q_i^\top$ or when $\|q_i\|$ is small.

Proposition 2: For all matrices $Q \in \mathbb{R}^{r\times n}$ and $i \in [r]$, leverage score $\tau(q_i^\top)$ is the smallest α such that:

$$q_i q_i^\top \preceq \alpha \cdot Q^\top Q. \qquad (23)$$

Optimization (22) is a least norm optimization. It is easy to show that when Q is a full-row rank matrix, its unique optimal solution can be obtained by introducing Lagrange multipliers and is in the form of $\hat{w} = Q(Q^\top Q)^{-1}q_i$. Therefore, $\tau_i = \|\hat{w}\|^2 = q_i^\top(Q^\top Q)^{-1}q_i$. When Q is not a full-rank matrix, $\tau_i = q_i^\top(Q^\top Q)^\dagger q_i$.

Although the dual algorithm as of [14] can be utilized to sample sensors in the time and place (sample rows of the observability matrix $\mathcal{O}(t,0)$) with probability proportional to their leverage score to obtain a (ϵ,d)-approximation of the fully-sensed system with high probability, computation of the exact leverage scores is too expensive (it requires solving systems in $\mathcal{O}^\top(t,0)\mathcal{O}(t,0)$ if it computed naively). However, in the following theorem, we will show that sampling with some overestimate approximations of the leverage score also suffices to obtain (ϵ,d)-approximation of the system.

Note that $\tau(\mathcal{O}(t,0))$ is the vector of the diagonal elemets of the projection matrix $\mathcal{O}(t,0)\mathcal{X}^\dagger(t,0)\mathcal{O}^\top(t,0)$, and $\tau_i = \tau(\mathcal{O}(t,0))(i)$.

Theorem 1 (Overestimates (ϵ,d)-approximation): Given an approximation factor $\epsilon \in (0,1)$, time horizon $t \geq n$, and the dynamics (10), let $\mathbf{u} = [u_i]$ be a vector of leverage score overestimates of the rows of the observability matrix, i.e., $u_i \geq \tau(\mathcal{O}(t,0))(i)$ for all $i \in [tp]$, $c > 3$ be a fixed constant, and the diagonal elements of the sparsification matrix Λ_s be $\Lambda_s(i,i) = \frac{1}{\sqrt{p_i}}$ with probability $p_i = \min(c\cdot u_i \cdot \log n/\epsilon^2, 1)$ and zero otherwise, then this sparsification matrix Λ_s produces scheduling S which is (ϵ,d)-approximation of (10) with probability at least $1-n^{1-c/3}$, and the average number of active sensors at each time d is at most $c\cdot\|\mathbf{u}\|_1\cdot\log n/t\epsilon^2$.

Applying the generalization of the Chernoff bound for matrices (which is a variant of [25, Corollary 5.2]) to the sequences of matrices defined as $M_i = \frac{\mathfrak{o}_i\mathfrak{o}_i^\top}{p_i}$, together with the result of Proposition 2 will give the proof.

Theorem 1 shows that to obtain a (ϵ,d)-approximation of the fully-sensed system, having a rough overestimate of the leverage scores of the rows of the observability matrix would be sufficient. However, the results of this theorem can not be simply modify and extend to the restrictive settings such as the semi-streaming setting or online setting because the entire observability matrix is needed beforehand to obtain the rough overestimations.

In the restrictive settings, however, the simple natural idea is just relying on the partial data that have already collected to approximate the leverage scores. Kelner et al. in [26] exploit similar idea to obtain a spectral approximation of a graph in the semi-streaming setting. Their idea was natural and conceptually simple. Their algorithm receives rows of the vertex edge incidence matrix (edges) one-by-one. The algorithm then rejects each row (edge) with the probability depending on it's leverage score with respect to the rows (edges) seen so far. The problem is the more rows are received, the better estimates can be obtained for the leverage scores of the previously sampled rows. This is more or less clear form the definition of leverage score (22), but it was proven rigorously in [27, Theorem 11]. Kelner et al. also adjusted their algorithm accordingly to skirt this problem. In summary, their final algorithm performs as this: as the rows of the vertex edge incidence matrix arrive, they are added to a small set of previously sampled rows called sparsifier set (the initial sparsifier set is sampled from the vertex edge incidence matrix of the initial graph by using the sampling procedure of [24]). Whenever the sparsifier set becomes too big (too big here means bigger than the storage space $\tilde{O}(n)$ where n is the number of columns of the vertex edge incidence matrix), it gets resparsified again. To skirt the problem addressed earlier, the algorithm considers not only the rows added but also the rows already exist in the sparsifier set in the resparsification steps.

