Stabilizability of Linear Dynamical Systems Using Sparse Control Inputs

Chandrasekhar Sriram[1], Geethu Joseph[2], and Chandra R. Murthy[1]

***Abstract*—Stabilizability of a linear dynamical system (LDS) refers to the existence of control inputs that drive the system state to zero. In this work, we analyze both theoretical and algorithmic aspects of the stabilizability of an LDS using sparse control inputs with potentially time-varying supports. We show that an LDS is stabilizable using sparse control inputs if and only if it is stabilizable (using unconstrained inputs). For a stabilizable LDS, we present an algorithm to determine the sparse control inputs that steer the system state to zero. We show that all stabilizable LDSs are also sparse mean square stabilizable when the process noise has zero mean and bounded second moment. For such an LDS, we devise a method to sequentially estimate the sparse control inputs to stabilize the LDS in the mean square sense. We prove that a detectable and stabilizable LDS is sparse stabilizable through output feedback and develop an algorithm for finding the corresponding sparse control inputs. Finally, we analyze the stabilizability of an LDS using sparse control inputs with common support. Our results shed light on the conditions under which a given LDS is stabilizable using sparse control inputs and the design of the corresponding control inputs.**

***Index Terms*—Sparsity, state control, stabilizability, output feedback, control input design, piecewise sparsity.**

I. Introduction

Sparse control of a linear dynamical system (LDS) is a new research area that deals with control inputs having very few nonzero entries compared to their dimension. Sparse control inputs are suitable for many resource-constrained systems such as networked systems, opinion dynamics, and environment control systems [1]–[4]. For example, in a networked LDS, the controller and plant (or actuator) communicate over a data rate-limited channel [5]. Sparse control inputs are suitable in this setting because sparse vectors admit compact representations, reducing the bandwidth requirements [2], [6]. In the social network opinion dynamics control problem, an agent (paid blogger, marketing staff, election candidate) manipulates the network opinion by influencing the opinion of a few individuals [3], [7]. Here, the opinion of people in the social network is the time-evolving state (for example, the DeGroot model), and the agents' influence is modeled using sparse control inputs. Motivated by these applications, this paper considers the design and analysis of sparse controllers, focusing on the stabilizability of an LDS with sparse control inputs.

The stability and stabilizability of an LDS are well-studied topics in control theory [8]. An LDS is said to be asymptotically stable if its state decays to zero with time, starting from any initial state. Asymptotic stability implies that even if the state is perturbed, it asymptotically settles back to the desired trajectory. If a system is not asymptotically stable, we seek the possibility of rendering the system stable by applying suitable control inputs [8]. When these inputs are applied, the system states do not significantly change under small perturbations in the state trajectory. We explore the possibility of stabilizing an LDS using sparse control inputs, which we henceforth refer to as *sparse stabilizability*.

[1]C. Sriram and C. R. Murthy are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India. Emails: {chandrasekhars, cmurthy}@iisc.ac.in

[2]G. Joseph is with the Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft Technical University, Delft 2628 CD, The Netherlands. Email: g.joseph@tudelft.nl

Using sparse control inputs for stabilizing a system is a nascent research topic. The use of sparse control inputs for stabilizing continuous-time linear systems via state feedback was introduced in [9], where the inputs were designed using a row-sparse feedback gain matrix. The estimation of the gain matrix was cast into a nonconvex optimization problem via a Lyapunov inequality, and solved using algorithms based on convexification. The idea of the row-sparse gain matrix was extended to discrete-time linear systems in [10]. Following this line of research, a few studies addressed the design of sparse feedback with other structural constraints [11]–[14]. However, most of the investigations were devoted to the numerical implementation of the design algorithms, and little attention has been paid to the theoretical aspects of sparse stabilizability. Also, using a row-sparse feedback matrix imposes the extra constraint of common support (set of indices corresponding to its nonzero entries) for all the control inputs, which is needlessly restrictive. Our paper bridges this gap by presenting a comprehensive analysis that includes theory and algorithms concerning the sparse stabilizability of an LDS when the support of the control inputs can vary over time. Our framework is motivated by the advantages of sparse control inputs with time-varying support demonstrated in [15]–[17].

We address the sparse stabilizability problem via the classical Popov-Belevitch-Hautus (PBH) test for the stabilizability of LDS [8]. Consequently, we do not design a row-sparse feedback gain matrix to induce sparsity. Instead, we directly design the sparse control inputs that drive the system state to zero. This, in turn, allows us to use control inputs with time-varying supports. The fewer constraints on the input lead to more flexibility in stabilizing the system. Moreover, our formulation exploits the sparse recovery algorithms from the compressed sensing literature to design sparse control inputs. This approach is fundamentally different from existing studies that use the linear matrix inequality-based approach that relies on the Lyapunov inequality [10]–[13]. Our specific contributions are as follows:

- In Section II, we show that all stabilizable LDSs are s-sparse stabilizable for any $s \geq 1$ (see Theorem 2). We also design an algorithm for finding a sequence of s-sparse control inputs such that the system state decays to zero with time (see Algorithm 1). Our algorithm is based on a sparse vector recovery method that exploits an inherent underlying structure called the piecewise-sparsity pattern in the control inputs.
- In Section III-A, we extend our results to a noisy LDS where we address the notion of mean square stabilizability of the system. We show that all stabilizable LDSs are also s-sparse mean square stabilizable for any $s \geq 1$ (see Theorem 4). This result leads to a procedure that sequentially estimates the sparse control inputs to ensure the mean square stabilizability of a noisy LDS (see Algorithm 2).
- In Section III-B, we consider a more restricted system where the system state is accessible only via the system output. For such systems, we show that if the system is stabilizable and detectable, it is also sparse stabilizable (see Theorem 5). We also present an algorithm to determine sparse control inputs that ensure stabilizability through output feedback (see Algorithm 3).
- In Section III-C, we analyze an LDS where the supports of the control inputs remain unchanged across all the time instants (similar

to [10]–[14]). We derive the necessary and sufficient conditions for the stabilizability (see Theorem 6) and show that verifying these conditions is an NP-complete problem (see Proposition 1). This result establishes that all stabilizable LDSs are not necessarily sparse stabilizable when the input supports are time-invariant.

In a nutshell, our work gives theoretical guarantees on the sparse stabilizability of an LDS and provides algorithmic solutions to compute the control inputs to ensure sparse stabilizability. We note that several past works have developed algorithms based on sparse signal recovery to estimate control inputs with time-invariant support [18], [19]. However, to the best of our knowledge, ours is the first algorithm that exploits the piecewise-sparsity structure in the control input design.

Notation: The ℓ_0-norm, which counts the number of nonzero elements in a vector, is denoted by $\|\cdot\|_0$. The operator $\|\cdot\|$ denotes the induced ℓ_2-norm for matrices and Euclidean norm for vectors. We use $\boldsymbol{A}_{\mathcal{S}}$ to denote the submatrix of $\boldsymbol{A}$ formed by the columns indexed by the set $\mathcal{S}$. Also, $\mathsf{Rank}\{\cdot\}$, $\rho(\cdot)$ and $(\cdot)^\dagger$ represent the rank, spectral radius, and pseudo-inverse of a matrix, respectively.

II. Characterization of Sparse Stabilizability

In this section, we characterize the notion of sparse stabilizability of an LDS. For this, we consider a discrete-time linear time-invariant system governed by the following equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k, \tag{1}$$

where $k \geq 0$ is the integer time index. Here, $\boldsymbol{x}_k \in \mathbb{R}^n$ is the system state and $\boldsymbol{u}_k \in \mathbb{R}^m$ is the control input at time k. Also, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the state transition matrix and $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ is the input matrix.

The LDS defined by (1) is said to be *controllable* if there exists a finite integer K and a sequence of control inputs $\{\boldsymbol{u}_k\}_{k=0}^{K-1}$ that can drive the system state from any initial state $\boldsymbol{x}_{\text{init}}$ to any final state $\boldsymbol{x}_{\text{final}}$, i.e., $\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}}$ and $\boldsymbol{x}_K = \boldsymbol{x}_{\text{final}}$. A slightly weaker notion than controllability is called stabilizability. The LDS defined by (1) is said to be *stabilizable* if there exists a sequence of control inputs $\{\boldsymbol{u}_k\}_{k=0}^{\infty}$ such that $\lim_{k\to\infty}\|\boldsymbol{x}_k\| = 0$, for any initial state $\boldsymbol{x}_0 \in \mathbb{R}^n$. From the above notion of stabilizability, the definition of s-sparse stabilizability naturally follows:

Definition 1: The LDS defined by (1) is said to be *s-sparse stabilizable* if, for any initial state $\boldsymbol{x}_0 \in \mathbb{R}^n$, there exists a control input sequence $\{\boldsymbol{u}_k, \|\boldsymbol{u}_k\|_0 \leq s\}_{k=0}^{\infty}$ such that $\lim_{k\to\infty}\|\boldsymbol{x}_k\| = 0$. Similarly, an LDS which is controllable using s-sparse control inputs is called s-sparse controllable [15].

