Pointwise-Sparse Actuator Scheduling for Linear Systems With Controllability Guarantee

Ballotta, Luca; Joseph, Geethu; Rahul Thete, Irawati

DOI
10.1109/LCSYS.2024.3475886

Publication date
2024

Document Version
Final published version

Published in
IEEE Control Systems Letters

Citation (APA)
Ballotta, L., Joseph, G., & Rahul Thete, I. (2024). Pointwise-Sparse Actuator Scheduling for Linear Systems With Controllability Guarantee. *IEEE Control Systems Letters*, *8*, 2361-2366.
https://doi.org/10.1109/LCSYS.2024.3475886

IEEE CONTROL SYSTEMS LETTERS, VOL. 8, 2024

Pointwise-Sparse Actuator Scheduling for Linear Systems With Controllability Guarantee

Luca Ballotta, Geethu Joseph, *Member, IEEE*, and Irawati Rahul Thete

***Abstract*—This letter considers the design of sparse actuator schedules for linear time-invariant systems. An actuator schedule selects, for each time instant, which control inputs act on the system in that instant. We address the optimal scheduling of control inputs under a hard constraint on the number of inputs that can be used at each time. For a sparsely controllable system, we characterize sparse actuator schedules that make the system controllable, and then devise a greedy selection algorithm that guarantees controllability while heuristically providing low control effort. We further show how to enhance our greedy algorithm via Markov chain Monte Carlo-based randomized optimization.**

***Index Terms*—Actuator scheduling, control design, energy-aware control, greedy algorithm, sparse control.**

I. Introduction

SPARSITY constraints in control inputs arise in several large-scale systems, such as networked control with limited bandwidth [1], budget-limited influence in marketing strategies [2], sparse damping control in power grids [3], and drug treatment in biological networks [4]. Motivated by these applications, we address the challenge of designing control with few actuators without significantly compromising performance.

Two important sparse control paradigms are maximum hands-off control, which reduces active (nonzero) control periods [5], and sparse feedback, which reduces the nonzero elements or rows in the feedback gain matrix [6], [7]. Some works have investigated sparsity directly in actuator use. An early approach selects a few inputs, which remain constant over time, to ensure controllability, tightly constraining the system [8]. Recent work minimizes the average number of inputs over a time horizon, allowing for time-varying inputs [3]. However, this strategy can lead to non-sparse inputs at certain times, which is unsuitable for applications like networked systems where bandwidth constraints must be satisfied at all times. Alternatively, this letter limits the number of control inputs active at every time [9], referred to as *sparse actuator control*.

The theoretical foundation of sparse actuator control has been studied in [9], [10], [11], [12], but sparse control input design is not well studied in the literature. A straightforward method uses sparse recovery algorithms to drive the system from a given state to a desired state [12], which requires distinct designs for state deviations. The standard, naive greedy algorithm identifies a sparse actuator schedule to transition the system from any initial state to any desired state [13]. Due to its heuristic nature, it does not assure controllability for all systems that are controllable under sparsity constraints. We address this literature gap and propose efficient algorithms for the design of sparse control inputs that formally guarantee controllability.

Contribution: We first show that the naive greedy algorithm may fail to ensure controllability (Section III-A), which motivates our study. Then, we analytically characterize sparse actuator schedules that make the system controllable (Section III-B) and devise an improved greedy algorithm that searches among such schedules (Section III-C), thus guaranteeing controllability if the system is controllable under the sparsity constraint. Also, we propose to improve the algorithm using a Markov chain Monte Carlo (MCMC)-based approach, albeit with increased computational complexity (Section III-D). Finally, in Section IV, we numerically study the performance of our greedy algorithm on two use cases where the naive greedy algorithm fails, as well as the improvement provided by MCMC.

II. Sparse Actuator Scheduling Problem

Consider the discrete-time linear dynamical system $(\boldsymbol{A}, \boldsymbol{B})$ with matrices $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ and state evolution

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

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ and $\boldsymbol{u}(k) \in \mathbb{R}^m$ respectively denote state and control input at time k. The system $(\boldsymbol{A}, \boldsymbol{B})$ is s-sparse controllable if it is controllable when $\|\boldsymbol{u}(k)\|_0 \leq s\ \forall k \geq 0$, *i.e.,* when at most s input channels are active (nonzero) at all times, where a channel is an element of $\boldsymbol{u}(k)$. We aim to design sparse inputs to drive the system from any initial state to a desired one. We first review the existing literature.

Received 16 July 2024; revised 13 September 2024; accepted 4 October 2024. Date of publication 7 October 2024; date of current version 18 October 2024. This work was supported in part by the EU Horizon Program through the Project TWAIN under Grant 101122194. Recommended by Senior Editor S. Tarbouriech. *(Corresponding author: Luca Ballotta.)*

Luca Ballotta is with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: l.ballotta@tudelft.nl).

Geethu Joseph is with the Signal Processing Systems Group, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: g.joseph@tudelft.nl).

Irawati Rahul Thete was with the Signal Processing Systems Group, Delft University of Technology, 2628 CD Delft, The Netherlands. She is now with the APP-IPP-MPP, Texas Instruments, Bengaluru 560093, India (e-mail: irawati.thete@gmail.com).

Digital Object Identifier 10.1109/LCSYS.2024.3475886

TABLE I
COMMON ENERGY-AWARE CONTROL PERFORMANCE METRICS [15]

$\frac{1}{n}\,\mathrm{Tr}(\boldsymbol{W}^{-1})$	average energy to reach a unit-norm state
$(\lambda_{\min}(\boldsymbol{W}))^{-1}$	maximal energy to reach a unit-norm state
$\sqrt[n]{\det(\boldsymbol{W})}$	volume of ellipsoid reached with unit energy

A. Preliminaries on Sparse Controllability

A simple test for s-sparse controllability is as follows, where $\mathrm{rk}\{\boldsymbol{A}\}$ denotes the rank of matrix $\boldsymbol{A}$.

Proposition 1 [9, Th. 1]: System (1) is s-sparse controllable if it is controllable and $s \geq \max\{n - \mathrm{rk}\{\boldsymbol{A}\}, 1\}$.

The next result ensures s-sparse controllability with fixed active input channels, following an *actuator schedule*, regardless of the initial and desired states.

Definition 1: Let $[m] \doteq \{1, \ldots, m\}$ with $[0] \doteq \emptyset$. An *actuator schedule* $\mathcal{S}$ on m inputs with horizon h is an ordered tuple

$$\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_{h-1})\colon \mathcal{S}_k \subseteq [m]\ \forall k = 0, \ldots, h-1. \quad (2)$$

We denote the set of all actuator schedules by $\mathcal{T}_m^h$.