However, the probability of sampling a row does not only depend on earlier rows in the stream due to pruning steps. Authors of [28] address this and state that this difficult dependency issue seems to break the argument in [26] where they claim that the distribution of their algorithm is same as the distribution of one round sampling by leverage scores. Therefore, one may not ensure that a spectral approximation is obtained. Later in [29], authors add an extra challenge in analyzing the pruning routines. They declare these dependencies also prevent using matrix concentration inequalities. They then resolve both of these issues by proposing an algorithm that only relies on the sampling probabilities computed based on the current graph.

Although the algorithms proposed in [29] can be reformed and then employed for the purpose of our online sensor scheduling, we prefer to break this dependency issues by doing *online sampling*. That is, when we see a row (sensor's contribution), we either sample it or do not, and we never update our decision. In this way, our algorithm is truly a Markov chain—i.e., sampling $\mathfrak{o}_i^\top$ only depends on the choices for $\mathfrak{o}_j^\top$ with $j < i$, and not for $\mathfrak{o}_j^\top$ with $j \geq i$. We add a small coefficient of identity matrix to the process to make sure that the online sampling probabilities are bounded[4]. We will show that the whole process allows us to rigorously argue that a sparse schedule with guaranteed performance bounds is obtained. We then reform all will be discussed to an online sensor scheduling framework that results in a (ϵ, δ, d)-approximation of the fully-sensed dynamics (10).

D. Main Online Sampling Result

In this section, we adapt the so-called *ridge leverage score* to achieve (ϵ, δ, d)-approximation. We show how using ridge leverage score allows for online sampling and consequently avoids the resparsification routines. Thus, the sampling probability of a sensor only depends on the previous sensors in the stream. Ridge leverage score was previously introduced for approximate kernel ridge regression [30], and it has been used as the sampling probabilities in the context of spectral approximation and online sampling [28], [31] and iteratively computing regular leverage scores [32]. We extend their applicability to online sensor scheduling.

In the ridge leverage scores, we are interested in computing the leverage scores for $Q^\top Q + \lambda I$ instead of $Q^\top Q$, where $\lambda > 0$ is some small number. In the literature of machine learning these quantities are also called *λ-ridge leverage scores* and defined as the following [33]:

$$\tau_i^\lambda := q_i^\top (Q^\top Q + \lambda I)^{-1} q_i, \tag{24}$$

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

One might simply update Theorem 1 for λ-ridge leverage scores to obtain (ϵ, δ, d)-approximation of dynamics (10). Let us state it formally in the following theorem.

Theorem 2 (Overestimates (ϵ, δ, d)-approximation): Given an approximation factor $\epsilon \in (0, 1)$, additive approximation factor $\delta > 0$, $\lambda := \delta/\epsilon$, time horizon $t \geq n$, and the dynamics (10), let $\mathbf{L} = [\ell_i]$ be a vector of λ-ridge leverage score overestimates of the rows of the observability matrix, i.e., $\ell_i \geq \mathfrak{o}_i^\top (\mathcal{X}(t, 0) + \lambda I)^{-1} \mathfrak{o}_i$ for all $i \in [tp]$, $c > 3$ be a fixed constant, and the diagonal elements of the sparsification matrix Λ_s be $\Lambda_s(i, i) = \frac{1}{\sqrt{p_i}}$ with probability $p_i := \min(c \cdot \ell_i \cdot \log n/\epsilon^2, 1)$ and zero otherwise, then this sparsification matrix Λ_s produces scheduling S which is (ϵ, δ, d)-approximation of (10), i.e., $\mathcal{X}_s(t, 0) = \mathcal{O}^\top (t, 0) \Lambda_s^2 \mathcal{O}(t, 0) \approx_{\epsilon,\delta} \mathcal{X}(t, 0)$, with probability at least $1 - n^{1-c/3}$. Moreover, the average number of active sensors at each time d is at most $c \cdot \|\mathbf{L}\|_1 \cdot \log n/t\epsilon^2$.