This section characterizes sparse stabilizability using a canonical form of the LDS in (1) that decomposes the system into stable and unstable parts. Using this form, we derive two sets of results:

1) *sparse stabilizability test:* conditions that are jointly necessary and sufficient for an LDS to be s-sparse stabilizable, for any $s \geq 1$;
2) *design of sparse control inputs:* an algorithm to find control inputs $\left\{\boldsymbol{u}_k, \|\boldsymbol{u}_k\|_0 \leq s\right\}_{k=0}^{\infty}$ for an s-sparse stabilizable system that asymptotically drive the system state to zero.

We start with the canonical form in the next subsection.

A. Canonical Form of an LDS

We construct the canonical form using the real Jordan decomposition of the state transition matrix, $\boldsymbol{A}$, which is as follows:

$$\boldsymbol{A} = \boldsymbol{V}^{-1}\begin{bmatrix}\boldsymbol{S}_{(1)} \in \mathbb{R}^{n_1\times n_1} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{S}_{(2)} \in \mathbb{R}^{n-n_1\times n-n_1}\end{bmatrix}\boldsymbol{V}, \tag{2}$$

where $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S}_{(1)}$ is an invertible matrix consisting of the Jordan blocks corresponding to the eigenvalues of $\boldsymbol{A}$ whose absolute values are greater than or equal to 1. Also, $\boldsymbol{S}_{(2)}$ is a stable matrix formed by the remaining Jordan blocks, i.e., $\rho(\boldsymbol{S}_{(2)}) < 1$. Further, we partition the rows of $\boldsymbol{V}$ as $\boldsymbol{V}_{(1)} \in \mathbb{R}^{n_1\times n}$ and $\boldsymbol{V}_{(2)} \in \mathbb{R}^{n-n_1\times n}$,

$$\boldsymbol{V} = \begin{bmatrix}\boldsymbol{V}_{(1)}^{\mathsf{T}} & \boldsymbol{V}_{(2)}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}. \tag{3}$$

Substituting (2) into (1) and premultiplying by $\boldsymbol{V}$, we get

$$\boldsymbol{V}_{(1)}\boldsymbol{x}_{k+1} = \boldsymbol{S}_{(1)}\boldsymbol{V}_{(1)}\boldsymbol{x}_k + \boldsymbol{V}_{(1)}\boldsymbol{B}\boldsymbol{u}_k \tag{4}$$

$$\boldsymbol{V}_{(2)}\boldsymbol{x}_{k+1} = \boldsymbol{S}_{(2)}\boldsymbol{V}_{(2)}\boldsymbol{x}_k + \boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_k. \tag{5}$$

The above relations represent two LDSs whose states are two low dimensional projections of the original state $\boldsymbol{x}_k$ namely, $\boldsymbol{V}_{(1)}\boldsymbol{x}_k \in \mathbb{R}^{n_1}$ representing the unstable part; and $\boldsymbol{V}_{(2)}\boldsymbol{x}_k \in \mathbb{R}^{n-n_1}$ representing the stable part (since $\rho(\boldsymbol{S}_{(2)}) < 1$). However, both LDSs share a common control input $\boldsymbol{u}_k$.

Furthermore, the new LDSs have some interesting properties related to stabilizability. Suppose that there exist an integer $K < \infty$ and a set of control inputs $\left\{\boldsymbol{u}_k, \|\boldsymbol{u}_k\|_0 \leq s\right\}_{k=0}^{K-1}$ such that $\boldsymbol{V}_{(1)}\boldsymbol{x}_K = \boldsymbol{0}$. Then, if we choose $\boldsymbol{u}_k = \boldsymbol{0}$, for all $k \geq K$, from (4) and (5), for $k \geq K$, we arrive at

$$\|\boldsymbol{x}_k\| \leq \left\|\boldsymbol{V}^{-1}\right\|\left(\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_k\right\| + \left\|\boldsymbol{V}_{(2)}\boldsymbol{x}_k\right\|\right) \tag{6}$$

$$= \left\|\boldsymbol{V}^{-1}\right\|\left(\left\|\boldsymbol{S}_{(1)}^{k-K}\boldsymbol{V}_{(1)}\boldsymbol{x}_K\right\| + \left\|\boldsymbol{S}_{(2)}^{k-K}\boldsymbol{V}_{(2)}\boldsymbol{x}_K\right\|\right) \tag{7}$$

$$\leq C\gamma^{k-K}\left\|\boldsymbol{V}_{(2)}\boldsymbol{x}_K\right\|, \tag{8}$$

for some real numbers $C > 0$ and $0 \leq \gamma < 1$ that depend on the state matrix $\boldsymbol{A}$. The last step is because $\rho(\boldsymbol{S}_{(2)}) < 1$. In other words, the above choice of *feedforward control inputs* asymptotically drives the system state $\boldsymbol{x}_k$ to zero, stabilizing the LDS.

In the sequel, we use the above formulation to derive the s-sparse stabilizability test and design s-sparse control inputs.

B. Necessary and Sufficient Conditions

We first note that s-sparse stabilizability is a stronger notion than stabilizability and a weaker notion than s-sparse controllability. So, the conditions for stabilizability are necessary for s-sparse stabilizability, and the conditions for s-sparse controllability are sufficient for s-sparse stabilizability. These conditions are as follows:

Theorem 1: Consider the LDS defined by (1).

1) [8, Theorem 14.2] The LDS is stabilizable if and only if for all $\lambda \in \mathbb{C}$ such that $|\lambda| \leq 1$, it holds that

$$\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I} - \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\right\} = n. \tag{9}$$

2) [15, Theorem 1] The LDS is s-sparse controllable if and only if (9) holds for all $\lambda \in \mathbb{C}$, and $s \geq n - \mathsf{Rank}\{\boldsymbol{A}\}$.

Building upon the above result, we next derive the stabilizability test.

Theorem 2: For $1 \leq s \leq m$, the LDS defined by (1) is s-sparse stabilizable if and only if $\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I} - \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\right\} = n \;\forall\, |\lambda| \geq 1$.

Proof: See Appendix I. ■

Theorem 2 leads to the following corollary connecting sparse stabilizability and sparse controllability.

Corollary 1: For the LDS in (1), the following are equivalent:

1) The LDS is stabilizable.
2) The LDS is s-sparse stabilizable for any given $1 \leq s \leq m$.
3) The unstable part of the LDS given by (4) is s-sparse controllable for any given $1 \leq s \leq m$.

The resulting relationship between sparse controllability, controllability, and sparse stabilizability is depicted by Figure 1. It is interesting to note that any stabilizable system is also s-sparse stabilizable. This is unlike its counterpart in controllability, where the notions of controllability and sparse controllability are different [15]. Corollary 1

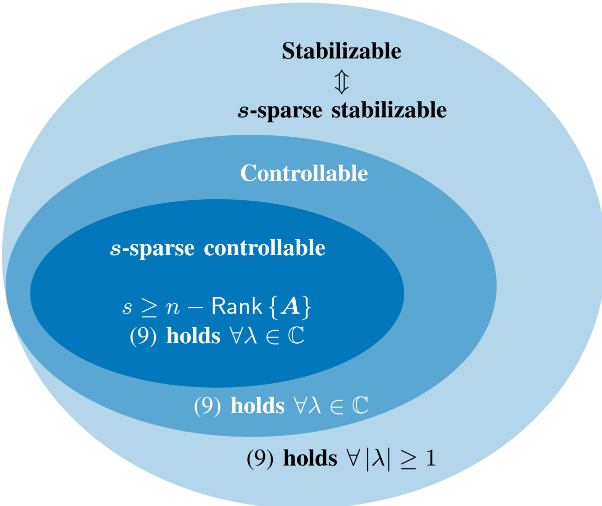


Fig. 1: Venn diagram showing the relation between sparse controllability, controllability, sparse stabilizability and stabilizability, and the necessary and sufficient conditions that each set satisfies.

also asserts that even if a stabilizable LDS is not s-sparse controllable, it's s-sparse uncontrollable part is still stabilizable. Here, the s-sparse uncontrollable part of an LDS refers to a projected system state $\boldsymbol{x}_k$ that cannot be driven to an arbitrary desired value from any initial value using s-sparse control inputs [15, Section V].

Having derived the conditions for sparse stabilizability, we next discuss the design of sparse control inputs satisfying these conditions.

C. Design of Sparse Control Inputs

Before we discuss the algorithm for finding sparse control inputs, we note that if the LDS defined by (1) is stabilizable, one can find a matrix $\boldsymbol{K} \in \mathbb{R}^{m\times n}$ such that $\rho(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}) < 1$. Hence, by selecting the (unconstrained) control inputs $\boldsymbol{u}_k = -\boldsymbol{K}\boldsymbol{x}_k$, we get

$$\lim_{k\to\infty} \boldsymbol{x}_k = \lim_{k\to\infty} (\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K})^{k-1}\boldsymbol{x}_0 = \boldsymbol{0}. \tag{10}$$

However, this feedback mechanism cannot be applied in our case, as $-\boldsymbol{K}\boldsymbol{x}_k$ may not be s-sparse. Hence, we develop an algorithm that computes a sequence of sparse control inputs for stabilizing an LDS.