In words, the ith input channel $\boldsymbol{u}_i(k)$ is active at time k only if $i \in \mathcal{S}_k$. Also, for given sparsity $s \in [m]$, an *s-sparse actuator schedule* fulfills the condition $|\mathcal{S}_k| \leq s$ for each time k, meaning that at most s inputs are active at every time.

Under he actuator schedule $\mathcal{S}$, the actual state evolution is

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}_{\mathcal{S}_k}\boldsymbol{u}_{\mathcal{S}_k}(k), \quad k = 0, \ldots, h-1, \quad (3)$$

where $\boldsymbol{B}_{\mathcal{I}}$ denotes the submatrix of $\boldsymbol{B}$ composed by columns with index in set $\mathcal{I}$ and $\boldsymbol{u}_{\mathcal{S}_k}(k)$ stacks the elements of $\boldsymbol{u}(k)$ with index in $\mathcal{I}$. We define the *$\mathcal{S}$-reachability matrix* as

$$\boldsymbol{\Phi}_{\mathcal{S}}^h \doteq \left[\boldsymbol{A}^{h-1}\boldsymbol{B}_{\mathcal{S}_0}\ \boldsymbol{A}^{h-2}\boldsymbol{B}_{\mathcal{S}_1}\ \ldots\ \boldsymbol{B}_{\mathcal{S}_{h-1}}\right]. \quad (4)$$

Proposition 2 [14, Ths. 2.4 and 2.13]: If system (1) is s-sparse controllable and $h \geq h^*$, where

$$\frac{n}{\min\{\mathrm{rk}\{\boldsymbol{B}\}, s\}} \leq h^* \leq n - \min\{\mathrm{rk}\{\boldsymbol{B}\}, s\} + 1, \quad (5)$$

there exists an s-sparse schedule $\mathcal{S} \in \mathcal{T}_m^h$ that makes system (3) s-sparse controllable, *i.e.,* for which $\mathrm{rk}\{\boldsymbol{\Phi}_{\mathcal{S}}^h\} = n$.

Using the above insights, we formulate our problem next.

B. Sparse Control Design Problem Formulation

In light of Proposition 2, we aim to find an s-sparse actuator schedule $\mathcal{S}$ that makes the resulting system (3) controllable.

To measure the control effort of a schedule $\mathcal{S}$, we consider the *$\mathcal{S}$-controllability Gramian* $\boldsymbol{W}_{\mathcal{S}}^h = (\boldsymbol{\Phi}_{\mathcal{S}}^h)^\top \boldsymbol{\Phi}_{\mathcal{S}}^h$, which is used in the literature to quantify the control energy required to reach a target state through (3) in h steps [15]. We denote the control metric by ρ and list common energy-aware performance metrics in Table I. Therefore, our design problem is as follows.

Problem 1 (Optimal Sparse Actuator Schedule): Given system (1), a time horizon h, a sparsity level s, and a cost function ρ, find an s-sparse schedule $\mathcal{S}^*$ as

$$\mathcal{S}^* \in \arg\min_{\mathcal{S}\in\mathcal{T}_m^h} \ \rho\left(\boldsymbol{W}_{\mathcal{S}}^h\right) \quad (6a)$$

$$\text{s.t } |\mathcal{S}_k| \leq s \quad \forall k \in \{0, \ldots, h-1\} \quad (6b)$$

$$\mathrm{rk}\left\{\boldsymbol{W}_{\mathcal{S}}^h\right\} = n. \quad (6c)$$

In problem (6), the combinatorial constraint (6b) makes the schedule (*i.e.,* the control inputs) point-wise s-sparse, and the rank constraint (6c) ensures that the chosen schedule makes the system s-sparse controllable. Propositions 1 and 2 imply that problem (6) is feasible if the following conditions hold.

Assumption 1: The sparsity level and the time horizon respectively satisfy $s \geq \max\{n - \mathrm{rk}\{\boldsymbol{A}\}, 1\}$ and $h \geq h^*$.

In the following, we assume that Assumption 1 holds and present design algorithms with polynomial complexity to solve Problem 1 ensuring the constraints (6b)–(6c) are met.

III. DESIGN ALGORITHMS

This section tackles Problem 1 via algorithms with tractable (polynomial) computational complexity. First, we explain why the naive greedy algorithm may fail to meet the constraints in (6). In Section III-B we characterize sparse schedules that ensure controllability and, building on the analysis, in Section III-C we devise our main design algorithm with formal controllability guarantees. Finally, in Section III-D we propose a stochastic optimization approach to improve performance.

A. Drawback of the Naive Greedy Algorithm

A classic approach to reduce cost (6a) subject to a budget constraint on the inputs is greedy selection, which we refer to as a naive greedy algorithm [13], [15]. However, imposing the point-wise s-sparsity constraint in (6b) to the greedy selection may yield an uncontrollable system, as in the example below.

Example 1: We choose a system with $n = 5$, $m = 7$ where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (7)$$

The system is s-sparse controllable with $s = n - \mathrm{rk}\{\boldsymbol{A}\} = 1$. We choose time horizon $h = n = 5$ and cost function $\rho(\cdot) = \mathrm{Tr}\big((\cdot)^{-1}\big)$. To avoid singular matrix inversion, the cost is computed as $\mathrm{Tr}((\boldsymbol{W}_{\mathcal{S}}^h)^{-1} + \epsilon \boldsymbol{I})$ with a small slack $\epsilon > 0$ [3], [13]. In the first iteration, the last column of the reachability matrix $\boldsymbol{\Phi}^h$ gives the lowest cost, so the algorithm selects $\mathcal{S}_4 = 7$. Due to the sparsity constraint, subsequent iterations restrict the greedy search to the first 28 columns of $\boldsymbol{\Phi}^h$. However, the last row of these columns is entirely zero. Thus, the last row of $\boldsymbol{\Phi}_{\mathcal{S}}^h$ has only zeros, yielding an uncontrollable system and failure of the naive greedy algorithm.

In short, the naive greedy algorithm need not output a feasible solution of the control design problem (6). We next examine the feasible set of (6) to guide a greedy selection.

B. Characterizing Feasible Actuator Schedules

We start with a necessary condition for feasibility.