Proof: Theorem 1 says that if we sample the rows of the observability matrix with probabilities proportional to their overestimates leverage score the (ϵ, d)-approximation is obtained with high probability, or the following holds

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

We define $\mathcal{O}^\lambda(t, 0) := \mathcal{O}(t, 0) \oplus \sqrt{\lambda} \cdot I$, so ${\mathcal{O}^\lambda}^\top (t, 0)\mathcal{O}^\lambda(t, 0) = \mathcal{O}^\top (t, 0)\mathcal{O}(t, 0) + \lambda I$. Then, if we sample the rows of matrix $\mathcal{O}^\lambda(t, 0)$ with their leverage

[4] If we do not add the identity matrix, we might take n samples.

scores, we get

$$(1-\epsilon)(\mathcal{X}(t,0)+\lambda I) \preceq \mathcal{X}_s(t,0)+\lambda I \preceq (1+\epsilon)(\mathcal{X}(t,0)+\lambda I). \tag{25}$$

One can observe that the rows of matrix $\sqrt{\lambda}\cdot I$ are all sampled since their leverage scores are one. Finally, moving λI from the middle side, combining the similar terms, and putting $\lambda = \delta/\epsilon$ give us the theorem. ∎

Although Theorem 2 shows that the slightly different performance bounds will be obtained by utilizing λ-ridge leverage scores, it can not be applied yet because firstly, we still do not know how the overestimates of the λ-ridge leverage scores can be computed, and secondly, the sum of these overestimate scores $\|\mathbf{L}\|_1$ is unknown. We will take care of these questions one-by-one and show that how the solutions proposed put forward a framework to design an online scheduling of the sensors.

The current definition of λ-ridge leverage scores stated in (24) can not be employed in an online setting to obtain even the exact values of these scores, since the entire of the observability matrix is not given to us in advance, and their rows are streamed in one at a time. To align this definition with our online goal, we update and redefine it in an online fashion.

Definition 4 (Online λ-ridge leverage score, $\overline{\tau}_i^\lambda$): Let $\mathcal{O}_{i-1}(t,0)$ for $i \in [tp]$ be the fraction of the observability matrix that includes it's first $i-1$ rows, then we define the *online λ-ridge leverage score* as

$$\overline{\tau}_i^\lambda := \min(\mathfrak{o}_i^\top(\mathcal{X}_{i-1}(t,0)+\lambda I)^{-1}\mathfrak{o}_i, 1), \tag{26}$$

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 $i-1$ rows seen thus far, and $\mathfrak{o}_i^\top$ is the i-th row of the observability matrix $\mathcal{O}(t,0)$. It is worth to highlight that the time and place of the corresponding sensor just received by the i-th row $\mathfrak{o}_i^\top$ are the respective pair $(k,j) = (\lfloor \frac{i}{p} \rfloor, i - p \times \lfloor \frac{i}{p} \rfloor)$, where p is the total number of sensors at each time.

The following lemma shows that this online notation of λ-ridge leverage score is exactly what we need to find the overestimates.

Lemma 1: Online λ-ridge leverage scores $\overline{\tau}_i^\lambda$, overestimate the regular λ-ridge leverage scores τ_i^λ for all $i = 1,\ldots,tp$, i.e.,

$$\overline{\tau}_i^\lambda \geq \tau_i^\lambda. \tag{27}$$

The proof is omitted due to space limitations. Lemma 1 provides us with the overestimates, so we only need to approximate the sum of the overestimate scores to apply Theorem 2 in the online setting. Lemma 2 gives us this approximation and bounds this sum. To prove this lemma, we relied on two pieces of math machinery; the relation between the determinant of a matrix and its rank-one perturbation[5], and what upper bound is for $\det(Q^\top Q)$.

Lemma 2: Let $\mathbf{L} = [\ell_i]$ be a vector of λ-ridge leverage score overestimates obtained by calculating online λ-ridge leverage scores $\overline{\tau}_i^\lambda$ for the rows of the observability matrix, according to Lemma 1, $\overline{\tau}_i^\lambda = \ell_i \geq \mathfrak{o}_i^\top(\mathcal{X}(t,0)+\lambda I)^{-1}\mathfrak{o}_i$ for $i \in [tp]$, then $\|\mathbf{L}\|_1 = \sum_{i\in[tp]} \ell_i \leq 2n\log(1+\|\mathcal{O}(t,0)\|^2/\lambda)$.