From (8), designing sparse control inputs for stabilizing the system is the same as finding an integer $K < \infty$ and sparse control inputs $\left\{\boldsymbol{u}_k, \|\boldsymbol{u}_k\|_0 \leq s\right\}_{k=0}^{K-1}$ that drive $\boldsymbol{V}_{(1)}\boldsymbol{x}$ to zero. The existence of such sparse inputs is guaranteed by Corollary 1. Equivalently, we seek sparse inputs that drive the unstable part of the state defined by (4) to zero. We utilize the following result that bounds the number of nonzero sparse control inputs, K, required to drive the state of a sparse controllable LDS to zero.

Theorem 3 ([15, Theorem 3]): If the LDS defined by (1) is s-sparse controllable, the number of s-sparse control inputs, K_*, required to ensure controllability is bounded as follows:

$$\frac{n}{R^*_{B,s}} \leq K_* \leq \min\left\{q\left\lceil\frac{\text{Rank}\{\boldsymbol{B}\}}{s}\right\rceil, n - R^*_{B,s}+1\right\} \leq n, \tag{11}$$

where q is the order of the minimal polynomial of $\boldsymbol{A}$, and $R^*_{B,s} \triangleq \min\{\text{Rank}\{\boldsymbol{B}\}, s\}$.

From Theorem 3, the maximum number of sparse vectors required to steer $\boldsymbol{V}_{(1)}\boldsymbol{x}_k$ to zero is given by

$$K^* = \min\left\{q_1\left\lceil\frac{R_1}{s}\right\rceil, n_1 - \min\{R_1, s\}+1\right\} \leq n_1, \tag{12}$$

where we define q_1 as the order of the minimal polynomial of $\boldsymbol{S}_{(1)}$ and $R_1 \triangleq \text{Rank}\left\{\boldsymbol{V}_{(1)}\boldsymbol{B}\right\} < n_1$. However, from (4), we also have

$$\boldsymbol{V}_{(1)}\boldsymbol{x}_{K^*} = \boldsymbol{S}_{(1)}^{K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_0 + \sum_{k=0}^{K^*-1}\boldsymbol{S}_{(1)}^{K^*-k-1}\boldsymbol{V}_{(1)}\boldsymbol{B}\boldsymbol{u}_k. \tag{13}$$

Thus, we solve for K^* sparse control inputs from

$$\boldsymbol{R}\tilde{\boldsymbol{u}}_{0:K^*-1} = -\boldsymbol{S}_{(1)}^{K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_0, \tag{14}$$

where $\boldsymbol{R} \in \mathbb{R}^{n_1\times K^*m}$ and $\tilde{\boldsymbol{u}}_{k_1:k_2} \in \mathbb{R}^{m(k_2-k_1+1)\times 1}$ are

$$\boldsymbol{R} \triangleq \begin{bmatrix}\boldsymbol{S}_{(1)}^{K^*-1}\boldsymbol{V}_{(1)}\boldsymbol{B} & \boldsymbol{S}_{(1)}^{K^*-2}\boldsymbol{V}_{(1)}\boldsymbol{B} & \ldots & \boldsymbol{V}_{(1)}\boldsymbol{B}\end{bmatrix} \tag{15}$$

$$\tilde{\boldsymbol{u}}_{k_1:k_2} \triangleq \begin{bmatrix}\boldsymbol{u}_{k_1}^\mathsf{T} & \boldsymbol{u}_{k_1+1}^\mathsf{T} & \ldots & \boldsymbol{u}_{k_2}^\mathsf{T}\end{bmatrix}^\mathsf{T}, \quad k_2 \geq k_1 \geq 0. \tag{16}$$

Here, the unknown vector $\tilde{\boldsymbol{u}}_{0:K^*-1}$ is formed by concatenating K^* number of s-sparse vectors. This sparsity structure is referred to as a *piecewise sparse structure*. Directly solving for a piecewise sparse vector from (14) incurs exponential computational complexity [20]. However, approximate piecewise sparse recovery algorithms with polynomial complexity are available in the compressed sensing literature. Some examples are the piecewise orthogonal matching pursuit [21] and piecewise inverse scale space algorithm [22].

The overall algorithm is summarized in Algorithm 1. By applying the control inputs obtained using Algorithm 1, the unstable part of the system state $\boldsymbol{V}_{(1)}\boldsymbol{x}_k$ is driven to zero at time $k = K^*$ while the stable part $\boldsymbol{V}_{(2)}\boldsymbol{x}_k$ goes to zero as the time index k goes to infinity.

Algorithm 1 Design of control inputs for s-sparse stabilization

Inputs: $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, and $\boldsymbol{x}_0 \in \mathbb{R}^n$

1: Compute $\boldsymbol{V}_{(1)} \in \mathbb{R}^{n_1\times n}$ and $\boldsymbol{S}_{(1)} \in \mathbb{R}^{n_1\times n_1}$ using the real Jordan decomposition of $\boldsymbol{A}$ as given in (2) and (3)
2: Compute K^* from (12)
3: Compute $\boldsymbol{R}$ using (15)
4: Compute $\{\boldsymbol{u}_k\}_{k=0}^{K^*-1}$ by solving (14) using a piecewise sparse recovery algorithm [21], [22]
5: Choose $\boldsymbol{u}_k = \boldsymbol{0}$, for $k \geq K^*$

We make a few observations about our algorithm:

1) Computational Complexity: Algorithm 1 involves computing the Jordan form and recovering a piecewise sparse vector from (14). The complexity in Jordan decomposition of an $n \times n$ matrix is $\mathcal{O}(n^5)$ [23]. The complexity in solving for a K^*m dimensional piecewise sparse vector with sparsity K^*s using n_1 equations via POMP is $\mathcal{O}(K^*mn_1 + K^{*3}ms^2)$ [21]. Since $K^* \leq n_1 \leq n$ (see (12)), the overall complexity of Algorithm 1 is $\mathcal{O}(n^5 + n^3s^2m)$.

2) Comparison with Stabilization via State Feedback: As mentioned earlier (see (10)), one method to stabilize an LDS (with no constraints on the control input) is via system state feedback control,

$$\boldsymbol{u}_k = -\boldsymbol{K}\boldsymbol{x}_k = -\boldsymbol{K}(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{B})^{k-1}\boldsymbol{x}_0, \tag{17}$$

for all values of k. However, if we relax the sparsity constraint (i.e., when $s = m$), Algorithm 1 gives $\tilde{\boldsymbol{u}}_{0:K^*-1} = -(\boldsymbol{R})^\dagger\boldsymbol{S}_{(1)}^{K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_0$ as the solution. Thus, in both approaches, the control inputs are linear functions of $\boldsymbol{x}_0$. However, our approach has a finite number of nonzero control inputs, whereas the state feedback can potentially have infinitely many such inputs.

3) Number of Nonzero Control Inputs: From Theorem 3, we know that $K^* \leq n_1$, which implies that, for any stabilizable LDS, we need at most n_1 nonzero sparse control inputs to drive the system state to zero. Similarly, Theorem 3 also implies that we need at least $\frac{n_1}{\min\{R_1,s\}}$ nonzero sparse control inputs to drive the state to zero.

4) Decay of System State $\mathbf{x}_k$*:* If the LDS is stabilizable, we can find s-sparse control inputs $\{\boldsymbol{u}_k\}_{k=0}^{K^*-1}$ such that for any $k>0$,

$$\|\boldsymbol{x}_k\| \leq \max\left\{\max_{0\leq k\leq K^*}\|\boldsymbol{x}_k\|, C\gamma^{k-K^*}\left\|\boldsymbol{V}_{(2)}\boldsymbol{x}_{K^*}\right\|\right\} \tag{18}$$

$$< c\gamma^k\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_0\right\|, \tag{19}$$

where $c>0$ and $0\leq\gamma<1$. Here, we also use (8) and (12), which implies that $K^*\leq n_1<\infty$. Consequently, similar to stabilization via state feedback in (10), the norm of the system state exponentially decays to zero, and the decay rate depends on $\boldsymbol{A}$ (via γ).

5) Dependence on Sparsity: Note that K^* decreases with the sparsity level s. As s increases, the control inputs become less restricted, and we can drive the unstable part $\boldsymbol{V}_{(1)}\boldsymbol{x}_k$ to zero more quickly. On the other hand, the sparsity level does not affect whether or not the system is s-sparse stabilizable. However, choosing a small value s may increase the time needed to drive the unstable part of the system state to zero, thereby lengthening the time taken to drive the exponentially decaying system state to a desired small value.