Lemma 1: Let $\mathcal{S}$ be a feasible solution of Problem 1 satisfying Assumption 1. Then, for $k = 1, \ldots, h-1$, it holds

$$\mathrm{rk}\left\{\left[\boldsymbol{A}^k \quad \boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k}} \quad \cdots \quad \boldsymbol{B}_{\mathcal{S}_{h-1}}\right]\right\} = n. \quad (8)$$

Proof: From the rank condition, we deduce that

$$
\begin{aligned}
n &= \mathrm{rk}\left\{\left[\boldsymbol{A}^{h-1}\boldsymbol{B}_{\mathcal{S}_0} \quad \dots \quad \boldsymbol{B}_{\mathcal{S}_{h-1}}\right]\right\}\\
&\leq \mathrm{rk}\left\{\left[\boldsymbol{A}^{k} \;\; \boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k}} \quad \dots \quad \boldsymbol{B}_{\mathcal{S}_{h-1}}\right]\right\} \leq n,
\end{aligned}
\tag{9}
$$

and the equality (8) immediately follows by comparison. ■

From the relation (8) with $k = 1$, we derive

$$
\mathbb{R}^n = \mathrm{Col}\left\{\left[\boldsymbol{A}\ \boldsymbol{B}_{\mathcal{S}_{h-1}}\right]\right\} = \mathrm{Col}\{\boldsymbol{A}\} \oplus \mathrm{Col}\left\{\boldsymbol{B}_{\mathcal{S}_{h-1}}\right\}, \tag{10}
$$

where $\mathrm{Col}\{\cdot\}$ denotes the column space and $\oplus$ denotes the sum of subspaces. As $\mathbb{R}^n = \mathrm{Col}\{\boldsymbol{A}\} \oplus \ker\{\boldsymbol{A}^\top\} = \mathrm{Col}\{\boldsymbol{A}\} \oplus \mathrm{Col}\left\{\boldsymbol{B}_{\mathcal{S}_{h-1}}\right\}$, where $\ker\{\cdot\}$ denotes the null space, we get

$$
\ker\left\{\boldsymbol{A}^\top\right\} = \mathrm{proj}_{\ker\{\boldsymbol{A}^\top\}}\left\{\mathrm{Col}\left\{\boldsymbol{B}_{\mathcal{S}_{h-1}}\right\}\right\}, \tag{11}
$$

where $\mathrm{proj}_{\mathcal{C}}\{\mathcal{D}\}$ denotes the projection of the (sub)space $\mathcal{D}$ onto the subspace $\mathcal{C}$. For $k = 2$, we write the relation (8) as

$$
\begin{aligned}
\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\} &= \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\}}\left\{\mathrm{Col}\left\{\boldsymbol{A}\boldsymbol{B}_{\mathcal{S}_{h-2}}\right\}\right\}\\
&\quad \oplus \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\}}\left\{\mathrm{Col}\left\{\boldsymbol{B}_{\mathcal{S}_{h-1}}\right\}\right\}.
\end{aligned}
\tag{12}
$$

Since $\ker\{\boldsymbol{A}^\top\} \subseteq \ker\{(\boldsymbol{A}^2)^\top\}$, (11) and (12) jointly yield

$$
\begin{aligned}
&\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\} \ominus \ker\left\{\boldsymbol{A}^\top\right\}\\
&\quad = \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\} \ominus \ker\left\{\boldsymbol{A}^\top\right\}}\left\{\mathrm{Col}\left\{\boldsymbol{A}\boldsymbol{B}_{\mathcal{S}_{h-2}}\right\}\right\}\\
&\qquad \oplus \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\} \ominus \ker\left\{\boldsymbol{A}^\top\right\}}\left\{\mathrm{Col}\left\{\boldsymbol{B}_{\mathcal{S}_{h-1}}\right\}\right\},
\end{aligned}
\tag{13}
$$

where for any two subspaces $\mathcal{C} \subseteq \mathcal{D}$, $\mathcal{D} \ominus \mathcal{C} \subseteq \mathcal{D}$ denotes the orthogonal complement to $\mathcal{C}$ in $\mathcal{D}$. The dimension of the subspace $\ker\{(\boldsymbol{A}^2)^\top\} \ominus \ker\{\boldsymbol{A}^\top\}$ is the difference between the dimensions of $\ker\{(\boldsymbol{A}^2)^\top\}$ and $\ker\{\boldsymbol{A}^\top\}$, given by

$$
\left(n - \mathrm{rk}\left\{\boldsymbol{A}^2\right\}\right) - (n - \mathrm{rk}\{\boldsymbol{A}\}) \leq n - \mathrm{rk}\{\boldsymbol{A}\} \leq s \tag{14}
$$

where we use the Sylvester rank inequality. Hence, regardless of the channels in $\mathcal{S}_{h-1}$, we can find a set $\mathcal{S}_{h-2}$ such that

$$
\begin{aligned}
&\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\} \ominus \ker\left\{\boldsymbol{A}^\top\right\}\\
&\quad = \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^2\right)^\top\right\} \ominus \ker\left\{\boldsymbol{A}^\top\right\}}\left\{\mathrm{Col}\left\{\boldsymbol{A}\boldsymbol{B}_{\mathcal{S}_{h-2}}\right\}\right\}.
\end{aligned}
\tag{15}
$$

Extending the same idea, we obtain sufficient conditions for controllability. To this aim, we define

$$
R \doteq \min\{k \geq 0 : \ \mathrm{rk}\{\boldsymbol{A}^k\} = \mathrm{rk}\{\boldsymbol{A}^{k+1}\}\}. \tag{16}
$$

Lemma 2: Let s and h satisfy Assumption 1. Then, for all $k \in [R]$, there exists an index set $\mathcal{S}_{h-k}$ such that

$$
\begin{aligned}
&\ker\left\{\left(\boldsymbol{A}^k\right)^\top\right\} \ominus \ker\left\{\left(\boldsymbol{A}^{k-1}\right)^\top\right\}\\
&\quad = \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^k\right)^\top\right\} \ominus \ker\left\{\left(\boldsymbol{A}^{k-1}\right)^\top\right\}}\left\{\mathrm{Col}\left\{\boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k}}\right\}\right\}.
\end{aligned}
\tag{17}
$$

Further, if $R > 0$, the sets satisfy

$$
\begin{aligned}
\ker\left\{\left(\boldsymbol{A}^R\right)^\top\right\} &= \mathrm{proj}_{\ker\left\{\left(\boldsymbol{A}^R\right)^\top\right\}}\left\{\mathrm{Col}\left\{\left[\boldsymbol{A}^{R-1}\boldsymbol{B}_{\mathcal{S}_{h-R}}\right.\right.\right.\\
&\qquad \left.\left.\left.\boldsymbol{A}^{R-2}\boldsymbol{B}_{\mathcal{S}_{h-R+1}} \quad \cdots \quad \boldsymbol{B}_{\mathcal{S}_{h-1}}\right]\right\}\right\}.
\end{aligned}
\tag{18}
$$

Algorithm 1: s-Sparse Greedy Selection

Input: Matrices $\boldsymbol{A}, \boldsymbol{B}$, sparsity s, horizon h, cost ρ.
Output: Schedule $\mathcal{S}$.

