2025 American Control Conference (ACC)
July 8-10, 2025. Denver, CO, USA

Finite-Horizon Discrete-Time LQR with Sparse Inputs

Rupam Kalyan Chakraborty[1], Vaibhav Katewa[2] and Chandra R. Murthy[3]

***Abstract*— The Linear Quadratic Regulator (LQR) is a classical problem in optimal control theory which deals with operating a linear dynamical system with optimized cost. In this work, we study the discrete-time LQR problem with sparsity constraints on the inputs. This problem has a combinatorial complexity. We develop a convex optimization-based approach to relax the problem into a semidefinite program which can be solved with polynomial complexity. We explore two cases for input sparsity: fixed temporal support and time-varying support. Moreover, we also solve the minimum-energy control problem with sparse inputs. Finally, using numerical simulations, we show that our algorithms give near-optimum performance with very good accuracy and time complexity.**

I. Introduction

Sparse control of Linear Dynamical Systems (LDS) has gained considerable interest in the recent years in the control community [1]–[8]. It is an area which deals with efficient control of LDS using control inputs with small number of active actuators compared to the total available actuators. An LDS with sparse control models various practical applications, including networked control systems [1]–[3], opinion dynamics manipulation [5], [6] and computer vision [7], [8]. For example, in networked control systems where the communication with the actuators happens over bandwidth-limited channels, minimizing the number of nonzero control inputs reduces the amount of communication between the controller and the actuators. Since sparse vectors admit compact representations, they can be used in such energy or bandwidth-limited scenarios. In this paper, we focus on the Linear Quadratic Regulator (LQR) problem with sparsity constraints on the control inputs. This is a variant of the classical LQR problem since our goal is not only to minimize a quadratic cost function but also to enforce sparsity in the control inputs.

Related work: The authors in [9] consider sparse control inputs design problem for a continuous-time LDS by minimizing a quadratic cost for a finite time horizon. They use ℓ_1 and total variation regularization to promote sparsity in the solutions. This approach can be extended to the discrete-time case by discretizing the system dynamics that results in a set of coupled nonlinear difference equations, which are difficult to solve. Also, the regularization based approach requires tuning of the regularizer term using trial and error to obtain the desired level of sparsity. In contrast, our regularization-free approach does not require tuning and inherently achieves the exact desired level of sparsity. Sparse actuator scheduling for continuous-time systems by maximizing the trace of the controllability Gramian is studied in [10], [11]. The problem of achieving optimal $\mathcal{H}_2$ norm performance for discrete-time LDS through the design of sparse and structured feedback matrices is addressed in [12], [13]. In [14], [15], an ADMM-based algorithm is developed for leader selection in stochastically-forced dynamic networks. Paper [16] explores optimal control problem that minimizes the ℓ_1 norm of outputs and inputs of a discrete-time LDS using linear programming that results in an idle or deadbeat solution. Further, ℓ_1 norm-regularized Model Predictive Control (MPC) has been studied using ADMM in [17]. Finally, [18] proposes an information-theoretic regularization for LQR problem in networked control systems that results in a sparse feedback matrix. However, to the best of our knowledge no work has been done on discrete-time finite-horizon LQR problem with sparsity constraints on the inputs. Finding the sparse supports of control inputs of a given sparsity level that optimizes the LQR cost is a NP-hard problem in general. We develop a novel convex optimization based algorithm to solve the problem in polynomial time with near-optimal performance. The core contribution of this work is to use techniques from convex optimization theory to solve sparsity-constrained LQR problem.

Notation: Boldface capital letters denote matrices, boldface small letters denote vectors and calligraphic letters denote sets. The i^{th} entry of a vector $\boldsymbol{a}$ is denoted by $\boldsymbol{a}_i$. $(\boldsymbol{a})_{\mathcal{I}}$ indicates the subvector of $\boldsymbol{a}$ formed by choosing the entries of $\boldsymbol{a}$ indexed by the set of indices $\mathcal{I}$. $\boldsymbol{I}_n$ and $\boldsymbol{0}_n$ denote the $n \times n$ identity matrix and zero matrix, respectively. The ℓ_0 norm of $\boldsymbol{a}$ is denoted by $\|\boldsymbol{a}\|_0$, and it is the number of nonzero entries in $\boldsymbol{a}$. For a sequence of vectors $\{\boldsymbol{a}(k)\}_{k=0}^{N-1}$, $\tilde{\boldsymbol{a}}_N = [\boldsymbol{a}(0)^T, \boldsymbol{a}(1)^T, \cdots, \boldsymbol{a}(N-1)^T]^T$ denotes the concatenation of the vectors in a column vector. $\mathrm{Blkdiag}(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_N)$ denotes a block diagonal matrix with square matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N$ along its diagonal. $\mathrm{diag}(\boldsymbol{a})$ denotes a diagonal matrix with entries of $\boldsymbol{a}$ along the diagonal. The operator $\mathrm{diag}(\boldsymbol{A})$ is the vector containing the diagonal entries of $\boldsymbol{A}$. $\mathbb{S}^n$ represents the set of $n \times n$ symmetric matrices, and the notation $\boldsymbol{X} \succeq \boldsymbol{Y}$ implies that the matrix $\boldsymbol{X} - \boldsymbol{Y}$ is positive semidefinite. $\lambda_{\min}(\boldsymbol{A})$ denotes the minimum eigenvalue of $\boldsymbol{A}$. For two sets $\mathcal{A}$ and $\mathcal{B}$, the XOR operation $\mathcal{A} \oplus \mathcal{B} \triangleq (\mathcal{A} \cup \mathcal{B}) \setminus (\mathcal{A} \cap \mathcal{B})$ yields the elements that are not common between $\mathcal{A}$ and $\mathcal{B}$. $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$. $\lceil \cdot \rceil$ denotes the ceiling operation.

[1]Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft 2628 CD, The Netherlands. Email: r.k.chakraborty@tudelft.nl. The work of R. K. Chakraborty was carried out when he was at the Indian Institute of Science, Bengaluru, India.

[2]Robert Bosch Center for Cyber-Physical Systems, Indian Institute of Science, Bengaluru 560012, India. Email:vkatewa@iisc.ac.in.

[3]Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, India. Email: cmurthy@iisc.ac.in.