This completes our discussion on the algebraic characterization and design of control inputs for sparse stabilizability. Our analysis thus far makes three assumptions: (1) the system is noiseless; (2) the initial state $\boldsymbol{x}_0$ is known, (3) the supports of sparse control inputs can be time-varying. In the following section, we extend the idea of sparse stabilizability to address these three restrictive assumptions.

III. Variants of Sparse Stabilizability

In this section, we build upon the notion of sparse stabilizability and present three (more stringent) variants of sparse stabilizability.

A. Mean Square Stabilizability of Noisy Systems

We consider a noisy LDS given by the following equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{v}_k, \tag{20}$$

where $\boldsymbol{x}_k$, $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined in (1). Also, $\boldsymbol{v}_k\in\mathbb{R}^n$ is the process noise at time k such that for any integer $k\geq 0$,

$$\mathbb{E}\{\boldsymbol{v}_k\}=\mathbf{0},\;\; \mathbb{E}\left\{\boldsymbol{v}_k\boldsymbol{v}_k^{\mathsf{T}}\right\}=\boldsymbol{\Sigma}_v\in\mathbb{R}^{n\times n} \text{ with } \|\boldsymbol{\Sigma}_v\|<\infty. \tag{21}$$

Also, we assume that the noise terms $\{\boldsymbol{v}_k\}_{k=0}^{\infty}$ are independent across time k and are independent of the initial state $\boldsymbol{x}_0$.

The stabilizability of noisy LDSs is characterized by the notion of *mean square stabilizability* [24]. The system is said to be mean square stable if $\sup_{k=1,2,\ldots}\mathbb{E}\left\{\|\boldsymbol{x}_k\|^2\right\}<\infty$, for any $\boldsymbol{x}_0\in\mathbb{R}^n$. Here, the expectation is computed with respect to the additive noise $\boldsymbol{v}_k$. We extend the above notion of stabilizability for sparse control inputs as follows.

Definition 2: The LDS given by (20) and (21) is said to be s-sparse mean square stabilizable if for any initial state $\boldsymbol{x}_0\in\mathbb{R}^n$, there exists a sequence of s-sparse control inputs $\{\boldsymbol{u}_k\}_{k=0}^{\infty}$ such that

$$\sup_{k=1,2,\ldots}\mathbb{E}\left\{\|\boldsymbol{x}_k\|^2\right\}<\infty. \tag{22}$$

If $\rho(\boldsymbol{A})<1$, the LDS is mean square stable [24]. Hence, a stabilizable system can use the feedback-based control $\boldsymbol{u}_k=-\boldsymbol{K}\boldsymbol{x}_k$ so that $\rho(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{B})<1$ which ensures stabilizability. Here, $\boldsymbol{K}$ is defined in (10). Similarly, to achieve s-sparse mean square stabilizability, we extend the idea of Algorithm 1. As discussed in Section II-A, we can decompose the system state into stable and unstable parts. However, if we apply the sparse control inputs returned by Algorithm 1, the unstable part of the system state may not go to zero at time $k=K^*$ (given by (12)). This behavior is because of the extra additive terms due to the noise $\boldsymbol{v}_k$. To handle the noise term, we estimate the sparse control inputs after every L^* time steps where

$$L^* = \min\left\{k\geq K^*: \left\|\boldsymbol{S}_{(2)}^k\right\|<1\right\}. \tag{23}$$

We note that $\left\{k\geq K^*: \left\|\boldsymbol{S}_{(2)}^k\right\|<1\right\}\neq\emptyset$ since $\rho(\boldsymbol{S}_{(2)})<1$. We sequentially select the sparse control inputs that ensure that the unstable part $\boldsymbol{V}_{(1)}\boldsymbol{x}_k$ is bounded. The stable part $\boldsymbol{V}_{(2)}\boldsymbol{x}_k$ is also bounded because of the condition on $\left\|\boldsymbol{S}_{(2)}\right\|$ in (23). Thus, if such a sequence of sparse control inputs exists, the system is guaranteed to be s-sparse mean square stabilizable. The formal statement and proof of the above arguments are given below.

Theorem 4: The LDS defined by (20) and (21) is s-sparse mean square stabilizable, for all $1\leq s\leq m$, if

$$\text{Rank}\left\{\left[\lambda\boldsymbol{I}-\boldsymbol{A}\quad \boldsymbol{B}\right]\right\}=n,\; \forall|\lambda|\geq 1. \tag{24}$$

Proof: See Appendix II. ■

Theorem 4 ensures that all stabilizable systems are s-sparse mean square stabilizable. Also, the constructive proof of Theorem 4 leads to the design algorithm described by Algorithm 2. Here, unlike the noiseless system where we used a feedforward control based on the initial state, we use feedback control where the control inputs are estimated using

$$\boldsymbol{R}\tilde{\boldsymbol{u}}_{rL^*:rL^*+K^*-1} = -\boldsymbol{S}_{(1)}^{K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*} \tag{25}$$

$$\tilde{\boldsymbol{u}}_{rL^*+K^*:(r+1)L^*-1}=\mathbf{0}, \tag{26}$$

at time $k=rL^*$ for every integer $r\geq 0$. Consequently, the number of nonzero sparse control inputs is not bounded when the LDS is noisy.

Algorithm 2 Design of control inputs for s-sparse stabilization of a noisy LDS

Inputs: $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\boldsymbol{B}\in\mathbb{R}^{n\times m}$
1: Compute $\boldsymbol{V}_{(1)}\in\mathbb{R}^{n_1\times n}$ and $\boldsymbol{S}_{(1)}\in\mathbb{R}^{n_1\times n_1}$ using the real Jordan decomposition of $\boldsymbol{A}$ as given in (2) and (3)
2: Compute K^* from (12)
3: Compute $\boldsymbol{R}$ using (15)
4: Compute L^* using (23)
5: **for** $r=0,1,\ldots$ **do**
6: Compute $\{\boldsymbol{u}_k\}_{k=rL^*}^{rL*+K^*-1}$ by solving (25) using a piecewise sparse recovery algorithm [21], [22]
7: Choose $\boldsymbol{u}_{rL^*+k}=\mathbf{0}$, for $k=K^*,K^*+1,\ldots,L^*-1$
8: **end for**

The computational complexity of Algorithm 2 is similar to Algorithm 1, except that that POMP is repeated after every L^* steps. The complexity of Steps 1-4 of Algorithm 2 is $\mathcal{O}(n^5+mn^2)$ and the complexity of the POMP step is $\mathcal{O}(n^3s^2m)$.

Comparison with conventional non-sparse stabilization: It is known [8] that the state feedback matrix $\boldsymbol{K}$ that stabilizes the system in the noiseless case can also be used for mean square stabilization of the system described by (20). The complexity of computing $-\boldsymbol{K}\boldsymbol{x}_k$ is $\mathcal{O}(mn)$, in each time step. Our proposed algorithm computes inputs for only L^* steps; it cannot be directly compared with the conventional approach that requires a matrix-vector multiplication to compute the inputs at each time step.

B. Stabilizability Through Output Feedback

Stabilizability through output feedback refers to the design of control inputs to stabilize an LDS using the knowledge of its external outputs only [8]. For this, we consider the LDS defined as follows.

$$\boldsymbol{x}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\boldsymbol{u}_k \quad\text{and}\quad \boldsymbol{y}_k=\boldsymbol{C}\boldsymbol{x}_k, \tag{27}$$

where $\boldsymbol{y}_k \in \mathbb{R}^p$ is the system output, and $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ is the output matrix. We note that this problem is harder because the outputs may not give us the full information about the system state. For such systems wherein the state vector is not accessible, we modify the notion of sparse stabilizability as follows:

Definition 3: The LDS defined by (27) is said to be s-sparse stabilizable through output feedback if for any integer $k \geq 0$, we can find an s-sparse control input $\boldsymbol{u}_k$ from the present and past output sequence $\{\boldsymbol{y}_t\}_{t=1}^k$ such that $\lim_{k\to\infty}\|\boldsymbol{x}_k\| = 0$, for any $\boldsymbol{x}_0 \in \mathbb{R}^n$.

From Section II-C, we recall that designing s-sparse control inputs for stabilizing the LDS requires only the knowledge of $\boldsymbol{V}_{(1)}\boldsymbol{x}_0$ and the system matrices. Therefore, for our system, if we can estimate the $\boldsymbol{V}_{(1)}\boldsymbol{x}_0$ from the output $\boldsymbol{y}_k$, then we can design the desired control inputs. In other words, if the unstable part (given by (4)) of an s-sparse stabilizable LDS is observable, the system is s-sparse stabilizable through output feedback. This observation yields the following result.