1 $\mathcal{S} \leftarrow [[m]] * h$; // initialize full schedule
2 $\mathtt{rk}_W \leftarrow 0$; // rank of controllability Gramian
3 $\boldsymbol{C} \leftarrow [\,]$; // column space of controllability Gramian
4 $\mathtt{cand} \leftarrow [\emptyset] * h$; // channels to be selected later
5 **for** $k = 1, \dots, h-1$ **do** // input at $k=0$ is unconstrained
6   $\mathcal{S}_k \leftarrow \emptyset$; // select necessary channels at time k
7   **if** $\mathtt{rk}_W < n$ **then**
8     $\mathcal{K}_k \leftarrow \ker\{(\boldsymbol{A}^{h-k})^\top\} \ominus \ker\{(\boldsymbol{A}^{h-1-k})^\top\}$;
9     $\mathcal{S}_k \leftarrow \mathtt{greedy\_k}(\{j : \boldsymbol{B}_j \not\perp \mathcal{K}_k\}, k, \mathcal{S}, \epsilon)$;
10     $\mathtt{rk}_W \leftarrow \mathtt{rk}_W + |\mathcal{S}_k|$;
11     $\boldsymbol{C} \leftarrow [\boldsymbol{A}^{h-1-k}\boldsymbol{B}_{\mathcal{S}_k}\ \boldsymbol{C}]$;
12   **if** $|\mathcal{S}_k| < s$ **then**
13     $\mathtt{cand}[k] \leftarrow [m] \setminus \mathcal{S}_k$;
14 $\mathcal{S}_0 \leftarrow \emptyset$; $\mathtt{cand}[0] \leftarrow [m]$; // reset schedule at time 0
15 **for** $i = 1, \dots, n - \mathtt{rk}_W$ **do** // ensure controllability
16   $\mathtt{cand}, \mathcal{S} \leftarrow \mathtt{greedy}(\mathtt{cand}, \mathcal{S}, \epsilon, \mathtt{rk})$;
17 **while** $\exists k : |\mathcal{S}_k| < s$ **do** // decrease cost
18   $\mathtt{cand}, \mathcal{S} \leftarrow \mathtt{greedy}(\mathtt{cand}, \mathcal{S}, 0)$;
19   **if** *$\mathcal{S}$ doesn't change* **then**
20   **break**;
21 **return** $\mathcal{S}$.

Proof: See Appendix A. ■

If $\boldsymbol{A}$ is invertible, then $R = 0$ and the result is trivial. Lemma 2 leads to a sufficient condition for feasibility, as established by the following result.

Theorem 1: Consider system (1) and sparsity level s and time horizon h satisfying Assumption 1. Then, there exists a feasible solution $\mathcal{S}$ of Problem 1 that satisfies (17) for $k \in [R]$.

Proof: See Appendix B. ■

Based on Theorem 1, we devise an improved greedy algorithm to find a schedule with formal controllability guarantee.

C. Improved s-Sparse Greedy Algorithm

We outline our selection procedure in Algorithm 1. The schedule is initialized full, *i.e.,* all input channels scheduled at all times (Line 1). This allows us to reduce the risk of dealing with singular $\mathcal{S}$-controllability Gramian and thus to compute the cost function more robustly in the first phase of the algorithm, which ensures that all channels that are necessary for controllability are scheduled, This phase is implemented in the "for" loop at Line 5. Specifically, for each time step $k > 1$, Line 8 computes the subspace $\mathcal{K}_k$ that has to be spanned by input channels active at time k, and Line 9 greedily selects the best channels among the candidates. For example, at time $k = h-1$, $\mathcal{K}_{h-1} = \ker\{\boldsymbol{A}^\top\}$ and Line 9 populates $\mathcal{S}_{h-1}$

Algorithm 2: Subroutine `greedy_k`

Input: Channels $\texttt{cand}$, step k, schedule $\mathcal{S}$, $\epsilon \geq 0$.
Output: Updated schedule $\mathcal{S}_k$ at time step k.

1 **while** $\mathit{cand} \neq \emptyset$ **do**
2   $c^* \leftarrow \arg\min_{c \in \texttt{cand}} \rho(\boldsymbol{W}_{\mathcal{S}}^h + \boldsymbol{\phi}_c^k + \epsilon \boldsymbol{I})$;
3   $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup \{c^*\}$; $\texttt{cand} \leftarrow \texttt{cand} \setminus \{c^*\}$;
4   **foreach** $c \in \mathit{cand}$ **do**
5   **if** $\mathrm{proj}_{\mathcal{K}_k}\{\boldsymbol{A}^{h-1-k}\boldsymbol{B}_c\} \parallel \mathrm{proj}_{\mathcal{K}_k}\{\boldsymbol{A}^{h-1-k}\boldsymbol{B}_{\mathcal{S}_k}\}$ **then**
6       $\texttt{cand} \leftarrow \texttt{cand} \setminus \{c\}$;
7 **return** $\mathcal{S}_k$.

Algorithm 3: Subroutine `greedy`

Input: Channels $\texttt{cand}$, schedule $\mathcal{S}$, $\epsilon \geq 0$, flag $\texttt{rk}$.
Output: Updated $\texttt{cand}$, $\mathcal{S}$.