979-8-3315-6937-2/$31.00 ©2025 AACC

II. Problem Formulation

We consider a discrete-time LDS, with state transition matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and input matrix $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ whose dynamics are governed by

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

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ and $\boldsymbol{u}(k) \in \mathbb{R}^m$ are the state and input at time k, respectively. We denote the initial condition $\boldsymbol{x}(0)$ by $\boldsymbol{x}_0$. We consider the LQR problem for the system in (1) where the goal is to minimize a quadratic cost on system's state and control inputs by choosing optimal control inputs. We aim to solve the LQR problem in presence of sparsity constraints on the control inputs. Specifically, we constrain the control inputs to be s-sparse (s is a positive integer), that is, $\boldsymbol{u}(k)$ can have at most s non-zero entries for every time instant $k \in \{0, 1, \cdots, N\}$. This sparse LQR problem is formulated as the following optimization problem:

$$\begin{aligned}\min_{\{\boldsymbol{u}(k)\}_{k=0}^{N-1}} \; J = \sum_{k=0}^{N-1} (\boldsymbol{x}(k)^T\boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{u}(k)^T\boldsymbol{R}\boldsymbol{u}(k)) + \\ + \boldsymbol{x}(N)^T\boldsymbol{Q}\boldsymbol{x}(N) \qquad (2)\\ \text{s.t. } \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k), \; \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \|\boldsymbol{u}(k)\|_0 \le s \;\; \forall k = 0, \cdots, N-1,\end{aligned}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n} \succeq \boldsymbol{0}_n$ and $\boldsymbol{R} \in \mathbb{R}^{m\times m} \succ \boldsymbol{0}_m$. The LQR problem without any sparsity constraints on the inputs can be solved using dynamic programming, which results in a backward Riccati recursion. However, an alternate way to solve the problem is to "unroll" the dynamics to write the state in terms of the control inputs, and then formulate it as an unconstrained optimization problem where the cost is a quadratic function of the inputs. Although this approach has high computational complexity and does not reveal the feedback structure of the optimal control inputs, it is required for developing the solution for the sparsity-constrained problem.

By "unrolling" the dynamics, the state $\boldsymbol{x}(k)$ can be expressed in terms of the initial condition $\boldsymbol{x}_0$ and sequence of control inputs $\{\boldsymbol{u}(i)\}_{i=0}^{k-1}$ as

$$\boldsymbol{x}(k) = \boldsymbol{A}^k\boldsymbol{x}_0 + \tilde{\boldsymbol{C}}_k\tilde{\boldsymbol{u}}_k, \tag{3}$$

where $\tilde{\boldsymbol{C}}_k \triangleq [\boldsymbol{A}^{k-1}\boldsymbol{B}, \boldsymbol{A}^{k-2}\boldsymbol{B}, \cdots, \boldsymbol{A}\boldsymbol{B}, \boldsymbol{B}]$ is the controllability matrix and $\tilde{\boldsymbol{u}}_k$ denotes the concatenated input sequence. Further, concatenating the state vectors for time instants $0, \cdots, N$, we get

$$\tilde{\boldsymbol{x}}_{N+1} = \tilde{\boldsymbol{O}}_N\boldsymbol{x}_0 + \tilde{\boldsymbol{\Gamma}}_N\tilde{\boldsymbol{u}}_N, \quad \text{where} \tag{4}$$

$$\tilde{\boldsymbol{O}}_N \triangleq \begin{bmatrix}\boldsymbol{I}_n \\ \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N\end{bmatrix}, \tilde{\boldsymbol{\Gamma}}_N \triangleq \begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & . & \dots & \boldsymbol{B}\end{bmatrix}.$$

Next, we compactly express the LQR cost in (2) in terms of $\tilde{\boldsymbol{x}}_{N+1}$ and $\tilde{\boldsymbol{u}}_N$ as

$$J = \tilde{\boldsymbol{x}}_{N+1}^T\bar{\boldsymbol{Q}}\tilde{\boldsymbol{x}}_{N+1} + \tilde{\boldsymbol{u}}_N^T\bar{\boldsymbol{R}}\tilde{\boldsymbol{u}}_N, \tag{5}$$

where $\bar{\boldsymbol{Q}} = \text{Blkdiag}(\boldsymbol{Q}, \cdots, \boldsymbol{Q})$ with $N+1$ blocks and $\bar{\boldsymbol{R}} = \text{Blkdiag}(\boldsymbol{R}, \cdots, \boldsymbol{R})$ with N blocks. Substituting (4) in the cost (5), we get

$$\begin{aligned}J &= (\tilde{\boldsymbol{O}}_N\boldsymbol{x}_0 + \tilde{\boldsymbol{\Gamma}}_N\tilde{\boldsymbol{u}}_N)^T\bar{\boldsymbol{Q}}(\tilde{\boldsymbol{O}}_N\boldsymbol{x}_0 + \tilde{\boldsymbol{\Gamma}}_N\tilde{\boldsymbol{u}}_N) + \tilde{\boldsymbol{u}}_N^T\bar{\boldsymbol{R}}\tilde{\boldsymbol{u}}_N \\ &= \tilde{\boldsymbol{u}}_N^T(\tilde{\boldsymbol{\Gamma}}_N^T\bar{\boldsymbol{Q}}\tilde{\boldsymbol{\Gamma}}_N + \bar{\boldsymbol{R}})\tilde{\boldsymbol{u}}_N + 2\boldsymbol{x}_0^T\tilde{\boldsymbol{O}}_N^T\bar{\boldsymbol{Q}}\tilde{\boldsymbol{\Gamma}}_N\tilde{\boldsymbol{u}}_N \\ &\qquad + \boldsymbol{x}_0^T\tilde{\boldsymbol{O}}_N^T\bar{\boldsymbol{Q}}\tilde{\boldsymbol{O}}_N\boldsymbol{x}_0 \\ &= \tilde{\boldsymbol{u}}_N^T\boldsymbol{G}\tilde{\boldsymbol{u}}_N + 2\boldsymbol{h}^T\tilde{\boldsymbol{u}}_N + c,\end{aligned} \tag{6}$$

where $\boldsymbol{G} \triangleq \tilde{\boldsymbol{\Gamma}}_N^T\bar{\boldsymbol{Q}}\tilde{\boldsymbol{\Gamma}}_N + \bar{\boldsymbol{R}}$, $\boldsymbol{h} \triangleq \tilde{\boldsymbol{\Gamma}}_N^T\bar{\boldsymbol{Q}}^T\tilde{\boldsymbol{O}}_N\boldsymbol{x}_0$ and $c \triangleq \boldsymbol{x}_0^T\tilde{\boldsymbol{O}}_N^T\bar{\boldsymbol{Q}}\tilde{\boldsymbol{O}}_N\boldsymbol{x}_0$. Thus, the sparsity constrained LQR problem is reformulated as

$$\begin{aligned}\min_{\tilde{\boldsymbol{u}}_N} \; & \tilde{\boldsymbol{u}}_N^T\boldsymbol{G}\tilde{\boldsymbol{u}}_N + 2\boldsymbol{h}^T\tilde{\boldsymbol{u}}_N + c \\ s.t. \; & \|\boldsymbol{u}(k)\|_0 \le s \;\; \forall k = 0, \cdots, N-1.\end{aligned} \tag{7}$$

III. Solution to the Sparsity Constrained LQR Problem

In this section, we derive a relaxed solution to the sparsity constrained LQR problem (7). We consider two sparsity patterns in the control inputs: (a) fixed temporal support, and (b) time-varying support.