Theorem 5: The LDS defined by (27) is s-sparse stabilizable through output feedback for any $1 \leq s \leq m$ if for all $\lambda \in \mathbb{C}$ such that $|\lambda| \geq 1$,

$$\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\right\} = \mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{A}\\ \boldsymbol{C}\end{bmatrix}\right\} = n. \tag{28}$$

Proof: The proof is straightforward because the condition ensures that the LDS is s-sparse stabilizable (see Theorem 2) and detectable (i.e., the part of the system state that cannot be estimated from the outputs is stable) [8, Theorem 16.5]. ■

Theorem 5 implies that all systems that are sparse stabilizable through output feedback are s-sparse stabilizable through output feedback, and vice versa, for all $1 \leq s \leq m$. Let the reduced column echelon form of the observability matrix be

$$\begin{bmatrix}(\boldsymbol{C}\boldsymbol{A}^{n-1})^\mathsf{T} & (\boldsymbol{C}\boldsymbol{A}^{n-2})^\mathsf{T} & \ldots & \boldsymbol{C}^\mathsf{T}\end{bmatrix}^\mathsf{T} = \begin{bmatrix}\tilde{\boldsymbol{C}} & \boldsymbol{0}\end{bmatrix}\boldsymbol{T}, \tag{29}$$

where $\tilde{n} \geq n_1$ is the rank of the observability matrix, $\tilde{\boldsymbol{C}} \in \mathbb{R}^{np\times\tilde{n}}$ has full column rank, and $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ represents the invertible transformation that leads to the echelon form. Then, the observable part of the initial state is given by $\boldsymbol{T}_{(1)}\boldsymbol{x}_0 \in \mathbb{R}^{\tilde{n}}$ where $\boldsymbol{T}_{(1)} \in \mathbb{R}^{\tilde{n}\times n}$ represents the submatrix of $\boldsymbol{T}$ formed by the first $\tilde{n}$ rows. As we noted earlier, to design of the sparse control inputs, we compute $\boldsymbol{T}_{(1)}\boldsymbol{x}_0$ as

$$\boldsymbol{T}_{(1)}\boldsymbol{x}_0 = \tilde{\boldsymbol{C}}^\dagger\begin{bmatrix}\boldsymbol{y}_1^\mathsf{T} & \boldsymbol{y}_2^\mathsf{T} & \ldots & \boldsymbol{y}_n^\mathsf{T}\end{bmatrix}^\mathsf{T}. \tag{30}$$

However, detectability of the LDS implies that the row space of $\boldsymbol{V}_{(1)}$ is a subset of that of $\boldsymbol{T}_{(1)}$, and hence, we compute $\boldsymbol{V}_{(1)}\boldsymbol{x}_0$ from $\boldsymbol{T}_{(1)}\boldsymbol{x}_0$. Finally, we estimate the sparse control inputs that can drive the unstable part $\boldsymbol{S}_{(1)}^n\boldsymbol{V}_{(1)}\boldsymbol{x}_0$ of the system state (at time $k=n$) to zero using

$$\boldsymbol{R}\tilde{\boldsymbol{u}}_{n:n+K^*-1} = -\boldsymbol{S}_{(1)}^{n+K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_0. \tag{31}$$

We present the pseudocode of our algorithm in Algorithm 3.

We note that the number of nonzero sparse control inputs that stabilize the LDS does not change even if we stabilize the LDS through output feedback. However, for stabilization through output feedback, the unstable part is driven to zero at time $k = n + K^*$, whereas it goes to zero at time $k = K^*$ when the initial state is known. From computational point of view, Algorithm 3 needs one extra step to compute $\boldsymbol{V}_{(1)}\boldsymbol{x}_0 = \boldsymbol{V}_{(1)}\boldsymbol{T}_{(1)}^\dagger\boldsymbol{T}_{(1)}\boldsymbol{x}_0$. Thus, the overall complexity of this algorithm is $\mathcal{O}(n^5 + n^3ms^2 + n^3p)$.

The notion of sparse stabilizability through output feedback can be extended to noisy LDSs. In this case, the LDS is represented as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{v}_k \quad \text{and} \quad \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{32}$$

Algorithm 3 Design of control inputs for s-sparse stabilization through output feedback

Inputs: $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$
Choose $\boldsymbol{u}_k = \boldsymbol{0}$ for $k = 1, 2, \ldots, n-1$
Compute $\tilde{\boldsymbol{C}} \in \mathbb{R}^{np\times\tilde{n}}$ and $\boldsymbol{T}_{(1)} \in \mathbb{R}^{\tilde{n}\times n}$ using the column reduced echelon form of the observability matrix given in (29)
Compute $\boldsymbol{T}_{(1)}\boldsymbol{x}_0$ using (30)
Compute $\boldsymbol{V}_{(1)}\boldsymbol{x}_0 = \boldsymbol{V}_{(1)}\boldsymbol{T}_{(1)}^\dagger\boldsymbol{T}_{(1)}\boldsymbol{x}_0$
Compute K^* from (12)
Compute $\boldsymbol{R}$ using (15)
Compute $\{\boldsymbol{u}_k\}_{k=n}^{n+K^*-1}$ by solving (31) using a piecewise sparse recovery algorithm [21], [22]
Choose $\boldsymbol{u}_k = \boldsymbol{0}$ for $k \geq n + K^*$

where $\boldsymbol{v}_k \in \mathbb{R}^m$ and $\boldsymbol{w}_k \in \mathbb{R}^p$ represent the process noise and observation noise, respectively. Here, we combine Algorithms 2 and 3 to sequentially estimate the unstable part of the state and design the corresponding sparse control inputs. Specifically, in Step 6 of Algorithm 2, we estimate the unstable part of the state. The estimation can be done via the least squares method or Kalman filtering if the initial state is Gaussian distributed with known statistics. We omit the details to avoid repetition.

Comparison with conventional non-sparse stabilization: We know that, if the system described by (27) is detectable, one can find a matrix $\boldsymbol{L}$ such that $\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}$ is a stable matrix [8]. Using $\hat{\boldsymbol{x}}_k = \boldsymbol{L}\boldsymbol{y}_k$, we can estimate the state with an exponential reduction in the norm of the error in the estimate. Using this estimate, it can be shown that we can stabilize the system through output feedback. In contrast, our proposed approach is a non-linear feedback technique.

C. Time-invariant Support

In this subsection, we analyze the stabilizability of an LDS using sparse control inputs that share the same support. Let $\mathcal{S} \subset \{1, 2, \ldots, m\}$ be the common support set of all the control inputs where $|\mathcal{S}| \leq s$. Then, the state evolution model (1) is equivalent to

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}_\mathcal{S}\boldsymbol{u}_{k,\mathcal{S}}, \tag{33}$$

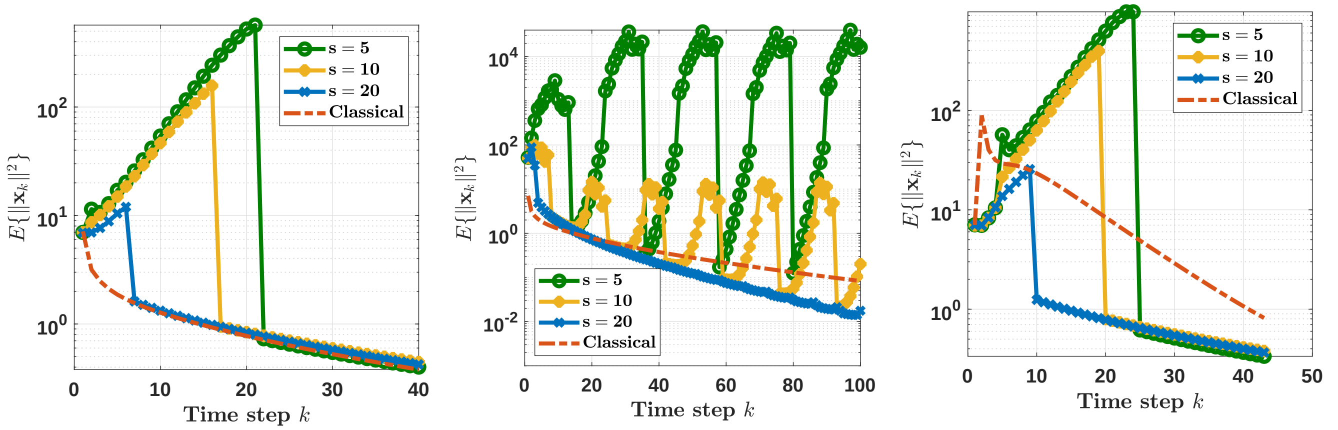
where $\boldsymbol{B}_\mathcal{S} \in \mathbb{R}^{n\times|\mathcal{S}|}$ and $\boldsymbol{u}_{k,\mathcal{S}}\mathbb{R}^{|\mathcal{S}|}$ represents the entries of $\boldsymbol{u}_k$ indexed by $\mathcal{S}$. We note that (33) represents an unconstrained LDS described by $(\boldsymbol{A}, \boldsymbol{B}_\mathcal{S})$. As a consequence, the necessary and sufficient conditions for stabilizability and the design of sparse inputs are straightforward.