Finally, in the last theorem of this paper, we show how we can design a sampling strategy that utilizes the online λ-ridge leverage scores to on-the-fly construct a sensor scheduling S that results in rigorous guarantees on the the quality of (ϵ,δ,d)-approximation of the dynamics (10).

Theorem 3 (Online sampling): Assume dynamics (10), time horizon $t \geq n$, approximation factor $\epsilon \in (0,1)$, additive approximation factor $\delta > 0$, and a fixed positive constant c are given. Then, Algorithm 1 produces scheduling (12) that at most contains on average $(2cn/t\epsilon^2)\cdot\log n\cdot\log(\epsilon\cdot\lambda_{\max}(\mathcal{X}(t,0))/\delta+1)$ active sensors at each time and results in (ϵ,δ,d)-approximation of (10) with probability at least $1-n^{-c/3}$.

Proof: This theorem is just the combination of the results presented in Theorem 2, Lemma 1, and Lemma 2. ∎

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

Input: The stream of rows $\{\mathfrak{o}_i^\top\}_{i=1}^{tp}$ of the observability matrix defined in (20), the approximation factor $\epsilon \in (0,1)$, the additive approximation factor $\delta > 0$, and the fixed positive constant c.

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

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

By performing online sampling, Algorithm 1 avoids pruning routines of [26]. Therefore, it circumvents the dependency issues since the sampling probability of a sensor only depends on earlier sensors in the stream. However, we will lose the chance of getting fewer sensors similar to the streaming setup described in [26]. Additionally, we must add λI to the process to ensure that the sampling probabilities are bounded in the online setting. This modification adds an additive approximation factor δ to the performance guarantee and changes the entire system's observability status. This change can be better observed and explained in terms of *observability ellipsoid*. The controllability/observability ellipsoids are generally used to illustrate which states are easy to control/observe and which ones are harder.

Definition 5 (Observability ellipsoid): An intuitive way

[5] This relation is proposed in [34, Lemma 1.1].

to define the observability ellipsoid is $\{z \mid \|\mathcal{O}(t,0)z\| \leq 1\}$ where $z = x(0) \in \mathbb{R}^n$ is the initial conditions, or the observability ellipsoid is the set of (initial) states that lead to measurements inside a unit ball. One may obtain the equivalent representation as $\{z \mid z^\top \mathcal{X}(t,0)z \leq 1\}$. However, the major axis of this ellipsoid represents the direction that is *most difficult* to observe. To be consistent with the name, we prefer to define the observability ellipsoid as

$$\varepsilon_o = \{z \mid z^\top \mathcal{X}^{-1}(t,0)z \leq 1\}.$$

This definition has the desired interpretation: longer axes correspond to the directions that are *easier* to observe. Since $\mathcal{X}(t,0)$ is a positive definite matrix (it is assumed to be observable), the observability ellipsoid has principal axes corresponding to its eigenvectors, and the length of each principal axis is the square root of the corresponding eigenvalue.

Appending $\sqrt{\lambda}I$ to the observability matrix in Theorem 2 changes the observability Gramian matrix to $\mathcal{X}(t,0) + \lambda I$. Based on the observability ellipsoid's discussion, one can conclude that adding λI to the observability Gramian matrix $\mathcal{X}(t,0)$ increases the area of the ellipsoid and thus the observability because it increases each eigenvalue of the observability Gramian matrix $\mathcal{X}(t,0)$ by λ. One may also infer that it corresponds to *soft-thresholding* the eigenvalues of the Gramian matrix, so that eigenvalues below λ are partially ignored.

IV. Concluding Remarks

In this paper, inspired by a growing body of work on online algorithms for machine learning and big data analysis, we developed an extremely simple framework to on-the-fly randomly select on average a constant number of active sensors to approximate certain observability measures. In the proposed algorithm, each row of the observability matrix of a given large-scale LTV system arrives 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 by assigning a weight to the schedule, 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.

References

[1] A. Chakrabortty and M. D. Ilić, *Control and optimization methods for electric smart grids*. Springer, 2011, vol. 3.

[2] J. Ruths and D. Ruths, "Control profiles of complex networks," *Science*, vol. 343, no. 6177, pp. 1373–1376, 2014.

[3] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035006, 2016.

[4] K. Fitch and N. E. Leonard, "Optimal leader selection for controllability and robustness in multi-agent networks," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1550–1555.

[5] M. Siami and N. Motee, "Network abstraction with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, 2018.