A. Fixed Temporal Support

We consider the case when the inputs are constrained to have a fixed common support of s non-zero entries for all time steps $k = 0, \ldots, N-1$. We introduce the support vector $\boldsymbol{w}$ to represent the active/non-active entries (support) of $\boldsymbol{u}(k)$: $\boldsymbol{w} = [w_1, w_2, \ldots, w_m]^T, \quad w_i \in \{0, 1\}$ where $w_i = 1$ implies that the i^{th} entry of $\boldsymbol{u}_k$ is allowed to be nonzero, and $w_i = 0$ implies that the entry is constrained to be zero. Note that since the support is fixed over time, $\boldsymbol{w}$ does not depend on k. In order to capture the nonzero entries of $\boldsymbol{u}(k)$'s we define a selection matrix $\boldsymbol{S} \in \{0, 1\}^{\|\boldsymbol{w}\|_0 \times m}$. It is the submatrix of $\text{diag}(\boldsymbol{w})$ after removing all rows corresponding to the zero entries of the $\boldsymbol{u}(k)$'s. Hence, $\boldsymbol{S}\boldsymbol{u}(k) = (\boldsymbol{u}(k))_{\mathcal{S}} \in \mathbb{R}^{\|\boldsymbol{w}\|_0}$ is the nonsparse vector containing the nonzero entries of $\boldsymbol{u}(k)$ where $\mathcal{S}$ is the support set of $\boldsymbol{u}(k)$. The selection matrix $\boldsymbol{S}$ has following properties:

$$\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{I}_{\|\boldsymbol{w}\|_0} \text{ and } \boldsymbol{S}^T\boldsymbol{S} = \text{diag}(\boldsymbol{w}). \tag{8}$$

Next, we concatenate all vectors $\{\boldsymbol{S}\boldsymbol{u}(k)\}_{k=0}^{N-1}$ and denote it as $\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} \triangleq \bar{\boldsymbol{S}}\tilde{\boldsymbol{u}}_N$, where

$$\bar{\boldsymbol{S}} = \text{Blkdiag}(\boldsymbol{S}, \cdots, \boldsymbol{S}) \text{ with } N \text{ blocks.} \tag{9}$$

With this, we write the cost in (6) in terms of $\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}$ as

$$\begin{aligned}J(\boldsymbol{S}) &= \tilde{\boldsymbol{u}}_N^T\bar{\boldsymbol{S}}^T\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T\bar{\boldsymbol{S}}\tilde{\boldsymbol{u}}_N + 2\boldsymbol{h}^T\bar{\boldsymbol{S}}^T\bar{\boldsymbol{S}}\tilde{\boldsymbol{u}}_N + c \\ &= \tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}^T(\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T)\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} + 2(\bar{\boldsymbol{S}}\boldsymbol{h})^T\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} + c.\end{aligned} \tag{10}$$

Further, we rewrite the sparsity constraint (7) in terms of the support vector $\boldsymbol{w}$ to get the following optimization problem

$$\begin{aligned}\min_{\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}, \boldsymbol{w}} \; & \tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}^T(\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T)\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} + 2(\bar{\boldsymbol{S}}\boldsymbol{h})^T\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} + c \\ s.t. \; & \boldsymbol{1}^T\boldsymbol{w} \le s, \; \text{ and } \; \boldsymbol{w} \in \{0, 1\}^m\end{aligned} \tag{11}$$

where $\mathbf{1}$ denotes a column vector of all ones and there exists a one-to-one mapping between $\boldsymbol{w}$ and $\boldsymbol{S}$ as evident from (8)-(9) and hence cost in (11) is a function of $\boldsymbol{w}$ and $\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}$.

We first solve the above problem for a fixed support vector $\boldsymbol{w}$, and subsequently find the optimal support vector.

Lemma 1: Let $\boldsymbol{w}$ be a given binary support vector of control inputs that satisfies the sparsity constraint $\|\boldsymbol{w}\|_0 = s$. Further, consider a decomposition of the form $\boldsymbol{G} = a\boldsymbol{I} + \boldsymbol{L}$ where $0 < a < \lambda_{\min}(\boldsymbol{G})$ is a scalar and $\boldsymbol{L}$ is a symmetric positive definite matrix. Define

$$\bar{\boldsymbol{w}} \triangleq [\boldsymbol{w}^T, \boldsymbol{w}^T, \cdots, \boldsymbol{w}^T]^T \in \mathbb{R}^{mN\times 1}. \tag{12}$$

Then, the optimal cost of problem (11) is given by

$$J^*(\boldsymbol{S}) = c - \boldsymbol{h}^T\Big(\boldsymbol{L}^{-1} - \boldsymbol{L}^{-1}\big(\boldsymbol{L}^{-1} + a^{-1}\operatorname{diag}(\bar{\boldsymbol{w}})\big)^{-1}\boldsymbol{L}^{-1}\Big)\boldsymbol{h}. \tag{13}$$

Proof: To begin, note that since $\boldsymbol{G} \succ \mathbf{0}$, we can always find $(a, \boldsymbol{L})$ such that the decomposition $\boldsymbol{G} = a\boldsymbol{I} + \boldsymbol{L}$ holds. Next, since $\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}$ is a non-sparse vector, we compute the gradient of $J(\boldsymbol{S})$ in (10) with respect to $\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}$ and set it to $\mathbf{0}$ to get the optimal inputs as

$$\tilde{\boldsymbol{u}}^*_{\bar{\boldsymbol{S}}_N} = -(\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T)^{-1}\bar{\boldsymbol{S}}\boldsymbol{h}. \tag{14}$$

Note that $\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T$ is invertible since $\bar{\boldsymbol{S}}$ has full row rank and $\boldsymbol{G}$ is invertible. Substituting (14) in (10), we get the optimal cost

$$J^*(\boldsymbol{S}) = -\boldsymbol{h}^T\bar{\boldsymbol{S}}^T(\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T)^{-1}\bar{\boldsymbol{S}}\boldsymbol{h} + c. \tag{15}$$