Theorem 6: For any given $1 \leq s \leq m$, the LDS defined by (1) is s-sparse stabilizable with time invariant support if and only if there exists a set $\mathcal{S} \subset \{1, 2, \ldots, m\}$ such that $|\mathcal{S}| = s$ and

$$\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{A} & \boldsymbol{B}_\mathcal{S}\end{bmatrix}\right\} = n, \ \forall |\lambda| \geq 1. \tag{34}$$

Further, if (34) holds for some $\mathcal{S}$, there exists a matrix $\boldsymbol{K}_{(s)} \in \mathbb{R}^{s\times n}$ such that $\rho\left(\boldsymbol{A} - \boldsymbol{B}_\mathcal{S}\boldsymbol{K}_{(s)}\right) < 1$, and the s-sparse control inputs with $\boldsymbol{u}_{k,\mathcal{S}} = -\boldsymbol{K}_{(s)}\boldsymbol{x}_k$ and $\boldsymbol{u}_{k,\mathcal{S}^\complement} = \boldsymbol{0}$ asymptotically drive the system state to zero.

Clearly, the condition described by (34) is more stringent than the sparse stabilizability condition in Theorem 2. Hence, we conclude that restricting the sparse control inputs to have common support can make the LDS non-stabilizable. Further, using the equivalent unconstrained system in (33), we can also arrive at results similar to those in Sections III-A and III-B. We omit the details to avoid repetition. Although Theorem 6 gives a necessary and sufficient condition, unlike sparse stabilizability with time-varying supports, this condition cannot be verified in polynomial time.



(a) Stabilization of noiseless LDS (b) Mean-square stabilization of noisy LDS (c) Stabilization of noiseless LDS through output feedback

Fig. 2: Comparison of stabilization of an LDS with $n = m = 50$, $p = 12$, and $n_1 = n_2 = 25$ using sparse inputs with sparsity levels $s = 5, 10$ and 20. The behavior of conventional stabilization using non-sparse inputs is also shown.

Proposition 1: For the LDS defined by (1), finding a set $\mathcal{S} \subset \{1, 2, \ldots, m\}$, such that $|\mathcal{S}| = s$ and (34) holds, is NP-complete.

Proof: We know that (34) seeks a subset of s columns of $\boldsymbol{B}$ such that for every left-eigenvector $\boldsymbol{z} \in \mathbb{C}^n$ of $\boldsymbol{A}$ with eigenvalue $\lambda \geq 1$, there exists at least one column of $\boldsymbol{B}$ from the subset which is not orthogonal to $\boldsymbol{z}$. This can be represented using a bipartite graph. The vertices of the graph on the left represent the columns of $\boldsymbol{B}$. The vertices of the graph on the right represent the left-eigenvectors of $\boldsymbol{A}$ with the corresponding eigenvalue greater than unity. Also, an edge from a left vertex to any right vertex exists if the corresponding column of $\boldsymbol{B}$ and left-eigenvector are not orthogonal. The task is then to check whether a subset of left-vertices of size at most s covers all of the right-vertices. In other words, verifying (34) is equivalent to the decision version of the hitting set problem or, equivalently, the set cover problem [25]. Since the decision version of the hitting set problem is NP-complete [25], the verification of (34) is also NP-complete. ■

Note that we can use approximate algorithms such as the greedy algorithm and layering [25] that solve the hitting set problem to estimate $\mathcal{S}$ satisfying (34). Once $\mathcal{S}$ is estimated, the design of sparse control inputs is straightforward from the classical stabilizability results for unconstrained control inputs. In other words, this approach leads to a row sparse feedback gain matrix whose nonzero rows are indexed by $\mathcal{S}$. We note that the existing methods for sparse stabilizing feedback using the linear matrix inequality [10]–[13] also lead to a row sparse gain matrix. However, the two methods have their roots in two equivalent concepts: the PBH stabilizability test (our approach) and the Lyapunov inequality (existing approach).

IV. Simulation Results

In this section, we illustrate the performance of our algorithms using numerical results. We choose the system dimensions as $n = m = 50$, $p = 12$, and $n_1 = n_2 = 25$. For the state transfer matrix $\boldsymbol{A}$, we choose $\boldsymbol{V}$ in (2) as $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U} \end{bmatrix}$. Here, $\boldsymbol{U} \in \mathbb{R}^{25\times 25}$ is a random orthogonal matrix obtained via the eigendecomposition of a symmetric random Gaussian matrix. The diagonal entries of $\boldsymbol{S}_{(1)}$ and $\boldsymbol{S}_{(2)}$ are drawn independently from uniform distributions with supports $(1, 1.5)$ and $(-1, 1)$, respectively. We choose the input matrix $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_c^\mathsf{T} & \boldsymbol{0} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{50\times 50}$ and the output matrix $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_c & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{12\times 50}$. Here, the entries of $\boldsymbol{B}_c \in \mathbb{R}^{25\times 50}$ and $\boldsymbol{C}_c \in \mathbb{R}^{12\times 25}$ are drawn from the standard Gaussian distribution. Using Theorem 4, we first verified that the system is stabilizable through output feedback, and thus, our algorithms can stabilize it.

Using the above realization, we compute sparse inputs for s-sparse stabilization, s-sparse mean-square stabilization, and s-sparse stabilization through output feedback using Algorithms 1, 2 and 3, respectively. Figure 2 shows the squared ℓ_2 norm of the state vector in each case, averaged across 100 random initial vectors $\boldsymbol{x}_0$ drawn from the standard Gaussian distribution. The process noise vectors $\boldsymbol{v}_k$ (only for mean square stabilization) drawn i.i.d. from the zero-mean Gaussian distribution with variance 10^{-6}. We compare with classical (non-sparse) stabilization, where the input is a linear function of the state vector for the noiseless and noisy LDS and a linear function of the output vector for stabilization through output feedback [8].

Figure 2a shows that for the noiseless LDS stabilized using sparse inputs, the norm of the state vector first increases when nonzero inputs are applied to drive the controllable part of the state vector $\boldsymbol{V}_{(1)}\boldsymbol{x}_k$ to zero. After the controllable part becomes zero, we apply zero control input and the uncontrollable part, which is asymptotically stable, decays down to $\boldsymbol{0}$. For mean-square stabilization in Figure 2b, the controllable part of the state is not driven to zero due to the process noise. Therefore, the norm of the state vector corresponding to the unstable part increases periodically. Stabilization through output feedback in Figure 2c shows a similar trend as that in Figure 2a. However, in this case, we first observe the output for some time to estimate the initial state. Due to this estimation period, it takes longer to drive the controllable part to zero. From Figures 2a and 2c, we also infer that as s decreases, we need a longer time to drive the controllable part to zero. This inference is in agreement with (12). Similarly, Figure 2b shows that as s increases, the system becomes less constrained, and thus, $\mathbb{E}\left\{\|\boldsymbol{x}_k\|^2\right\}$ decreases. Finally, the state vector norm curve is smooth for the classical (non-spare) stabilization schemes. In contrast, for our algorithms, the state vector norm does not change smoothly for small values of s.

V. Conclusions

We studied the problem of stabilizing an LDS using sparse control inputs. We first proved that all stabilizable systems are sparse stabilizable. Further, we provided bounds on the number of nonzero sparse inputs required to stabilize the LDS and derived an algorithm to design the sparse control inputs. We also developed similar results for three other notions of sparse stabilizability, namely, mean square stabilizability for noisy LDSs; stabilizability through output feedback when the initial state is unknown; and stabilizability when inputs share a common support. Extending our analysis to study the stabilizability of an LDS when its control inputs have both sparsity and energy constraints is a promising direction for future work.

Appendix I
Proof of Theorem 2

From Theorem 1, the necessity of the rank condition in the theorem is straightforward. Consequently, it is enough to show the sufficiency of the condition, and we prove this using the canonical form discussed in Section II-A. From (8), we know that to prove the sufficiency of the condition in Theorem 2, it suffices to show that under this condition, there exist an integer $K<\infty$ and a set of control inputs $\left\{\boldsymbol{u}_k, \|\boldsymbol{u}_k\|_0\leq s\right\}_{k=0}^{K-1}$ such that $\boldsymbol{V}_{(1)}\boldsymbol{x}_K=\mathbf{0}$. For this, it is enough to verify s-sparse controllability of the unstable part of system defined by the matrix tuple $(\boldsymbol{S}_{(1)},\boldsymbol{V}_{(1)}\boldsymbol{B})$ (see (4)). However, when the condition in Theorem 2 holds for all λ such that $|\lambda|\geq 1$, we have

$$n=\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\right\} \tag{35}$$

$$=\mathsf{Rank}\left\{\boldsymbol{V}\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}^{-1} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{I}\end{bmatrix}\right\} \tag{36}$$

$$=\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{S}_{(1)} & \mathbf{0} & \boldsymbol{V}_{(1)}\boldsymbol{B}\\ \mathbf{0} & \lambda\boldsymbol{I}-\boldsymbol{S}_{(2)} & \boldsymbol{V}_{(2)}\boldsymbol{B}\end{bmatrix}\right\}, \tag{37}$$

which follows from (5). Therefore, for any $\lambda\in\mathbb{C}$ with $|\lambda|\geq 1$,

$$\mathsf{Rank}\left\{\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{S}_{(1)} & \boldsymbol{V}_{(1)}\boldsymbol{B}\end{bmatrix}\right\}=n_1. \tag{38}$$