1 $k^*, c^* \leftarrow \arg\min_{k:|\mathcal{S}_k|<s, c \in \texttt{cand}[k]} \rho(\boldsymbol{W}_{\mathcal{S}}^h + \boldsymbol{\phi}_c^k + \epsilon \boldsymbol{I})$;
2 **if** *not* rk *and* $\rho(\boldsymbol{W}_{\mathcal{S}}^h + \boldsymbol{\phi}_{c^*}^{k^*} + \epsilon \boldsymbol{I}) \geq \rho(\boldsymbol{W}_{\mathcal{S}}^h + \epsilon \boldsymbol{I})$ **then**
3   **return** $\texttt{cand}$, $\mathcal{S}$;
4 $\mathcal{S}_{k^*} \leftarrow \mathcal{S}_{k^*} \cup \{c^*\}$; $\texttt{cand}[k^*] \leftarrow \texttt{cand}[k^*] \setminus \{c^*\}$;
5 **if** rk **then**
6   $\boldsymbol{C} \leftarrow [\boldsymbol{A}^{h-1-k^*}\boldsymbol{B}_{c^*}\ \boldsymbol{C}]$;
7   **foreach** $k : |\mathcal{S}_k| < s,\ c \in \mathit{cand}[k]$ **do**
8   **if** $\mathrm{rk}\{[\boldsymbol{A}^{h-1-k}\boldsymbol{B}_c\ \boldsymbol{C}]\} = \mathrm{rk}\{\boldsymbol{C}\}$ **then**
9       $\texttt{cand}[k] \leftarrow \texttt{cand}[k] \setminus \{c\}$;
10 **return** $\texttt{cand}$, $\mathcal{S}$.

until condition (11) is met. The feasibility of this selection is formally supported by Theorem 1, which states that one can find indices $\mathcal{S}_k$ such that $\boldsymbol{B}_{\mathcal{S}_k}$ spans $\mathcal{K}_k$ for every time k.

The selection at Line 9 of Algorithm 1 is executed by the subroutine `greedy_k` outlined in Algorithm 2, where we define the "contribution" to the controllability Gramian of channel c scheduled at time k as $\boldsymbol{\phi}_c^k \doteq (\boldsymbol{A}^{h-1-k}\boldsymbol{B}_c)^\top \boldsymbol{A}^{h-1-k}\boldsymbol{B}_c$. Here, after a channel is selected (Line 3), the candidate channels that are not independent of those already selected are removed (Lines 5, and 6), guaranteeing that the schedule $\mathcal{S}_k$ spans one more direction of $\mathcal{K}_k$ after each iteration of the "while" loop of Algorithm 2 and eventually spans all of $\mathcal{K}_k$ so as to fulfill (8). By the initialization of $\mathcal{S}$, the parameter ϵ in Algorithm 2 can be zero if $\mathcal{K}_k$ has dimension one, but it must be positive otherwise to avoid a singular matrix in ρ.

After including all necessary input channels to span the kernels of $\boldsymbol{A}, \boldsymbol{A}^2, \dots, \boldsymbol{A}^R$ so as to fulfill Theorem 1, Algorithm 1 ensures controllability through the selected schedule with the "for" loop at Line 15. This is achieved via the subroutine `greedy` in Algorithm 3, which is the classic greedy selection but with the option to check the rank of the controllability Gramian through the flag `rk`. When `rk` is raised (Line 15), all candidate channels that cannot increase the rank of the controllability Gramian accrued so far are removed at Lines 8, and Line 9. This ensures that the "for" loop at Line 15 of Algorithm 1 eventually makes the $\mathcal{S}$-controllability Gramian full rank.

Finally, the last "while" loop at Line 17 adds channels till the cost cannot be further decreased or the selected schedule

Algorithm 4: s-Sparse MCMC

Input: Matrices $\boldsymbol{A}, \boldsymbol{B}$, sparsity s, horizon h, cost ρ, parameters $\epsilon \geq 0$, $T_{\mathrm{in}} > 0$, $T_{\min} > 0$, $\alpha \in (0, 1)$, $\texttt{it}$, initial schedule $\mathcal{S}_0$.
Output: Schedule $\mathcal{S}$.

1 $\mathcal{S} \leftarrow \mathcal{S}_0$;
2 $\rho_{\min} \leftarrow \rho(\boldsymbol{W}_{\mathcal{S}}^h + \epsilon \boldsymbol{I})$;
3 $\texttt{rk}_W \leftarrow \mathrm{rk}\{\boldsymbol{W}_{\mathcal{S}}^h\}$; `// optional, to check controllability`
4 $T \leftarrow T_{\mathrm{in}}$;
5 **while** $T > T_{min}$ **do**
6   **for** $\mathit{i} = 1, \dots, \mathit{it}$ **do**
7     $\mathcal{S}' \leftarrow \mathcal{S}$;
8     $k, c_k \sim U(\mathcal{S})$; `// sample from current schedule`
9     $c_k' \sim U([m] \setminus \mathcal{S}_k)$; `// sample candidate channel`
10     $\mathcal{S}_k' \leftarrow (\mathcal{S}_k \setminus \{c_k\}) \cup \{c_k'\}$;
11   **if** *check rank* **then**
12   **while** $\mathrm{rk}\{\boldsymbol{W}_{\mathcal{S}'}^h\} < \mathit{rk}_W$ **do**
13   **go to** Line 8 or **continue**;
14     $\rho_{\mathrm{curr}} \leftarrow \rho(\boldsymbol{W}_{\mathcal{S}'}^h + \epsilon \boldsymbol{I})$;
15     $p \leftarrow \mathrm{e}^{-\frac{1}{T}(\rho_{\mathrm{curr}} - \rho_{\min})}$;
16   **if** $p > n \sim U([0, 1])$ **then** `// w.p.` $\min\{1, p\}$
17       $\mathcal{S}_k \leftarrow \mathcal{S}_k'$;
18       $\rho_{\min} \leftarrow \rho_{\mathrm{curr}}$;
19   **if** *check rank* **then**
20         $\texttt{rk}_W \leftarrow \mathrm{rk}\{\boldsymbol{W}_{\mathcal{S}'}^h\}$;
21   $T \leftarrow \alpha T$;
22 **return** $\mathcal{S}$.

has reached the s-sparsity limit.[1] Notably, this last selection need not care about the rank of the controllability Gramian, and the cost can be computed exactly by setting $\epsilon = 0$.

Assuming that computing the cost $\rho(\boldsymbol{W}_{\mathcal{S}}^h)$ requires $O(n^\beta)$ operations, the computational complexity of Algorithm 1 is $O(hm \max\{n^{\beta+1}, n^4\})$, the same of the naive greedy. The detailed analysis is provided in the technical report [16].