Next, using $\boldsymbol{G} = a\boldsymbol{I} + \boldsymbol{L}$, we get

$$\begin{aligned}\bar{\boldsymbol{S}}^T(\bar{\boldsymbol{S}}\boldsymbol{G}\bar{\boldsymbol{S}}^T)^{-1}\bar{\boldsymbol{S}} &= \bar{\boldsymbol{S}}^T\left(a\boldsymbol{I}_{\|\bar{\boldsymbol{w}}\|_0} + \bar{\boldsymbol{S}}\boldsymbol{L}\bar{\boldsymbol{S}}^T\right)^{-1}\bar{\boldsymbol{S}} \\ &\overset{(a)}{=} \boldsymbol{L}^{-1} - \boldsymbol{L}^{-1}\left(\boldsymbol{L}^{-1} + a^{-1}\bar{\boldsymbol{S}}^T\bar{\boldsymbol{S}}\right)^{-1}\boldsymbol{L}^{-1} \\ &\overset{(b)}{=} \boldsymbol{L}^{-1} - \boldsymbol{L}^{-1}\left(\boldsymbol{L}^{-1} + a^{-1}\operatorname{diag}(\bar{\boldsymbol{w}})\right)^{-1}\boldsymbol{L}^{-1},\end{aligned} \tag{16}$$

where in step (a), we have used the matrix inversion lemma, and step (b) holds due to the property of $\boldsymbol{S}$ given in (8). Substituting (16) in (15), we get the optimal cost (13). ■

Note that the decomposition of $\boldsymbol{G}$ is not unique and (16) holds for all possible pairs $(a, \boldsymbol{L})$ and they give rise to same cost and hence the solution of the optimization is unaffected.

The expression (13) provides the optimal LQR cost for a given support vector $\boldsymbol{w}$. Next, we optimize this cost over all possible support vectors that satisfy the sparsity constraints. Hence, we get the following optimization problem

$$\begin{aligned}\min_{\boldsymbol{w}} \ & \boldsymbol{h}^T\boldsymbol{L}^{-1}\left(\boldsymbol{L}^{-1} + a^{-1}\operatorname{diag}(\bar{\boldsymbol{w}})\right)^{-1}\boldsymbol{L}^{-1}\boldsymbol{h} \\ \text{s.t.} \ & \mathbf{1}^T\boldsymbol{w} \le s, \ \text{ and } \ \boldsymbol{w} \in \{0,1\}^m,\end{aligned} \tag{17}$$

where we have omitted the constant term $c - \boldsymbol{h}^T\boldsymbol{L}^{-1}\boldsymbol{h}$ in the cost that does not affect the optimization.

The above cost contains an inverse term, which makes it a non-linear function of the optimization variable, and therefore, difficult to analyze. To address this, we convert it into a linear optimization problem by change of variables. We reformulate the above problem by introducing an auxiliary variable $\boldsymbol{V} \in \mathbb{S}^{mN}$ as follows

$$\min_{\boldsymbol{w},\boldsymbol{V}} \ \boldsymbol{h}^T\boldsymbol{V}\boldsymbol{h} \tag{18}$$

$$\text{s.t.} \ \boldsymbol{V} \succeq \boldsymbol{L}^{-1}\left(\boldsymbol{L}^{-1} + a^{-1}\operatorname{diag}(\bar{\boldsymbol{w}})\right)^{-1}\boldsymbol{L}^{-1}, \tag{18a}$$

$$\mathbf{1}^T\boldsymbol{w} \le s, \ \text{ and } \ \boldsymbol{w} \in \{0,1\}^m.$$

Note that the optimization problem (17) is equivalent to (18). Further, we apply the Schur complement theorem to (18a), and convert it to a linear matrix inequality (LMI). With this, the optimization problem is reformulated as

$$\min_{\boldsymbol{w},\boldsymbol{V}} \ \boldsymbol{h}^T\boldsymbol{V}\boldsymbol{h} \tag{19}$$

$$\text{s.t.} \ \begin{bmatrix} \boldsymbol{V} & \boldsymbol{L}^{-1} \\ \boldsymbol{L}^{-1} & \boldsymbol{L}^{-1} + a^{-1}\operatorname{diag}(\bar{\boldsymbol{w}}) \end{bmatrix} \succeq 0, \tag{19a}$$

$$\mathbf{1}^T\boldsymbol{w} \le s, \ \text{ and } \ \boldsymbol{w} \in \{0,1\}^m \tag{19b}$$

Problem (19) is combinatorial and NP-hard due to the boolean constraint (19b) on $\boldsymbol{w}$. Thus, we relax this constraint using semidefinite relaxation (SDR) [19], [20]. Since $w_i^2 = w_i$, the boolean constraints on $\boldsymbol{w}$ are relaxed as $\operatorname{diag}\left(\boldsymbol{w}\boldsymbol{w}^T\right) = \boldsymbol{w}$. This also forces w_i to take non-negative values, and hence, $\boldsymbol{w} \succeq \mathbf{0}$ is ensured. We introduce an auxiliary variable $\boldsymbol{W}$ as a proxy for $\boldsymbol{w}\boldsymbol{w}^T$, which enforces a rank-one constraint $\boldsymbol{W} = \boldsymbol{w}\boldsymbol{w}^T$. This rank-one constraint is also nonconvex. Hence, we relax it to $\boldsymbol{W} \succeq \boldsymbol{w}\boldsymbol{w}^T$, which is convex. Using the Schur complement theorem, this constraint is equivalent to

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{w} \\ \boldsymbol{w}^T & 1 \end{bmatrix} \succeq 0. \tag{20}$$

Thus, the boolean constraints on $\boldsymbol{w}$ are relaxed and replaced by (20) and $\operatorname{diag}(\boldsymbol{W}) = \boldsymbol{w}$. Further, the sparsity level is enforced by $\operatorname{tr}(\boldsymbol{W}) \le s$. Thus, the resulting relaxed problem is given as

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{W},\boldsymbol{V}} \ & \boldsymbol{h}^T\boldsymbol{V}\boldsymbol{h} \\ \text{s.t.} \ & \text{LMI in (19)}, \\ & \operatorname{tr}(\boldsymbol{W}) \le s, \operatorname{diag}(\boldsymbol{W}) = \boldsymbol{w}, \begin{bmatrix} \boldsymbol{W} & \boldsymbol{w} \\ \boldsymbol{w}^T & 1 \end{bmatrix} \succeq 0.\end{aligned} \tag{21}$$

The above problem is a semidefinite program (SDP) problem that can be solved numerically in an efficient manner (it has polynomial complexity), for instance, using the interior-point method. The solution of the SDP yields a vector $\boldsymbol{w}$ in which s entries are very close to 1 and remaining entries are close to 0. This demonstrates that the convex relaxation effectively enforces sparsity at the desired level s in the solution. After obtaining the solution, we map the s largest entries of $\boldsymbol{w}$ to 1 and all other entries to 0 to satisfy the sparsity level. Finally, we use the resulting $\boldsymbol{w}$ to compute the optimal control inputs using (14).