Further, since the all eigenvalues of $\boldsymbol{S}_{(1)}$ have absolute values greater than 1, (38) holds for all $\lambda\in\mathbb{C}$ and

$$\mathsf{Rank}\left\{\boldsymbol{S}_{(1)}\right\}=n_1>n_1-s,\ \forall\, s\geq 1. \tag{39}$$

Invoking Theorem 1, we conclude that the LDS described by (4) is s-sparse controllable $\forall\, s\geq 1$. Thus, we establish the sufficiency of the condition in Theorem 2, and the proof is complete. ∎

Appendix II
Proof of Theorem 4

The proof relies on the canonical form described in Section II-A. Similar to (4) and (5), here we have

$$\boldsymbol{V}_{(1)}\boldsymbol{x}_{k+1}=\boldsymbol{S}_{(1)}\boldsymbol{V}_{(1)}\boldsymbol{x}_k+\boldsymbol{V}_{(1)}\boldsymbol{B}\boldsymbol{u}_k+\boldsymbol{V}_{(1)}\boldsymbol{v}_k \tag{40}$$

$$\boldsymbol{V}_{(2)}\boldsymbol{x}_{k+1}=\boldsymbol{S}_{(2)}\boldsymbol{V}_{(2)}\boldsymbol{x}_k+\boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_k+\boldsymbol{V}_{(2)}\boldsymbol{v}_k. \tag{41}$$

Using the above decomposition and (6), we obtain

$$\sup_{k=1,2,\ldots}\mathbb{E}\left\{\|\boldsymbol{x}_k\|^2\right\}\leq 2\left\|\boldsymbol{V}^{-1}\right\|^2\left(\sup_{k=1,2,\ldots}\mathbb{E}\left\{\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_k\right\|^2\right\}\right.$$
$$\left.+\sup_{k=1,2,\ldots}\mathbb{E}\left\{\left\|\boldsymbol{V}_{(2)}\boldsymbol{x}_k\right\|^2\right\}\right). \tag{42}$$

Therefore, to prove the desired result, it is enough to prove that there exists a sequence of sparse control inputs such that

$$\sup_{k=1,2,\ldots}\mathbb{E}\left\{\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_k\right\|^2\right\}<\infty \tag{43}$$

$$\sup_{k=1,2,\ldots}\mathbb{E}\left\{\left\|\boldsymbol{V}_{(2)}\boldsymbol{x}_k\right\|^2\right\}<\infty. \tag{44}$$

Hence, the rest of the proof is devoted to choosing a sequence of s-sparse control inputs that satisfy (25) and (26) and showing that the choice of inputs guarantees the relations (43) and (44).

We start with the sparse control inputs defined by (25) and (26). From Theorem 2 and Corollary 1, we know that (24) ensures s-sparse controllability of the LDS defined by (4). As a result, there exist index sets $\{\mathcal{S}_i\subset\{1,2,\ldots,m\}\}_{i=1}^{K^*}$ such that

$$|\mathcal{S}_i|\leq s \text{ and } \mathsf{Rank}\left\{\boldsymbol{R}_{\mathcal{S}}\right\}=n_1, \tag{45}$$

where the index set $\mathcal{S}=\cup_{i=1}^{K^*}\{l+(i-1)m, l\in\mathcal{S}_i\}$. Also, K^*, L^* and $\boldsymbol{R}\in\mathbb{R}^{n_1\times K^*m}$ are defined in (12), (22) and (15), respectively. Hence,

$$\boldsymbol{R}_{\mathcal{S}}=\begin{bmatrix}\boldsymbol{S}_{(1)}^{K^*-1}\boldsymbol{V}_{(1)}\boldsymbol{B}_{\mathcal{S}_1} & \boldsymbol{S}_{(1)}^{K^*-2}\boldsymbol{V}_{(1)}\boldsymbol{B}_{\mathcal{S}_2}\ldots\boldsymbol{V}_{(1)}\boldsymbol{B}_{\mathcal{S}_{K^*}}\end{bmatrix}. \tag{46}$$

For any integer $r\geq 0$, we choose the s-sparse control inputs as

$$\tilde{\boldsymbol{u}}_{rL^*:rL^*+K^*-1,\mathcal{S}}=(\boldsymbol{R}_{\mathcal{S}})^{\dagger}\boldsymbol{S}_{(1)}^{K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}\in\mathbb{R}^{K^*|\mathcal{S}|} \tag{47}$$

$$\tilde{\boldsymbol{u}}_{rL^*:rL^*+K^*-1,\mathcal{S}^{\complement}}=\mathbf{0}\in\mathbb{R}^{K^*(m-|\mathcal{S}|)}, \tag{48}$$

where $\tilde{\boldsymbol{u}}_{rL^*:rL^*+K^*-1}\in\mathbb{R}^{K^*m}$ is defined in (16).

Using the above choice of control inputs, we next establish (43). For this, we notice from (25) that

$$-\boldsymbol{S}_{(1)}^{K^*}\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}=\sum_{i=0}^{K^*-1}\boldsymbol{S}_{(1)}^{K^*-i-1}\boldsymbol{V}_{(1)}\boldsymbol{B}\boldsymbol{u}_{rL^*+i}, \tag{49}$$

for any integer $r\geq 0$. Also, from (4), for any $r\geq 0$ and $1\leq k\leq L^*$,

$$\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*+k}=\boldsymbol{S}_{(1)}^{k}\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}$$
$$+\sum_{i=0}^{k-1}\boldsymbol{S}_{(1)}^{k-i-1}\boldsymbol{V}_{(1)}\left[\boldsymbol{B}\boldsymbol{u}_{rL^*+i}+\boldsymbol{v}_{rL^*+i}\right]. \tag{50}$$

With $k=L^*$ in (50) and using (26) and (49), we arrive at

$$\boldsymbol{V}_{(1)}\boldsymbol{x}_{(r+1)L^*}=\sum_{i=0}^{L^*-1}\boldsymbol{S}_{(1)}^{L^*-i-1}\boldsymbol{V}_{(1)}\boldsymbol{v}_{rL^*+i}, \tag{51}$$

for any integer $r\geq 0$. Thus, using (21) and the fact that the noise vectors $\boldsymbol{v}_k$ are independent, the expected value of the ℓ_2 norm square of the unstable part of the state is

$$\mathbb{E}\left\{\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_{(r+1)L^*}\right\|^2\right\}=\sum_{i=0}^{L^*-1}\mathsf{Tr}\left\{\boldsymbol{D}_{(i)}\right\}<\infty, \tag{52}$$

where we define $\boldsymbol{D}_{(i)}\in\mathbb{R}^{n_1\times n_1}$ as

$$\boldsymbol{D}_{(i)}\triangleq\boldsymbol{S}_{(1)}^{L^*-i-1}\boldsymbol{V}_{(1)}\boldsymbol{\Sigma}_v\left(\boldsymbol{S}_{(1)}^{L^*-i-1}\boldsymbol{V}_{(1)}\right)^{\mathsf{T}}. \tag{53}$$

Similarly, from (50), for any integer $1\leq k\leq L^*-1$,

$$\mathbb{E}\left\{\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*+k}\right\|^2\right\}\leq(k+1)\left[\max_{i=0,1,\ldots,k}\left\|\boldsymbol{S}_{(1)}^{i}\right\|^2\right]$$
$$\times\left[\mathbb{E}\left\{\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}\right\|^2\right\}+\left\|\boldsymbol{V}_{(1)}\boldsymbol{B}\right\|^2\sum_{i=0}^{k-1}\mathbb{E}\left\{\|\boldsymbol{u}_{rL^*+i}\|^2\right\}\right.$$
$$\left.+\sum_{i=0}^{k-1}\mathsf{Tr}\left\{\boldsymbol{D}_{(i)}\right\}<\infty.\right. \tag{54}$$

Here, we use the following fact from (26), (47), and (48),

$$\sum_{i=0}^{k-1}\mathbb{E}\left\{\|\boldsymbol{u}_{rL^*+i}\|^2\right\}\leq\sum_{i=0}^{K^*}\mathbb{E}\left\{\|\boldsymbol{u}_{rL^*+i}\|^2\right\} \tag{55}$$

$$\leq\left\|(\boldsymbol{R}_{\mathcal{S}})^{\dagger}\boldsymbol{S}_{(1)}^{K^*}\right\|^2\mathbb{E}\left\{\left\|\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}\right\|^2\right\} \tag{56}$$

$$=\left\|(\boldsymbol{R}_{\mathcal{S}})^{\dagger}\boldsymbol{S}_{(1)}^{K^*}\right\|^2\left[\sum_{i=0}^{L^*-1}\mathsf{Tr}\left\{\boldsymbol{D}_{(i)}\right\}\right]<\infty, \tag{57}$$

where (57) is due to (52). Combining (54) and (57), we prove (43).