[6] ——, "Growing linear dynamical networks endowed by spectral systemic performance measures," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, 2018.

[7] V. Latora, V. Nicosia, and G. Russo, *Complex networks: principles, methods and applications*. Cambridge University Press, 2017.

[8] Y. Tian, K. Khosoussi, and J. P. How, "A resource-aware approach to collaborative loop-closure detection with provable performance guarantees," *The International Journal of Robotics Research*, vol. 40, no. 10-11, pp. 1212–1233, 2021.

[9] I. Rajapakse, M. Groudine, and M. Mesbahi, "What can systems theory of networks offer to biology?" *PLoS computational biology*, vol. 8, no. 6, p. e1002543, 2012.

[10] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[11] D. Georges, "The use of observability and controllability gramians or functions for optimal sensor and actuator location in finite-dimensional systems," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 4. IEEE, 1995, pp. 3319–3324.

[12] P. Müller and H. Weber, "Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems," *Automatica*, vol. 8, no. 3, pp. 237–246, 1972.

[13] R. Vafaee and M. Siami, "Learning-based sensor selection with guaranteed performance bounds," in 2022 American Control Conference, accepted.

[14] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1686–1701, 2020.

[15] M. Siami and A. Jadbabaie, "A separation theorem for joint sensor and actuator scheduling with guaranteed performance bounds," *Automatica*, vol. 119, p. 109054, 2020.

[16] K. Manohar, J. N. Kutz, and S. L. Brunton, "Optimal sensor and actuator selection using balanced model reduction," *arXiv preprint arXiv:1812.01574*, 2018.

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

[18] V. Tzoumas, "Resilient submodular maximization for control and sensing," Ph.D. dissertation, University of Pennsylvania, 2018.

[19] Y. Dong and J. Huang, "Leader-following connectivity preservation rendezvous of multiple double integrator systems based on position measurement only," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2598–2603, 2014.

[20] K. Fitch and N. E. Leonard, "Joint centrality distinguishes optimal leaders in noisy networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 4, pp. 366–378, 2015.

[21] M. Siami and N. Motee, "Systemic measures for performance and robustness of large–scale interconnected dynamical networks," in *Proceedings of the 53rd IEEE Conference on Decision and Control*, Dec. 2014.

[22] A. W. Marcus, D. A. Spielman, and N. Srivastava, "Interlacing families ii: Mixed characteristic polynomials and the kadison—singer problem," *Annals of Mathematics*, pp. 327–350, 2015.

[23] M. Rudelson and R. Vershynin, "Sampling from large matrices: An approach through geometric functional analysis," *Journal of the ACM (JACM)*, vol. 54, no. 4, pp. 21–es, 2007.

[24] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.

[25] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of computational mathematics*, vol. 12, no. 4, pp. 389–434, 2012.

[26] J. A. Kelner and A. Levin, "Spectral sparsification in the semi-streaming setting," *Theory of Computing Systems*, vol. 53, no. 2, pp. 243–262, 2013.

[27] M. B. Cohen, C. Musco, and C. Musco, "Input sparsity time low-rank approximation via ridge leverage score sampling," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2017, pp. 1758–1777.

[28] M. B. Cohen, C. Musco, and J. Pachocki, "Online row sampling," *arXiv preprint arXiv:1604.05448*, 2016.

[29] R. Kyng, J. Pachocki, R. Peng, and S. Sachdeva, "A framework for analyzing resparsification algorithms," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2017, pp. 2032–2043.

[30] A. El Alaoui and M. W. Mahoney, "Fast randomized kernel methods with statistical guarantees," *stat*, vol. 1050, p. 2, 2014.

[31] M. Kapralov, Y. T. Lee, C. Musco, C. P. Musco, and A. Sidford, "Single pass spectral sparsification in dynamic streams," *SIAM Journal on Computing*, vol. 46, no. 1, pp. 456–477, 2017.

[32] M. Li, G. L. Miller, and R. Peng, "Iterative row sampling," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 2013, pp. 127–136.

[33] A. Alaoui and M. W. Mahoney, "Fast randomized kernel ridge regression with statistical guarantees," *Advances in neural information processing systems*, vol. 28, 2015.

[34] J. Ding and A. Zhou, "Eigenvalues of rank-one updated matrices with some applications," *Applied Mathematics Letters*, vol. 20, no. 12, pp. 1223–1226, 2007.