B. Time-Varying Support

Here, we allow the support of the control inputs to vary with time. However, the sparsity level s remains fixed at every time instant. This is more general than the fixed temporal support setting in the previous subsection. Let $\boldsymbol{w}_k$ denote the support vector of $\boldsymbol{u}(k)$ and define $\bar{\boldsymbol{w}} = [\boldsymbol{w}_0^T, \boldsymbol{w}_1^T, \cdots \boldsymbol{w}_{N-1}^T]^T$. Let $\boldsymbol{S}_k$ denote the selection matrix at time k that is constructed from $\boldsymbol{w}_k$ similarly as before.

Following similar steps as in Section III-A, the optimization problem for this case can be formulated as

$$\begin{aligned} \min_{\{\boldsymbol{w}_k\}_{k=0}^{N-1}, \boldsymbol{V}} \quad & \boldsymbol{h}^T \boldsymbol{V} \boldsymbol{h} \\ \text{s.t.} \quad & \begin{bmatrix} \boldsymbol{V} & \boldsymbol{L}^{-1} \\ \boldsymbol{L}^{-1} & \boldsymbol{L}^{-1} + a^{-1}\operatorname{diag}(\bar{\boldsymbol{w}}) \end{bmatrix} \succeq 0, \\ & \mathbf{1}^T \boldsymbol{w}_k \leq s \text{ and } \boldsymbol{w}_k \in \{0,1\}^m \ \forall k = 0, \cdots N-1. \end{aligned} \tag{22}$$

We relax the boolean constraints on $\boldsymbol{w}_k$ similar to Section III-A, and the final SDP is given as

$$\begin{aligned} \min_{\{\boldsymbol{w}_k, \boldsymbol{W}_k\}_{k=0}^{N-1}, \boldsymbol{V}} \quad & \boldsymbol{h}^T \boldsymbol{V} \boldsymbol{h} \\ \text{s.t.} \quad & \text{LMI in (22)}, \\ & \operatorname{tr}(\boldsymbol{W}_k) \leq s, \quad \operatorname{diag}(\boldsymbol{W}_k) = \boldsymbol{w}_k, \\ & \begin{bmatrix} \boldsymbol{W}_k & \boldsymbol{w}_k \\ \boldsymbol{w}_k^T & 1 \end{bmatrix} \succeq 0, \ \forall k = 0, \cdots, N-1. \end{aligned} \tag{23}$$

Note that the number of optimization variables and constraints in this case is more than in the fixed temporal support case, resulting in a higher computational complexity.

The SDP in (21) and (23) can be solved using standard convex optimization packages (e.g. CVX) that use an interior point solver. The computational complexity of SDP is a polynomial function of dimensions of optimization variables. The asymptotic complexity scales as $\mathcal{O}(m^{4.5})$ and $\mathcal{O}(N^{4.5}m^{4.5})$ [19] for fixed temporal support and time-varying support case respectively. This is a vast improvement over solving the problem by exhaustively searching for the optimal support, the complexity of which grows exponentially with Nm.

IV. Minimum Energy Control

In this section, we consider the minimum energy control problem, where the goal is to to transfer the state of the system from a given initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to a given final state $\boldsymbol{x}(N) = \boldsymbol{x}_f$ in N time steps using the least amount of control input energy. We wish to solve this problem under sparsity constraints on the inputs with fixed temporal support as considered previously in Section III-A. The cost function for this problem can be obtained by setting $\boldsymbol{Q} = \mathbf{0}_n$ and $\boldsymbol{R} = \boldsymbol{I}_m$ in the LQR cost (5), which yields the control energy $J = \sum_{k=0}^{N-1} \|\boldsymbol{u}(k)\|_2^2 = \|\tilde{\boldsymbol{u}}_N\|_2^2$. Further, using (3), the final state constraint is $\boldsymbol{x}(N) = \boldsymbol{A}^N \boldsymbol{x}_0 + \tilde{\boldsymbol{C}}_N \tilde{\boldsymbol{u}}_N = \boldsymbol{x}_f$. Thus, the minimum energy control problem[1] is formulated as

$$\begin{aligned} \min_{\tilde{\boldsymbol{u}}_N} \quad & \|\tilde{\boldsymbol{u}}_N\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{A}^N \boldsymbol{x}_0 + \tilde{\boldsymbol{C}}_N \tilde{\boldsymbol{u}}_N = \boldsymbol{x}_f \\ & \|\boldsymbol{u}(k)\|_0 \leq s \ \forall k = 0, \cdots, N-1. \end{aligned} \tag{24}$$

[1]It is possible to formulate the problem in LQR framework, where the equality constraint $\boldsymbol{x}(N) = \boldsymbol{x}_f$ is incorporated as a quadratic penalty term in the objective function. By minimizing the cost: $\|\tilde{\boldsymbol{u}}_N\|_2^2 + c\|\boldsymbol{x}(N) - \boldsymbol{x}_f\|^2$, ($c$ is a large weightage) minimum energy control problem can be approximately solved using LQR formulation presented in Section III.

We assume that the system satisfies the conditions of s-sparse controllability [21], which implies that the above problem is feasible for any $(\boldsymbol{x}_0, \boldsymbol{x}_f)$ under the sparsity constraints. We first solve the above problem for a fixed support vector $\boldsymbol{w}$ as defined above, and subsequently find the optimal $\boldsymbol{w}$.

Lemma 2: Let $\boldsymbol{w}$ be a given binary support vector that satisfies the sparsity constraint $\|\boldsymbol{w}\|_0 = s$. Let $\boldsymbol{d} \triangleq \boldsymbol{A}^N \boldsymbol{x}_0 - \boldsymbol{x}_f$. Then, the optimal cost of problem (24) is given by

$$J^* = \boldsymbol{d}^T \left(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \right)^{-1} \boldsymbol{d}, \tag{25}$$

where $\bar{\boldsymbol{w}}$ is as defined in (12).

Proof: Since the sparsity constraints are assumed to be satisfied, we ignore them in (24). Further, since all the inputs have a (given) common support vector $\boldsymbol{w}$, we have $\|\tilde{\boldsymbol{u}}_N\|_2^2 = \|\bar{\boldsymbol{S}}\tilde{\boldsymbol{u}}_N\|_2^2 = \|\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}\|^2$ where $\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}$ is a non-sparse vector and $\bar{\boldsymbol{S}}$ is as defined in (9). Also, $\tilde{\boldsymbol{C}}_N \tilde{\boldsymbol{u}}_N = \tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \bar{\boldsymbol{S}} \tilde{\boldsymbol{u}}_N = \left(\tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \right) \tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}$ since $\bar{\boldsymbol{S}}^T \bar{\boldsymbol{S}} = \operatorname{diag}(\bar{\boldsymbol{w}})$. Hence, (24) becomes

$$\begin{aligned} \min_{\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}} \quad & \|\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{A}^N \boldsymbol{x}_0 + \left(\tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \right) \tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} = \boldsymbol{x}_f. \end{aligned} \tag{26}$$