D. s-Sparse Markov Chain Monte Carlo

MCMC is a randomized algorithm that approximately solves combinatorial problems by sampling a Markov chain supported on the domain of the optimization variable. We summarize its workflow in Algorithm 4. Every sample $\mathcal{S}'$ replaces the current solution $\mathcal{S}$ with a probability that decreases exponentially with the cost gap (Line 16). MCMC is asymptotically optimal with infinite samples, *i.e.,* as the parameter T goes to zero and the iterations `it` go to infinity. In practice, T is progressively reduced down to $T_{\min} > 0$ (Line 5) and finite samples are drawn for each value of T (Line 6), causing approximation errors and a suboptimality gap that decreases with `it` and $T_{\min}^{-1}$ [17]. The interested readers are referred to [18] for details. We adapt

[1]Because the greedy selection is suboptimal, if the cost cannot be decreased in one iteration, one can randomly select channels to possibly drop the cost after multiple iterations. We do not explore this possibility in the current letter and leave a more comprehensive numerical evaluation for future work.

TABLE II
SCHEDULES AND COSTS FOR EXAMPLE 1

	$\mathcal{S}_0$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathrm{Tr}\left((\boldsymbol{W}_{\mathcal{S}}^h)^{-1}\right)$
fully actuated	[7]	[7]	[7]	[7]	[7]	1.7
naive greedy	1	4	1	2	7	uncontrollable
s-sparse greedy	1	4	4	4	4	5.0

MCMC to Problem 1 using the output of Algorithm 1 as warm start $\mathcal{S}_0$. To enforce s-sparsity, we restrict candidate samples $\mathcal{S}'$ to differ from the current solution $\mathcal{S}$ by only one channel in one time step k (Lines 8, and 9).

Although MCMC has polynomial complexity with respect to its parameters α, T_{in}, $T_{\min}$, and $\texttt{it}$, it generally converges slow in practice and requires a huge number of samples to achieve good solutions for large systems. Moreover, its randomized nature prevents formal controllability guarantees and the output schedule may violate constraint (6c). This issue can be overcome by imposing that candidate samples $\mathcal{S}'$ do not decrease the rank of the controllability Gramian (Line 11), which however may significantly slow down the algorithm.

IV. Numerical Experiments

We test our algorithms on two problem instances.[2]

A. Revisited Example 1

We use Algorithm 1 to solve the design of Example 1. The output schedule and cost are reported in Table II, together with the schedule output by the naive greedy algorithm and the cost of the fully actuated system for the sake of comparison. Our proposed s-sparse greedy algorithm picks the key input channel 4 that guarantees controllability according to the necessary condition (8) for all time steps $k = 1, 2, 3, 4$, thanks to the smart pre-selection at Lines 8, and 9 of Algorithm 1.

B. Experiment With Large-Scale Network

We benchmark Algorithm 1 on a large-scale system with $n = m = 50$, where $\boldsymbol{A}$ is the adjacency matrix of a random geometric graph scaled by the number of nodes and $\boldsymbol{B}$ is the identity matrix. This setup mimics a network system where node (subsystem) i can be directly controlled through the ith column of $\boldsymbol{B}$. We generate sparse random geometric graphs with nodes in the unit square $[0, 1]^2$ and radius 0.1, and choose time horizon $h = n = 50$. The nullity of $\boldsymbol{A}$ ranges from 10 to 20 and $\ker\{\boldsymbol{A}\} = \ker\{\boldsymbol{A}^k\}\ \forall k \geq 1$, meaning that $R = 1$ and the critical time step for controllability is only $k = h - 1$ based on Lemma 2. The naive greedy algorithm always yields uncontrollable systems when choosing the minimal admissible sparsity $s = n - \mathrm{rk}\{\boldsymbol{A}\}$, even when refining its output with s-sparse MCMC. In contrast, our s-sparse greedy algorithm successfully yields controllable systems on all tried instances.

Finally, we compare the costs achieved with the s-sparse greedy selection (Algorithm 1) and s-sparse MCMC

[2]The code used for experiments is available at https://github.com/lucaballotta/sparse-control

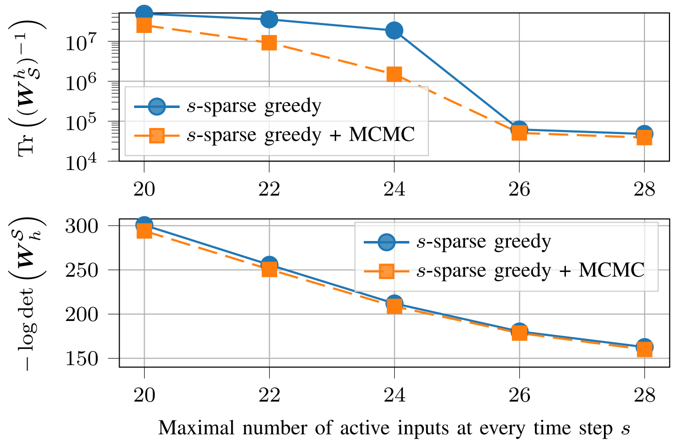

Fig. 1. Cost for experiment on large-scale network. The parameters of s-sparse MCMC are $T_{\text{in}} = 1$, $T_{\min} = 10^{-7}$, $\alpha = 0.1$, $\texttt{it} = 5000$.

(Algorithm 4) as the sparsity constraint s varies. Figure 1 illustrates the results for a system with $n - \mathrm{rk}\{\boldsymbol{A}\} = 20$ where s increases from 20 (the minimum for s-sparse controllability) to 28. Both curves are decreasing, meaning that selecting more inputs reduces the control effort. Also, while MCMC yields smaller costs thanks to its exploratory approach, the costs obtained with the s-sparse greedy are only slightly larger in most cases, especially with $\rho(\cdot) = -\log\det(\cdot)$, validating its effectiveness.

V. Conclusion

We investigated the sparse actuator scheduling problem for linear systems to ensure controllability with minimal control effort. We characterized feasible actuator schedules, designed a greedy algorithm with formal controllability guarantee, and enhanced it using MCMC-based randomized optimization. Numerical tests prove our algorithms effective even when the naive greedy selection fails. Future work could explore stricter sparsity constraints, such as slowly varying active input sets.