We next complete the proof by establishing (44). To this end, we obtain from (41) that for any integer $k > 0$,

$$\boldsymbol{V}_{(2)}\boldsymbol{x}_k = \boldsymbol{S}_{(2)}^k \boldsymbol{V}_{(2)}\boldsymbol{x}_0 + \sum_{i=0}^{k-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\left[\boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{v}_i\right]. \quad (58)$$

We note that the right-hand side of (58) is the sum of two terms: the first term is due to the initial state and process noise given by $\boldsymbol{S}_{(2)}^k \boldsymbol{V}_{(2)}\boldsymbol{x}_0 + \sum_{i=0}^{k-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{v}_i$; and the second term is due to the control inputs given by $\sum_{i=0}^{k-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_i$. However, we know that if no control inputs are applied, the LDS in (5) is mean square stable since $\rho(\boldsymbol{S}_{(2)}) < 1$. Thus, we arrive at

$$\sup_{k=1,2,\ldots} \mathbb{E}\left\{\left\|\boldsymbol{S}_{(2)}^k \boldsymbol{V}_{(2)}\boldsymbol{x}_0 + \sum_{i=0}^{k-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{v}_i\right\|^2\right\} < \infty. \quad (59)$$

Consequently, to prove (44) and establish the desired result, it is sufficient to prove that

$$\sup_{k=1,2,\ldots} \mathbb{E}\left\{\left\|\sum_{i=0}^{k-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_i\right\|^2\right\} < \infty. \quad (60)$$

Furthermore, from (47) and (48), we see that $\{\boldsymbol{u}_i\}_{i=rL^*}^{rL^*+K^*-1}$ is a function of $\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}$. Also, from (51), $\boldsymbol{V}_{(1)}\boldsymbol{x}_{rL^*}$ is a function of $\{\boldsymbol{v}_i\}_{i=(r-1)L^*}^{rL^*-1}$, for $r > 0$. Since the process noise vectors are independent across time and independent of $\boldsymbol{x}_0$, the sets of control inputs $\{\boldsymbol{u}_i\}_{i=rL^*}^{(r+1)L^*-1}$ are independent across different values of $r \geq 0$. Thus, we deduce that for any integers $r \geq 0$ and $1 \leq k \leq L^*$,

$$\begin{aligned}
&\mathbb{E}\left\{\left\|\sum_{i=0}^{rL^*+k-1} \boldsymbol{S}_{(2)}^{rL^*+k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_i\right\|^2\right\} \\
&\quad = \sum_{t=0}^{r-1}\mathbb{E}\left\{\left\|\boldsymbol{S}_{(2)}^{(r-t)L^*}\sum_{i=0}^{L^*-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_{tL^*+i}\right\|^2\right\} \\
&\qquad\qquad + \mathbb{E}\left\{\left\|\sum_{i=0}^{k-1} \boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{B}\boldsymbol{u}_{rL^*+i}\right\|^2\right\} \quad (61)\\
&\leq L^*\sum_{i=0}^{L^*-1}\left\|\boldsymbol{S}_{(2)}^{k-i-1}\boldsymbol{V}_{(2)}\boldsymbol{B}\right\|^2 \sum_{t=0}^{r}\left\|\boldsymbol{S}_{(2)}^{(r-t)L^*}\right\|^2\mathbb{E}\left\{\left\|\boldsymbol{u}_{tL^*+i}\right\|^2\right\}, \quad (62)
\end{aligned}$$

where the last step follows because of the sub-multiplicative property of norms. So, the desired result (60) is proved if the following holds:

$$\sum_{t=0}^{r}\left\|\boldsymbol{S}_{(2)}^{(r-t)L^*}\right\|^2\mathbb{E}\left\{\left\|\boldsymbol{u}_{tL^*+i}\right\|^2\right\} < \infty. \quad (63)$$

To prove the above relation, we use (57) to get

$$\begin{aligned}
&\sum_{t=0}^{r}\left\|\boldsymbol{S}_{(2)}^{(r-t)L^*}\right\|^2\mathbb{E}\left\{\left\|\boldsymbol{u}_{tL^*+i}\right\|^2\right\} \\
&\leq \sum_{t=0}^{r}\left\|\boldsymbol{S}_{(2)}^{L^*}\right\|^{2t}\left\|(\boldsymbol{R}_{\mathcal{S}})^{\dagger}\,\boldsymbol{S}_{(1)}^{K^*}\right\|^2\left[\sum_{i=0}^{L^*-1}\mathrm{Tr}\left\{\boldsymbol{D}_{(i)}\right\}\right] < \infty, \quad (64)
\end{aligned}$$

because $\left\|\boldsymbol{S}_{(2)}^{L^*}\right\| < 1$ due to (23). Hence, (63) is established, which concludes the proof. ■

References

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

[2] M. Nagahara and D. E. Quevedo, "Sparse representations for packetized predictive networked control," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 84–89, Jan. 2011.

[3] N. Wendt, C. Dhal, and S. Roy, "Control of network opinion dynamics by a selfish agent with limited visibility," *IFAC-PapersOnLine*, vol. 52, no. 3, pp. 37–42, Jan. 2019.

[4] A. Jadbabaie, A. Olshevsky, G. J. Pappas, and V. Tzoumas, "Minimal reachability is hard to approximate," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 783–789, May 2018.

[5] S. C. Tatikonda, "Control under communication constraints," Ph.D. dissertation, Dept. Elect. Comput. Sci., MIT, Cambridge, USA, 2000.

[6] Z. Li, Y. Xu, H. Huang, and S. Misra, "Sparse control and compressed sensing in networked switched systems," *IET Control Theory Appl.*, vol. 10, no. 9, pp. 1078–1087, May 2016.

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

[8] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.

[9] B. T. Polyak, M. V. Khlebnikov, and P. S. Shcherbakov, "Sparse feedback in linear control systems," *Autom. Remote. Control*, vol. 75, no. 12, pp. 2099–2111, Dec. 2014.

[10] A. Bykov and P. S. Shcherbakov, "Sparse feedback design in discrete-time linear systems," *Autom. Remote. Control*, vol. 79, no. 7, pp. 1175–1190, Jul. 2018.

[11] A. Bykov, P. Shcherbakov, and M. Ding, "A tractable nonconvex surrogate for the matrix l0-quasinorm: applications to sparse feedback design," *IFAC-PapersOnLine*, vol. 49, no. 13, pp. 53–58, Jan. 2016.

[12] A. Bykov and P. Shcherbakov, "Surrogates for the matrix l0-quasinorm in sparse feedback design: Numerical study of the efficiency," *Adv. Syst. Sci. Appl.*, vol. 18, no. 2, pp. 11–25, Aug. 2018.

[13] F. Ferrante, F. Dabbene, and C. Ravazzi, "On the design of structured stabilizers for LTI systems," *IEEE Control Syst. Lett.*, vol. 4, no. 2, pp. 289–294, Apr. 2020.

[14] M. Yagoubi and R. Chaibi, "A nonsmooth newton method for the design of state feedback stabilizers under structure constraints," *J. Franklin Inst.*, vol. 358, no. 1, pp. 800–813, Jan. 2021.

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

[16] L. F. Chamon, A. Amice, and A. Ribeiro, "Matroid-constrained approximately supermodular optimization for near-optimal actuator scheduling," in *Proc. IEEE Conf. on Decis. and Control*, Dec. 2019, pp. 3391–3398.

[17] C. Sriram, G. Joseph, and C. R. Murthy, "Control of linear dynamical systems using sparse inputs," in *Proc. ICASSP*, May 2020, pp. 5765–5769.

[18] M. Kafashan, A. Nandi, and S. Ching, "Relating observability and compressed sensing of time-varying signals in recurrent linear networks," *Neural Net.*, vol. 83, pp. 11–20, Jul. 2016.

[19] S. Sefati, N. J. Cowan, and R. Vidal, "Linear systems with sparse inputs: Observability and input recovery," in *Proc. Amer. Control Conf.*, Jul. 2015, pp. 5251–5257.

[20] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, 2013.

[21] K. Li, C. R. Rojas, T. Yang, H. Hjalmarsson, K. H. Johansson, and S. Cong, "Piecewise sparse signal recovery via piecewise orthogonal matching pursuit," in *Proc. ICASSP*. IEEE, Mar. 2016, pp. 4608–4612.

[22] Y. Zhong and C. Li, "Piecewise sparse recovery via piecewise inverse scale space algorithm with deletion rule," *J. Comput. Math.*, vol. 38, no. 2, Mar. 2020.

[23] T. Beelen and P. Van Dooren, "Computational aspects of the Jordan canonical form," *Reliable numerical computation*, pp. 57–72, 1990.

[24] P. Minero, L. Coviello, and M. Franceschetti, "Stabilization over markov feedback channels: the general case," *IEEE Trans. Automat. Control*, vol. 58, no. 2, pp. 349–362, Feb. 2013.

[25] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.