The Lagrangian of the above equality-constrained optimization problem is given by

$$\mathcal{L}(\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}, \lambda) = \|\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}\|_2^2 + \boldsymbol{\lambda}^T (\boldsymbol{d} + \tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}), \tag{27}$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier and $\boldsymbol{d} \triangleq \boldsymbol{A}^N \boldsymbol{x}_0 - \boldsymbol{x}_f$. Setting $\nabla_{\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N}} \mathcal{L} = 2\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} + \bar{\boldsymbol{S}} \tilde{\boldsymbol{C}}_N^T \boldsymbol{\lambda} = 0$, we get the optimal inputs as

$$\tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} = -\frac{1}{2} \bar{\boldsymbol{S}} \tilde{\boldsymbol{C}}_N^T \boldsymbol{\lambda}. \tag{28}$$

Further, $\nabla_\lambda \mathcal{L} = \mathbf{0} \implies \boldsymbol{d} + \tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \tilde{\boldsymbol{u}}_{\bar{\boldsymbol{S}}_N} = 0$, and using (28) we get

$$\boldsymbol{\lambda} = 2(\tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \bar{\boldsymbol{S}} \tilde{\boldsymbol{C}}_N^T)^{-1} \boldsymbol{d} = 2(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T)^{-1} \boldsymbol{d}. \tag{29}$$

Substituting (29) in (28), we obtain

$$\tilde{\boldsymbol{u}}^*_{\bar{\boldsymbol{S}}_N} = -\bar{\boldsymbol{S}} \tilde{\boldsymbol{C}}_N^T \left(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \right)^{-1} \boldsymbol{d}. \tag{30}$$

Finally, the optimal cost is given by

$$\begin{aligned} J^* &= \|\tilde{\boldsymbol{u}}^*_{\bar{\boldsymbol{S}}_N}\|^2 \\ &= \boldsymbol{d}^T \left(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \right)^{-1} \tilde{\boldsymbol{C}}_N \bar{\boldsymbol{S}}^T \\ &\qquad \cdot \bar{\boldsymbol{S}} \tilde{\boldsymbol{C}}_N^T \left(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \right)^{-1} \boldsymbol{d}. \end{aligned}$$

Using the property $\bar{\boldsymbol{S}}^T \bar{\boldsymbol{S}} = \operatorname{diag}(\bar{\boldsymbol{w}})$, we get (25). ■

Next, using the optimal cost in (25), we aim to find the optimal support vector that minimizes this cost and satisfies the sparsity constraints. Towards this, we formulate the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \boldsymbol{d}^T \left(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \right)^{-1} \boldsymbol{d} \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{w} \leq s, \text{ and } \boldsymbol{w} \in \{0,1\}^m. \end{aligned} \tag{31}$$

The cost in (31) contains inverse of a matrix, which makes it a non-linear function of the optimization variables, and therefore, difficult to analyze. To address this, we convert it into a linear optimization problem by change of variables. We introduce an auxiliary variable $\boldsymbol{Z} \in \mathbb{S}^n$ and reformulate (31) as

$$\min_{\boldsymbol{w},\boldsymbol{Z}} \ \boldsymbol{d}^T \boldsymbol{Z} \boldsymbol{d} \tag{32}$$

$$\text{s.t.} \ \left(\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T\right)^{-1} \preceq \boldsymbol{Z}, \tag{32a}$$

$$\mathbf{1}^T \boldsymbol{w} \leq s, \ \text{ and } \ \boldsymbol{w} \in \{0,1\}^m. \tag{32b}$$

Next, we introduce another auxiliary variable $\boldsymbol{V} \in \mathbb{S}^n$ such that the matrix inequality (32a) is expressed equivalently by the following two inequalities

$$\boldsymbol{V} \succeq \boldsymbol{Z}^{-1}, \tag{33}$$

$$\tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \succeq \boldsymbol{V}. \tag{34}$$

Next, applying the Schur complement theorem, (33) is equivalent to the following LMI

$$\begin{bmatrix} \boldsymbol{V} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{Z} \end{bmatrix} \succeq 0. \tag{35}$$

Finally, we use the same Semidefinite Relaxation (SDR) mentioned previously (below (19)) to the constraints (32b). With all these steps, we obtain the following SDP for the relaxed minimum energy control problem:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{Z},\boldsymbol{V},\boldsymbol{W}} \quad & \boldsymbol{d}^T \boldsymbol{Z} \boldsymbol{d} \\ \text{s.t.} \quad & \text{LMIs in (34) and (35)}, \\ & \operatorname{tr}(\boldsymbol{W}) \leq s, \quad \operatorname{diag}(\boldsymbol{W}) = \boldsymbol{w}, \\ & \begin{bmatrix} \boldsymbol{W} & \boldsymbol{w} \\ \boldsymbol{w}^T & 1 \end{bmatrix} \succeq 0. \end{aligned} \tag{36}$$

As before, the above SDP can be solved efficiently. After obtaining the solution, we map the s largest entries of $\boldsymbol{w}$ to 1 and all other entries to 0. The resulting $\boldsymbol{w}$ is used to compute the optimal control inputs using (30).

Remark 1 (Time-varying Support): The minimum energy control problem for time-varying support can be formulated in a similar manner as above. To avoid repetition, we do not present all the steps. The final SDP in this case is given as

$$\begin{aligned} \min_{\{\boldsymbol{w}_k,\boldsymbol{W}_k\}_{k=0}^{N-1},\boldsymbol{Z},\boldsymbol{V}} \quad & \boldsymbol{d}^T \boldsymbol{Z} \boldsymbol{d} \\ \text{s.t.} \quad & \tilde{\boldsymbol{C}}_N \operatorname{diag}(\bar{\boldsymbol{w}}) \tilde{\boldsymbol{C}}_N^T \succeq \boldsymbol{V}, \quad \begin{bmatrix} \boldsymbol{V} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{Z} \end{bmatrix} \succeq 0, \\ & \bar{\boldsymbol{w}} = [\boldsymbol{w}_0^T, \ldots, \boldsymbol{w}_{N-1}^T]^T, \\ & \operatorname{tr}(\boldsymbol{W}_k) \leq s, \quad \operatorname{diag}(\boldsymbol{W}_k) = \boldsymbol{w}_k, \\ & \begin{bmatrix} \boldsymbol{W}_k & \boldsymbol{w}_k \\ \boldsymbol{w}_k^T & 1 \end{bmatrix} \succeq 0 \quad \forall k = 0, \ldots, N-1. \end{aligned} \tag{37}$$

V. Simulation Results

In this section, we present empirical results for the SDPs developed for sparsity-constrained LQR and minimum energy control problems. To solve the SDPs, we use the CVX package [22] in MATLAB 2024a. To verify the effectiveness and accuracy of our proposed SDPs, we compare our relaxed solutions against the optimal solutions obtained via an exhaustive search. For the fixed temporal support case, we exhaustively explore $\binom{m}{s}$ support vectors and compute the optimal inputs for each of them. For the time-varying support case, we exhaustively search over $\binom{m}{s}^N$ support vectors. It is evident that exhaustive search becomes computationally infeasible even for moderate values of m and N due to the exponential increase in search space size. Hence, we keep the problem dimensions small to facilitate a comparison between our proposed methods and the true optimal solution obtained through the exhaustive search.