Appendix A
Proof of Lemma 2

We define $\boldsymbol{K}^{(k)} = \boldsymbol{I} - \boldsymbol{A}^k(\boldsymbol{A}^k)^\dagger$ as the projection matrix onto $\ker\{(\boldsymbol{A}^k)^\top\}$, where $(\cdot)^\dagger$ is the pseudoinverse. As the subspace orthogonal to $\ker\{(\boldsymbol{A}^k)^\top\}$ is $\mathrm{Col}\{\boldsymbol{A}^k\}$, we get

$$\ker\left\{\left(\boldsymbol{A}^k\right)^\top\right\} \ominus \ker\left\{\left(\boldsymbol{A}^{k-1}\right)^\top\right\} = \mathrm{Col}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\right\}. \quad (19)$$

We note that (17) trivially holds if $R = 0$ by setting $k = R + 1$. Then, to prove (17) for all $k \in [R]$ it suffices to prove that there exists an index set $\mathcal{S}_{h-k}$ for each $k \in [R]$ such that

$$\mathrm{Col}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\right\} = \mathrm{Col}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k}}\right\}. \quad (20)$$

Since the system is controllable, it holds

$$\begin{aligned}\mathrm{Col}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\right\} &= \mathrm{Col}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\boldsymbol{\Phi}^n\right\}\\ &= \mathrm{Col}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\boldsymbol{B}\right\}, \quad (21)\end{aligned}$$

because $\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\boldsymbol{A}^{i} = [\boldsymbol{I} - \boldsymbol{A}^{k}(\boldsymbol{A}^{k})^{\dagger}]\boldsymbol{A}^{k}\boldsymbol{A}^{i-1} = \boldsymbol{0}$, for $i > 0$. Applying the Sylvester rank inequality to (19) yields

$$\mathrm{rk}\left\{\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\right\} = \left[n - \mathrm{rk}\left\{\boldsymbol{A}^{k}\right\}\right] - \left[n - \mathrm{rk}\left\{\boldsymbol{A}^{k-1}\right\}\right]$$
$$= \mathrm{rk}\left\{\boldsymbol{A}^{k-1}\right\} - \mathrm{rk}\left\{\boldsymbol{A}^{k}\right\} \le n - \mathrm{rk}\{\boldsymbol{A}\} \le s. \quad (22)$$

From (21), we conclude that there exist s columns in $\boldsymbol{K}^{(k)}\boldsymbol{A}^{k-1}\boldsymbol{B}$, indexed by $\mathcal{S}_{h-k}$, such that (20) holds. Also, for any $\boldsymbol{z} \in \ker\{(\boldsymbol{A}^{R})^{\top}\}$, we have $\boldsymbol{z} = [\boldsymbol{I} - \boldsymbol{K}^{(1)}]\boldsymbol{z} + \boldsymbol{K}^{(1)}\boldsymbol{z}$. Here, $\boldsymbol{K}^{(1)}\boldsymbol{z} = \boldsymbol{K}^{(1)}\boldsymbol{B}_{\mathcal{S}_{h-1}}\boldsymbol{v}(h-1)$ for some $\boldsymbol{v}(h-1)$ due to (20) with $k = 1$. Hence, defining $\bar{\boldsymbol{z}}(1) \doteq [\boldsymbol{I} - \boldsymbol{K}^{(1)}][\boldsymbol{z} - \boldsymbol{B}_{\mathcal{S}_{h-1}}\boldsymbol{v}(h-1)]$, we have

$$\boldsymbol{z} = \bar{\boldsymbol{z}}(1) + \boldsymbol{B}_{\mathcal{S}_{h-1}}\boldsymbol{v}(h-1). \quad (23)$$

Recursively using (20), there exist $\{\boldsymbol{v}(k)\}_{k=1}^{R}$ such that

$$\boldsymbol{z} = \left[\boldsymbol{I} - \boldsymbol{K}^{(2)}\right]\bar{\boldsymbol{z}}(1) + \boldsymbol{K}^{(2)}\bar{\boldsymbol{z}}(1) + \boldsymbol{B}_{\mathcal{S}_{h-1}}\boldsymbol{v}(h-1)$$
$$= \bar{\boldsymbol{z}}(2) + \sum_{k=1}^{2}\boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k}}\boldsymbol{v}(h-2)$$
$$= \bar{\boldsymbol{z}}(R) + \sum_{k=1}^{R}\boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k}}\boldsymbol{v}(h-k), \quad (24)$$

where $\bar{\boldsymbol{z}}(k) = [\boldsymbol{I} - \boldsymbol{K}^{(k)}][\bar{\boldsymbol{z}}(k-1) - \boldsymbol{B}_{\mathcal{S}_{h-k}}\boldsymbol{v}(h-k)]$. Since $\boldsymbol{z} \in \ker\{(\boldsymbol{A}^{R})^{\top}\}$, multiplying (24) with $\boldsymbol{K}^{(R)}$ gives

$$\boldsymbol{K}^{(R)}\boldsymbol{z} = \boldsymbol{z} = \boldsymbol{K}^{(R)}\sum_{k=1}^{R}\boldsymbol{A}^{k-1}\boldsymbol{B}_{\mathcal{S}_{h-k-1}}\boldsymbol{v}(k), \quad (25)$$

proving the desired result (18).

Appendix B
Proof of Theorem 1

We begin by noting that $\mathbb{R}^{n} = \ker\{(\boldsymbol{A}^{R})^{\top}\} \oplus \mathcal{A}_{R}$, where $\mathcal{A}_{R} \doteq \mathrm{Col}\{\boldsymbol{A}^{R}\}$. From Lemma 2, it suffices to prove that there exists an s-sparse schedule $\mathcal{S} \in \mathcal{T}_{m}^{h}$ such that

$$\mathcal{A}_{R} = \mathrm{Col}\left\{\left[\boldsymbol{A}^{h-1}\boldsymbol{B}_{\mathcal{S}_0}\ \boldsymbol{A}^{h-2}\boldsymbol{B}_{\mathcal{S}_1}\ \ldots\ \boldsymbol{A}^{R}\boldsymbol{B}_{\mathcal{S}_{h-R-1}}\right]\right\}. \quad (26)$$