We randomly generate the system matrices and initial condition as

$$\boldsymbol{A} = \begin{bmatrix} 0.05 & -0.29 & -0.61 & -0.40 \\ 0.25 & 0.41 & 0.33 & -0.79 \\ 0.55 & 0.08 & -0.18 & 0.08 \\ 0.49 & -0.25 & 0.02 & -0.03 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 1.19 & -0.93 & 0.72 & -1.42 & 1.40 & 0.66 \\ 0.80 & -1.26 & -0.77 & 0.71 & 0.40 & 2.13 \\ 1.05 & 0.49 & 0.83 & -0.77 & 0.92 & 0.54 \\ -0.74 & 2.78 & -1.12 & 0.31 & -1.60 & -1.54 \end{bmatrix},$$

$$\boldsymbol{x}_0 = [-13.85, -19.56, 4.2, 4.01]^T,$$

and select $N = 4$. The sparsity level of the input is varied from $s = 1$ to $s = 6$.

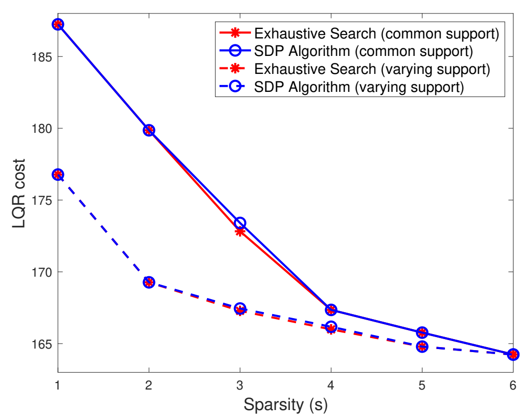

Fig. 1. Performance comparison of the LQR cost obtained from our SDP-based algorithm with the true optimal cost (obtained from exhaustive search) for different sparsity levels.

Figure 1 shows the LQR cost achieved by our SDP-based algorithm and the optimal cost obtained via exhaustive search for different sparsity levels s. We make the following observations from the figure: (i) Our SDP-based algorithm provides a near-optimum solution in both fixed and time-varying support cases, validating its effectiveness and accuracy. (ii) The optimal cost decreases with increasing s since the inputs become less restricted. (iii) The optimal cost achieved by time-varying support is smaller than the fixed support case since the latter case is more restrictive than the former, (iv) At $s = m$, all curves coincide as this case is

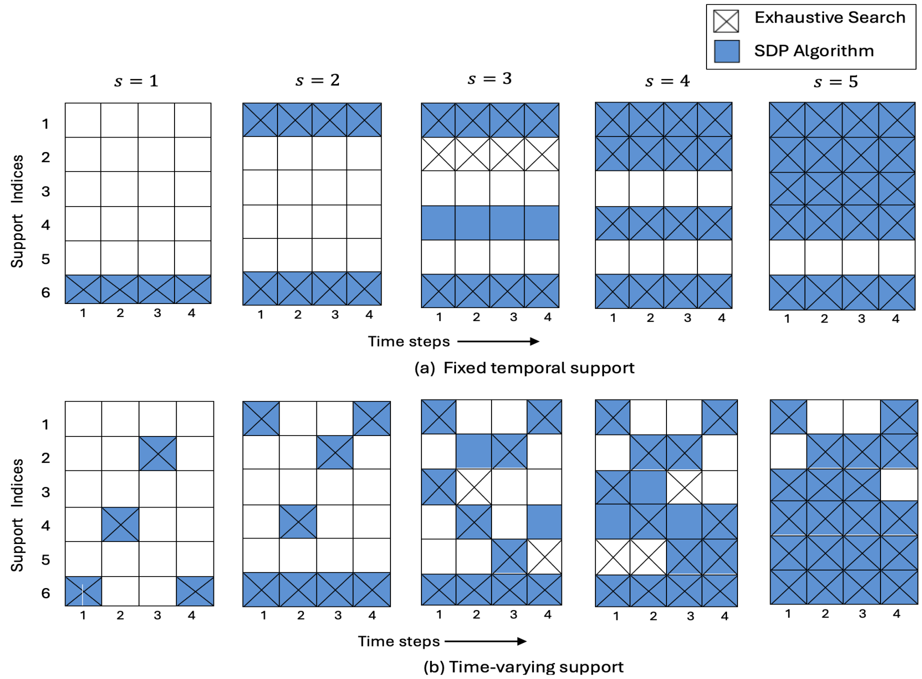

Fig. 2. Sparse supports obtained from our SDP-based algorithm and the exhaustive search. Horizontal axis represents time steps from $k = 0, \cdots, N-1$. k^{th} column of each table represents support vector of the input $\boldsymbol{u}_k$.

equivalent to the conventional LQR without any sparsity, (v) The SDP-based cost nearly coincides with the optimal cost at all s values, although there is a small gap $s = 3$ and $s = 4$.

Let $\mathcal{S}_k$ denote the support set of $\boldsymbol{u}(k)$, that is, the index set of entries of $\boldsymbol{u}(k)$ which are allowed to be nonzero. Figure 2 shows the support sets obtained by our SDP-based algorithm and the true optimal support sets obtained by the exhaustive search. We observe that both the support sets match in the majority of the cases, showing that our SDP-based algorithm performs well in obtaining the optimal support. The exceptions are when $s = 3$ for fixed temporal support case (SDP-based support set is $\{1, 4, 6\}$ and the optimal support set is $\{1, 2, 6\}$), and when $s = 3, 4$ for the time-varying support case. However, from the previous result, we see that the cost obtained by the SDP-based algorithm is only marginally away from the optimal cost obtained via exhaustive search, showing that the difference in the support identified by the two approaches does not significantly impact the performance.

Next, we quantify the accuracy of support sets obtained by our SDP-based algorithm. Let $\mathcal{S}_k$ and $\mathcal{S}_k^*$ denote the support sets obtained by the SDP-based algorithm and the exhaustive search. Then, the total number of incorrect support indices for $\boldsymbol{u}_k$ obtained by SDP-based algorithm is given by $|\mathcal{S}_k^* \oplus \mathcal{S}_k|/2$. Averaging over inputs at all time instants, we define the false support rate (FSR) as

$$\text{FSR} = \frac{1}{N \cdot s} \sum_{k=0}^{N-1} \frac{|\mathcal{S}_k^* \oplus \mathcal{S}_k|}{2}. \tag{38}$$

For fixed temporal support case, (38) reduces to $\frac{|\mathcal{S}^* \oplus \mathcal{S}|}{2 \cdot s}$. We compute the FSR for 100 trials, where in each trial $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{x}_0)$ are generated randomly from a Gaussian distribution. The FSR values averaged over all trials are reported in Table I. We observe that the FSR values are within 4% for all s (except for $s = 1$), i.e., the SDP-based algorithm achieves good accuracy in recovering the optimal support.

TABLE I. Percentage of incorrect support (FSR in %) obtained from SDP-based algorithm for different sparsity levels

Support Type	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$
Fixed temporal	4.0	3.5	3.66	3.75	1.8
Time-varying	11.7	4.0	3.91	2.31	0.85

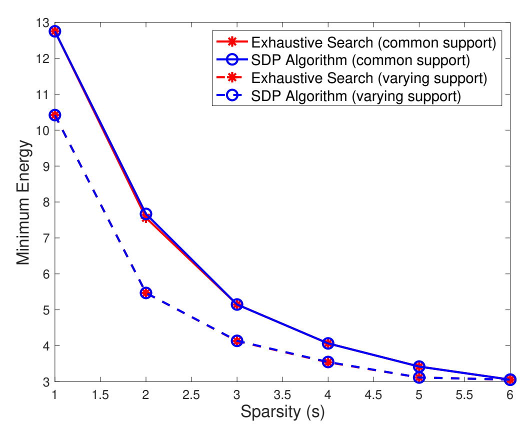

Fig. 3. Performance comparison of the minimum-energy control cost obtained from our SDP-based algorithm with the true optimal cost (obtained from exhaustive search) for different sparsity levels. In all cases system reaches it's final state $\boldsymbol{x}_f$.

Next, we show the performance of SDP-based algorithm for the minimum energy control problem. The parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{x}_0$ are the same as mentioned before and we set $\boldsymbol{x}_f = [-0.7132, -9.3830, 1.6136, -2.6818]^T$. Figure 3 shows the control energy obtained by the SDP-based algorithm and the optimal control energy obtained by the exhaustive search for different sparsity levels. Similar to the LQR case, we observe that the SDP-based algorithm obtains near optimum solution for all values of s. Further, all observations made for the LQR case hold true here as well.

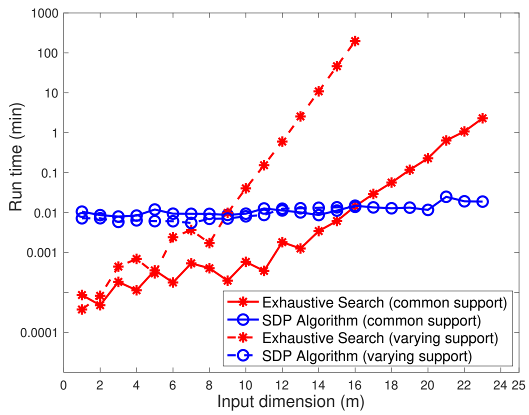

Fig. 4. Runtime comparison of SDP-based algorithm with exhaustive search for sparsity constrained LQR problem.

Finally, we show the superiority of SDP-based algorithms over exhaustive search-based approach in terms of runtime complexity. We fix $n = 4$, $N = 2$ and m is varied from 1 to 23. The sparsity level s is set at $\lceil \frac{m}{2} \rceil$. $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{x}_0)$ are generated randomly from standard normal distribution. In Figure 4, we plot the runtime of the SDP-based and exhaustive search algorithms for different input dimensions for sparsity-constrained LQR problem. Since the scale is logarithmic, the linear trend observed in the runtime of the exhaustive search-based approach indicates its exponential growth in complexity with respect to m. In contrast, the SDP-based algorithms exhibit a significantly slower increase in runtime compared to exhaustive search. While exhaustive search is feasible for small values of m, its runtime rapidly escalates for even moderate values of m. Therefore, for large-scale systems with high input dimensions, the use of SDP-based algorithms is preferred.

VI. Conclusion

We studied the sparsity-constrained LQR and sparsity-constrained minimum-energy control problem for a discrete-time linear dynamical system. Obtaining the optimal sparse control inputs for these problems is NP-hard in general due to combinatorial complexity. We developed convex relaxation-based approaches to reformulate the nonconvex optimization problem into a tractable semidefinite program that can be solved in polynomial time. Extending the work for the infinite horizon case and for linear quadratic Gaussian (LQG) control and Model Predictive Control (MPC) are interesting directions for future work.

References

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

[2] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1686–1701, Jun. 2020.

[3] 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.

[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] 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.

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

[7] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto, "Recognition of human gaits," in *Proc. IEEE CVPR*, vol. 2, Dec. 2001, pp. II–II.

[8] M. Raptis, K. Wnuk, and S. Soatto, "Spike train driven dynamical models for human actions," in *Proc. IEEE CVPR*, Jun. 2010, pp. 2077–2084.

[9] M. R. Jovanović and F. Lin, "Sparse quadratic regulator," in *2013 European Control Conference (ECC)*, 2013, pp. 1047–1052.

[10] A. Olshevsky, "On a relaxation of time-varying actuator placement," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 656–661, 2020.

[11] T. Ikeda and K. Kashima, "Sparsity-constrained controllability maximization with application to time-varying control node selection," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 321–326, 2018.

[12] F. Lin, M. Fardad, and M. R. Jovanović, "Sparse feedback synthesis via the alternating direction method of multipliers," in *2012 American Control Conference (ACC)*, 2012, pp. 4765–4770.

[13] ——, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.

[14] ——, "Algorithms for leader selection in large dynamical networks: Noise-corrupted leaders," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 2932–2937.

[15] ——, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1789–1802, 2014.

[16] C. V. Rao, "Sparsity of linear discrete-time optimal control problems with l_1 objectives," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 513–517, 2018.

[17] M. Annergren, A. Hansson, and B. Wahlberg, "An admm algorithm for solving ℓ_1 regularized mpc," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 4486–4491.

[18] J. Stefan and T. Tanaka, "Sparse lqr synthesis via information regularization," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 7345–7351.

[19] S. Liu, S. P. Chepuri, M. Fardad, E. Maşazade, G. Leus, and P. K. Varshney, "Sensor selection for estimation with correlated measurement noise," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3509–3522, 2016.

[20] Z.-q. Luo, W.-k. Ma, A. M.-c. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[21] 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.

[22] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," https://cvxr.com/cvx, Mar. 2014.