To this end, let the real Jordan canonical of $\boldsymbol{A}$ be

$$\boldsymbol{A} = \boldsymbol{P}^{-1}\begin{bmatrix}\boldsymbol{J} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{N}\end{bmatrix}\boldsymbol{P} = \begin{bmatrix}\boldsymbol{P}^{(1)}\\ \boldsymbol{P}^{(2)}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{J} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{N}\end{bmatrix}\begin{bmatrix}\boldsymbol{P}^{(1)}\\ \boldsymbol{P}^{(2)}\end{bmatrix}, \quad (27)$$

where $\boldsymbol{P} = [\boldsymbol{P}^{(1)\top}\ \boldsymbol{P}^{(2)\top}]^{\top}$ is an invertible matrix, with the columns of $\boldsymbol{P}^{(1)}$ and $\boldsymbol{P}^{(2)}$ are the generalized eigenvectors of $\boldsymbol{A}$ corresponding to the nonzero and zero eigenvalues of $\boldsymbol{A}$, respectively. Also, the square matrices $\boldsymbol{J} \in \mathbb{R}^{J\times J}$ and $\boldsymbol{N}$ are formed by the Jordan blocks of $\boldsymbol{A}$ corresponding to its nonzero and zero eigenvalues, respectively. We see that $\boldsymbol{N}^{R} = \boldsymbol{0}$, for any $k \ge R$. As a result, $J \doteq \mathrm{rk}\{\boldsymbol{J}\} \le n - R$, and we deduce

$$\mathcal{A}_{R} = \mathrm{Col}\left\{\boldsymbol{P}^{-1}\begin{bmatrix}\boldsymbol{J}^{R} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{P}\right\} = \mathrm{Col}\left\{\begin{bmatrix}\boldsymbol{J}^{R}\\ \boldsymbol{0}\end{bmatrix}\right\}. \quad (28)$$

Further, premultiplying (1) with $\boldsymbol{P}^{(1)}$ gives

$$\boldsymbol{P}^{(1)}\boldsymbol{x}(k+1) = \boldsymbol{J}\boldsymbol{P}^{(1)}\boldsymbol{x}(k) + \boldsymbol{P}^{(1)}\boldsymbol{B}\boldsymbol{u}(k). \quad (29)$$

The linear dynamical system $(\boldsymbol{J}, \bar{\boldsymbol{B}})$ with $\bar{\boldsymbol{B}} \doteq \boldsymbol{P}^{(1)}\boldsymbol{B}$ is s-sparse controllable for any $s \ge 1$, because $(\boldsymbol{A}, \boldsymbol{B})$ is controllable and $\boldsymbol{J}$ is invertible. Hence, by Proposition 2, there exist sets $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{J-1}$ such that $|\mathcal{S}_k| \le s$ and

$$\mathbb{R}^{J} = \mathrm{Col}\left\{\left[\boldsymbol{J}^{J-1}\bar{\boldsymbol{B}}_{\mathcal{S}_0}\quad \boldsymbol{J}^{J-2}\bar{\boldsymbol{B}}_{\mathcal{S}_1}\quad \cdots\quad \bar{\boldsymbol{B}}_{\mathcal{S}_{J-1}}\right]\right\}. \quad (30)$$

Consequently, from (28), we have

$$\mathcal{A}_{R} = \mathrm{Col}\left\{\begin{bmatrix}\boldsymbol{J}^{R+J-1}\bar{\boldsymbol{B}}_{\mathcal{S}_0} & \boldsymbol{J}^{R+J-2}\bar{\boldsymbol{B}}_{\mathcal{S}_1} & \cdots & \boldsymbol{J}^{R}\bar{\boldsymbol{B}}_{\mathcal{S}_{J-1}}\\ \boldsymbol{0}\,\boldsymbol{0} & & \cdots & \boldsymbol{0}\end{bmatrix}\right\}$$
$$= \mathrm{Col}\left\{\left[\boldsymbol{A}^{R+J-1}\boldsymbol{B}_{\mathcal{S}_0}\quad \boldsymbol{A}^{R+J-2}\boldsymbol{B}_{\mathcal{S}_1}\quad \cdots\quad \boldsymbol{A}^{R}\boldsymbol{B}_{\mathcal{S}_{J-1}}\right]\right\}. \quad (31)$$

Due to $h \ge n$, it follows $h - R \ge n - R \ge J$ and thus $R + J \le h$, implying (26) holds, and the proof is complete.

Acknowledgment

Views and opinions expressed in this letter are of the authors and may not reflect those of the funding institutions.

References

[1] W. P. M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nešić, "Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1781–1796, Aug. 2010.

[2] G. Joseph, B. Nettasinghe, V. Krishnamurthy, and P. K. Varshney, "Controllability of network opinion in Erdős–Rényi graphs using sparse control inputs," *SIAM J. Control Optim.*, vol. 59, no. 3, pp. 2321–2345, 2021.

[3] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Trans. Automat. Control*, vol. 66, no. 4, pp. 1686–1701, Apr. 2021.

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

[5] M. Nagahara, *Sparsity Methods for Systems and Control*. Norwell, MA, USA: Now Publ., 2020.

[6] M. R. Jovanović and N. K. Dhingra, "Controller architectures: Tradeoffs between performance and structure," *Eur. J. Control*, vol. 30, pp. 76–91, Jul. 2016.

[7] L. Ballotta, M. R. Jovanović, and L. Schenato, "Can decentralized control outperform centralized? The role of communication latency," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 3, pp. 1629–1640, Sep. 2023.

[8] A. Olshevsky, "Minimal controllability problems," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 3, pp. 249–258, Sep. 2014.

[9] G. Joseph and C. R. Murthy, "Controllability of linear dynamical systems under input sparsity constraints," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 924–931, Feb. 2021.

[10] G. Joseph, "Output controllability of a linear dynamical system with sparse controls," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 1, pp. 147–156, Mar. 2023.

[11] G. Joseph, "Controllability of a linear system with nonnegative sparse controls," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 468–473, Jan. 2022.

[12] C. Sriram, G. Joseph, and C. R. Murthy, "Stabilizability of linear dynamical systems using sparse control inputs," *IEEE Trans. Autom. Control*, vol. 68, no. 8, pp. 5014–5021, Aug. 2023.

[13] K. P. V. S. Kondapi, C. Sriram, G. Joseph, and C. R. Murthy, "Sparse actuator scheduling for discrete-time linear dynamical systems," 2024, *arXiv:2407.00385*.

[14] G. Joseph, "Sparse actuator control of discrete-time linear dynamical systems," *Found. Trends Syst. Control*, vol. 11, no. 3, pp. 186–284, 2024.

[15] G. Baggio, F. Pasqualetti, and S. Zampieri, "Energy-aware controllability of complex networks," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, no. 1, pp. 465–489, May 2022.

[16] L. Ballotta, G. Joseph, and I. R. Thete, "Pointwise-sparse actuator scheduling for linear systems with controllability guarantee," 2024, *arXiv:2407.12125*.

[17] G. Joseph, S. Moothedath, and J. Lin, "Minimal input structural modifications for strongly structural controllability," in *Proc. IEEE Conf. Decis. Control*, 2024.

[18] C. Sammut, "Markov chain monte carlo," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., Boston, MA, USA: Springer, 2010, pp. 639–642.